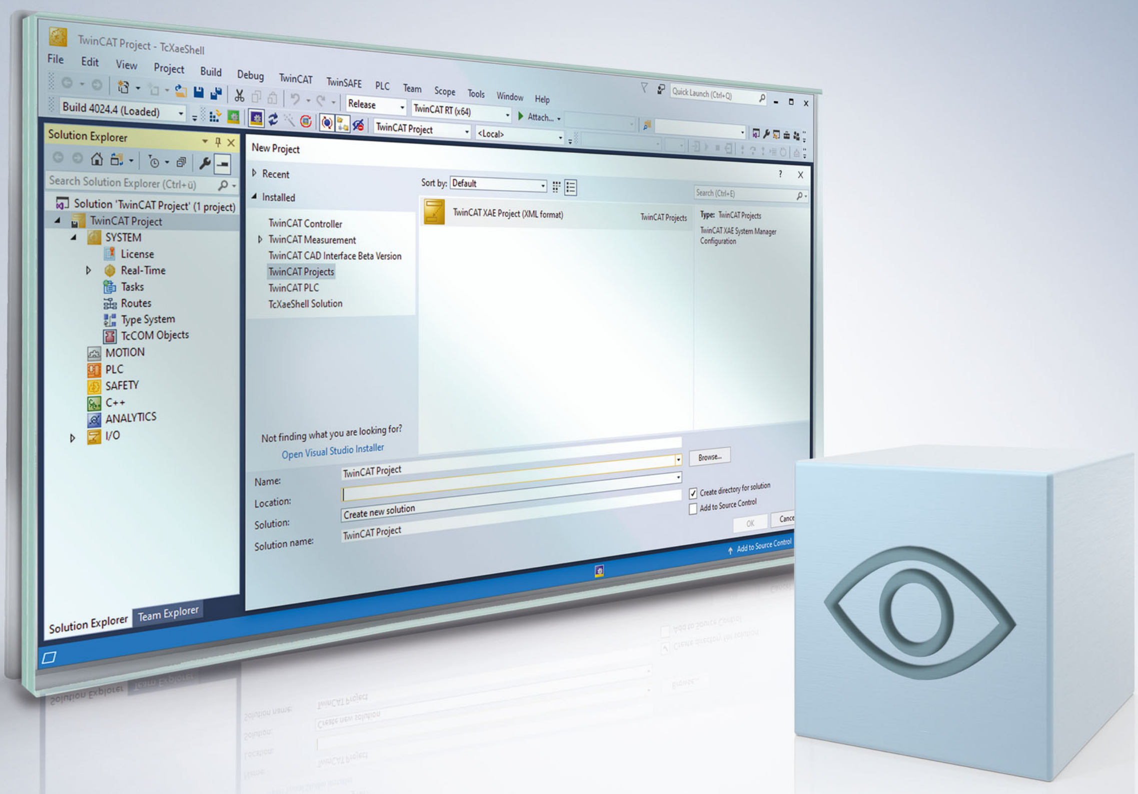
BECKHOFF New Automation Technology

手册 | ZH

TF7000 - TF7800

TwinCAT 3 | Vision



2024-07-16 | 版本： 2.0.1

目录

1 前言

1.1 文档说明

本说明仅适用于熟悉国家标准且经过培训的控制和自动化工程专家。
在安装和调试组件时，必须遵循文档和以下说明及解释。
操作人员应具备相关资质，并始终使用最新的生效文档。

相关负责人员必须确保所述产品的应用或使用符合所有安全要求，包括所有相关法律、法规、准则和标准。

免责声明

尽管本文档经过精心编制， 然而，所述产品正在不断开发中。
我们保留随时修订和更改本文档的权利，恕不另行通知。
不得依据本文档中的数据、图表和说明对已供货产品的修改提出赔偿。

商标

Beckhoff®、TwinCAT®、TwinCAT/BSD®、TC/BSD®、EtherCAT®、EtherCAT G®、EtherCAT G10®、EtherCAT P®、Safety over EtherCAT®、TwinSAFE®、XFC®、XTS® 和 XPlanar® 是德国倍福自动化有限公司的注册商标并已获得授权。
本文档中所使用的其它名称可能是商标名称，任何第三方为其自身目的而引用，都可能触犯商标所有者的权利。

正在申请的专利

涵盖 EtherCAT 技术，包括但不限于以下专利申请和专利：
EP1590927、EP1789857、EP1456722、EP2137893、DE102015105702
并在多个其他国家进行了相应的专利申请或注册。

EtherCAT® 是注册商标和专利技术，由德国倍福自动化有限公司授权使用。

版权所有

1.2 安全信息

安全规范

为了确保您的使用安全，请务必仔细阅读
并遵守本文档中每个产品的安全使用说明。

责任免除

所有组件在供货时都配有适合应用的特定硬件和软件配置。严禁未按文档所述修改硬件或软件配置，否则，德国倍福自动化有限公司对由此产生的后果不承担责任。

人员资格

本说明仅供熟悉适用国家标准的控制、自动化和驱动工程专家使用。

警示性词语

文档中使用的警示信号词分类如下。为避免人身伤害和财产损失，请阅读并遵守安全和警告注意事项。

人身伤害警告

⚠ 危险
存在死亡或重伤的高度风险。

⚠ 警告
存在死亡或重伤的中度风险。

⚠ 谨慎
存在可能导致中度或轻度伤害的低度风险。

财产或环境损害警告

注意
可能会损坏环境、设备或数据。

操作产品的信息

这些信息包括：
有关产品的操作、帮助或进一步信息的建议。

1.3 信息安全说明

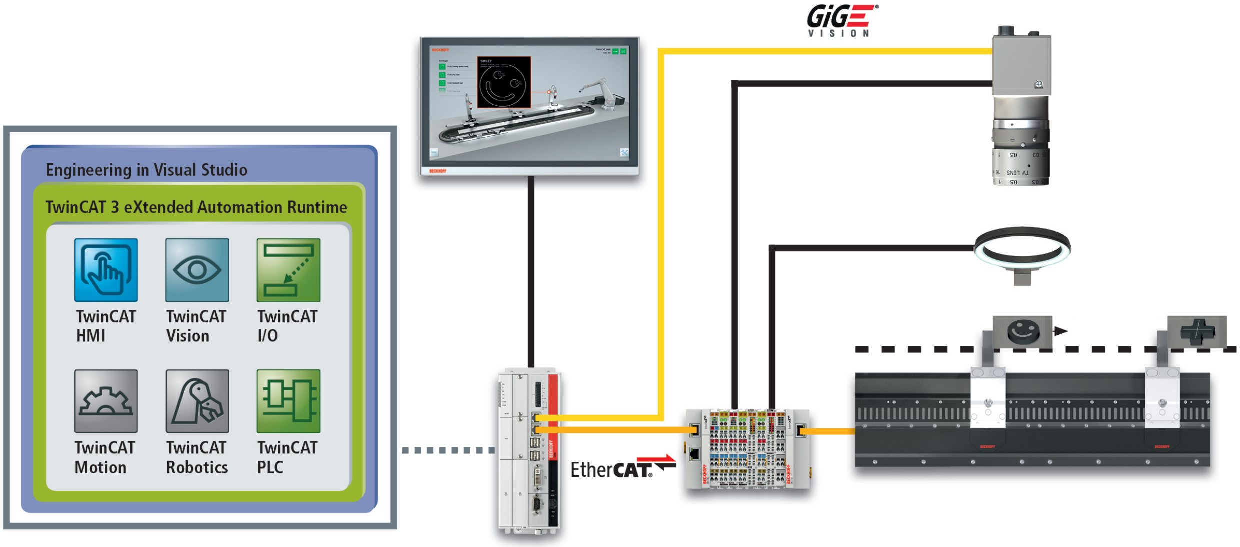
Beckhoff Automation GmbH & Co.KG（简称 Beckhoff）的产品，只要可以在线访问，都配备了安全功能，支持工厂、系统、机器和网络的安全运行。尽管配备了安全功能，但为了保护相应的工厂、系统、机器和网络免受网络威胁，必须建立、实施和不断更新整个操作安全概念。Beckhoff 所销售的产品只是整个安全概念的一部分。客户有责任防止第三方未经授权访问其设备、系统、机器和网络。它们只有在采取了适当的保护措施的情况下，方可与公司网络或互联网连接。

此外，还应遵守 Beckhoff 关于采取适当保护措施的建议。关于信息安全和工业安全的更多信息，请访问本公司网站 https://www.beckhoff.com/secguide。

Beckhoff 的产品和解决方案持续进行改进。这也适用于安全功能。鉴于持续进行改进，Beckhoff 明确建议始终保持产品的最新状态，并在产品更新可用后马上进行安装。使用过时的或不支持的产品版本可能会增加网络威胁的风险。

如需了解 Beckhoff 产品信息安全的信息，请订阅 https://www.beckhoff.com/secinfo 上的 RSS 源。

2 概述



TwinCAT Vision是TwinCAT 3 中的机器视觉功能模块。诸如目标检测、识别和测量等任务可直接在 PLC 中实时处理。通过将图像处理集成到 TwinCAT 平台中，可以实现高度同步的应用控制和极短的响应时间。应用开发由大量现有 TwinCAT Engineering 工具辅助。

图像处理序列可直接在 PLC 代码中使用 Tc3_Vision 库以 IEC61131-3 语言进行编程。另外，也可以在 TwinCAT 3 C++ 中进行编程。可以使用图像显示工具直接检查中间结果。

TwinCAT Vision 具有实时功能，因为图像处理算法直接在 TwinCAT 3 运行时环境中与控制器同步执行。此外，合适的算法可以在多个核上自动并行运行。

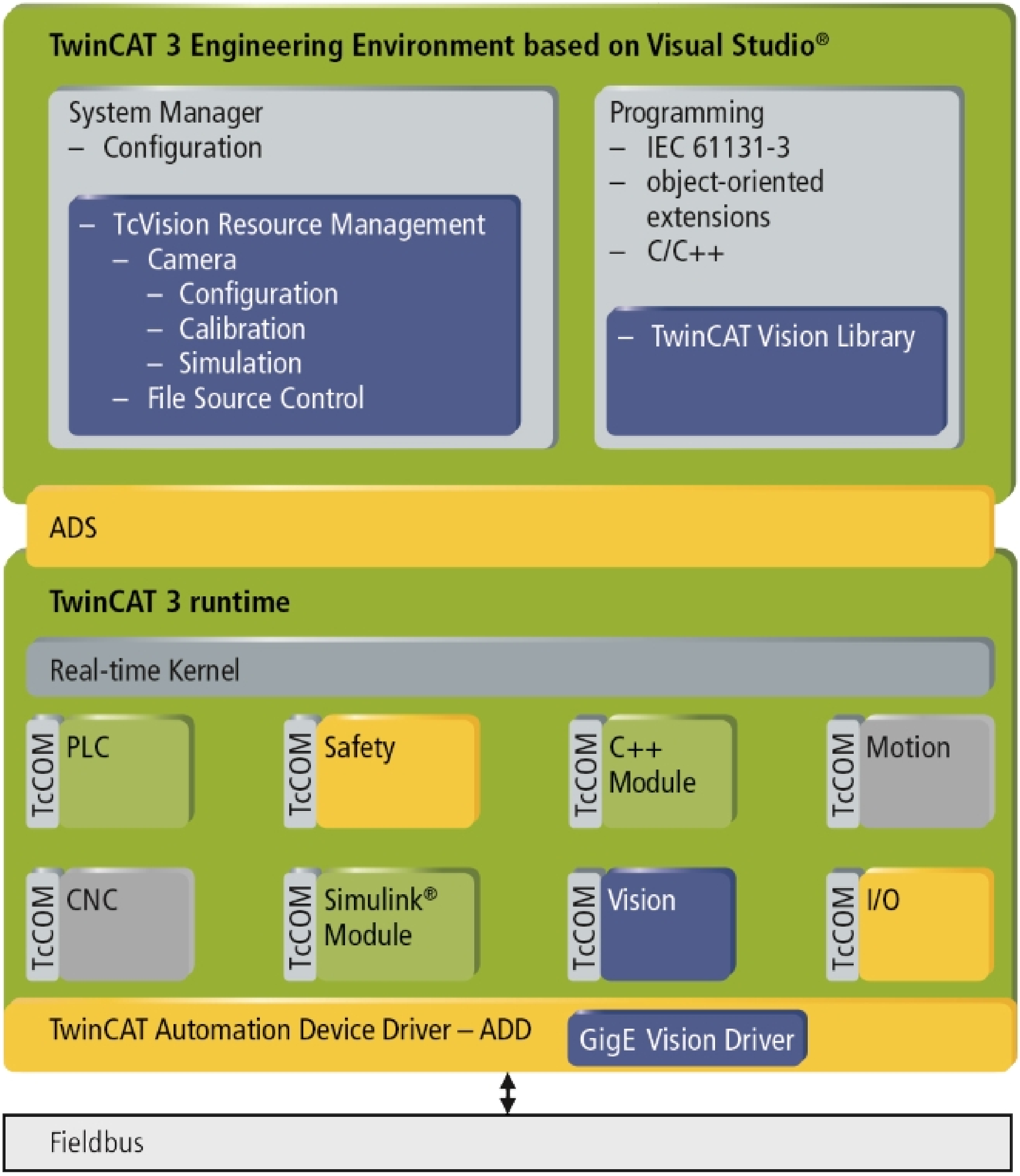
通过 GigE Vision 接口可连接大量工业相机。熟悉的 TwinCAT 开发环境可用于配置和标定相机，以及系统设置和调试。

各章提供有关以下主题的详细信息：

- 系统组件 [▶ 9]，用于了解 TwinCAT Vision 的系统结构
- 安装 [▶ 13]，提供有关安装和授权的指南
- 第一步 [▶ 23]，提供对 TwinCAT Vision 的简单介绍
- Development environment [▶ 48]，有关相机配置、标定和仿真的信息
- API 参考 [▶ 119]（软件文档）
- 一些图像处理序列的示例 [▶ 2639]
- 故障排除须知 [▶ 2737]

2.1 系统组件

与其他 TwinCAT 组件一样，TwinCAT Vision 也集成到基于 Visual Studio 的 TwinCAT 开发环境和具有实时功能的 TwinCAT 运行时环境中。下文对 TwinCAT Vision 系统的组件进行了说明。



配置助手

- 工业相机可通过 GigE Vision [▶ 107] 接口连接。助手 [▶ 63]可提供相机配置、标定和仿真服务。
- 图像可通过文件源 [▶ 110]元件从文件系统输入 PLC 运行时。
- 在 TwinCAT 实时设置中必须针对 TwinCAT Vision 进行核和路由器内存的配置 [▶ 48]。
- TwinCAT Vision 可以利用 TwinCAT 的多核支持来并行执行某些算法。相应设置在 Vision Job Pool [▶ 114] 中进行。

库

TwinCAT Vision 提供一个包含大量算法的库，可用于图像处理序列的编程。它还包含作为相机和文件系统接口的功能块。API 参考 [▶ 119]包含所有库元件的详细说明。

调试工具

TwinCAT Vision 提供以下图像处理序列调试机构：

- ADS Image Watch [▶ 115] 可用于实时可视化相机图像和中间结果。它通过 ADS 将图像从 PLC 运行时传输到开发环境，并在其中显示。
- 所有 TwinCAT Vision 功能均根据 ADS 返回代码 [▶ 2753]（请参见 HRESULT [▶ 122]）提供有关其执行状态的信息。
- 所有 TwinCAT 调试功能（如断点等）均可用于编程。

GigE Vision 驱动程序

TwinCAT Vision 包括一个具有实时功能的GigE Vision [▶ 107]驱动程序。它用于相机配置的开发环境和TwinCAT运行时。它通过UDP/IP与工业相机进行通信，基于GigE Vision标准。该标准用于查找和配置网络中的相机，并接收来自相机的图像数据。

TwinCAT Vision服务

对于某些功能，TwinCAT Vision 必须在运行时与外部组件进行实时通信。例如，文件源控制从文件系统加载图像。TwinCAT Vision Service [▶ 59]是 TwinCAT 运行时与外部 Vision 组件之间的一个集中的、基于 ADS 的通信接口。

TwinCAT HMI 视觉扩展

TwinCAT HMI 视觉扩展 [▶ 2599]是TwinCAT HMI的一个扩展，可将图像从 PLC 传输到 TwinCAT HMI。来自 PLC 的图像可以通过 HmiImageControl 直接显示在基于网络的 HMI 中。

TwinCAT HMI 视觉控制

TwinCAT HMI 视觉控制 [▶ 2601]包提供了视觉专用控制，作为TwinCAT HMI的扩展。例如，它包括一个控件，除了图像显示外，控件还提供许多工具和信息显示。这些功能和显示可以很容易地通过配置进行调整，并直接使用，而不需要额外编程。

2.2 授权模式

用于 TwinCAT Vision 的 TF7xxx 产品包括用于相机连接的许可证和一个图像处理库。如需生成许可证，请按照相应的指南 [▶ 20]进行。对于 TwinCAT Vision 的测试或开发，可以创建 7 天测试版本的许可证。

关于目前可用的产品许可证，请参考产品网站。

TF700x ｜ TC3 GigE Vision 连接器

TC3 GigE Vision 连接器能够连接 GigE Vision 相机。始终需要基本的 TF7000 许可证，并能连接两台相机。如果需要，可以用扩展许可证增加容量：

TF7000	基本许可证包括 2 个相机连接 （始终需要）
TF7001	2 个附加连接的扩展
TF7002	4 个附加连接的扩展
TF7003	8 个附加连接的扩展

扩展许可证可以合并，这样最多可以连接 2+2+4+8=16 台相机。

例如，11 台相机的连接需要以下许可证：

- TF7000 基本（包括 2 个相机连接）
- TF7001 2 个相机
- TF7003 8 个相机

➙ 现在总共可以连接 12 台相机。

TF7100-TF7800 | TC3 Vision 算法

TF7100-TF7800 产品许可在“Tc3_Vision”库中提供图像处理函数。对该库的任何使用均需要基本 TF7100 许可。它提供对基本算法的访问权限，如滤波操作、格式转换和轮廓跟踪。这已经使许多应用得以实现。扩展许可补充了表中所述的函数范围。函数与许可的分配可参见 API 参考 [▶ 410]。

TF7100	“Tc3_Vision”库的基本许可。包含用于解决图像处理任务的算法，如代数运算、滤波、傅立叶分析、彩色图像处理、分割、轮廓和 Blob 分析、结果显示以及用于读写相机参数的算法。
TF7200	用于匹配 2D 的扩展功能： 可根据学习到的引用、轮廓、特征点或其他属性（模板匹配/关键点检测和描述符匹配）查找和比较对象。常见的应用是对象分类。
TF7250	读码扩展： 检测和读取一维和二维码。
TF7260	OCR 扩展： 用于光学字符识别的函数，可识别图像区域中的字符并返回识别出的字符串。
TF7300	Metrology 2D 扩展： 用于光学测量几何对象属性（距离、半径等）的工具。 • 以亚像素精度对测量图像进行边缘定位 • 补偿光学几何畸变 （必须事先标定相机！） • 像素坐标与真实世界坐标之间的转换 （必须事先标定相机！）
TF7800	机器学习扩展： 用于培训和执行各种经典 ML 模型的函数。

自动确定所需 Vision 库许可证

只有在编译相应的 TwinCAT 项目时，才能确定所需的许可证。同时，进行检查，查看哪些功能块已声明以及哪些功能在代码中已使用。

因此，请注意，在项目变化的情况下，可能需要增加或省略所需的许可证（临时通知）。如果增加了新的许可证，必须通过**激活配置**加载更改；在这种情况下，无法进行在线更改。

3 安装

本章包含 TwinCAT Vision 的系统要求 [▸ 13]、版本概述 [▸ 13]、以及安装 [▸ 15]和授权 [▸ 20]说明。

3.1 系统要求

IPC 硬件

技术数据	描述
TwinCAT 3 平台等级	最低 P50 性能增强版，如 Intel 4 核 Atom CPU
网络适配器	TwinCAT 实时以太网兼容千兆网卡（用于连接 GigE Vision 相机） **至少应有一个专供相机使用的网络端口。建议每个相机使用单独的网络端口。**
主内存	纯运行时系统建议至少 4GB 开发系统建议至少 8GB 对于在运行时系统上使用记录/回放 [▸ 89]数据流的情况，建议至少 8GB
硬盘	建议使用读/写速率高的 SSD 硬盘，尤其是在 PLC 和硬盘之间需要交换大量数据的情况。

相机

技术数据	描述
接口	必须支持 GigE Vision [▸ 107]。（标有 GigE Vision 徽标） **TwinCAT GigE Vision 连接器已根据 GigE Vision 标准版本 2.0 获得认证。但这并不意味着支持该版本的所有相机或 GigE Vision 功能。**

软件

技术数据	描述
操作系统	Windows 10（仅 64 位！）
支持的 Visual Studio 版本	2017（社区版，专业版） 2019（社区版，专业版） 2022（社区版，专业版），TwinCAT 3.1 4026 及以上 TwinCAT XAE Shell (TcXaeShell) TwinCAT XAE Shell (TcXaeShell64)，TwinCAT 3.1 4026 及以上
TwinCAT	最低版本 3.1.4024.44
.NET 框架	最低版本 4.7.2（仅用于开发环境 [▸ 48]）

TwinCAT HMI Server 扩展

技术数据	描述
.NET	最低版本 .NET Desktop Runtime 6.0

3.2 版本概览

设置	注释
4.0.4.8	增加了 C++ API、OCR 模型和机器学习。
4.0.3.5	删除了在编译过程或激活配置期间对初始化命令的自动检查。
4.0.2.13	
4.0.1.3	安装更新设置后会删除存储库文件。 **如果您想保留该版本，请在安装新设置之前保存“%TWINCAT3DIR%\3.1\Repository\Beckhoff Automation GmbH”文件夹，并在安装后将其复制回去。**

注意

兼容性

如果在一个系统上安装了多个 TwinCAT Vision 版本，或者开发和运行时环境位于不同的系统上，则必须确保使用的所有组件版本均相同。

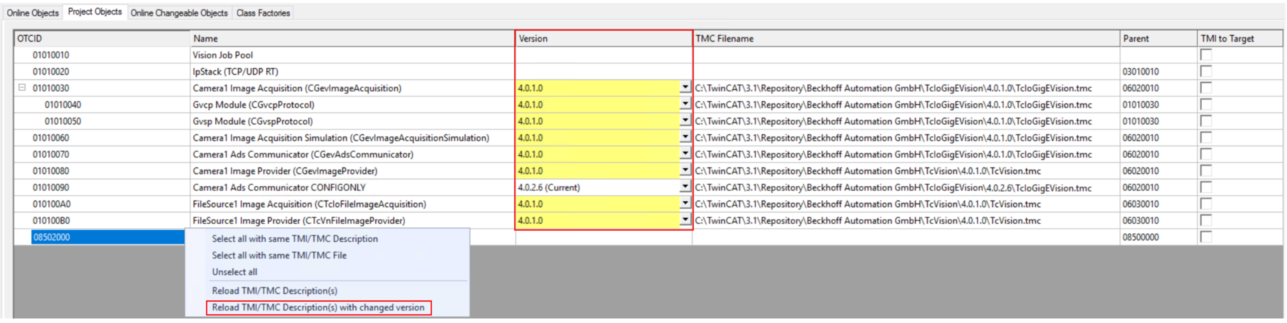
TwinCAT Vision Service 和工程扩展仅在最新版本中提供。因此，这些组件决定了要使用的 TcCOM 和库版本。不允许混合版本，否则会导致故障。

版本选择提示

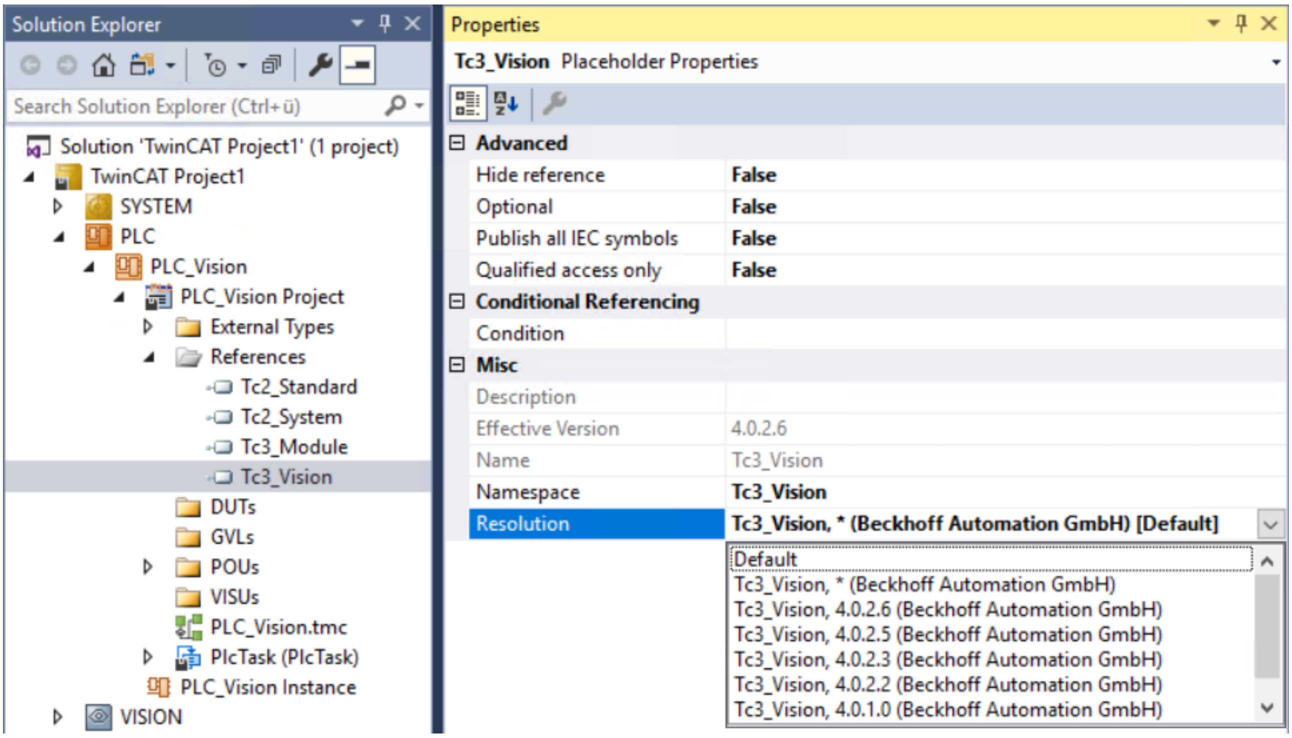
如果要使用特定版本的 TwinCAT Vision，则必须将项目的所有 TcCOM 对象更新到相应版本。Ads Communicator CONFIGONLY 是个例外。该模块仅在配置助手内部使用，必须始终拥有与扩展相匹配的最新版本。因此，该模块会在加载项目时自动设置最新版本。

要检查或更改 TcCOM 版本：

1. 打开 SYSTEM（系统） > TcCOM Objects（TcCOM 对象） > Project Objects（项目对象）窗口。
2. 要更改，请为所有对象选择所需版本。
3. 右键点击第一列中的一个对象 ID。
4. 选择 Reload TMI/TMC Description(s) with changed version重新加载 TMI/TMC 说明以及更改版本。



此外，PLC 项目中必须使用与所选 TcCOM 对象版本相对应的相同 Tc3_Vision 库版本。因此，在 Tc3_Vision 库的属性中检查有效使用的版本，或对其进行相应调整。



TwinCAT Vision Service 的当前版本号以及工程扩展的当前版本号可在 Vision 节点下的 服务配置 [▶ 59] 中找到。

注意

更新后的在线更改

如果在开发系统上更新 TwinCAT Vision，从而在使用旧版本的目标系统上进行了在线更改，这可能会导致接口指针（图像和容器）版本冲突。

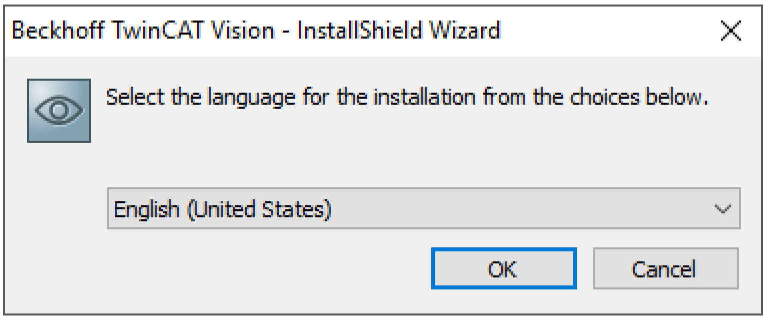
删除 TwinCAT 3.1 Build 4024 以下的旧版本

要删除旧版本，需要两个步骤。首先，您必须删除“%TWINCAT3DIR%\3.1\Repository\Beckhoff Automation GmbH”下的 Tc3_Vision、TcIoGigEVision 和 TcVision 目录中的相应版本文件夹。要删除相应的 PLC 库版本，必须通过库存储库对话框将其卸载。

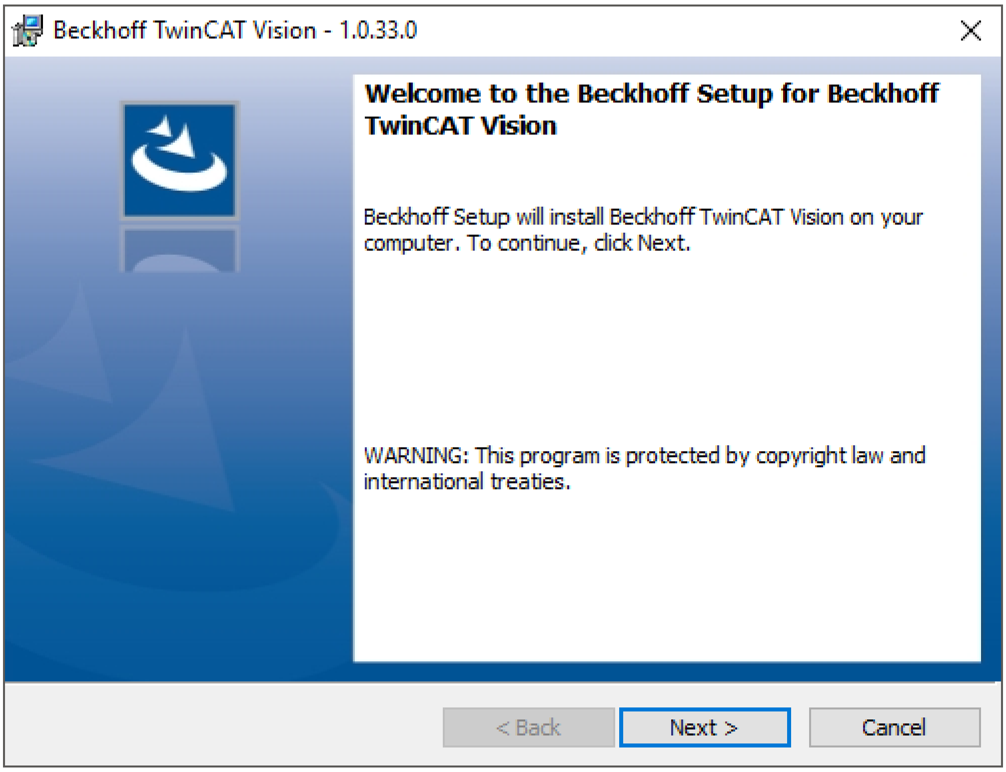
3.3 TwinCAT 3.1 Build 4024 安装说明

在 Windows 操作系统上安装 TwinCAT Vision 包含以下步骤：

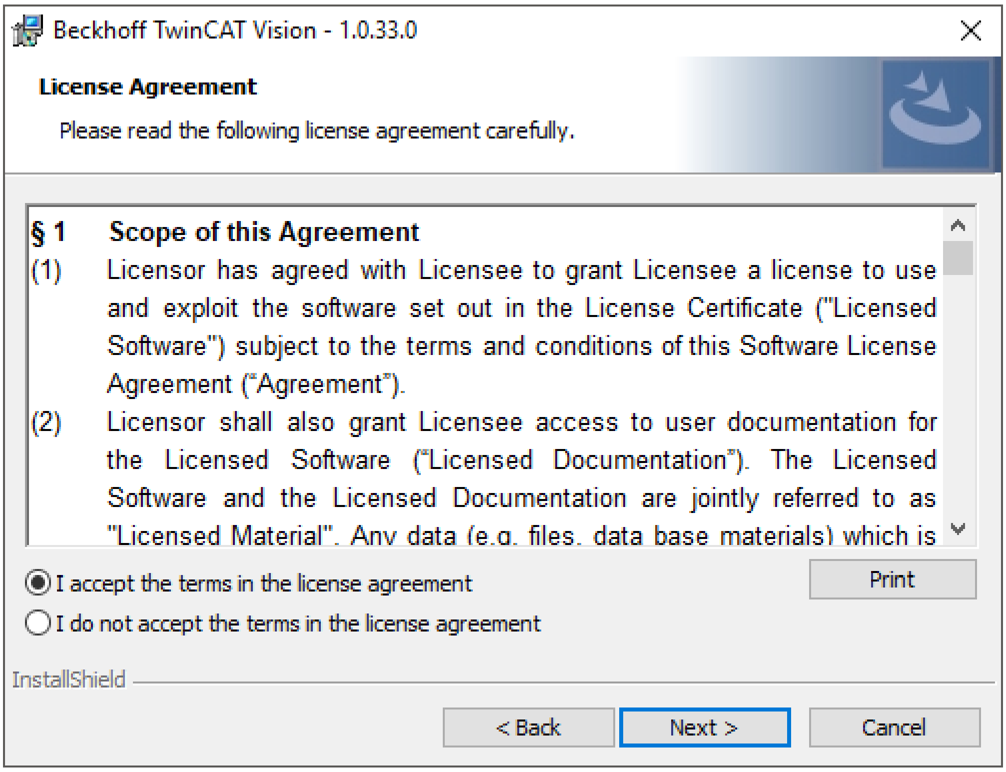
1. 确保 TwinCAT 处于配置模式，且未打开任何 Visual Studio 实例或 TcXaeShell。
2. 以管理员身份运行安装文件 TF7xxx-Vision.exe。为此，请右键点击文件并选择 **Run As Admin（以管理员身份运行）**。点击 **Yes（确定）**确认。
3. 选择安装语言。然后点击 **OK（确认）**。



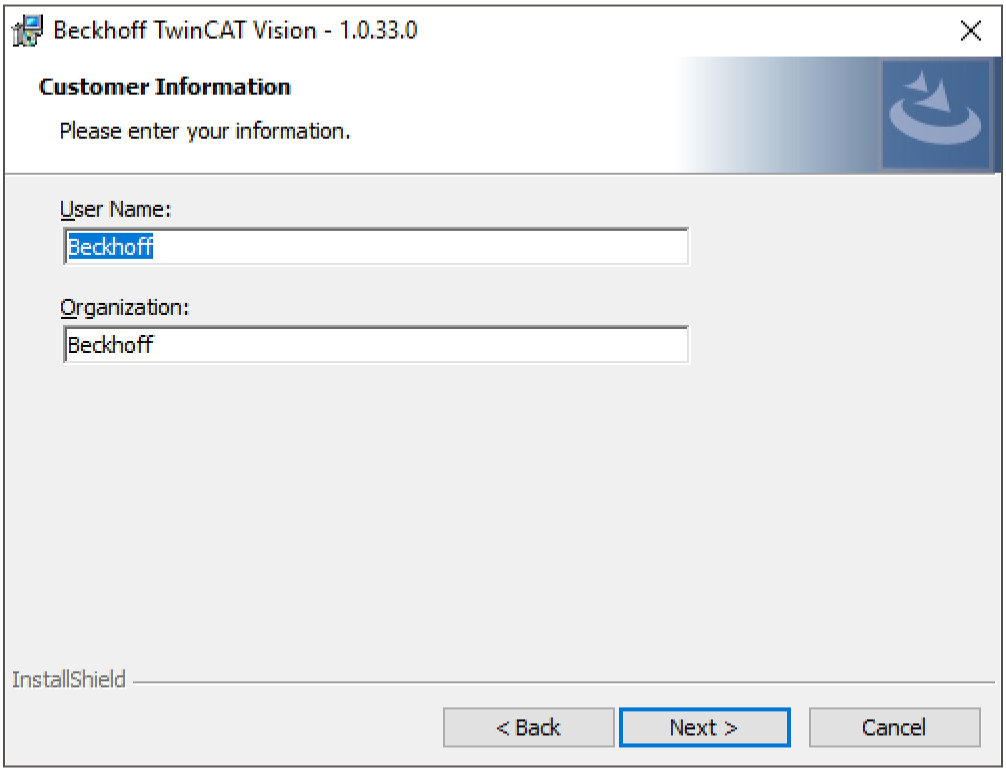
4. 等待安装准备过程结束，然后点击 Next（下一步）。



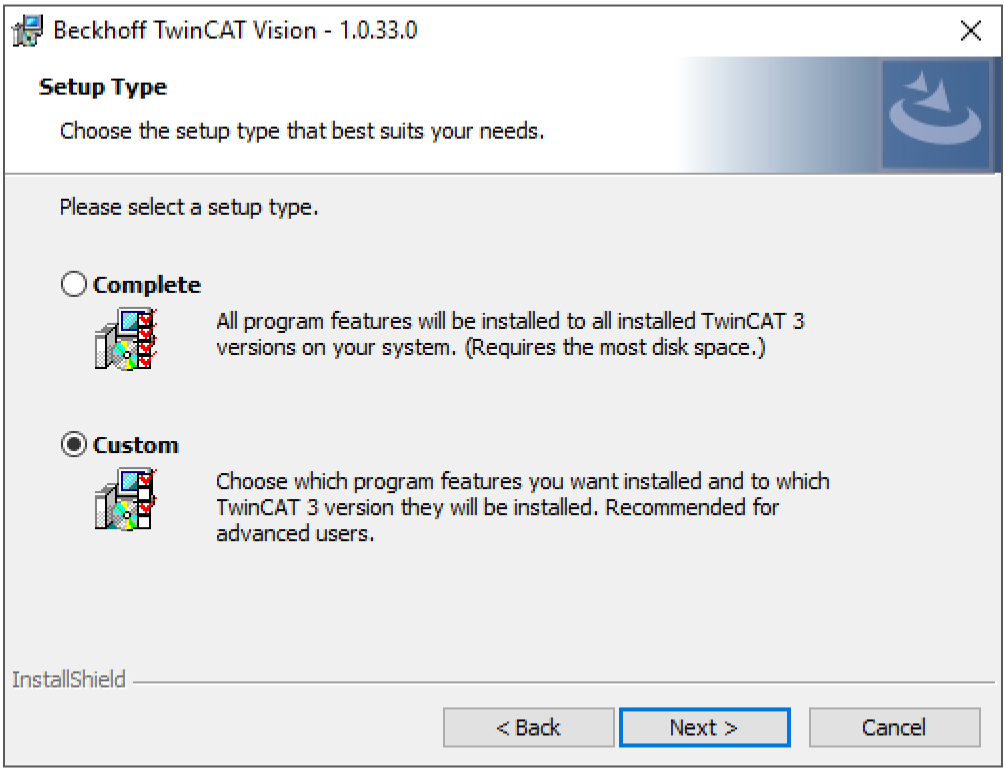
5. 通过勾选 I accept the terms of the license agreement（我接受许可协议的条款）选项，确保您同意许可协议。如有需要，您可以打印许可协议。然后点击 Next（下一步）。



6. 输入用户名和公司作为用户信息。然后点击 Next（下一步）。

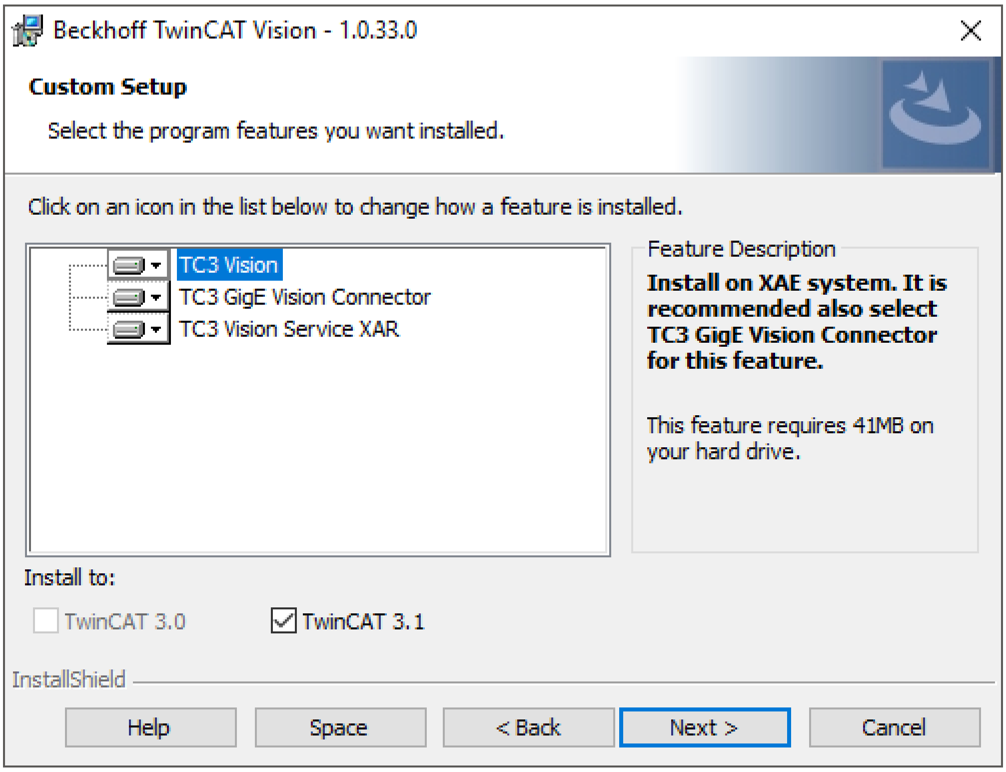


7. 在 Complete（完整）或 Custom（自定义）两种设置类型中进行选择。然后点击 Next（下一步）。



— 完整设置会安装所有 TwinCAT Vision 组件，需要最多的存储空间。建议 XAE 系统选择该选项。由于纯运行时系统上只能安装 Vision Service，其他选项无法选择，因此这种情况也推荐选择该选项。
— 自定义设置可让您在下一步中选择要安装的 TwinCAT Vision 组件。

8. 对于下列每个 TwinCAT Vision 组件，请选择是否安装。此外，请选择至少一个待安装所选组件的 TwinCAT 3 版本。然后点击 Next（下一步）。



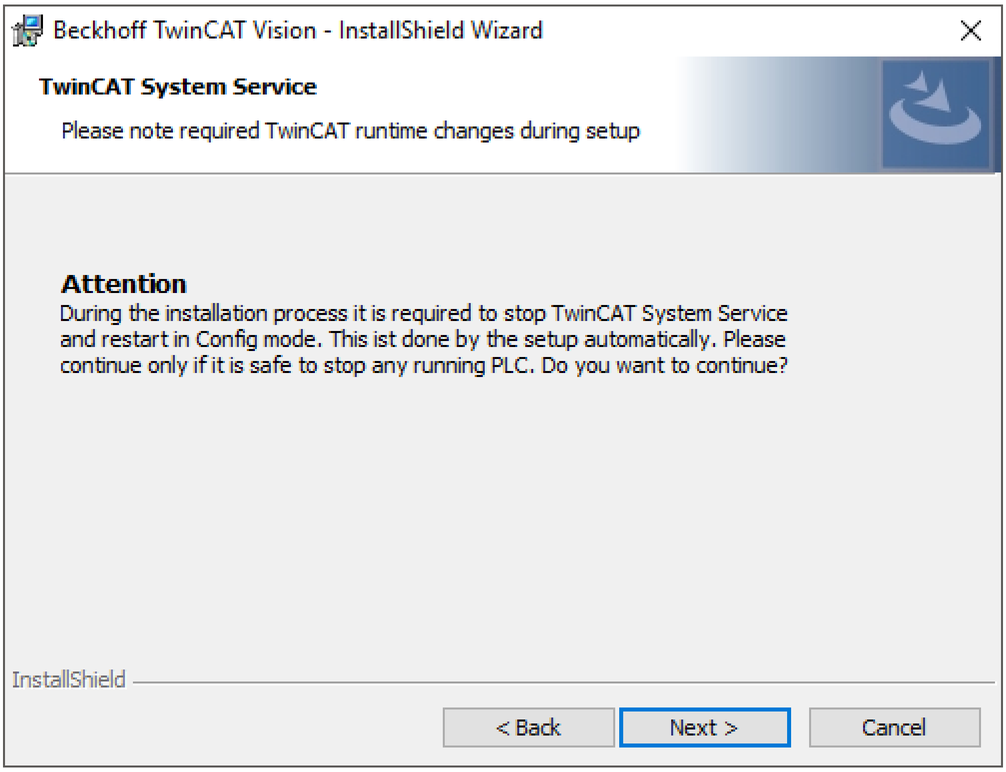
— TC3 Vision：为开发 TwinCAT Vision 应用所必需，并应安装在 XAE 系统上。

— TC3 GigE Vision 连接器：为使用 GigE Vision 设备进行开发所必需，并应安装在 XAE 系统上。
— TC3 Vision Service XAR：包含 TwinCAT Vision Service，且必须安装在 TwinCAT Vision 应用的目标系统上。

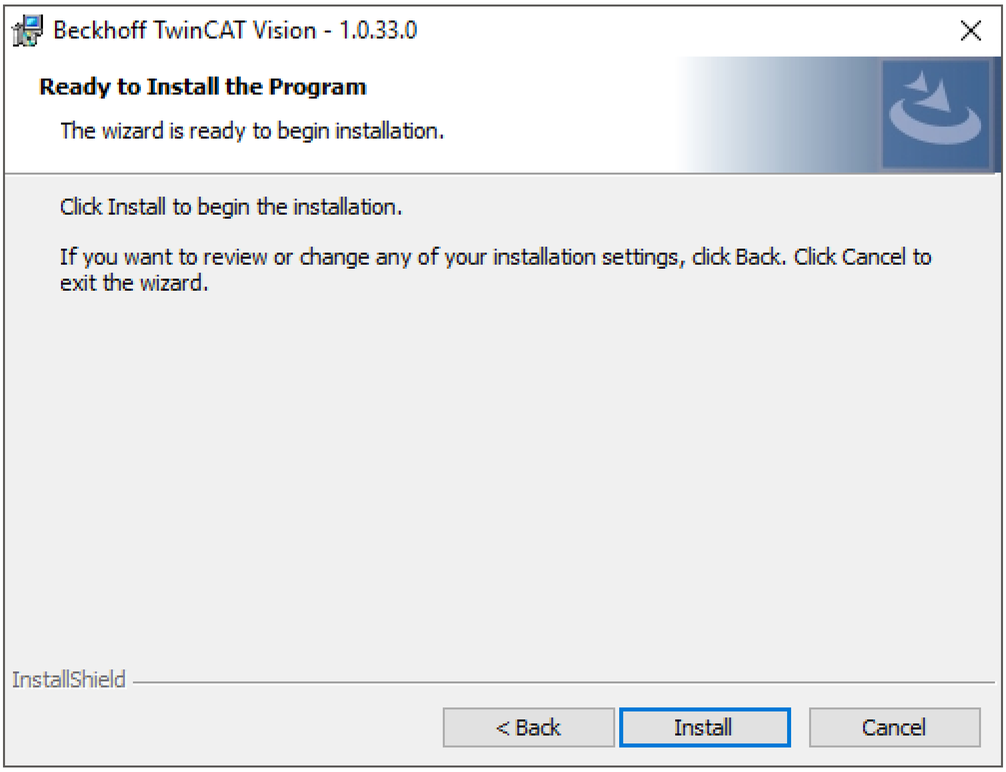
说明

目标系统上必须安装 Vision Service。相反，在激活 TwinCAT 项目时，TC3 Vision 和 TC3 GigE Vision 连接器的组件总是自动从开发系统复制到目标系统。因此，它们并不一定要安装在目标系统上。

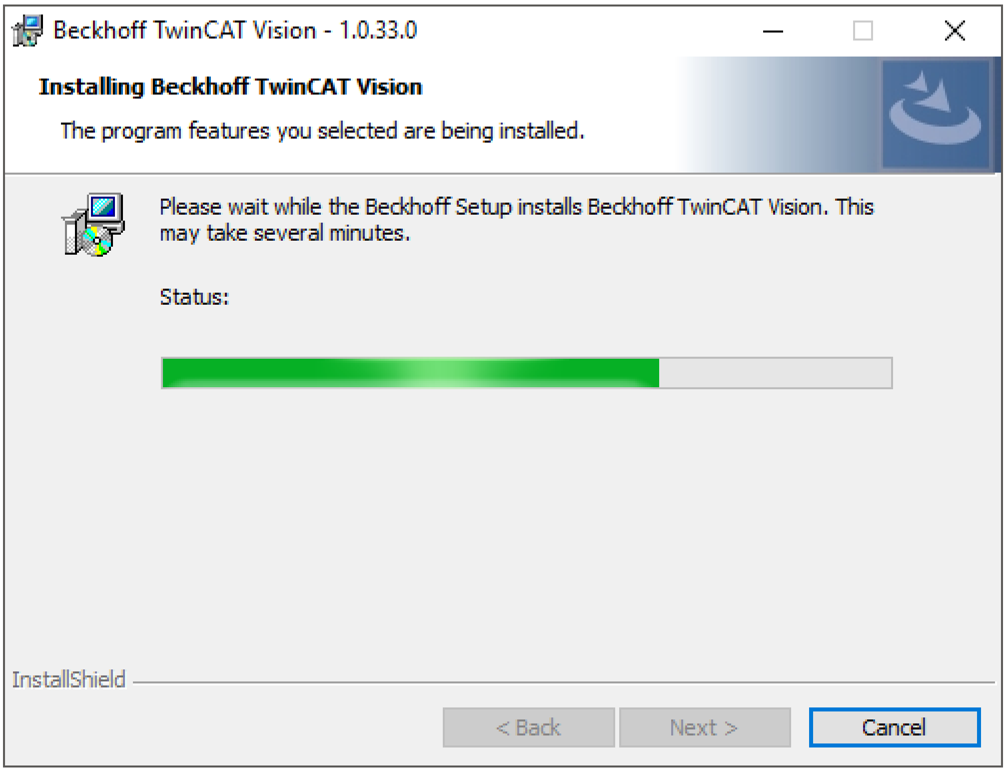
9. 请注意，在接下来的安装过程中，TwinCAT 系统服务将停止并以配置模式重新启动。只有当运行中的 PLC 系统可以停止时，才能继续安装。在这种情况下，点击 Next（下一步）。



10. 确保您同意所选的安装设置。然后点击 Next（下一步）。

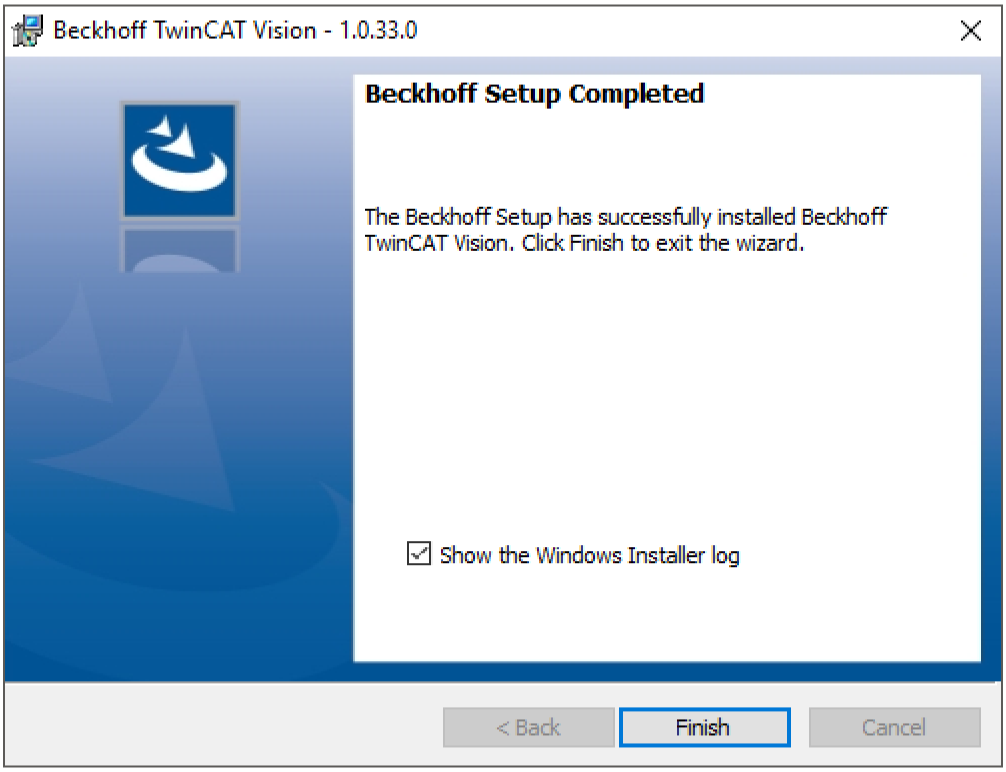


11. 等待所选 TwinCAT Vision 组件安装完毕。



12. 可能会出现一个对话框，显示为完成安装而必须关闭的应用列表。选择自动关闭应用还是在安装后重启 PC。

⇨ 安装现已完成。可以选择显示安装的日志文件。点击 Finish（完成）。



TwinCAT Vision 安装成功后必须先获得许可方可使用。该程序在许可 [▶ 20]一章中有所描述。

有关版本和更新现有项目的信息，请访问 版本概览 [▶ 13]。

如果安装失败，您有以下选择：

- 使用故障排除 [▶ 2737]解决问题。
- 联系客户支持部并提供安装日志。安装日志 *TF7xxx-Vision.exe.log* 位于路径：*C:\Users\<username>* 中的以下文件夹中

3.4

3.5 授权

TwinCAT 3 功能可以作为完整版或 7 天测试版激活。这两种许可证类型都可以通过 TwinCAT 3 开发环境（XAE）激活。

授权使用 TwinCAT 3 功能的完整版本

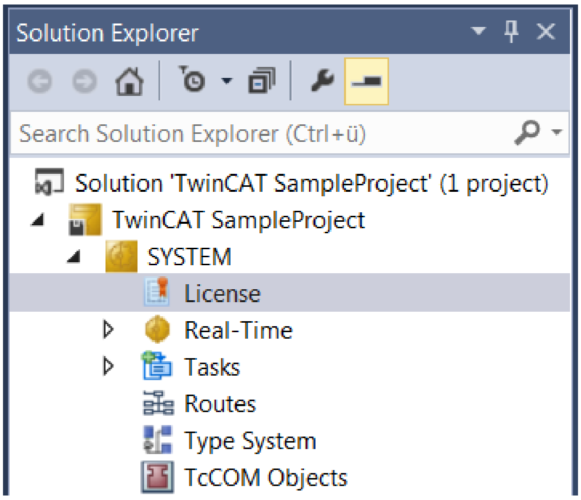
关于许可证完整版本的程序描述，可参见倍福信息系统的文档“TwinCAT 3 授权”。

授权一个 TwinCAT 3 功能的 7 天测试版本

对于TwinCAT 3 许可证加密狗无法启用 7 天测试版本。

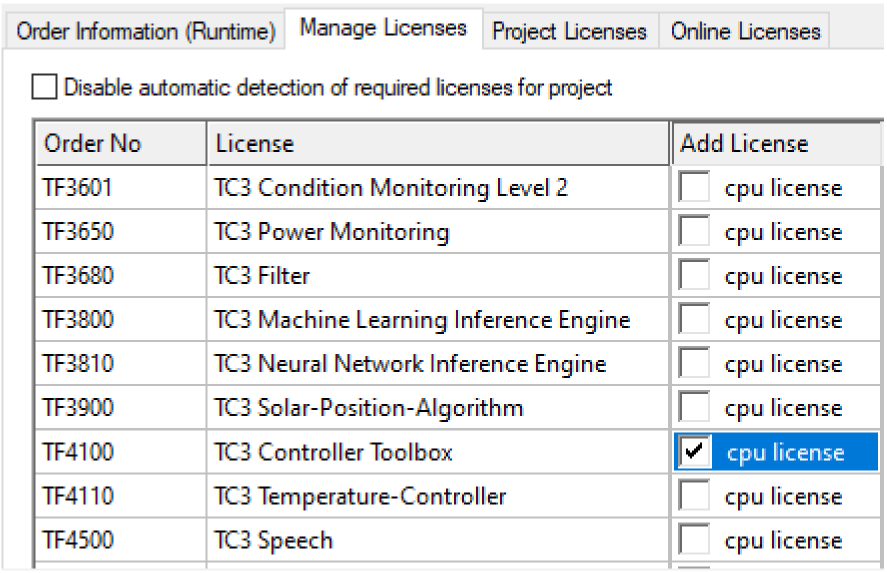
1. 启动 TwinCAT 3 开发环境（XAE）。
2. 打开一个现有的 TwinCAT 3 项目或创建一个新项目。
3. 如果想为一个远程设备激活授权，请设置所需的目标系统。为此，从工具栏的**选择目标系统**下拉列表中选择目标系统。
   ⇨ 授权设置总是指选定的目标系统。在目标系统上激活项目时，自动将相应的 TwinCAT 3 许可证复制到该系统中。

4. 在**解决方案资源管理器**中，双击**系统**子目录的**许可证**。



⇨ TwinCAT 3 许可证管理器打开。

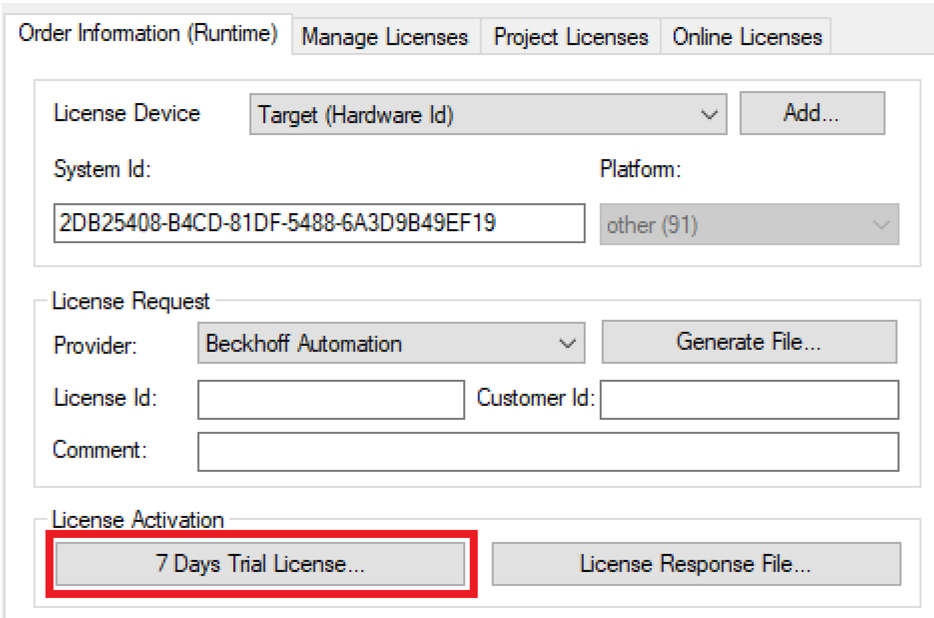
5. 打开**管理许可证**选项卡。在**添加许可证**栏中，选中希望添加到项目的许可证的复选框（例如“TF4100 TC3 控制器工具箱”）。



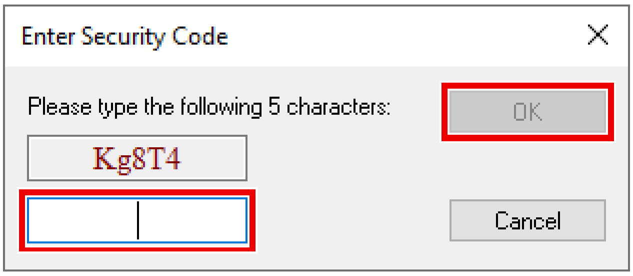
6. 打开**订单信息（运行时间）**选项卡。

⇨ 在许可证的表格概览中，以前选择的许可证显示为“缺失”状态。

7. 点击7 天试用许可证...，以激活 7 天试用许可证。



⇨ 一个对话框打开，提示输入对话框中显示的安全代码。



8. 准确输入显示的代码并确认输入。
9. 确认随后的对话框，表明激活成功。
   ⇨ 在许可证的表格概览中，许可证状态现在显示许可证的到期日。
10. 重新启动 TwinCAT 系统。

⇨ 7 天试用版启用。

4 第一步

以下指南对 TwinCAT Vision 进行了介绍（成功安装 [▶ 13]后）。

关于更多详细信息，可参见开发环境 [▶ 48]、API 参考 [▶ 119]和样本 [▶ 2639]。

4.1 构建项目

为 TwinCAT Vision 设置系统的视频示例：

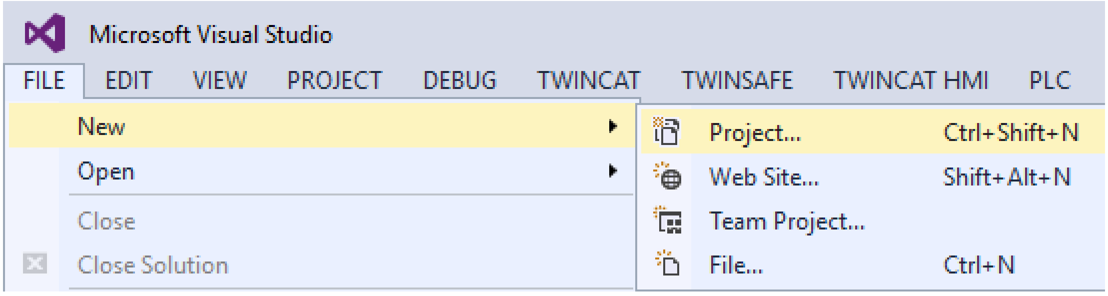
视频：https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/Resources/5906036107/.mp4

4.1.1 创建 TwinCAT 项目

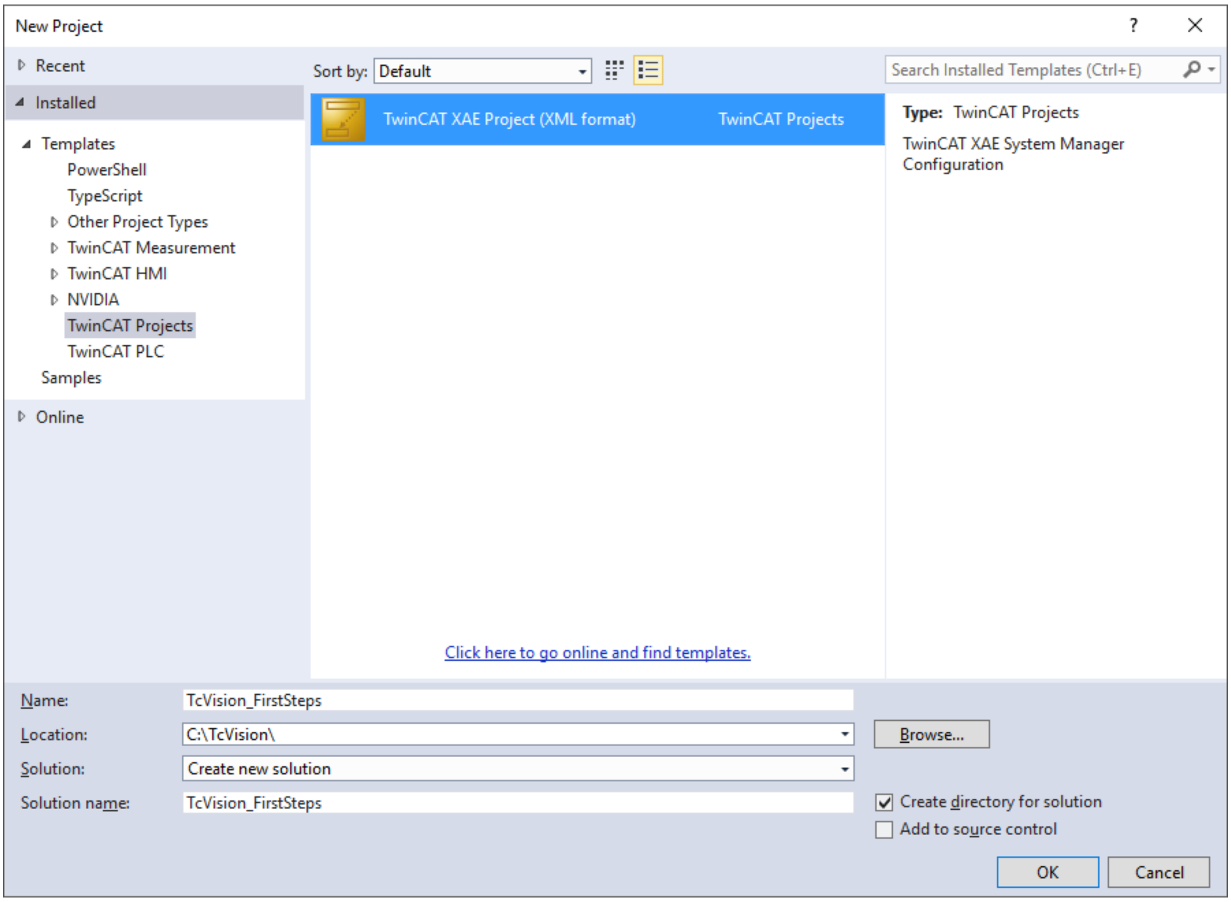
TwinCAT Vision 完全集成在 TwinCAT 系统中。因此，一个标准的 TwinCAT 项目是每个 TwinCAT Vision 应用的出发点：

1. 通过集成 TwinCAT 打开 Visual Studio。

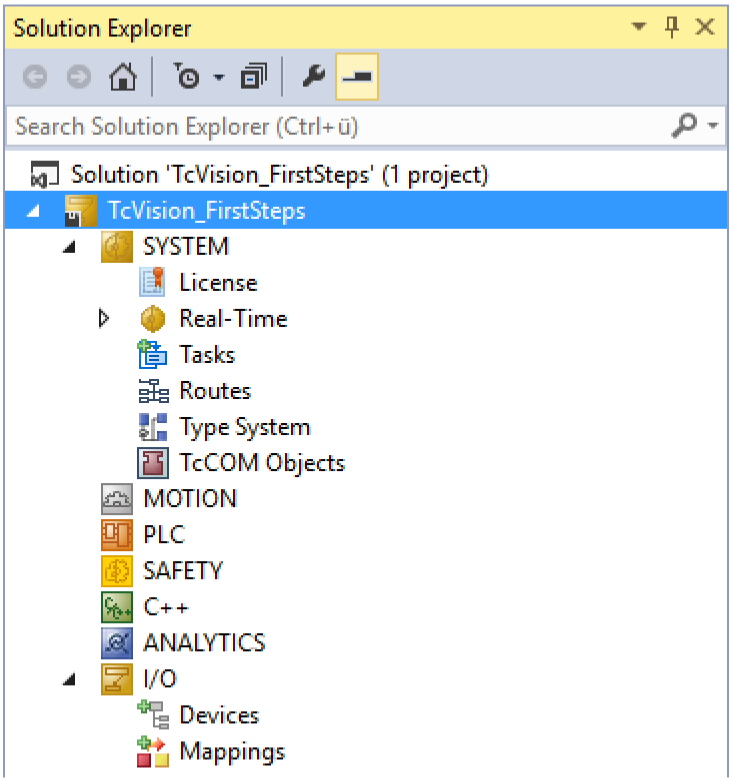
2. 打开新/项目对话框。



3. 选择名称和地点。



⇨ TwinCAT 项目已创建。



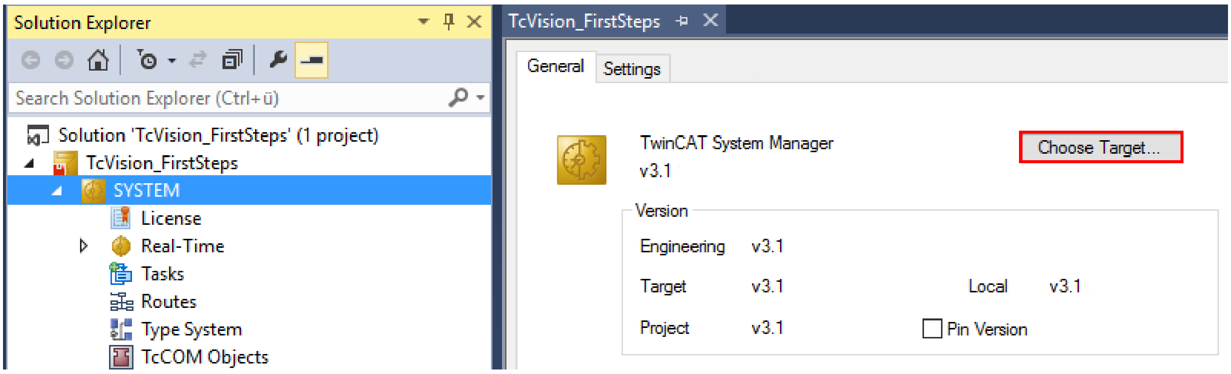
下一步

选择目标系统 [▶ 25]

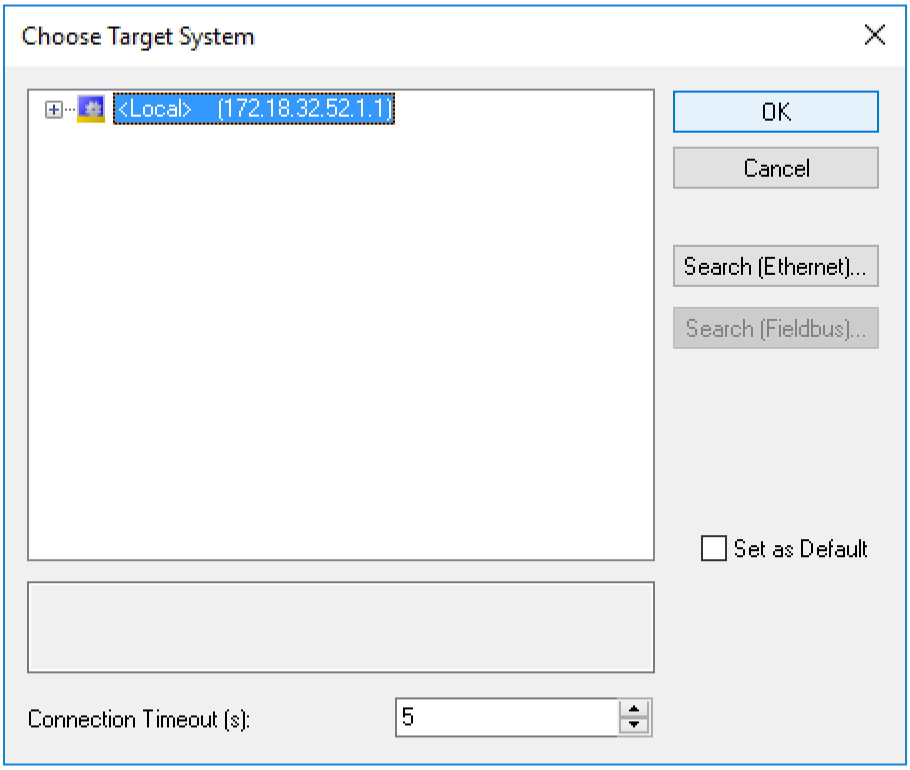
4.1.2 选择目标系统

TwinCAT 区分了开发系统和运行时间系统。如果你的开发系统不是你的运行时系统，创建一个通往运行时系统的路线并选择路线。例如，这可以通过以下步骤完成：

1. 点击**解决方案资源管理器** > **系统** > **通用** > **选择目标...**
   选择目标系统对话框打开，其中列出了所有从开发系统中存在路线的目标系统。

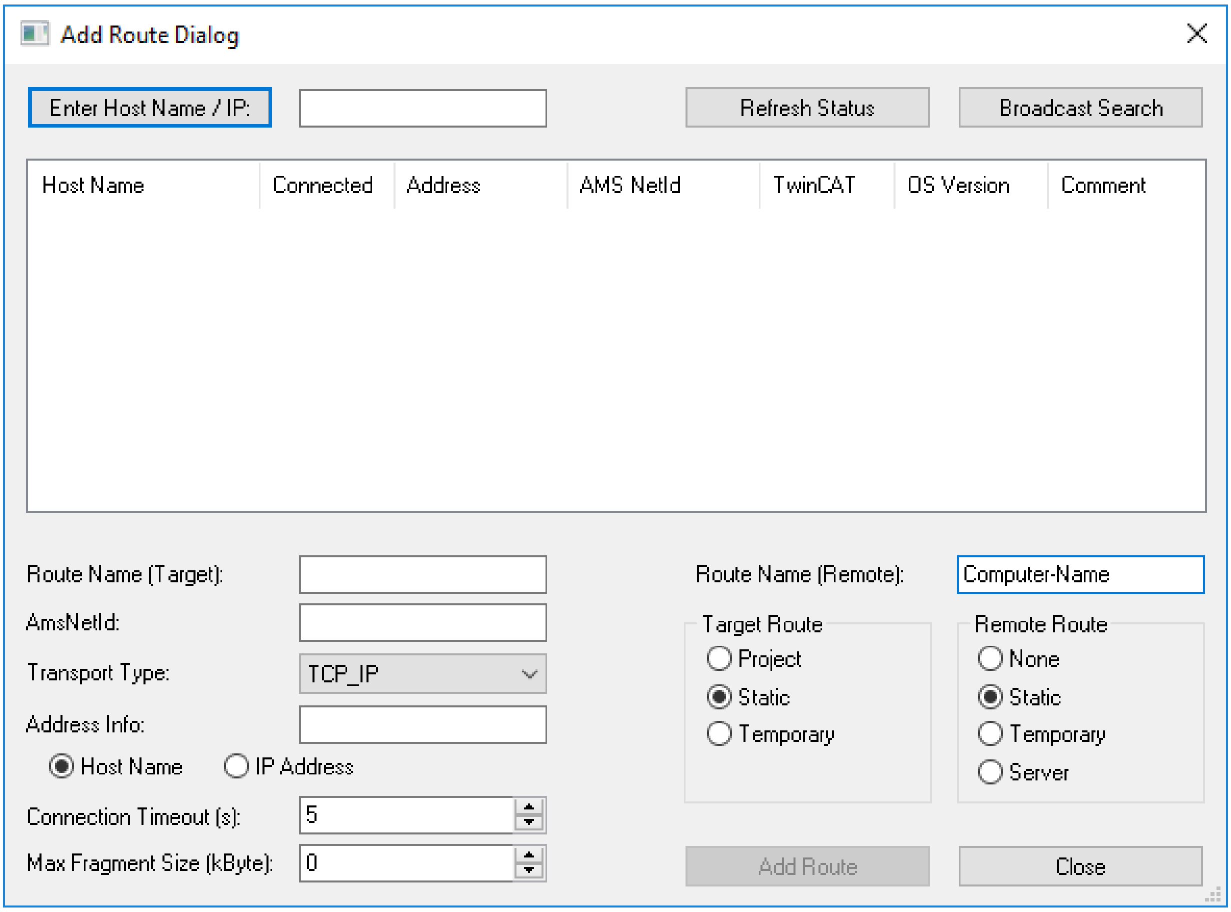


2. 如果目标系统尚未列出，即不存在通往系统的路线，点击**搜索**。

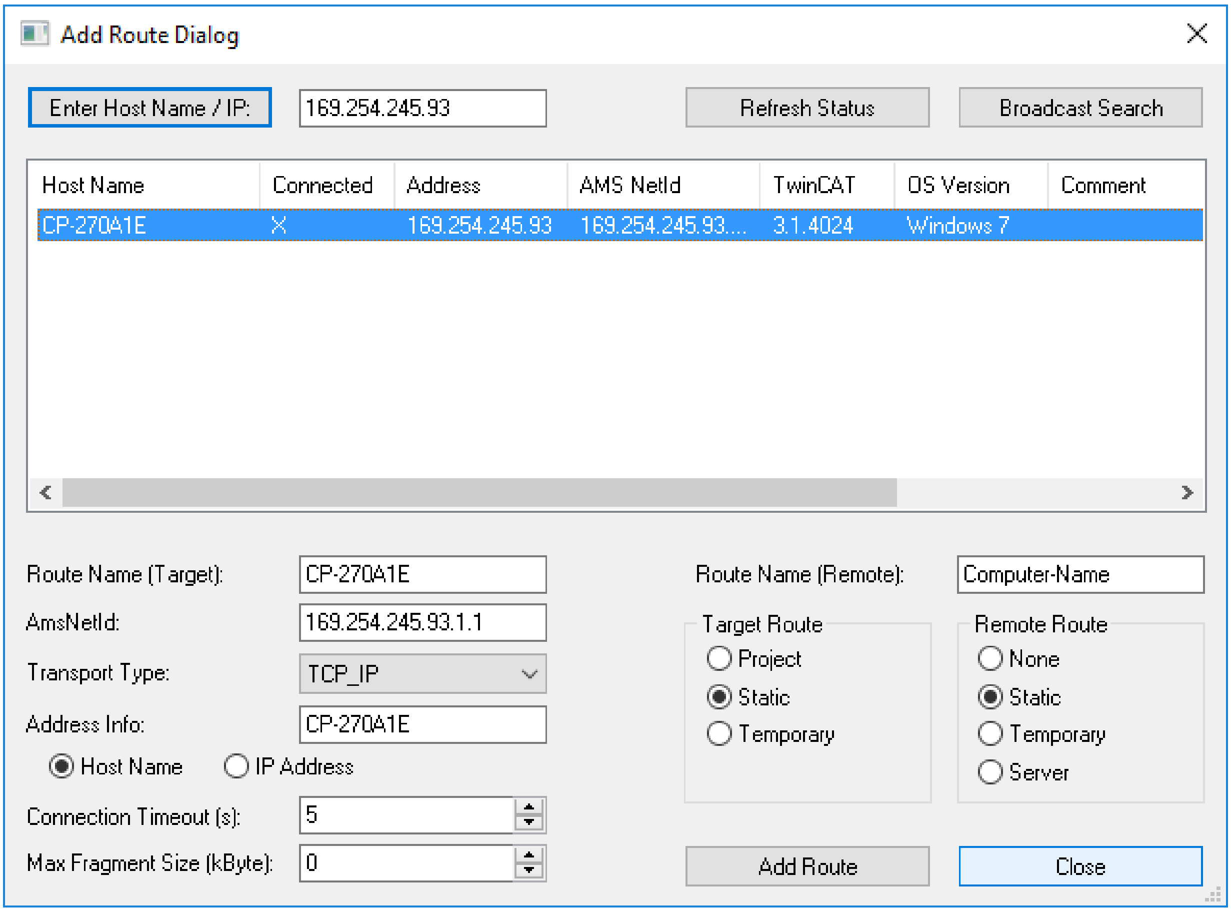


“添加路线”对话框打开，而在这里可以添加更多路线。

3. 在**添加路线对话框**中，点击**广播搜索**，以搜索网络中的 TwinCAT 系统。或者，通过**输入主机名称/IP**，搜索设备名称或 IP 地址。然后显示找到的任何系统。



4. 从列表中选择一个系统，然后点击**添加路线**，为其创建一个路线。显示的信息和设置选项在添加路线对话框章中有详细描述。新路线现在出现在**选择目标系统**对话框中。



5. 在**选择目标系统**对话框中选择目标系统，并通过**确定**确认。

下一步

系统配置 [▶ 27]

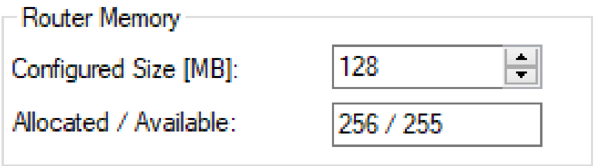
4.1.3 系统配置

系统配置取决于目标系统，详见 系统配置 [▶ 48] 章节。

增加路由器内存

例如，TwinCAT Vision 在创建图像时需要动态内存分配。这在路由器内存 [▶ 49]中进行。因此，必须配置足够大的路由器内存。

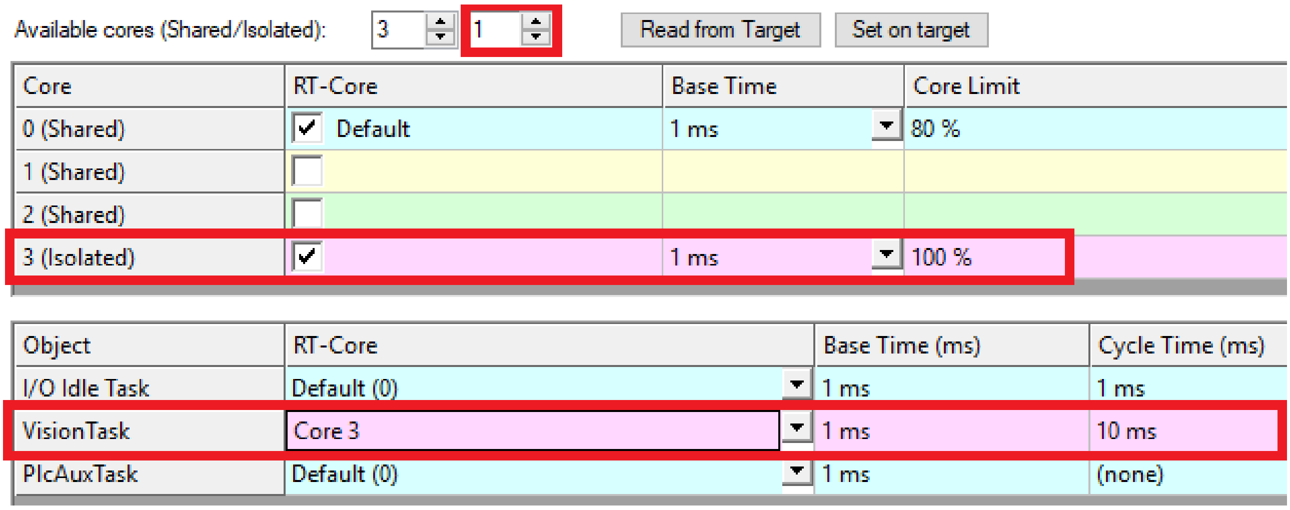
这可以在 TwinCAT 项目中的 SYSTEM（系统） > Real-Time（实时） > Settings（设置） > Router Memory (Mbyte)（路由器内存，兆字节）下完成。



然后必须激活配置  。如果路由器内存减少，则必须重新启动目标系统。

选择 CPU 核和任务

在多核系统中，TwinCAT 支持区分 Windows 系统分享核和独立核。使用 Windows 核时，操作系统和 TwinCAT 应用共享处理器时间。实时部分可限制为 10% 到 90%。与此相反，独立核可完全用于 TwinCAT 应用。因此，建议 Vision 应用使用独立核。在初始测试中，也可以使用分核而没有任何问题。



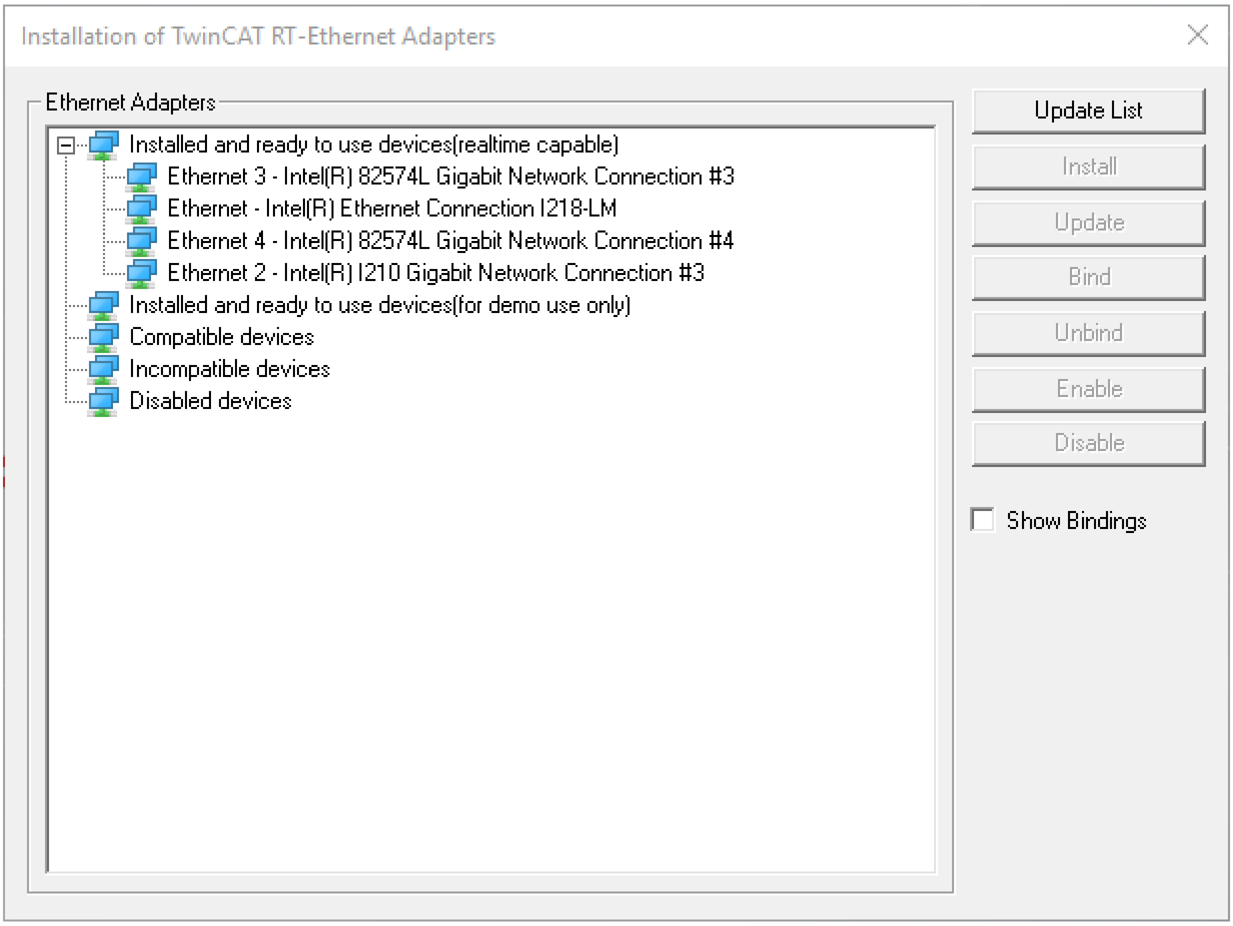
核配置在 TwinCAT 项目树中的 **SYSTEM（系统）** > **Real-Time（实时）** > **Settings（设置）**下进行。详细配置说明见 CPU 内核和任务 [▸ 55] 章节。本章还介绍了如何更改执行任务的循环时间。

然后必须激活配置  。此外，在重新定义分核和独立核时，必须重新启动目标系统。

安装实时以太网驱动程序

要将 GigE Vision 相机连接到 TwinCAT Vision，必须在所使用的网络适配器上安装 TwinCAT RT 以太网驱动程序。该驱动程序可确保 TwinCAT 在运行时期间使用网络适配器。详情请参见 网络适配器 [▸ 50] 章节。

在开发环境中，可通过 **TWINCAT > Show Realtime Ethernet Compatible Devices...（显示实时以太网兼容设备……）**调用安装助手。在目标系统上，可以通过文件 *C:\TwinCAT\3.1\System\TcRteInstall.exe* 启动该助手。可为打开对话框中的 **Compatible devices（兼容设备）**下所列的网络适配器安装 RT 以太网驱动程序。



确保要连接相机的所有网络适配器现在都在**已安装即用设备（支持实时功能）**列表中。

下一步

显示视觉节点并创建应用程序 [▶ 29]

4.2 设置视觉设备

使用文件源控制的视频示例：

视频：https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/Resources/5931182859/.mp4

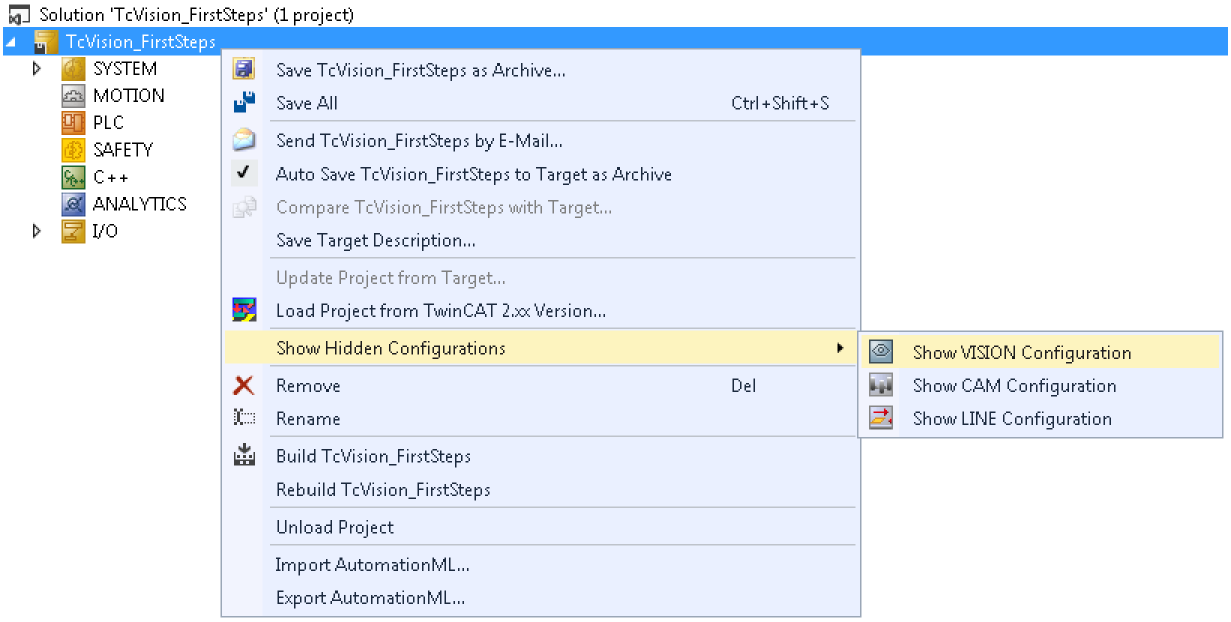
创建相机实例的视频示例：

视频：https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/Resources/5931185163/.mp4

4.2.1 显示视觉节点并创建应用程序

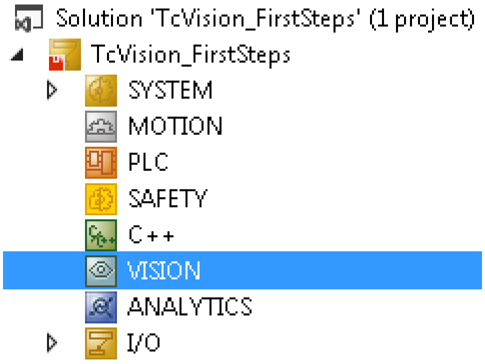
在 TwinCAT 项目中有一个可用于 Vision 配置的 Vision 节点。

1. 如果 TwinCAT 项目中未显示 Vision 节点，请右键点击 Solution Explorer（解决方案资源管理器）>Show Hidden Configurations（显示隐藏配置） > Show VISION Configuration（显示 VISION 配置）

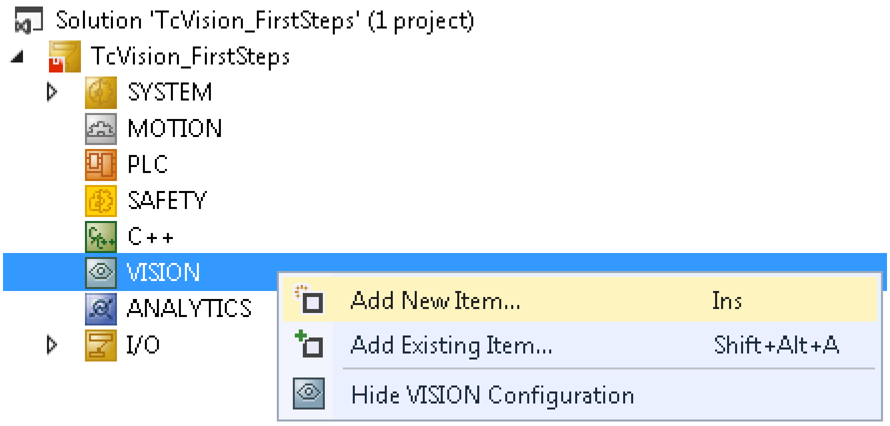


中的 TwinCAT 项目。

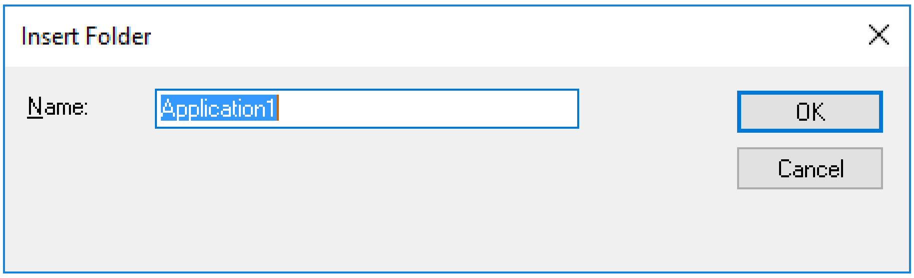
⇨ Vision 节点现在可见。



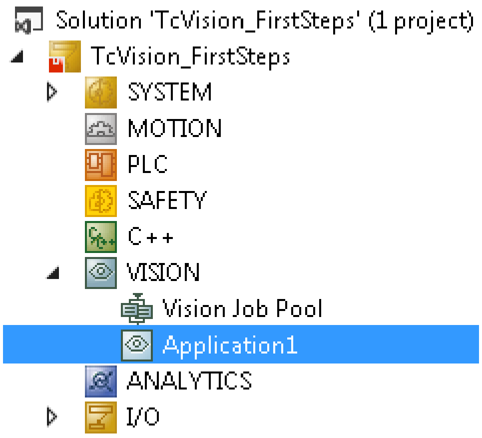
2. 在 Vision 节点内，文件夹（也称为应用）用于对图像处理设备进行分组。
右键点击 Vision Node（Vision 节点） > Add New Item（添加新项目）。



3. 选择一个名称并使用 OK（确定）确认。



⇨ 文件夹已创建



此外，您还可以在此找到更多信息：

- 视觉节点 [▶ 59]
- 应用节点 [▶ 62]

下一步

下一步可创建一个可用于将图像加载到 PLC 的 Vision 设备。既可以是真正的 GigE Vision 相机，也可以是用于加载现有图像文件的文件源控制器：

创建 GigE Vision 相机 [▶ 34] 或者 创建文件源控制 [▶ 31]

4.2.2 创建文件源控制

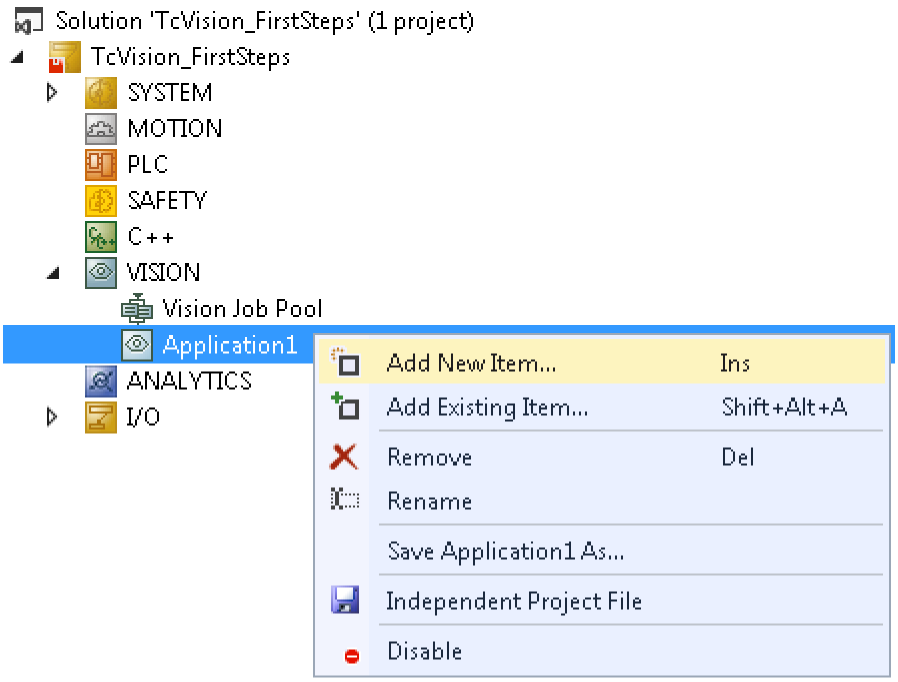
以下步骤中描述了创建文件源 [▶ 110]对象。这可以用来将存储的图像加载到 PLC 中。为此，目标系统必须处于配置模式。点击 TwinCAT XAE Base 工具栏中的符号  或点击TWINCAT 下 Visual Studio 菜单中的重新启动 TwinCAT（配置模式）。

或者，可以创建 GigE Vision 相机 [▶ 34]。

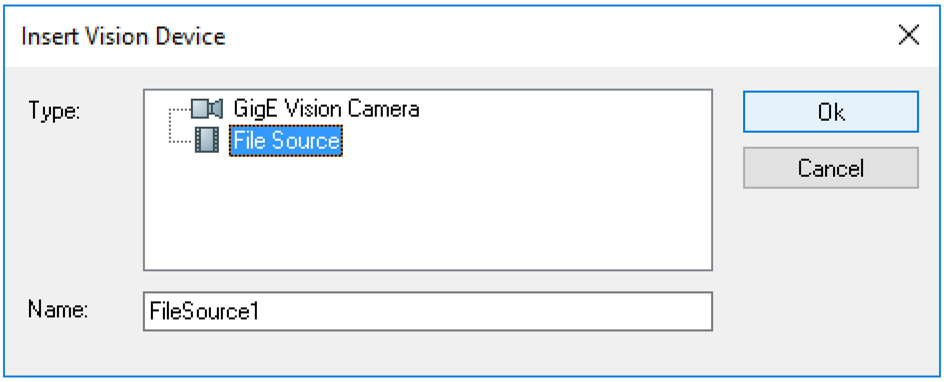
按以下步骤创建文件源对象：

1. 点击之前创建的视觉节点下的应用程序。

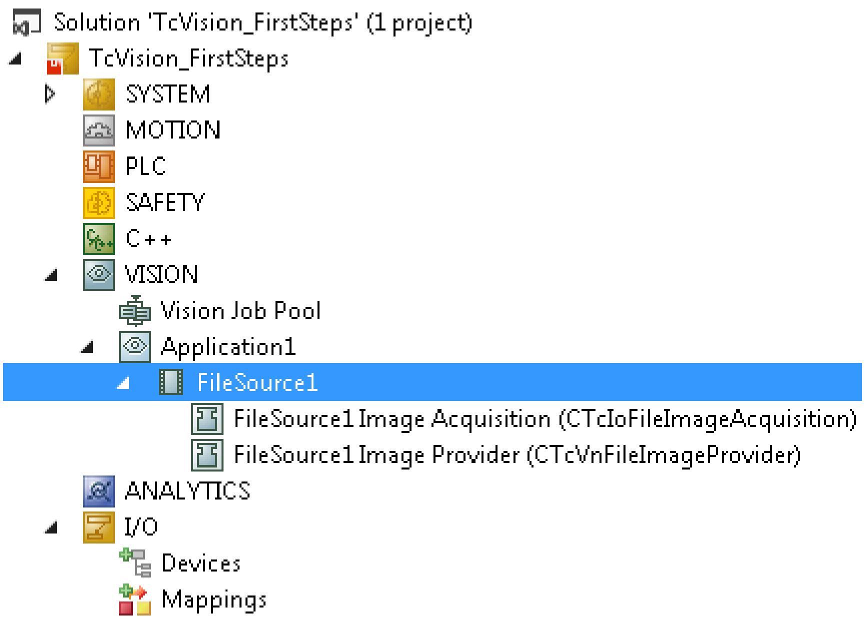
2. 右键单击并选择**添加新项目...**



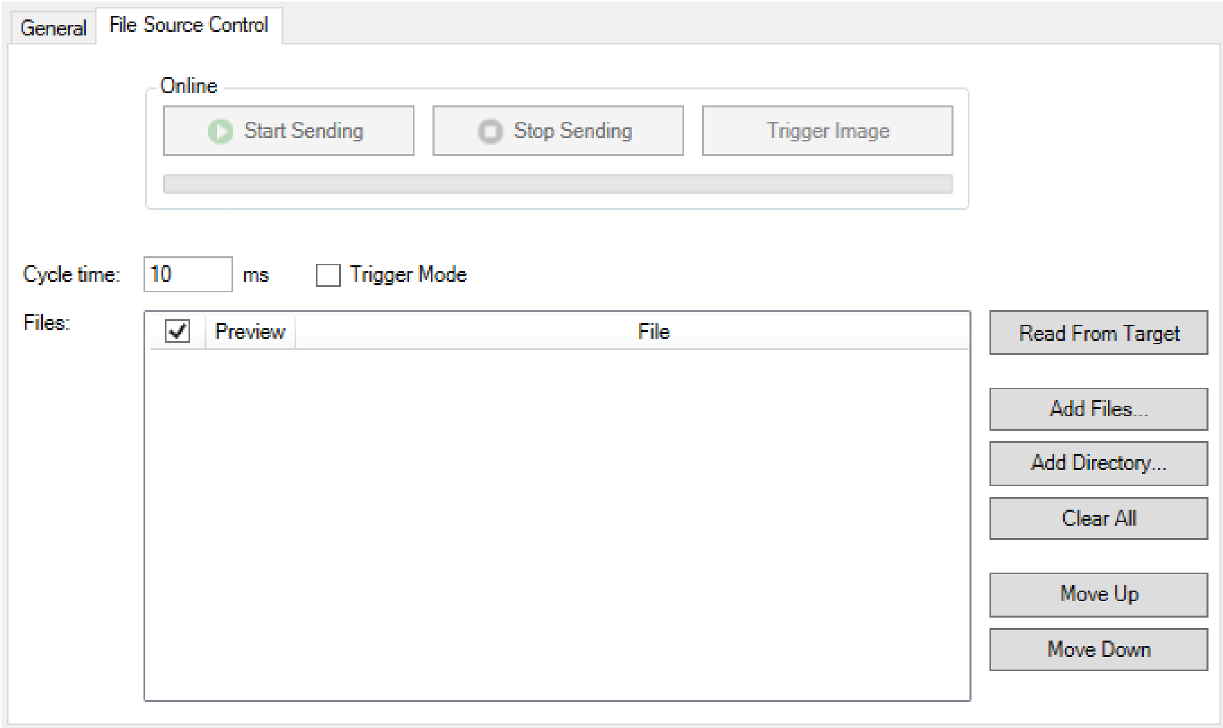
3. 选择设备类型**文件源**，必要时调整名称，点击**确定**。



⇨ 现在，文件源控制已经在应用程序下面创建，且可以使用。

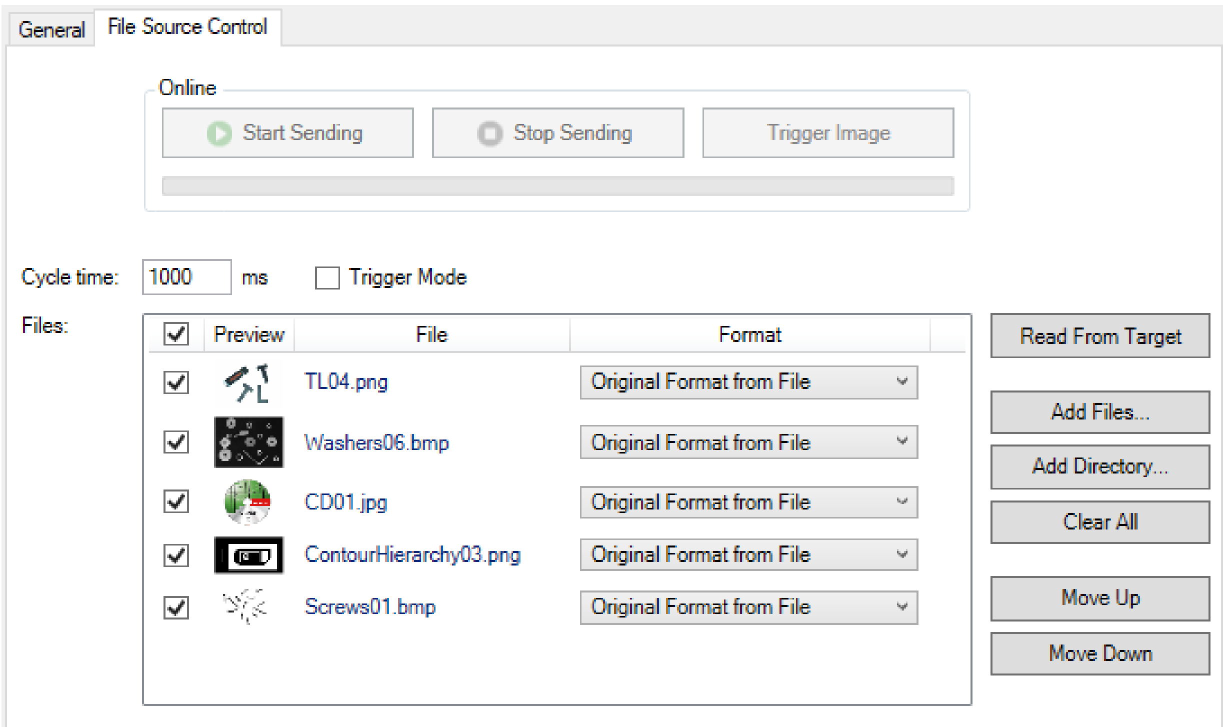


4. 双击项目树中应用程序节点下的**文件源对象**。在打开的窗口中，选择**文件源控制**选项卡，并实施相关设置。



5. 单击**从目标读取**。项目中的文件源控制 [▶ 111]与所选目标系统上存储的任何图像进行比较。
6. 通过**添加文件...**，通过**添加目录...**或通过拖放将图像添加到文件源控制。如有必要，可通过项目下载下载样本图片：第一步 [▶ 23]
7. 如有必要，调整文件源对象向 PLC 发送图像的周期时间。

⇨ 文件源对象可以像这样使用：



下一步

创建 PLC 项目 [▶ 40]

4.2.3 创建 GigE Vision 相机

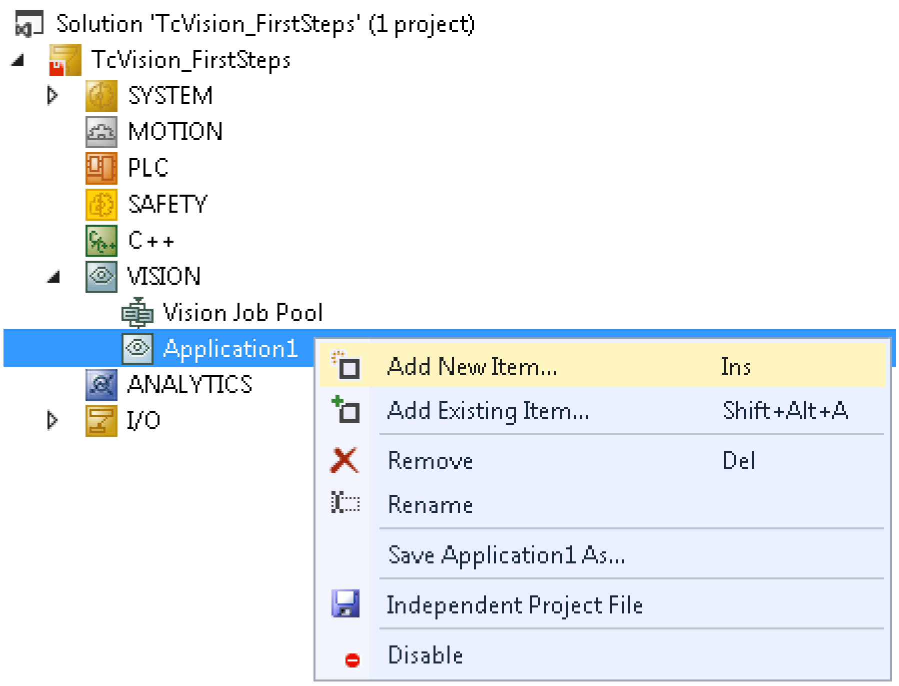
GigE Vision 相机 [▸ 63]对象的创建步骤如下所述。这需要将 GigE Vision 相机连接到 PC，并为相应的网络端口安装 TwinCAT RT 以太网适配器，具体如 系统配置 [▸ 27] 中所述。

另外，目标系统必须处于配置模式。点击 TwinCAT XAE Base 工具栏中的  符号，或点击 TWINCAT 下 Visual Studio 菜单中的 Restart TwinCAT (Config Mode （重新启动 TwinCAT，配置模式）。

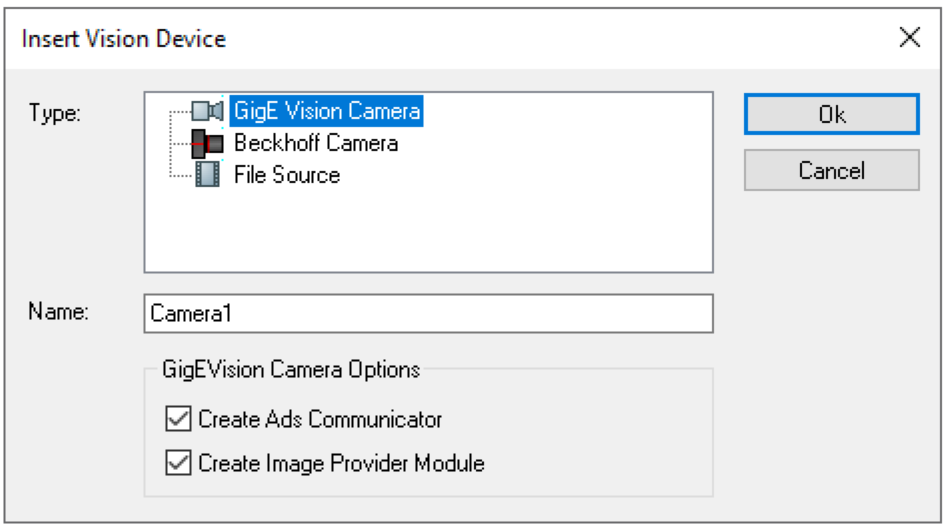
或者，您也可以创建文件源控制器 [▸ 31]。

相机创建步骤如下：

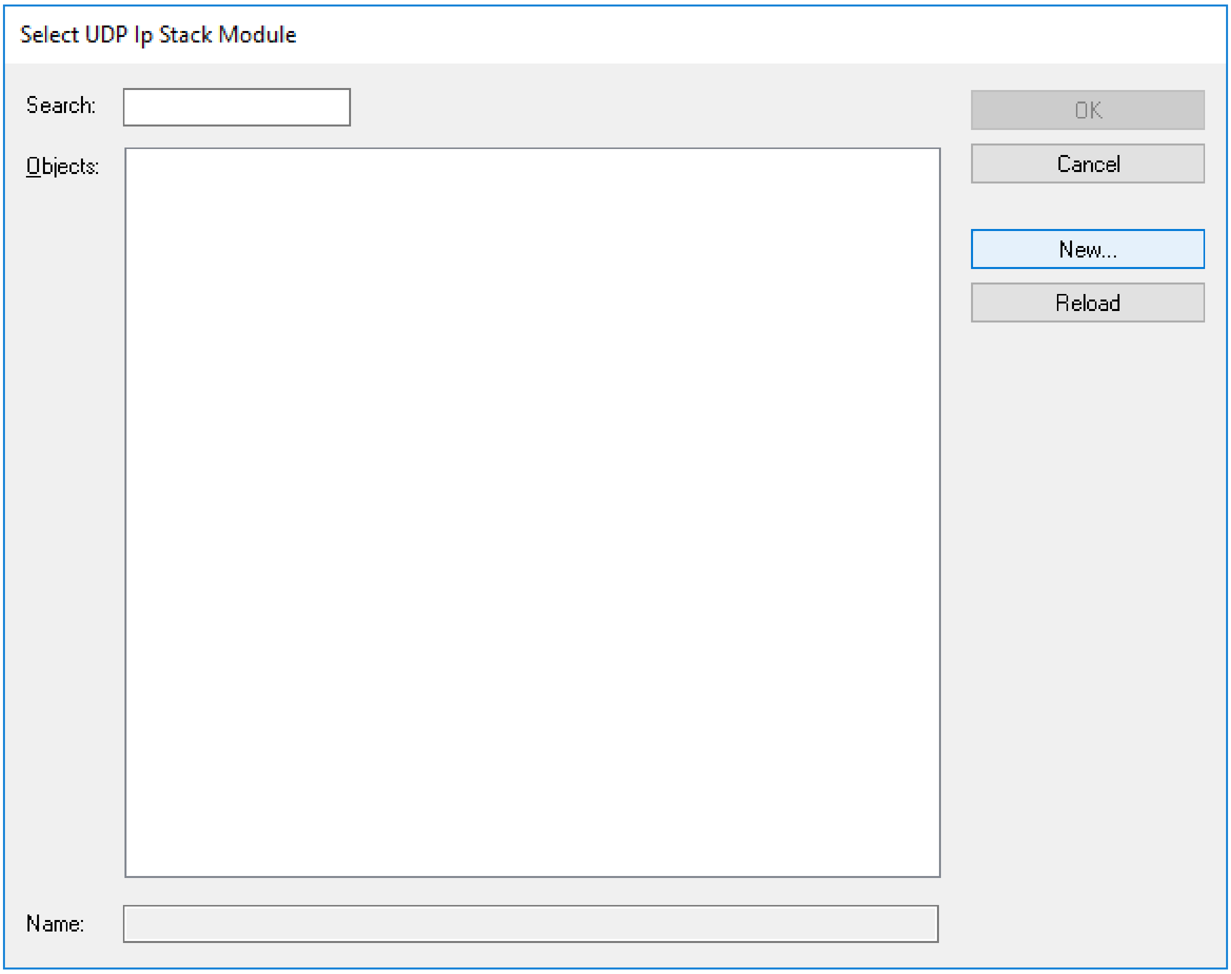
1. 点击先前创建 Vision 节点下的一个应用。
2. 右键点击并选择 Add new item... （创建新项目……）



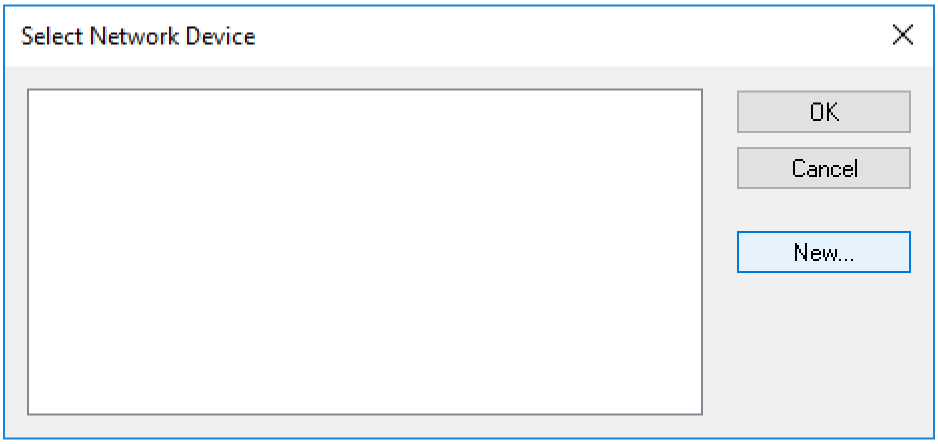
3. 选择设备类型 GigE Vision 相机。根据需要自定义相机名称和 GigE Vision 相机选项。记录数据流需要 Ads communicator；图像提供程序是相应 PLC 功能块的接口，因此通常一直需要（请参见相机详情 [▸ 108]）。点击 OK（确定）。



4. GigE Vision 相机通过 UDP/IP 进行通信。对于所使用的网络端口，需要在 IO 节点中安装一个带有相应 IP 栈的合适网络设备。点击 New... （新建……）进行创建。
只能针对网卡的每个物理端口创建一个带有一个 IP 栈的网络设备。

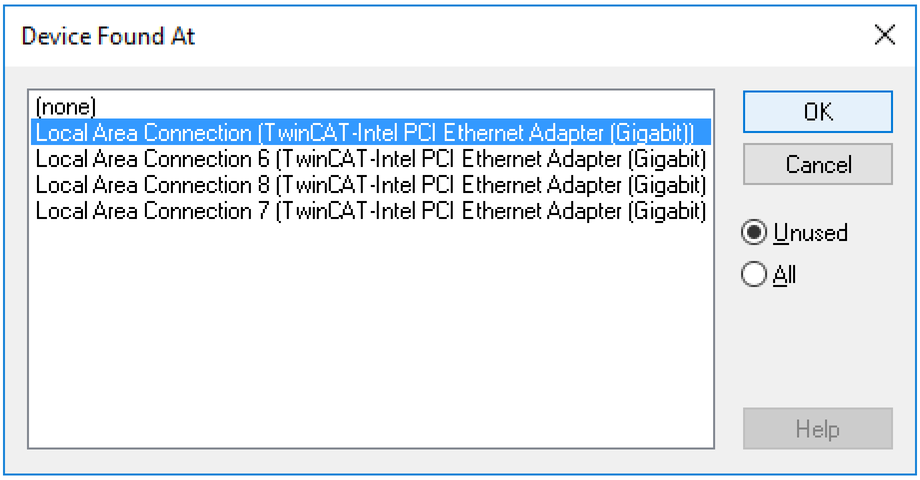


5. 要创建新网络设备，请点击 Select Network Device（选择网络设备）

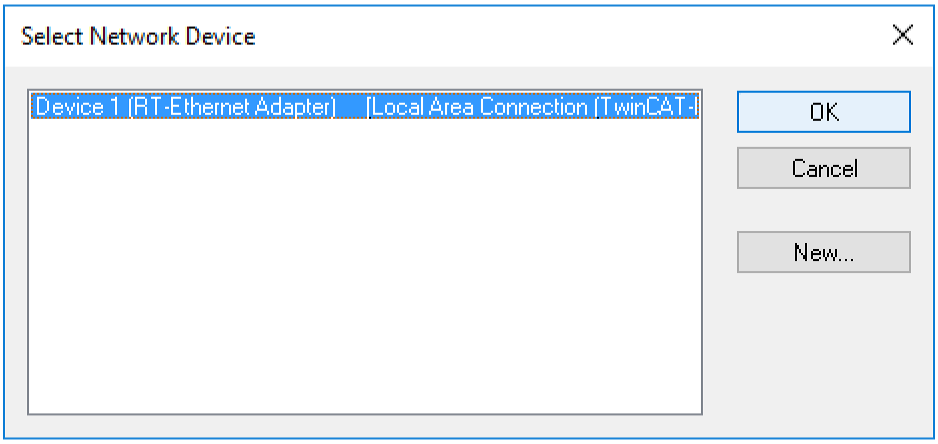


对话框中的 NEW... （新建……）

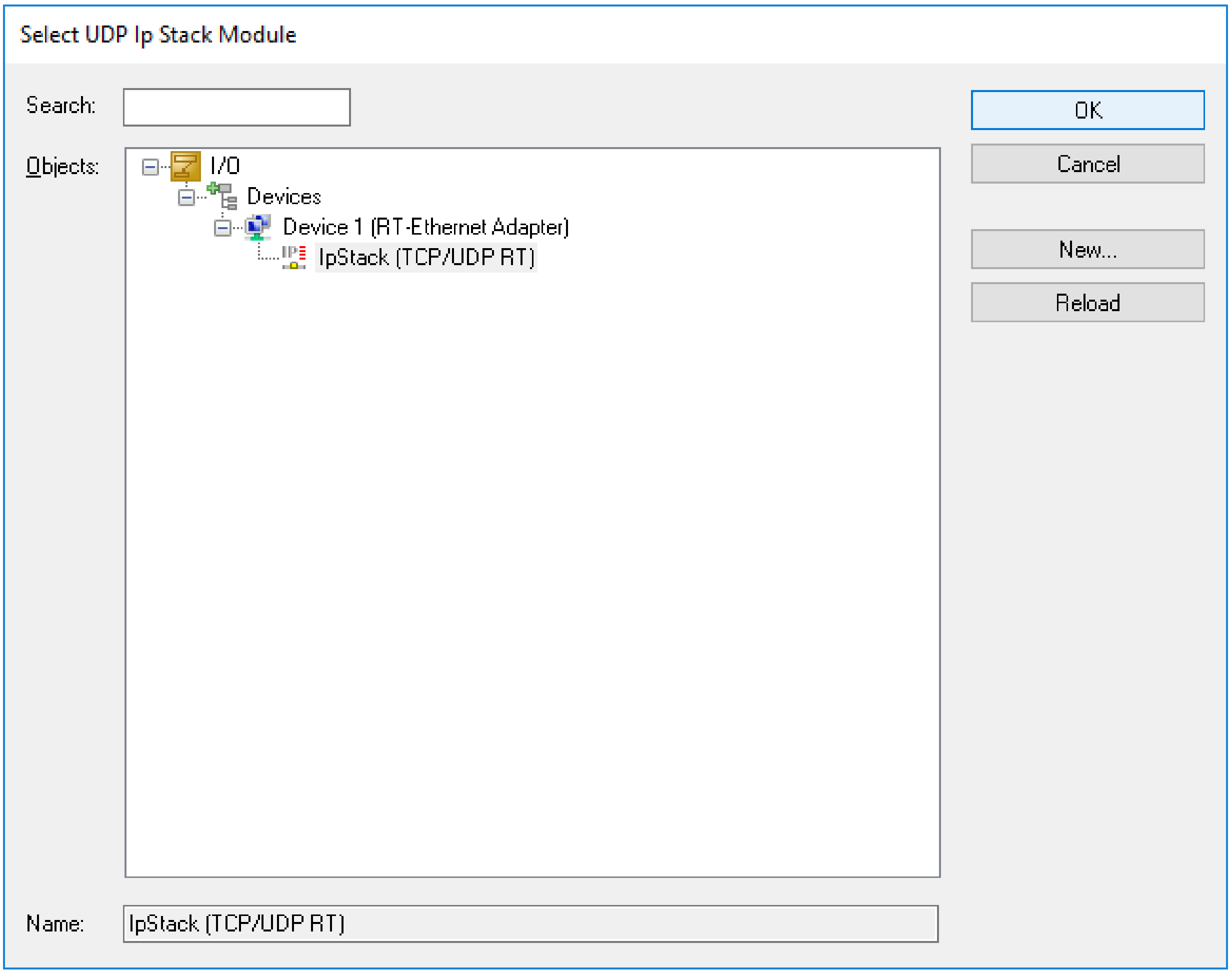
6. 在 Device Found At（设备找到weizh）对话框中，选择相机所连接的网络端口，然后点击 OK（确定）。只显示已安装 TwinCAT RT 以太网驱动程序的网络端口。



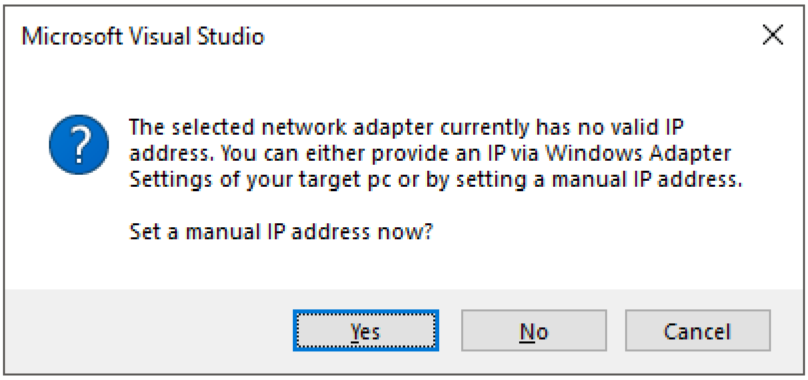
7. 现已在 I/O 配置中创建网络设备。在 Select Network Device（选择网络设备）对话框中选择它，然后点击 OK（确定）。这样就为设备创建了一个 IP 栈。



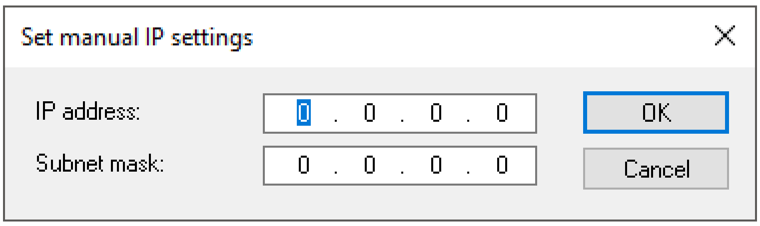
8. 选择 IP 栈，然后点击 OK（确定）。



9. 如果尚未为网络端口分配 IPv4 地址，请回答是否要分配固定 IP 地址。

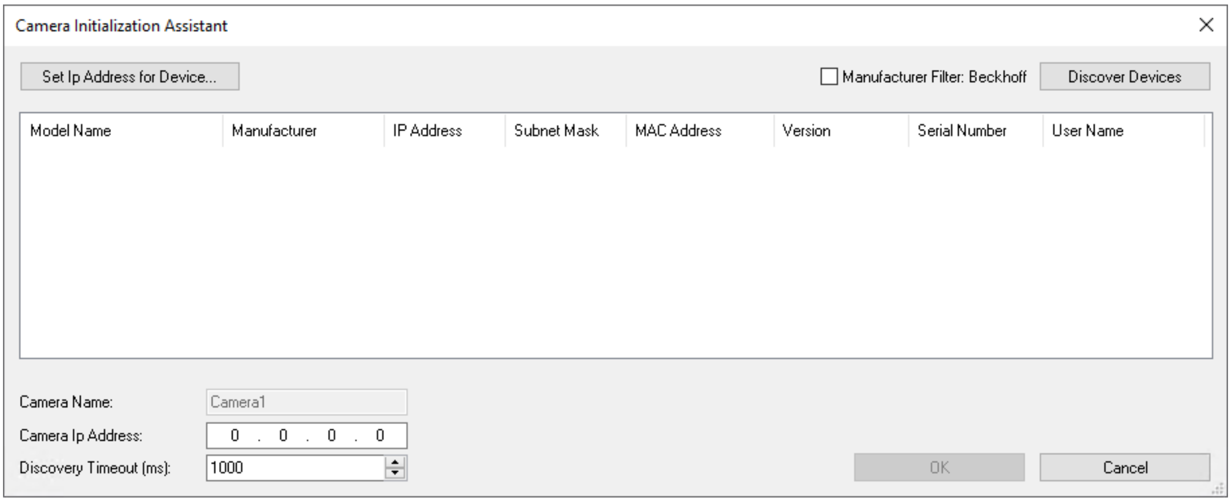


如果是，请在以下对话框中设置 IP 地址。

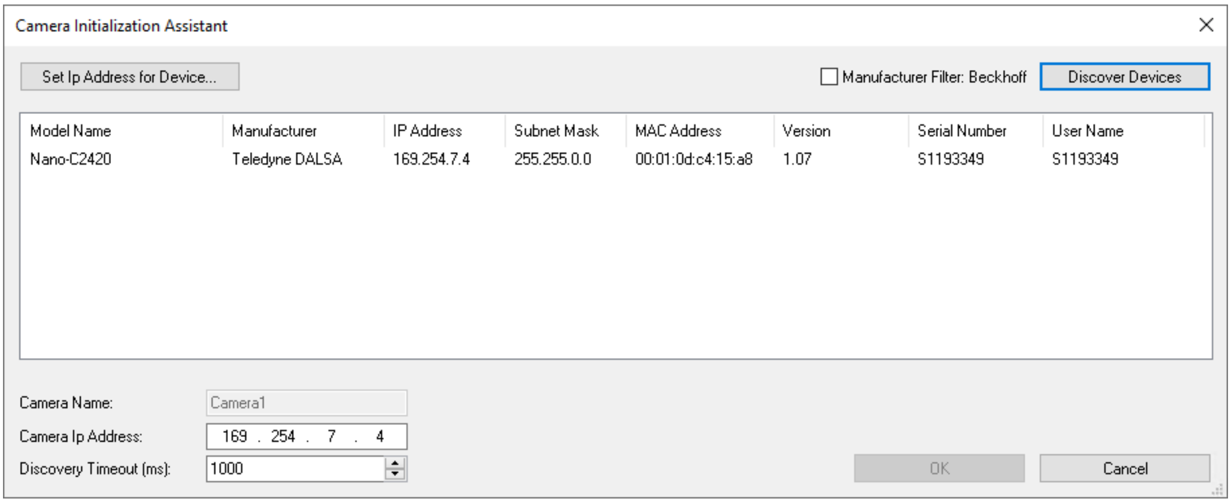


然后 IP 地址和子网掩码将保存在网络适配器的 TwinCAT IP 配置中（请参见网络配置 [▶ 53]）。

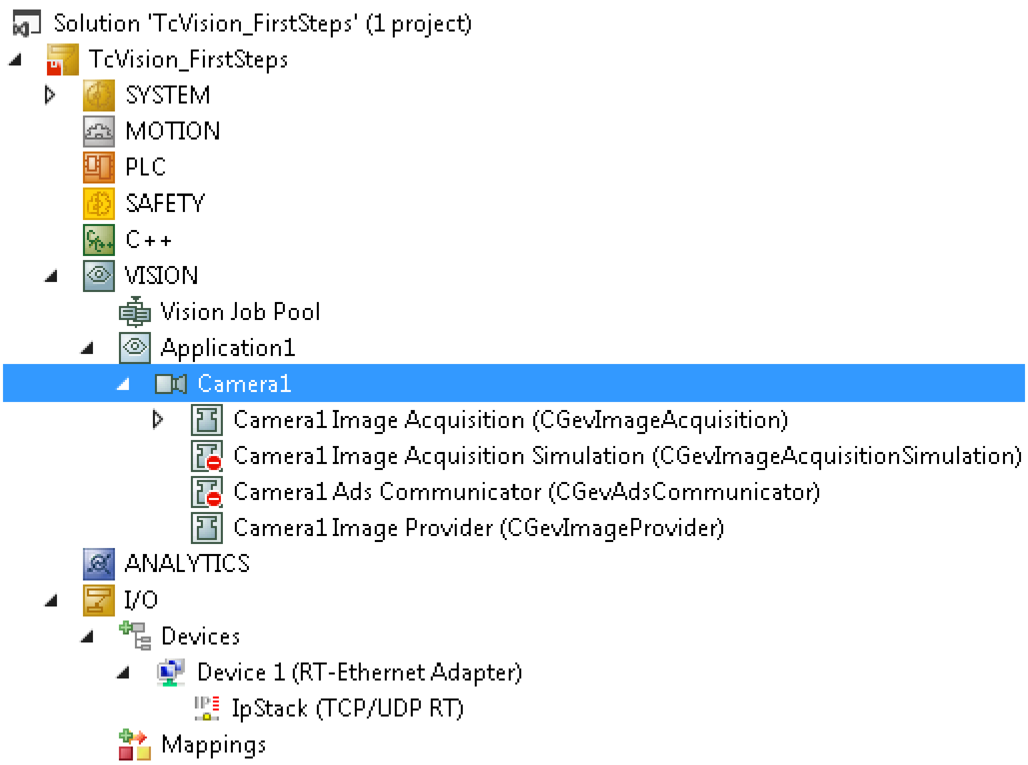
10. 在相机初始化助手中，点击 Discover Devices （发现设备）。所选网络端口将搜索连接相机。



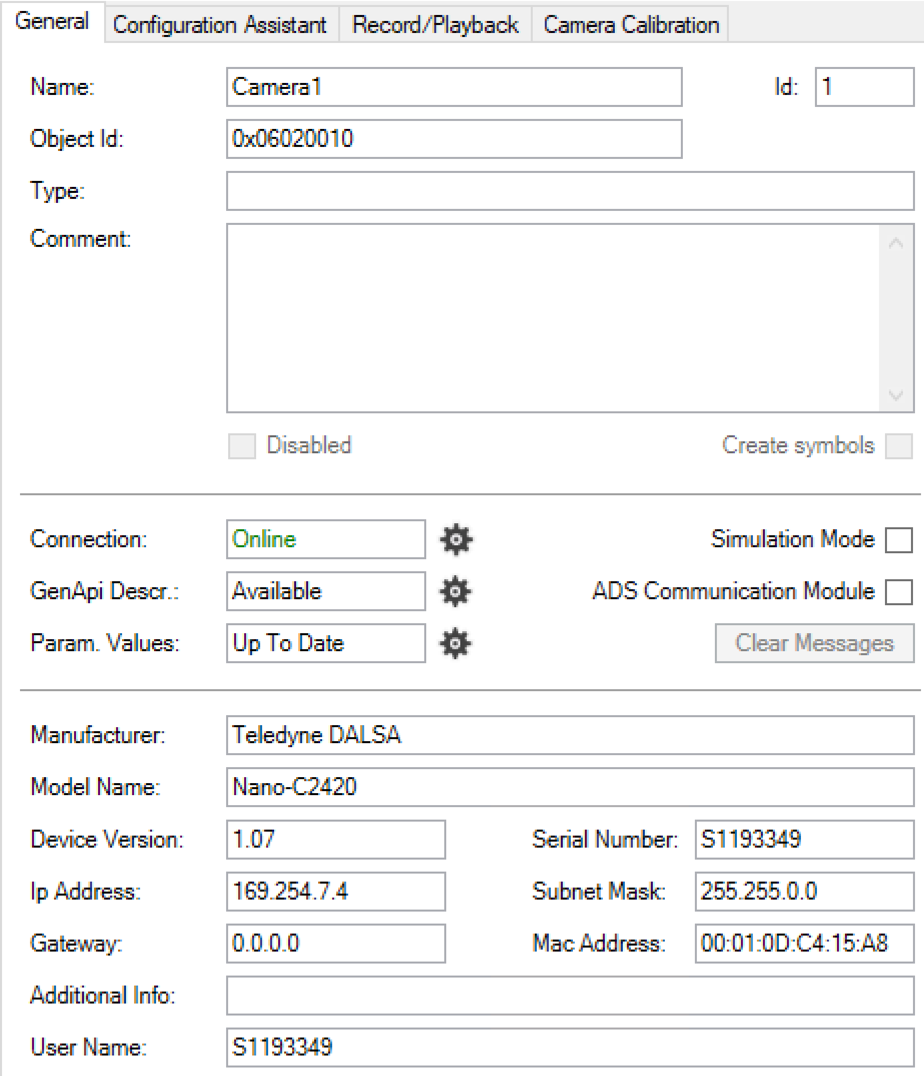
11. 显示找到的所有相机。选择所需的相机，然后点击 OK（确定）。



⇨ 相机现已在应用下创建并可使用。



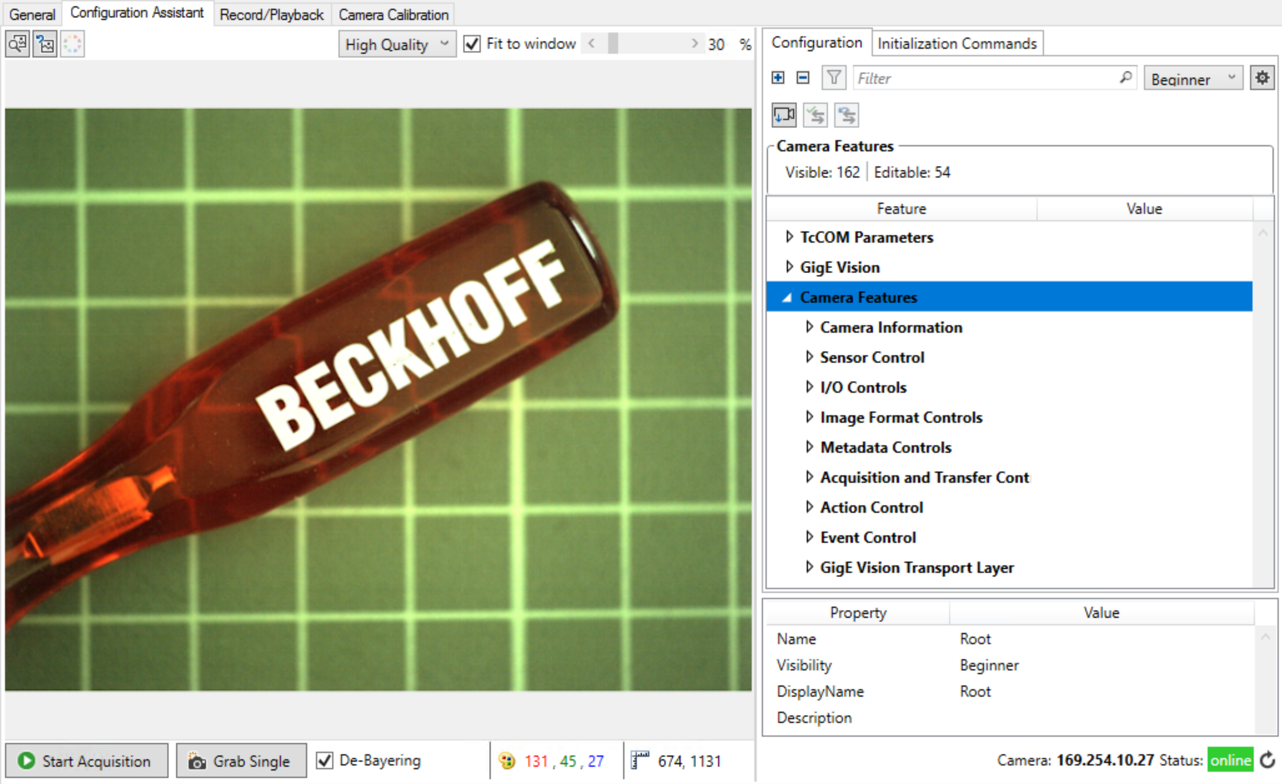
双击相机节点 [▶ 63]可打开一个窗口，在其中可验证相机连接。常规 [▶ 64] 选项卡提供有关相机的一般信息。



在 配置助手 [▶ 67] 中，您可以使用 Start Acquisition（开始采集）查看相机的实时图像，并对相机进行配置设置。

另请参见：相机配置样本 [▶ 2726]

如果要应用更改的参数并在每次启动时将其写入相机，请在最后创建 初始化命令 [▶ 77]。



下一步

创建 PLC 项目 [▶ 40]

还请参阅有关此

[▶ 110]

4.3 创建 PLC 项目

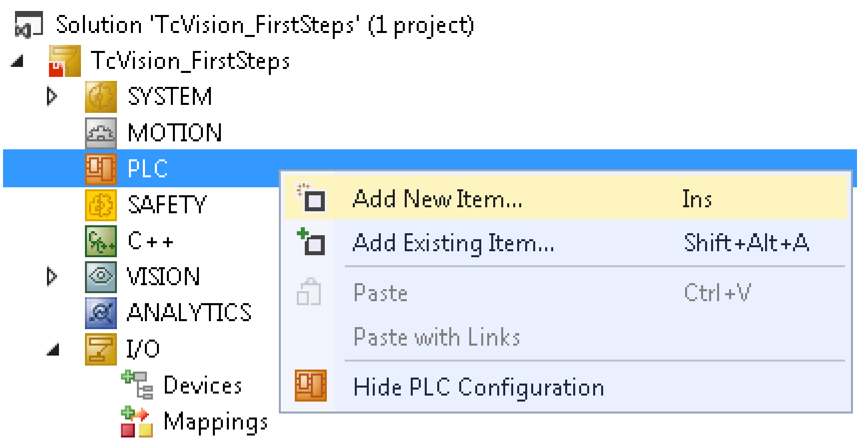
通过 TwinCAT Vision 创建 PLC 项目的视频示例：

视频：https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/Resources/5977084043/.mp4

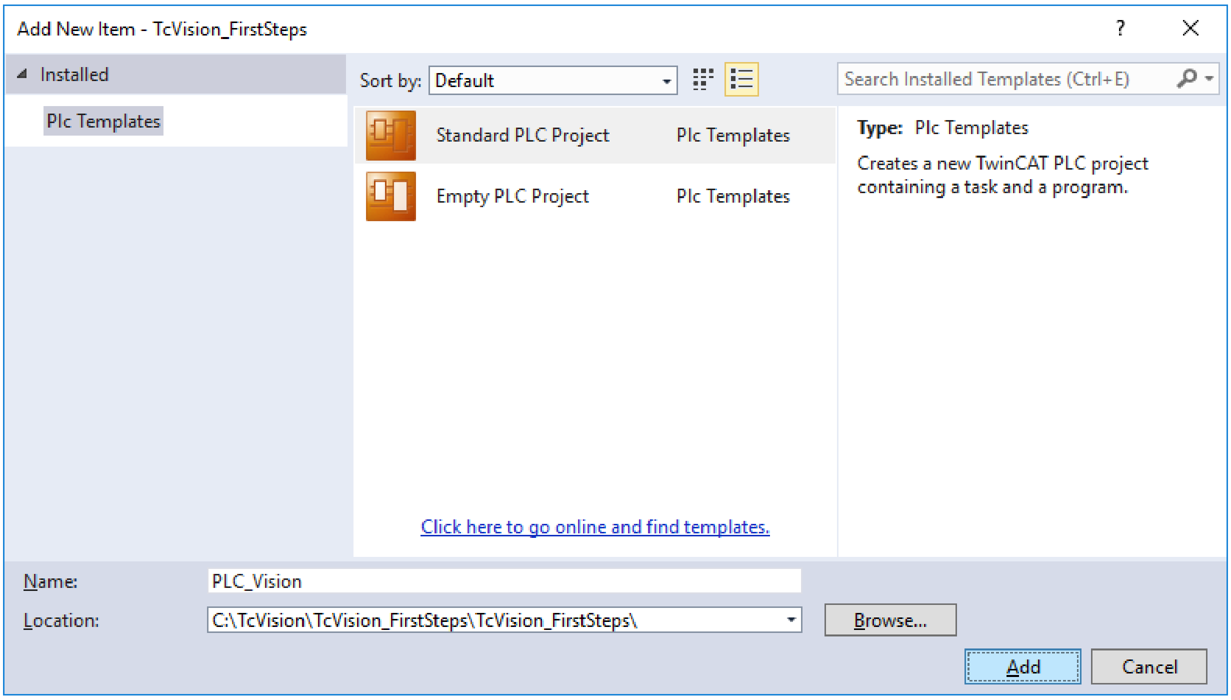
4.3.1 创建 PLC 项目

与一般机器控制的编程一样，图像处理的编程也在 PLC 中进行。为此，TwinCAT 项目必须通过 PLC 项目扩展。

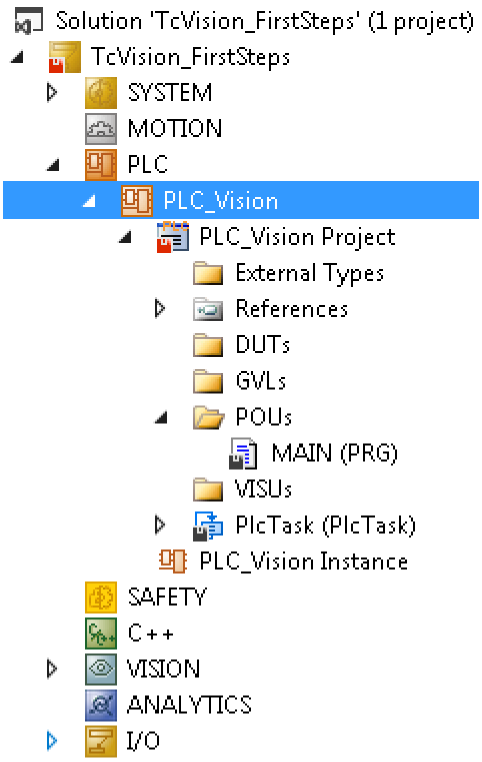
1. 右键单击解决方案资源管理器中的 PLC 节点，然后选择**添加新项目...**



2. 在**添加新项目**对话框中选择名称和位置。



⇨ PLC 项目已创建。



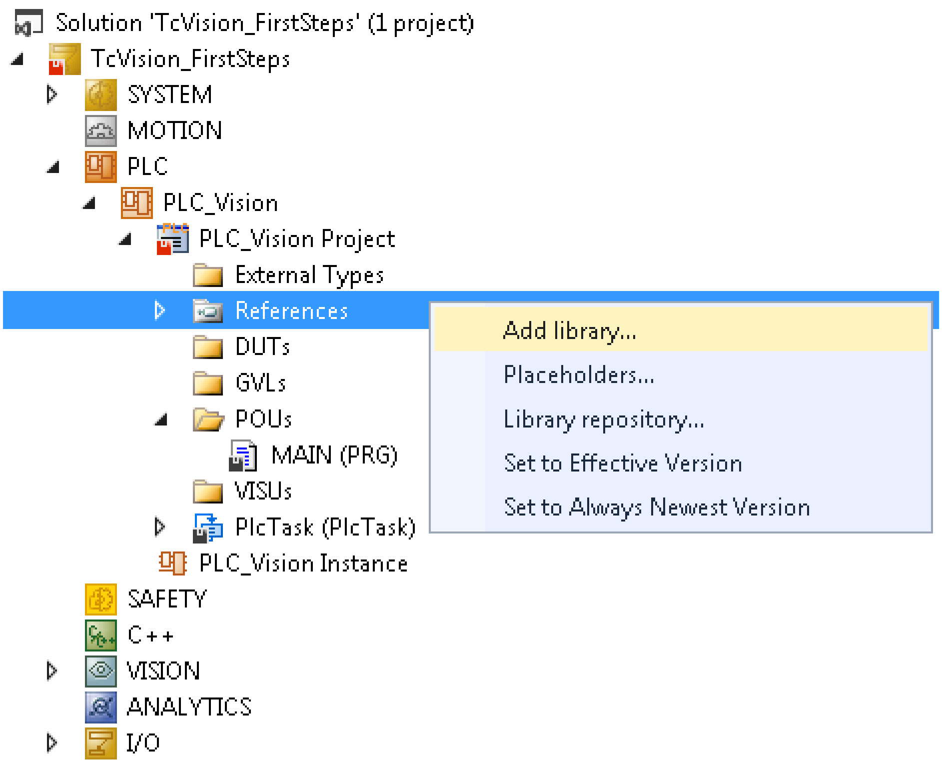
下一步

集成 PLC 库 [▶ 41]

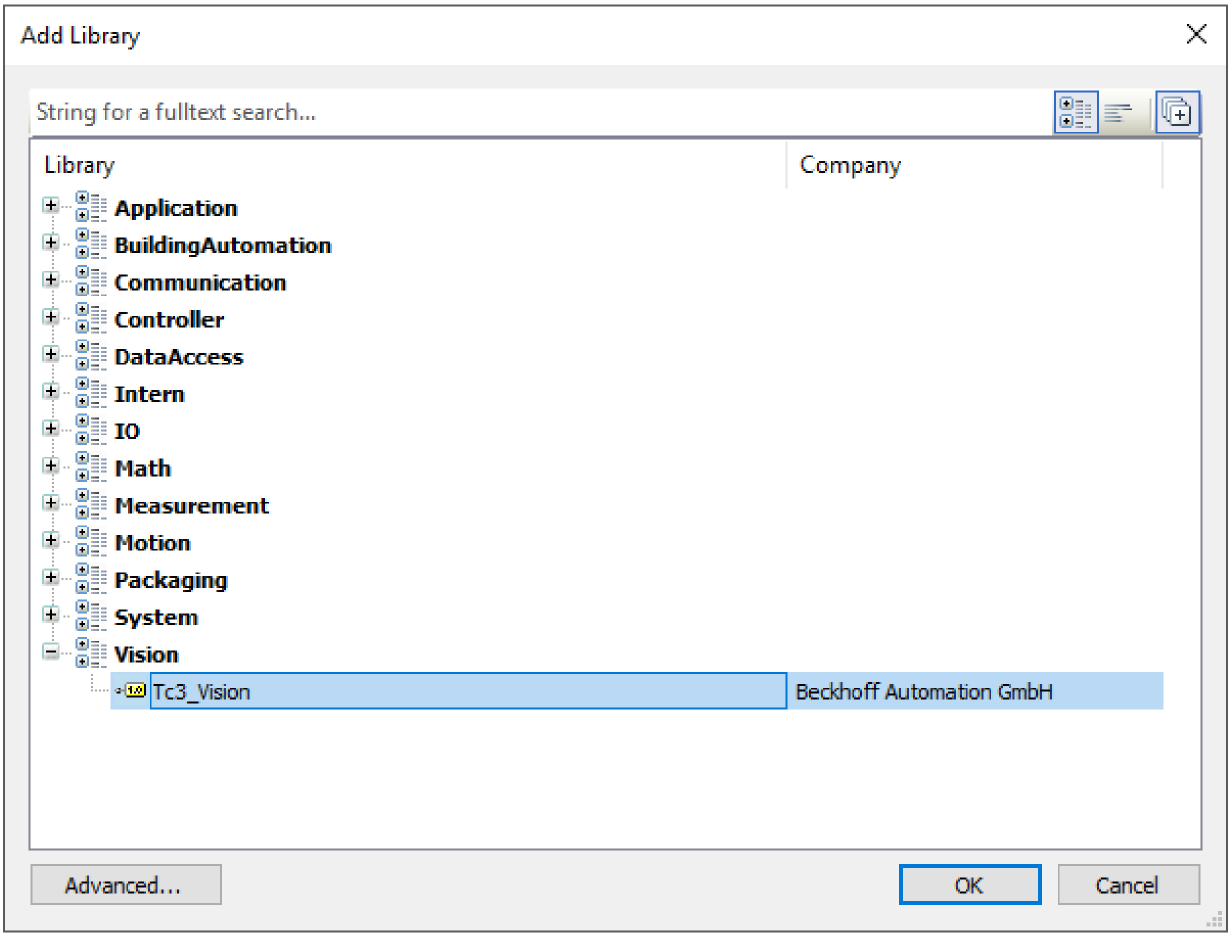
4.3.2 集成 PLC 库

Tc3_Vision 库包含图像处理所需的数据类型、功能和功能块。

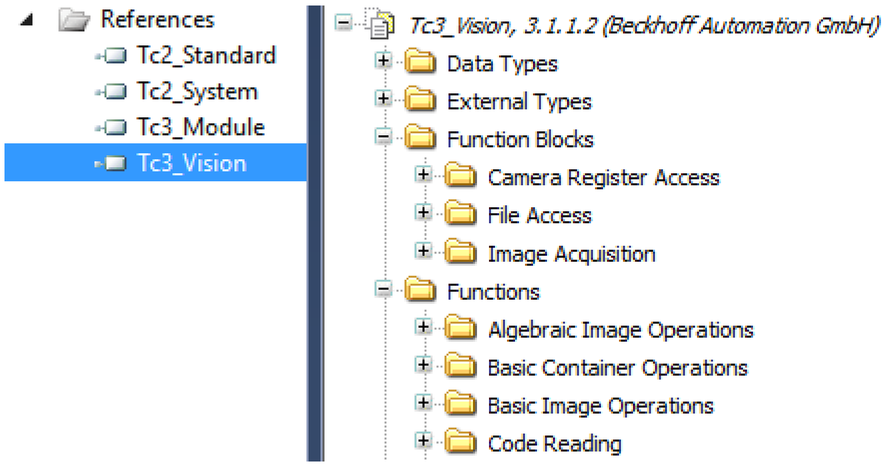
1. 如需将其添加到 PLC 项目中，右键单击**参考**，然后选择**添加库...**



2. 选择Tc3_Vision库，并通过**确定**进行确认。



⇨ 现在库已添加，且可以双击查看。



下一步

编写 PLC 程序 [▶ 43]

4.3.3 编写 PLC 程序

简单 PLC 程序用于从相机或文件源连续采集图像，包括以下内容：

变量

```
PROGRAM MAIN
VAR
    hr                  :    HRESULT;
    fbCamera            :    FB_VN_SimpleCameraControl;
    eState              :    ETcVnCameraState;
    ipImageIn           :    ITcVnImage;
    ipImageInDisp       :    ITcVnDisplayableImage;
    nNewImageCounter    :    UINT;
END_VAR
```

名称	类型	描述
hr	HRESULT	Tc3_Vision库的所有功能和许多方法都返回一个 HRESULT 类型的结果代码，用于错误处理。关于背后概念和单个代码含义的更多信息，可参见HRESULT [▶ 122]。
fbCamera	FB_VN_SimpleCameraControl	每个相机和文件源控制都由一个功能块实例表示。在最简单的情况下，这是一个FB_VN_SimpleCameraControl [▶ 1575]的实例，其中涵盖了相机和文件源控制的共同属性，且因此可以与两种设备类型相联系。另外，FB_VN_GevCameraControl [▶ 1550]可用于相机，而FB_VN_FileSourceControl [▶ 1541]用于文件源控制。
eState	ETcVnCameraState	每个视觉设备都有一个类型为ETcVnCameraState [▶ 147]的状态。
ipImageIn	ITcVnImage	处理的图像使用ITcVnImage [▶ 390]变量进行管理。在这个例子中，相机输入图像的访问通过ipImageIn获取。
ipImageInDisp	ITcVnDisplayableImage	显示的图像通过ITcVnDisplayableImages [▶ 390]管理。

带有状态机的程序代码

每个视觉设备都有一个内部状态，TCVN_CS_INITIAL是初始状态。由此，可以使用StartAcquisition方法将设备带入TCVN_CS_ACQUIRING状态。在这种状态下，图像可以被捕捉和传输。如果一个图像完全存储在存储器中，可以通过fbCamera.GetCurrentImage(ipImageIn)检索到。

为了检查一个图像是否被完全接收并可以被检索，IF 查询查询该方法的HRESULT以及图像内存区域的指针是否不为 0。之后，可以开始分析图像。例如，为了检查并表明可以检索到图像，计数器变量nNewImageCounter将递增。

F_VN_TransformIntoDisplayableImage [▶ 784]然后将图像指针移到ipImageInDisp，以显示图像。这样，就不能再通过ipImageIn检索图像了。另外，可以通过F_VN_CopyIntoDisplayableImage [▶ 753]创建副本，以显示和编辑图像。

```
eState := fbCamera.GetState();

IF eState = TCVN_CS_ERROR THEN
    hr := fbCamera.Reset();

ELSIF eState < TCVN_CS_ACQUIRING THEN
    hr := fbCamera.StartAcquisition();

ELSIF eState = TCVN_CS_ACQUIRING THEN
    hr := fbCamera.GetCurrentImage(ipImageIn);

    // Check if new Image was received
    IF SUCCEEDED(hr) AND ipImageIn <> 0 THEN
        nNewImageCounter := nNewImageCounter + 1;

        // Place to call vision algorithms
        hr := F_VN_TransformIntoDisplayableImage (ipImageIn, ipImageInDisp, hr);
    END_IF
END_IF
```

关于状态机的更多信息，可查看 API 章节图像采集 [▶ 1540]。

更多信息

- API 参考 [▶ 119]
- 图像采集的功能块 [▶ 1540]
- 图像 [▶ 130]

下一步

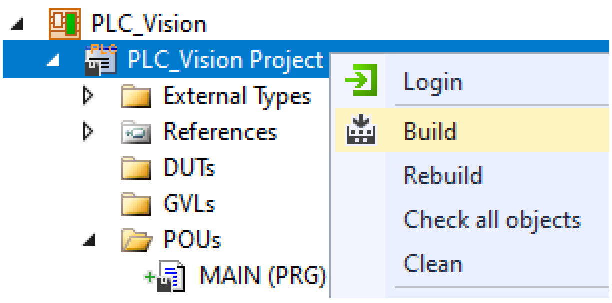
初始化功能块 [▶ 44]

4.3.4 初始化功能块

在配置了视觉设备和创建 PLC 程序之后，现在必须将视觉设备与 PLC 中适当的相机变量联系起来。

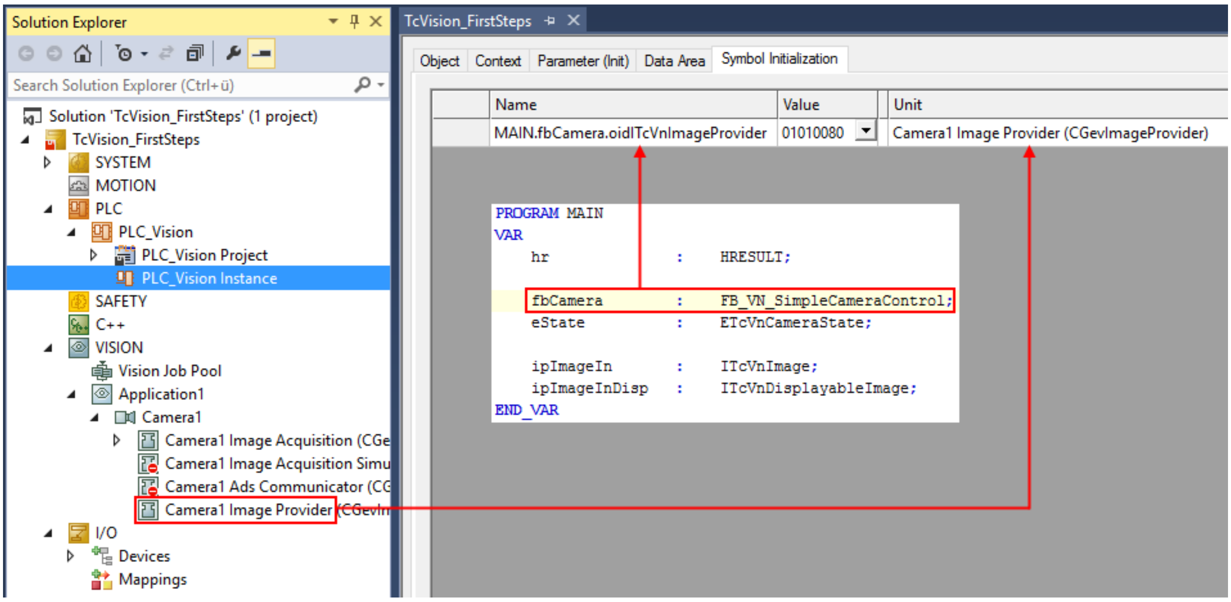
PLC 程序的功能块FB_VN_SimpleCameraControl [▶ 1575]的实例fbCamera必须与视觉设备的图像提供者连接。请注意：

1. 通过右键单击 PLC 项目并选择**构建**，建立PLC 程序 [▶ 43]。确保在操作过程中不会发生错误。如果发生错误，**符号初始化**选项卡出现在 PLC 实例和其下方fbCamera实例中。



2. 双击**PLC 项目实例**，然后打开**符号初始化**选项卡。

3. 通过“数值”栏的下拉菜单，将你的视觉设备的图像提供者分配给符号 MAIN.fbCamera.oidITcVnImageProvider。这是为图像中的相机对象显示的，但它与文件源对象的工作方式完全相同。



下一步

激活 TwinCAT 项目并启动 PLC 项目 [▶ 45]

4.4 运转

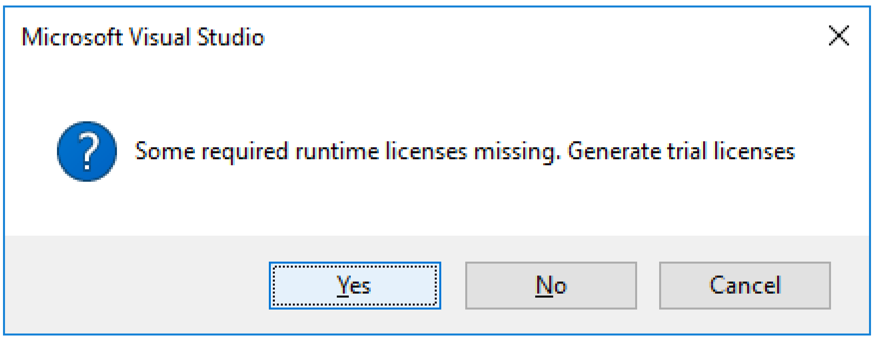
执行 TwinCAT Vision PLC 项目的视频示例：

视频：https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/Resources/5977086347/.mp4

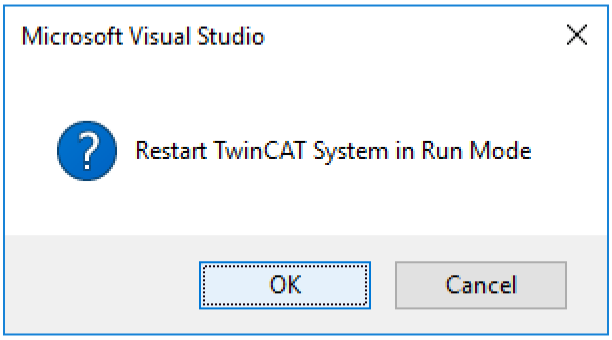
4.4.1 激活 TwinCAT 项目并启动 PLC 项目

可通过执行以下步骤执行项目：

1. 通过 或通过 Visual Studio 菜单TwinCAT >**激活配置**激活配置。
2. 如果目标系统上尚未激活许可证，会出现以下对话框。对于最初测试，可通过是，并输入安全代码，创建一个试用许可证。



3. 然后会问您是否要重新启动 TwinCAT。通过**是**确认。另外，可以通过 或通过 Visual Studio 菜单 TwinCAT >**重新启动 TwinCAT 系统**触发重启。



4. 如果 PLC 未设置为自动启动，请手动启动。为此，通过 或通过 Visual Studio 菜单的PLC >**登录**登录 PLC。

5. 然后通过 或通过 Visual Studio 菜单的PLC >**启动**启动 PLC。

下一步

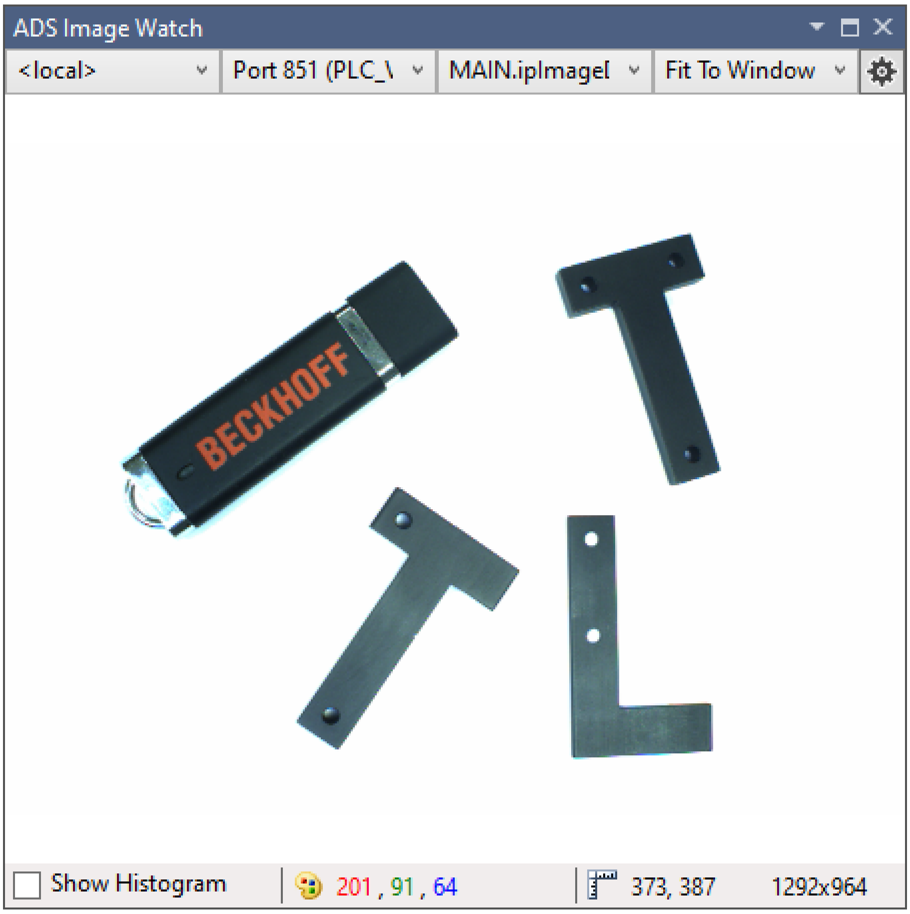
ADS 图像查看中的图像显示 [▶ 46]

4.4.2 ADS 图像查看中的图像显示

PLC 中类型为ITcVnDisplayableImage的图像可以显示在ADS 图像查看中。您可以在 Visual Studio 菜单 TwinCAT > Windows > ADS **图像查看**中找到该窗口。

如需选择待显示图像，可从左到右设置以下参数：

- 目标系统
- 端口
- 图像的 ADS 符号
- 图像缩放

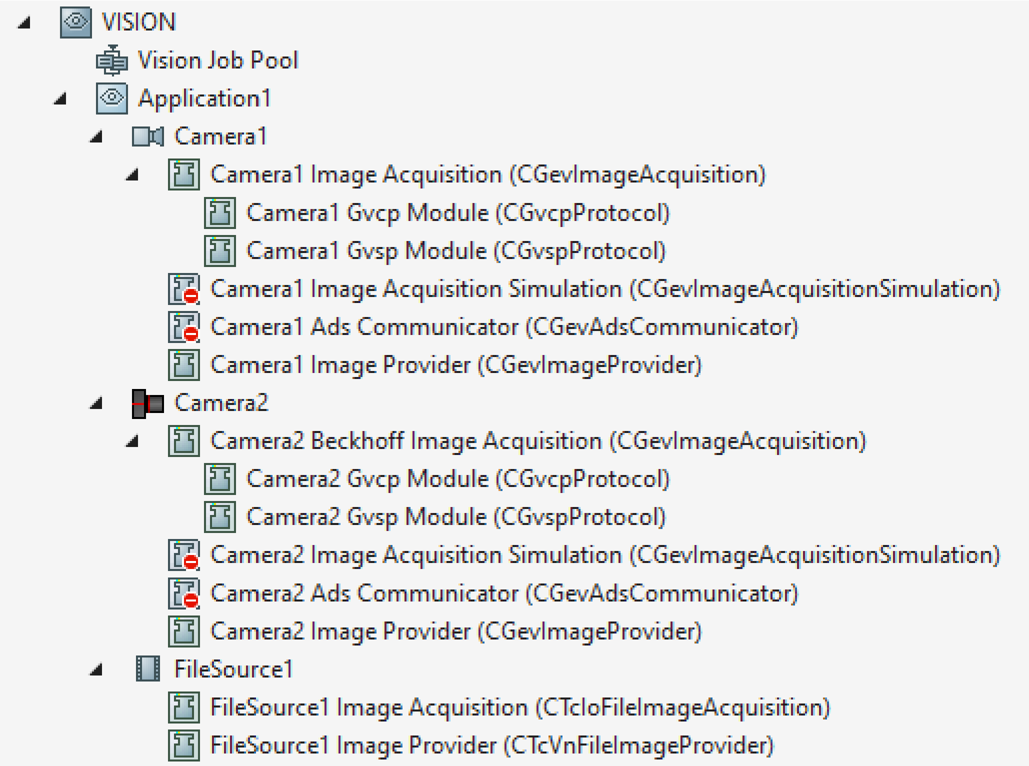


除了图像内容外，还可以在 ADS 图像查看中读取以下信息：

- 图像的直方图
- 当前光标位置的颜色/强度值
- 当前光标位置的像素坐标
- 图像大小

5 开发环境

TwinCAT Vision 完全集成在 TwinCAT 3 开发环境中。从本质上讲，TwinCAT Vision 通过另一个节点扩展 TwinCAT 系统，在该节点下可创建和配置 Vision 组件。使用了以下配置元件：



- VISION 节点 [▸ 59]提供影响系统上所有 Vision 组件的常规配置选项。它包括所有其他配置元件。
- Vision Job Pool [▸ 114] 可用于定义并行执行 Vision 功能的工作任务。每个系统都有一个单独的 Vision Job Pool。
- 一个应用节点 [▸ 62]为多个 Vision 设备（GigE Vision 相机和文件源）提供接口，形成一个应用单元。可以导出和导入应用的整个配置。每个 TwinCAT 项目可以有多个应用。
- GigE Vision 相机节点 [▸ 63]包含一个物理 GigE Vision 相机的配置。每个应用可以有多个 GigE Vision 相机节点。
- 文件源节点 [▸ 110]包含目标系统硬盘上的图像与 PLC 内图像采集模块之间接口的配置。每个应用可以有多个文件源节点。

i **与系统组件进行通信**

GigE Vision 相机节点和文件源节点均包含 TcCOM 模块，可用于与各种系统组件进行通信。

此外，开发环境还具有一个 ADS Image Watch [▸ 115] 窗口元件，可通过 ADS 显示来自 PLC 的图像。此外，为实现 Vision 组件的正确运行，需要常规系统配置 [▸ 48]。

为了能够在 TwinCAT HMI 中实时显示来自 PLC 的图像，可使用相应的 HMI 扩展 [▸ 2599]。

还请参阅有关此

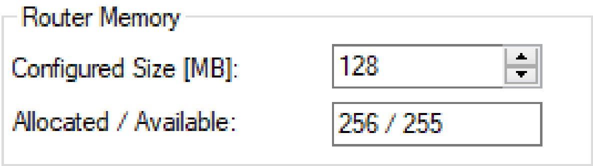
[▸ 110]

5.1 系统配置

以下章节描述了 TwinCAT Vision 特有的设置，应根据系统和项目进行或检查。

5.1.1 路由器内存

与传统控制系统相比，图像处理应用需要大量内存，因为即使是单一图像也可能包含几兆字节的数据。在 TwinCAT Vision 中，所有动态数据（请参见接口指针 [▶ 119]）均分配在路由器内存中。因此，相应地配置路由器内存大小尤为重要。可以在 Settings（设置）选项卡中实时节点下的 TwinCAT 系统配置中对其进行设置：



配置大小	设置当前项目所需的路由器内存。
分配	系统上当前分配的路由器内存。
可用	系统上当前分配路由器内存中仍可自由使用的部分。（其余部分已在 TwinCAT 中使用）。 **如果重启 TwinCAT 后可用内存没有完全恢复，则表明存在内存泄漏。在这种情况下，请重新启动系统并检查程序是否存在内存泄漏。**

TwinCAT 3.1 4024 中路由器内存的最大可调大小为 1024 MB。TwinCAT 3.1 4026 及以上版本d的可用大小基本上取决于可用主内存，且限制为 65535 MB。实际所需内存大小主要取决于 PLC 程序中动态内存的使用情况（如图像 [▶ 130]和容器 [▶ 132]）。

注意

路由器内存的大小

在配置路由器内存时，必须注意路由器内存将从主内存中减去，因此也不再可供操作系统使用。因此，我们建议在主内存大小小于 4 GB 的系统中避免使用 1024 MB 的大小。对于外部项目，在激活配置前一定要检查路由器内存设置，必要时减少内存。

估计所需的内存大小

路由器内存的所需大小取决于许多因素。通过计算动态分配的内存（通过接口指针 [▶ 119]），可以得到 TwinCAT 项目视觉部分的粗略估计。

一个图像的存储空间M（单位：字节）可以通过纳入图像大小（宽度w和高度h）、通道数C和元素类型的字节数 M = w * h * C * sizeof(ElementType)来计算。

对于一个字符串或矢量类型的容器，存储空间M（单位：字节）可以通过分配的元素数量N和元素类型的字节数 M = N * sizeof(ElementType)来计算。

必须注意数据的复制和发布方式。如果复制了一个图像或生成了一个新的图像，两个实例的存储空间当然必须包括在计算中。此外，路由器本身和其他项目部分的内存需求也必须包括在计算中。当通过 ADS 检索图像或其他数据时，这些都会被复制，所以也必须考虑到额外的存储空间。

注意

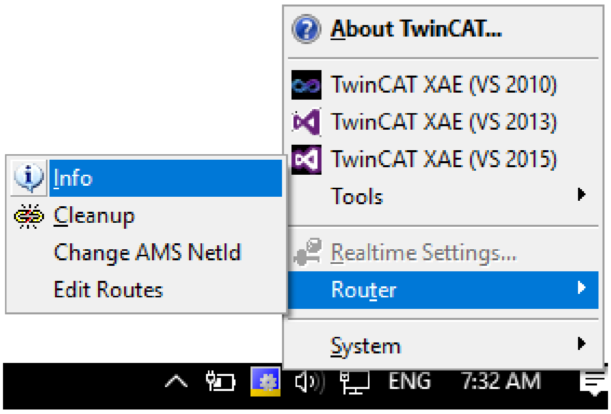
带有安全系数的内存值

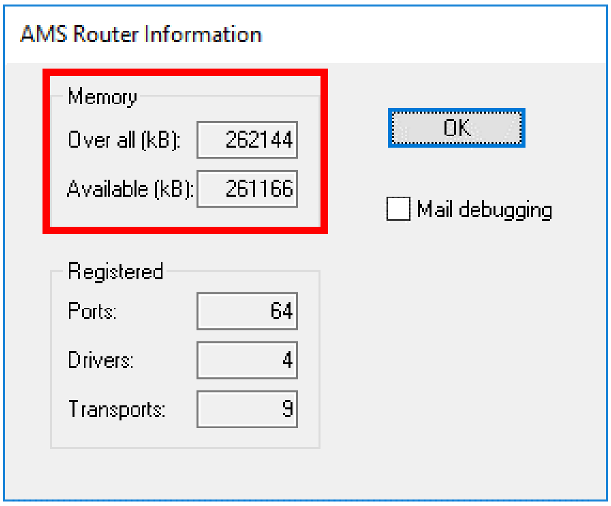
始终用一个安全系数来扩展计算的内存值。还要考虑到项目中其他模块的内存需求。

为了不在路由器内存中使用不必要的大量空间，你应该注意不要做任何不必要的深度拷贝，并再次释放不再需要的对象（见接口指针 [▶ 119]）。为了了解实际使用了多少内存，可以对路由器的内存进行实时监控，具体方法如下。

监测路由器的内存

即使在项目中设置了路由器内存，这也是一个全系统的设置。因此，可以在目标系统上查看项目中设置的、目前在系统上使用的路由器内存。为此，请右击 Windows 任务栏中的 TwinCAT 图标，并选择**路由器 > 信息**。





注意

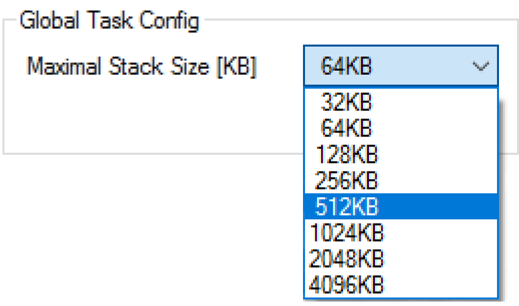
减少可用的路由器内存

可用的路由器内存可能通过异常或未释放的接口指针而减少。因此，在开发过程中应检查内存，特别是在出现错误的情况下。如果没有足够的内存，请重新启动整个系统，然后检查你的程序。

5.1.2 堆栈大小

默认的 64KB 大小通常对图像处理应用来说是足够的。在使用较大的图像格式或特殊功能时，如F_VN_Clahe [▸ 1195]，该数值太小，必须增加堆栈大小。那么建议直接使用 512 KB 或更大。

它可以在 TwinCAT 系统配置中实时节点下的**设置**选项卡中设置：



5.1.3 网络适配器

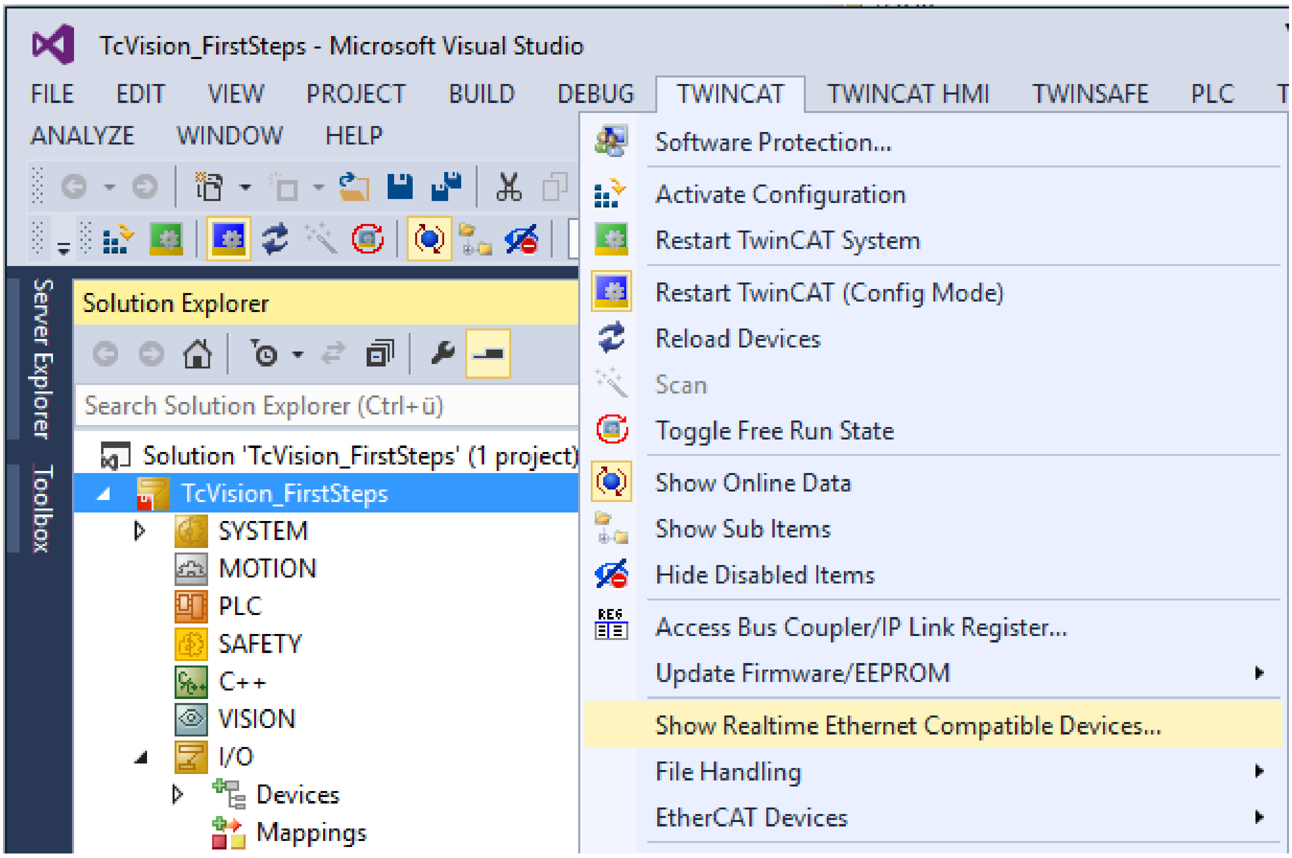
在选择、配置和使用网络适配器时，有几个方面需要考虑，下面将单独列出并加以说明：

- 首先，必须选择我们的 TwinCAT 实时以太网驱动程序所支持的合适网络适配器，以便安装驱动程序并用于实时操作。
- 然后，应针对该网络适配器禁用所有其他不需要的网络协议，以防止通信干扰。

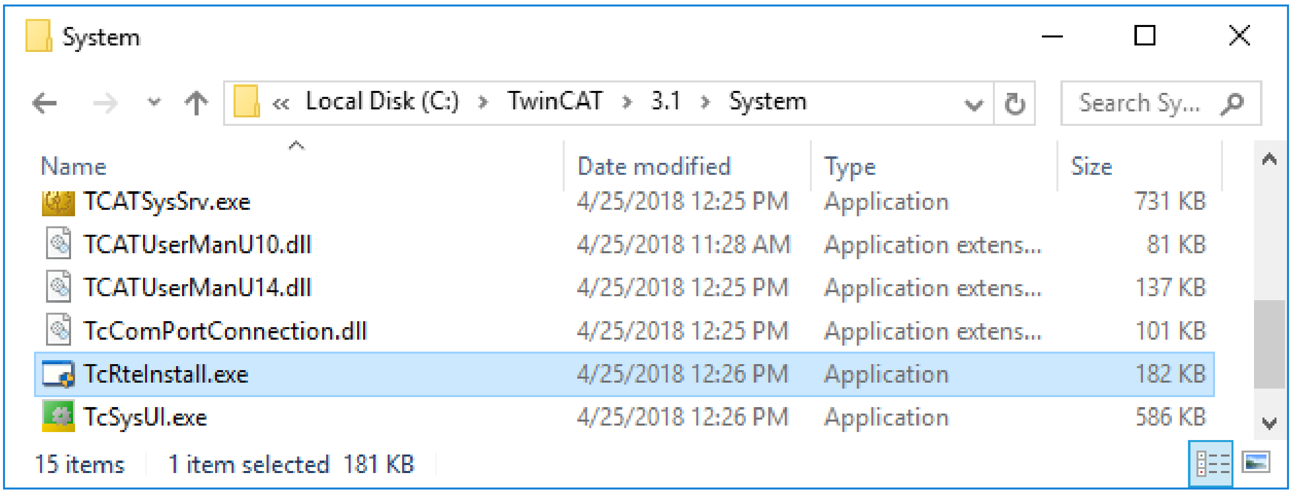
- 首次设置相机后，应在 TwinCAT 项目的 IP 栈中设置一个永久 IP 地址。
- 如果要在网络适配器的同一端口上操作多个相机，则必须在 IP 栈中进行额外设置。
- 最后，您将看到一些关于诊断选项的说明。

TwinCAT RT 以太网适配器

与连接 EtherCAT 设备类似，必须在相应的网络适配器上安装 TwinCAT 实时以太网驱动程序，才能实现与 GigE Vision 设备的连接。要打开相应的安装对话框，请点击 Visual Studio 菜单中的 TWINCAT > Show Realtime Ethernet Compatible Devices（**显示实时以太网兼容设备**）。



或者，您也可以在目标系统上调用文件 *C:\TwinCAT\3.1\System\TcRteInstall.exe*；此操作也会打开相应的安装助手。

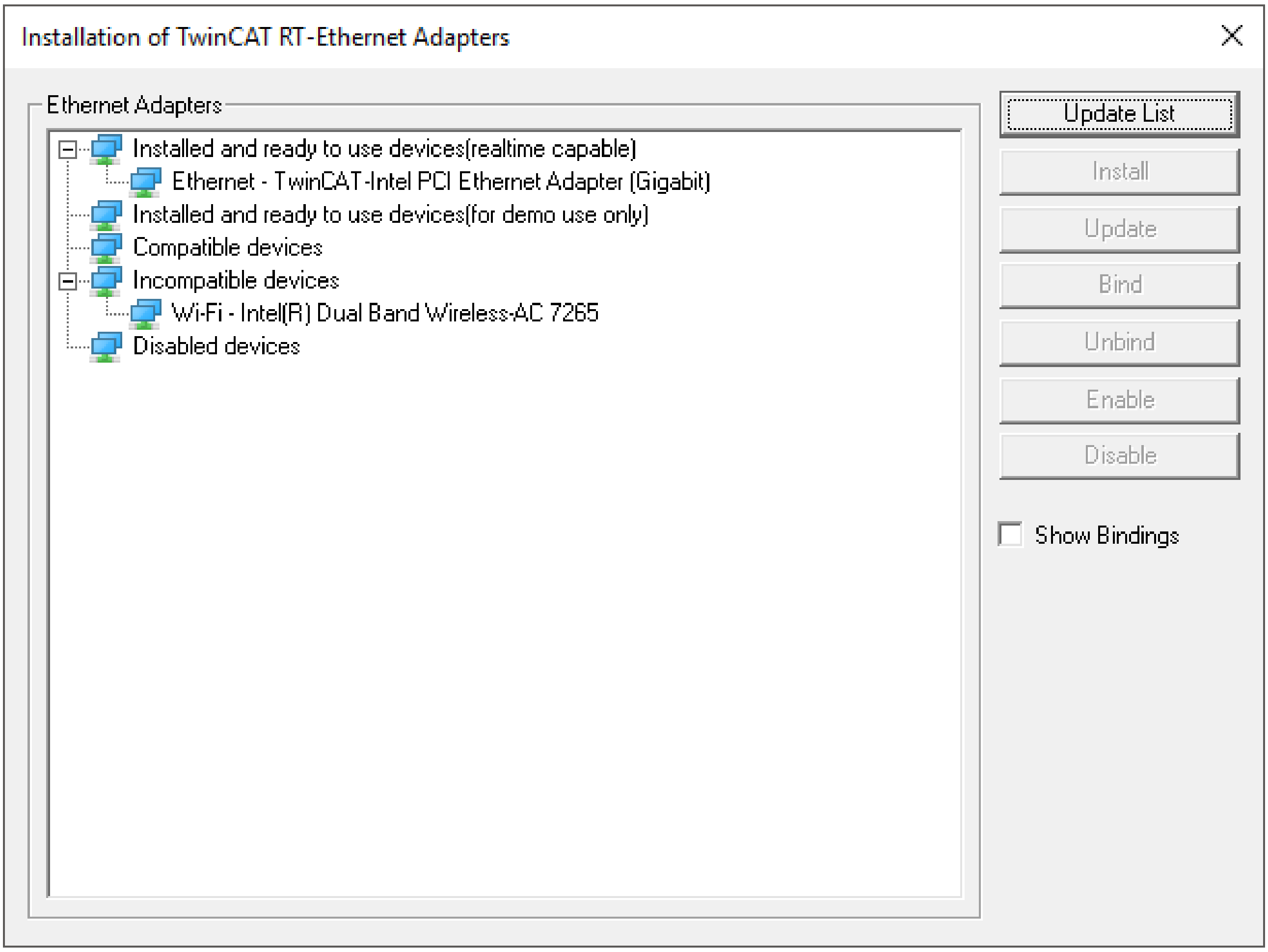


在安装助手中，确保 Installed and ready to use devices (realtime capable) （**已安装即用设备，支持实时功能**）下列出相应的网络适配器。如果不是这种情况，请从**兼容设备**列表中选择一个合适的适配器，然后点击 Install（**安装**）。

注意

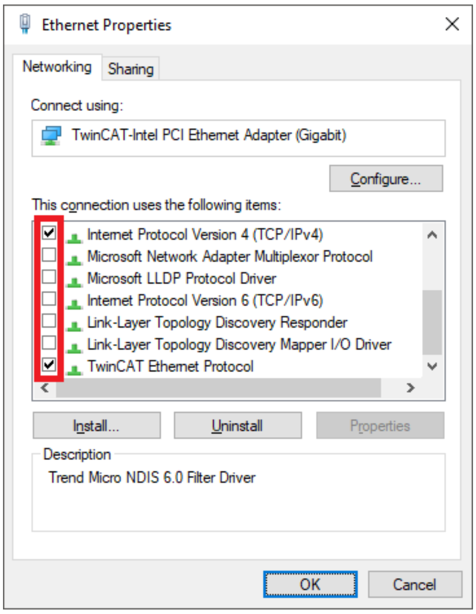
仅供演示使用

这种状态不足以保证 GigE Vision 设备的稳定连接连接和运行。如果安装后网络适配器只列在该组下，则必须使用其他适配器。



其他网络协议

建议将网络适配器的使用限制为 TwinCAT 以太网协议，并通过 TwinCAT 项目中的 IP 栈分配一个持久 IP 地址。因为如果多个协议都能访问网络适配器，它们发送的数据包可能会导致相机通信出错。相关设置可在网络适配器的 Windows 属性中找到。



只有在使用 DHCP 设置相机连接时才需要互联网协议版本 4 (IPv4)。首次设置相机时，自动分配链路本地地址 (LLA) 会有所帮助。在所有其他情况下或完成初始设置后，也可以禁用 IPv4 协议并在 TwinCAT 的 IP 栈 [▶ 53]中手动配置 IP 地址。

注意

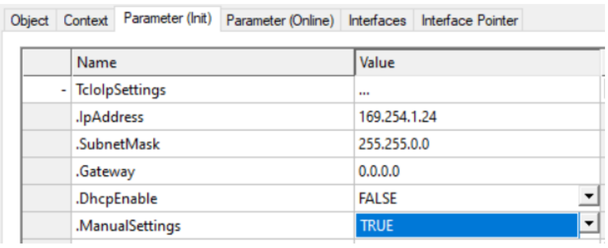
禁用其他网络协议

如果可能，禁用以太网适配器上除 TwinCAT 以太网协议之外的所有网络协议。否则，与 GigE Vision 相机的通信可能会出现错误和延迟。

指定持久 IP 地址

如果 PC 不应自动指定网络适配器的 IP 地址，而您希望手动指定一个持久 IP 地址，则可以在 TwinCAT 中如下进行操作：

1. 在 I/O 配置中打开 RT 以太网适配器的 IP 栈。
2. 选择 Parameter（参数）（Init）选项卡，点击 TcIoIpSettings 前面的**加号 (+)** 显示更多参数。
3. 指定参数 .IpAddress 和 .SubnetMask。
4. 将参数 .ManualSettings 设置为 TRUE。



如果在创建相机对象 [▶ 34]时手动分配 IP 地址，则会自动分配相同的设置。另外，也可以通过 Windows 网络适配器属性定义一个持久 IP 地址。

注意

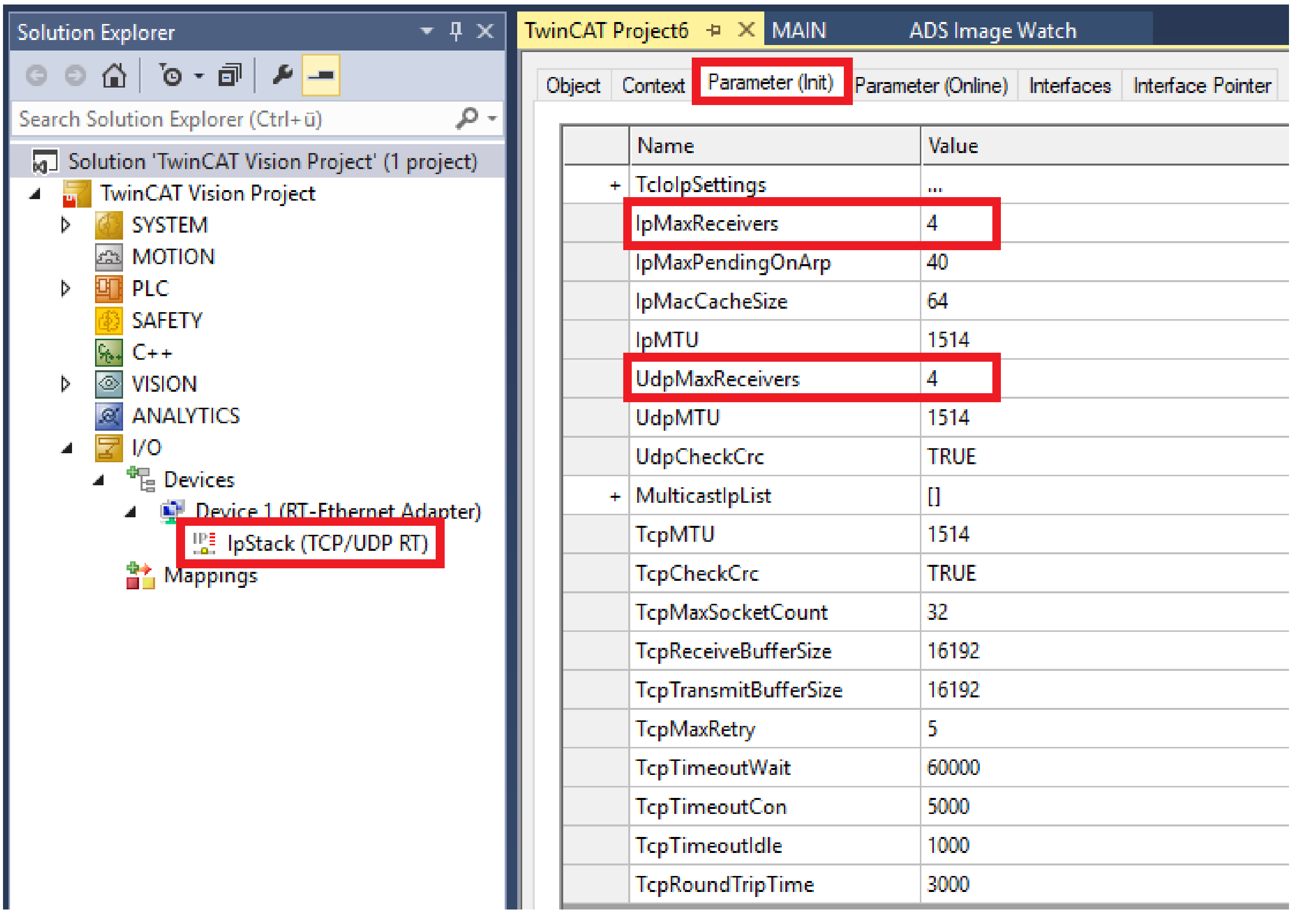
持久 IP 地址

为防止网络适配器的 IP 地址发生变化，请在 TwinCAT 项目的 IP 栈中指定一个持久 IP 地址。

在一个端口上操作多台相机

如果想要在相同 RT 以太网适配器上操作多台相机，必须确保 UDP/IP 设置允许操作。导航到 RT 以太网设备下的 UDP/IP 堆栈，并选择**参数（初始化）**选项卡。

在出现的参数列表中，设置IpMaxReceivers和UdpMaxReceivers，使其相当于通过该适配器连接的相机数量的两倍。这是因为每台相机有两个接收器（GVCP 和 GVSP 模块）。标准设置允许连接两台相机。为此，参数被设置为2 * 2 = 4。



高级设置

最大传输单元（MTU）	待传输帧的最大允许尺寸是 1514。因此，目前不支持巨型帧。

诊断

如果连接出现问题，特别是发送或接收的帧出现问题，你会发现 RT 以太网适配器中有一个包含统计信息的选项卡，而在 IpStack 中有包含单个计数器信息的**参数（在线）**选项卡。关于详细信息，可参见附录/故障排除，地址：相机通信 [▶ 2751]。

混杂模式

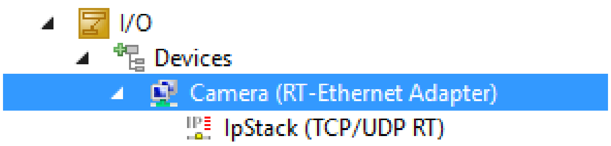
混杂模式是 TwinCAT RT 以太网适配器的一个设置。在使用Wireshark工具进行调试时，需要使用。在与 GigE Vision 设备进行通信时，这种模式可能会导致问题。

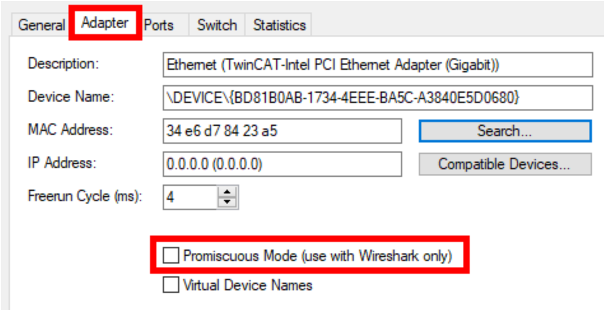
注意

避免混杂模式

在与 GigE Vision 设备进行通信时，这种模式可能会导致问题。因此，GigE Vision 设备应非“混杂模式”下运行！

你可以通过在 Visual Studio 中打开相应的 RT-Ethernet 适配器并切换到**适配器**选项卡来检查这个设置。**混杂模式（仅与 Wireshark 一起使用）**的复选框必须停用。





5.1.4 CPU 内核和任务

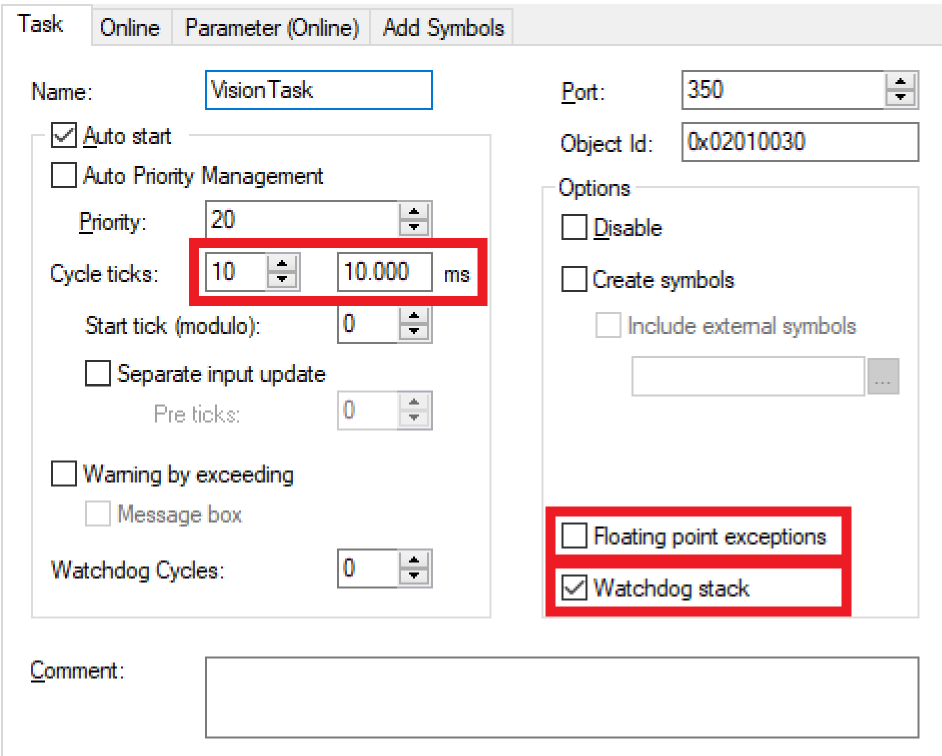
任务和 CPU 核的配置不为 TwinCAT Vision 所特有，对于所有 TwinCAT 组件均相同。有关如何操作的说明，请参见实时设置和任务设置。下文将介绍与 TwinCAT Vision 有关的问题和注意事项。

TwinCAT Vision 应用需要用于各种用途的各种任务：

- 执行 PLC 的循环任务
- 操作 Vision 设备的循环任务
- 用于并行 PLC 执行的工作任务
  （可选；此类任务将在 工作任务 [▶ 58] 章节中单独介绍）。

配置 PLC 任务

基本上，TwinCAT Vision 所需的所有循环任务均自动创建。创建 PLC 时，会创建一个循环时间为 10 ms 的任务 这意味着 PLC 中的图像处理序列每 10 ms 重复一次。您可以在 SYSTEM（系统） > Tasks（任务） > [TaskName]（[任务名称]）下更改任务设置。



循环刻度	循环刻度乘以所选 CPU 核的基准时间（默认为 1 ms）等于上述 PLC 的循环时间。
浮点异常	该设置指定 TwinCAT 是否检查浮点异常。 对于 Vision 应用，建议停用此选项。除此以外，这些错误信息没有说明异常的确切原因。
Watchdog（看门狗）栈	如果要在分配的 PLC 程序中使用 Watchdog（看门狗） [▶ 127]，则必须激活此设置。

⚠ 警告

浮点异常

对于主要执行 Vision 命令的 PLC 任务，强烈建议停用“Floating point exceptions（浮点异常）”选项。否则，某些 TwinCAT Vision API 功能可能会出现不必要的错误，导致系统中止。但是，如果运动控制或安全组件也在同一个 PLC 任务中执行，则必须激活此选项。

注意

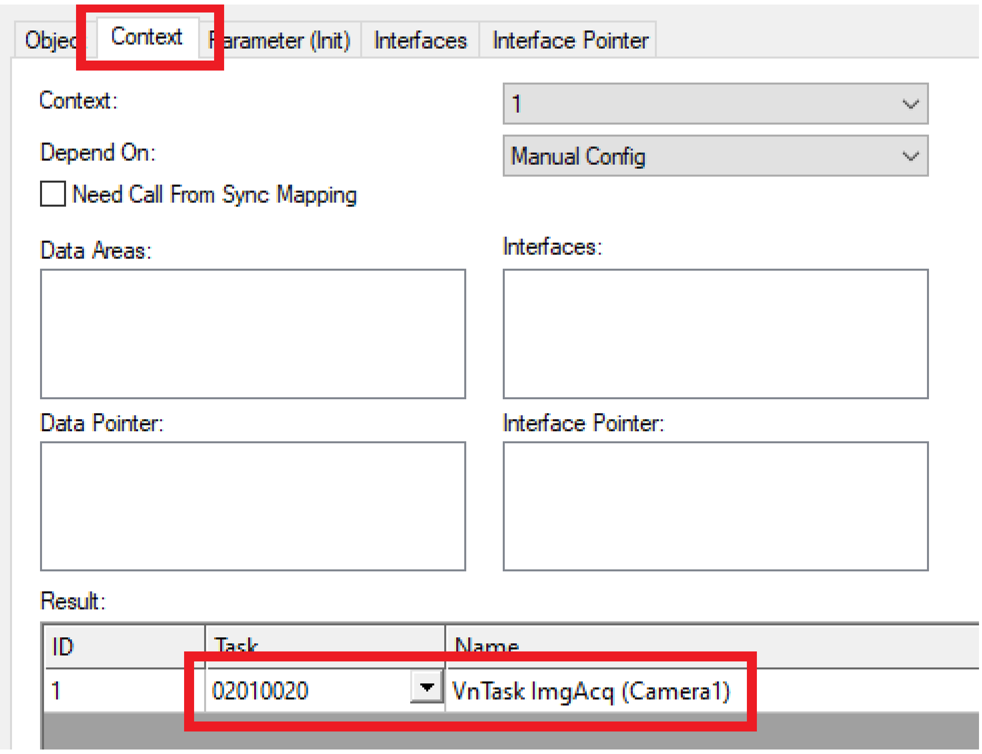
工作任务中的浮点异常

如果还为 Vision 应用使用工作任务 [▶ 58]，并且在 PLC 任务中禁用了“浮点异常”，则也必须在分配的工作任务中禁用此选项。否则仍会出现异常和系统崩溃。

设备任务配置

创建 Vision 设备时会自动生成所需任务。这些任务的循环时间可以如上所述进行调整。在大多数情况下，建议不要更改这些任务的循环时间。只有在特殊情况下才需要这样做。

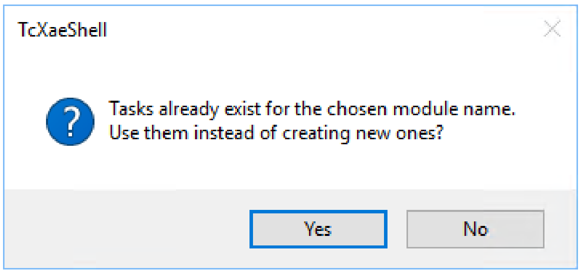
如果需要调整 Vision 设备与任务的链接，可在 GigE Vision 相机 [▶ 108]或文件源 [▶ 113]对象的相应 TcCOM 对象的 Context（上下文）选项卡中进行：



如果使用下列 TcCOM 对象，则必须将它们与循环任务相链接：

- 相机图像采集 [▶ 108]
- 相机图像采集仿真 [▶ 109]
- ADS 通讯器 [▶ 109]
- 文件图像采集 [▶ 114]

如果在创建 Vision 设备时所选设备名称下已经存在合适的任务，则可以选择重复使用这些任务。在这种情况下会出现以下窗口；点击 Yes（是）以确认重复使用：



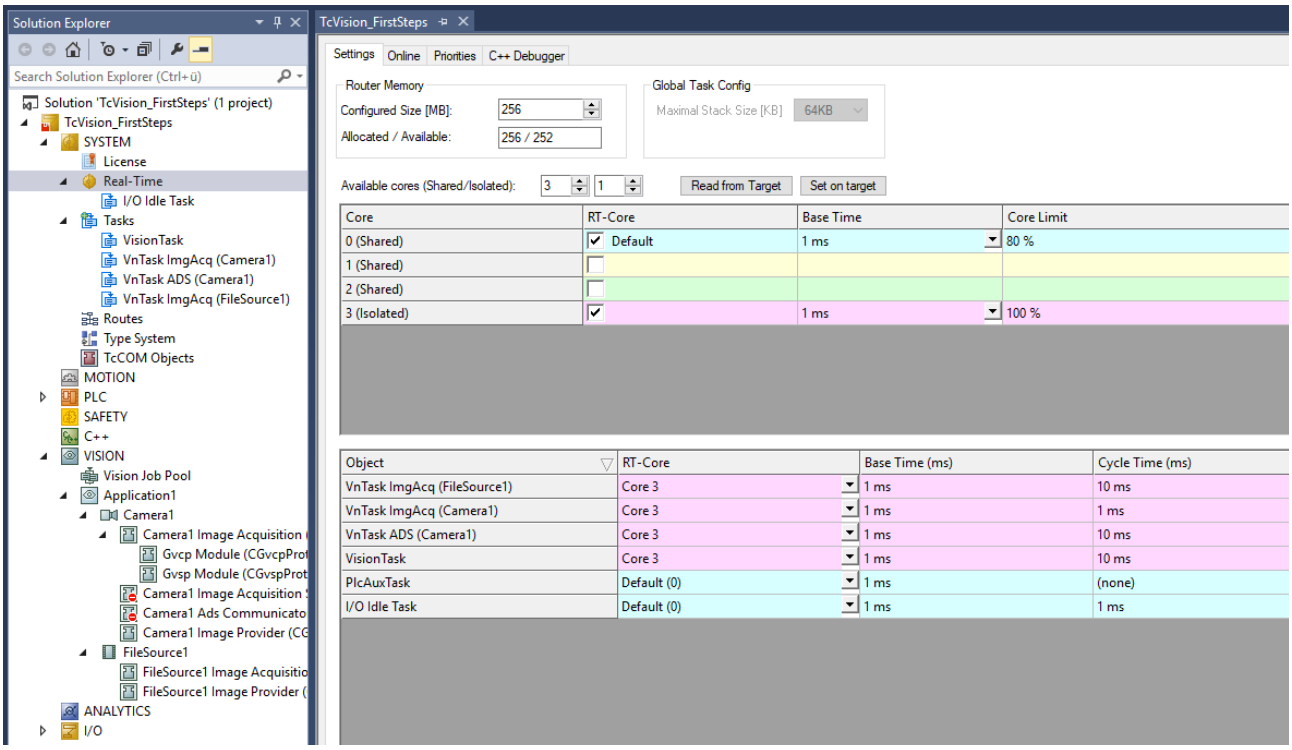
重复使用设备任务

在重复使用设备任务时，应确保它们不会同时被多个 Vision 设备使用。

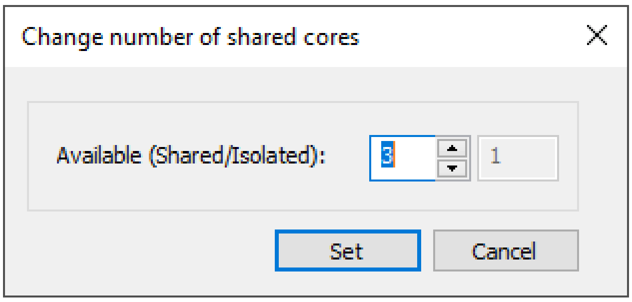
分配给 CPU 核

将任务分配给 CPU 核的操作在 TwinCAT 项目中的 SYSTEM（系统）> Real-Time（实时）> Settings（设置）下进行。在多核系统中，TwinCAT 支持区分 Windows 系统分享核和独立核。使用 Windows 核时，操作系统和 TwinCAT 应用共享处理器时间。实时部分可限制为 10% 到 90%。与此相反，独立核可完全用于 TwinCAT 应用。因此，建议 Vision 应用使用独立核。

要显示目标系统当前的 CPU 核配置，请点击 Read from Target（从目标读取）。



要更改目标系统的 CPU 核配置，请点击 Set on Target（在目标上设置）并选择所需的分享核和独立核分配。



之后必须重新启动目标系统。

建议

我们建议为 Vision 任务（设备和 PLC）至少使用一个专用独立核。如果由于保存或加载图像（通过相机流、文件源控制器或直接从 PLC）而导致 TcVnService 负载较大，我们建议将一个核完全留用于 Windows，或严格限制一个核的实时部分。

如果为一个核分配了多个任务，则必须考虑优先级。一般来说，应为循环时间较短的任务分配较高优先级。

5.1.5 工作任务

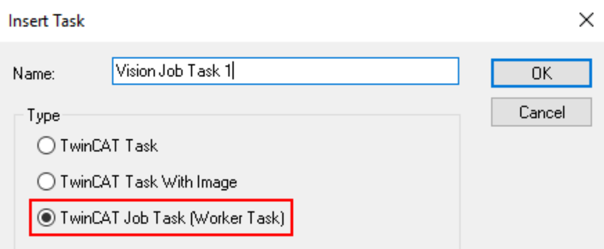
工作任务被 TwinCAT Vision 用于并行化功能。根据使用的系统、其配置、使用的 API 功能、图像内容和工作任务的数量，这可能导致执行时间的缩短。在API 参考资料 [▶ 119]中支持的功能通过注释“可以使用可用的 TwinCAT 工作任务来执行并行代码区域”进行标注。视觉工作池 [▶ 114]用于指定哪些工作任务被 TwinCAT Vision 使用。工作任务只需要创建和配置一次；此后，并行化将在运行时自动进行。

工作任务的优点

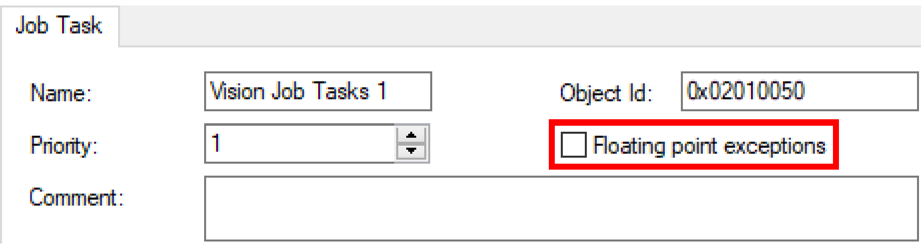
工作任务的实际优点必须在个别情况下检查。这取决于使用的系统、其配置、使用的 AP I功能、图像格式、图像内容和工作任务的数量。

使用

像其他任务一样，工作任务可以通过**解决方案资源管理器 > 系统** >创建，右键单击**任务 > 添加新的项目...**。在以下**插入任务**对话框，选择类型**TwinCAT 工作任务（工人任务）**。



像所有其他任务一样，然后可以通过**解决方案资源管理器 > 系统 > 实时**，进行相应的内核分配。请遵守限制和建议 [▶ 59]下的注意事项。



注意

浮点异常

该设置指定了 TwinCAT 是否检查浮点异常的问题。这里应选择相应的 PLC 任务的配置 [▶ 55]中的相同选项。否则会发生异常和系统崩溃。

哪些工作任务由 TwinCAT Vision 使用通过视觉工作池 [▶ 114]的参数列表进行分配。只要在视觉节点下创建了一个应用程序，就可以在解决方案资源管理器中的视觉节点下找到视觉工作池。

优先次序

视觉工作池访问的优先次序
如果几个并行执行的程序共享视觉工作池，具有最高任务优先次序的程序通常被授予对工作任务的访问权。然而，这种分配是动态的，也就是说，如果具有最高优先级的程序目前处于顺序处理部分，较低优先级的程序可以使用工作任务。一旦高优先级的程序可以并行处理任务，低优先级程序的任务就会被取代。然而，低优先级程序的主任务能够自行终止并行启动的任务，这样就不需要等待新的工作任务的可用性。

一个内核上的多个任务
如限制和建议一节所述，一个工作任务不应共享一个内核。否则，任务优先级适用。

限制

- 工作任务目前只能用于 TwinCAT Vision。
- 不得将一个以上的工作任务分配给同一个内核。
- 工作任务不应被分配到与 Vision PLC 任务相同的内核。
- 工作任务不应被分配到与 PlcAuxTask 相同的内核。

建议

- 每个工作任务都应分配给一个专门的独立内核。
- 然而，如果另一个循环任务与工作任务在同一内核上运行，循环任务的优先级应该高于工作任务。应该注意的是，这限制了并行化，且在最坏的情况下使其无法使用。
- 不得将工作任务分配给默认内核。

如果循环 TwinCAT 任务或 Windows 操作系统同时要求在内核上计算时间，由于工作任务的负载不同，分析会受到阻碍。

5.2 视觉节点

与图像处理相关的组件通过 VISION 节点及其子元件进行配置。它有用于服务配置 [▶ 59]和日志记录 [▶ 61]的选项卡。

显示 VISION 节点

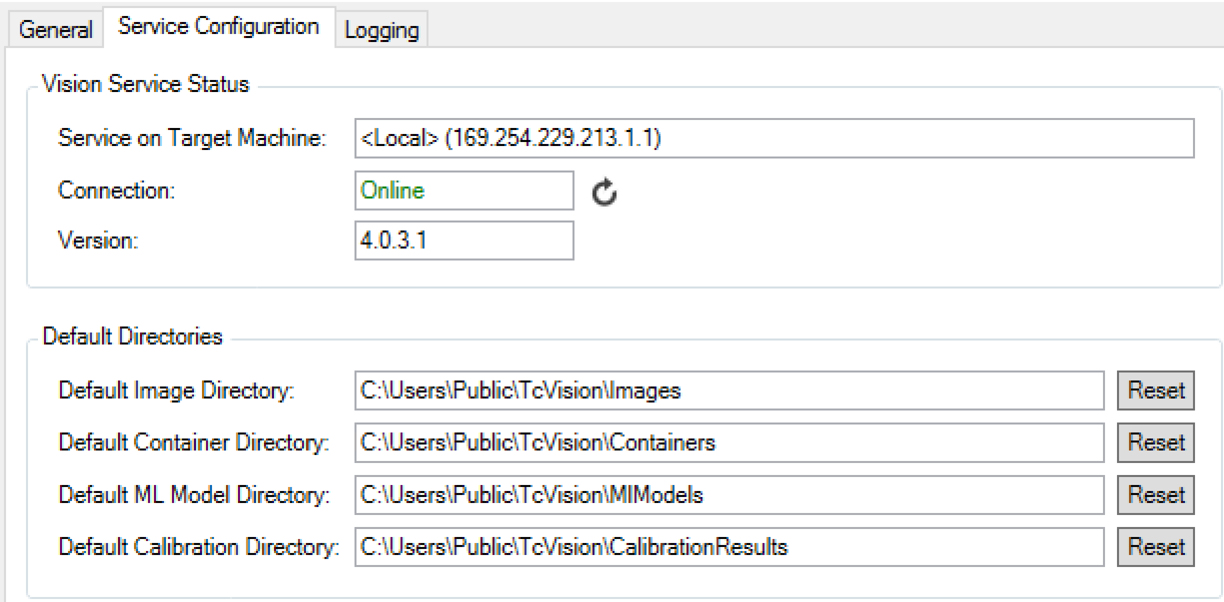
如果 TwinCAT 项目中未显示 VISION 节点，请右键点击 TwinCAT 项目并选择 **Show Hidden Configurations（显示隐藏配置）** > **Show VISION Configuration（显示 VISION 配置）**，以显示 Vision 节点。下一步是创建应用节点 [▶ 62]。

5.2.1 服务配置

Vision Service 为某些实时组件（如功能块）提供非实时对应组件。为此，Vision Service 执行一个 ADS 服务器，支持从实时环境和开发环境进行访问。

Vision Service 支持 TwinCAT Vision API 功能块从实时环境间接访问硬盘，例如加载或保存图像。

Vision Service 在 **VISION** > **Service Configuration（服务配置）**下进行配置。



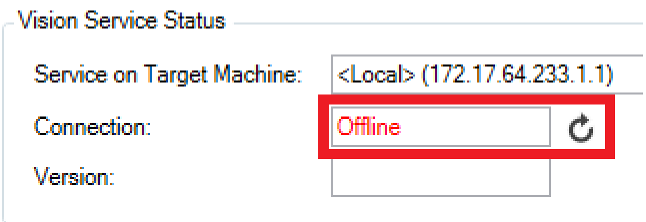
Vision Service 状态

目标机构上的服务	运行 TwinCAT Vision Service 的目标系统。
连接	开发环境与服务之间的连接状态。这是衡量服务是否正常运行的良好指标。
版本	TwinCAT Vision Service 的版本号。

首次使用前，必须在目标系统上安装 Vision Service，请参见设置说明 [▶ 15]。

手动启动 Vision Service

Vision Service 通常与 TwinCAT 一起自动启动。但是，如果 Vision Service 处于脱机状态，则可以通过重启 TwinCAT 来重新激活（点击  或  ）。如果这样做不起作用，或者 Service 反复启动失败，则应在 TwinCAT 中重新注册；请参见：视觉服务不启动 [▶ 2739]。



默认目录

这些路径定义目标系统上的点，默认情况下，TwinCAT Vision 应在这些点上搜索文件并保存。

默认图像目录	功能块读写图像时使用的默认路径： FB_VN_ReadImage [▶ 1529] FB_VN_WriteImage [▶ 1536]
默认容器目录	功能块读写容器时使用的默认路径： FB_VN_ReadContainer [▶ 1527] FB_VN_WriteContainer [▶ 1534]
默认 ML 模型目录	功能块读写 ML 模型时使用的默认路径： FB_VN_ReadMlModel [▶ 1530] FB_VN_WriteMlModel [▶ 1538]
默认标定目录	功能块读写标定结果时默认使用的路径： FB_VN_ReadCalibrationResult [▶ 1525] FB_VN_WriteCalibrationResult [▶ 1532]

所有重要路径的完整列表请参见 见重要路径 [▶ 2736] 部分。

Vision Service 日志

所有重要的服务事件均会写入目标系统上的日志文件 TcVnService.log，其位于 *C:\ProgramData\Beckhoff\TcVnService* 下。如果遇到有关 TwinCAT Vision Service 的支持问题，必须同时发送日志文件和行为发生时间。开发系统还配备一个适用于 Vision 节点下所有组件的日志记录器 [▶ 61]。

Vision Service 缓存设置

默认情况下，Vision Service 会缓存其读取的图像，以便在下次检索时更快提供请求图像。这种优化运行模式可满足大多数应用的要求，因此值得推荐。

如果图像内容发生变化，但文件名保持不变，则可以禁用图像缓存。为此，请打开目标系统上位于 *C:\ProgramData\Beckhoff\TcVnService\settings.json* 的 Vision Service 设置，将 updateCacheOnReadSize 参数的默认值从 0 更改为 1，然后保存文件。

退出 Vision Service

要更改设置，必须首先通过任务管理器终止 TwinCAT Vision Service (TcVnService.exe)，否则将无法应用更改。之后，可在配置模式下重新启动该服务。

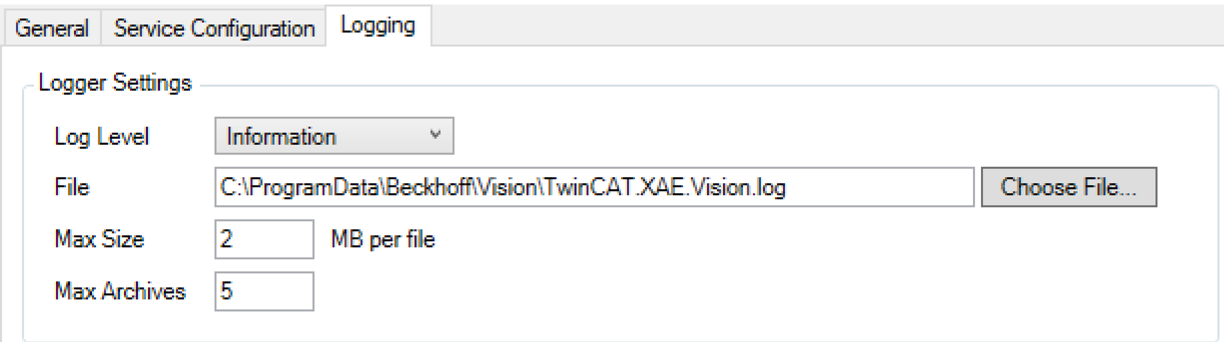
注意

数据不一致

确保同一文件不会被多个进程同时访问，否则会导致文件损坏或数据不一致。

5.2.2 记录

Vision 节点有一个日志记录器，根据日志级别的不同，它可以记录 VISION 节点下 TwinCAT Vision 组件的所有重要事件。设置可在 VISION > Logging（日志记录）下找到。



可提供下列设置：

日志级别

级别	描述
错误	只记录错误信息。
警告	记录错误信息和警告。
信息	记录错误信息、警告和重要事件。[默认]
详细资讯	记录错误信息、警告和所有重要事件。

日志文件

默认情况下，名称为 *TwinCAT.XAE.Vision.log* 的日志文件保存在 *C:\ProgramData\Beckhoff\Vision* 目录中。这可以在 Logging（日志记录）选项卡中更改。日志文件的最大大小和文件的最大数量也是如此。如果达到限制，将覆盖旧数据。

如想有关于 Vision 节点内行为的支持问询，请提供日志文件和行为发生时间。

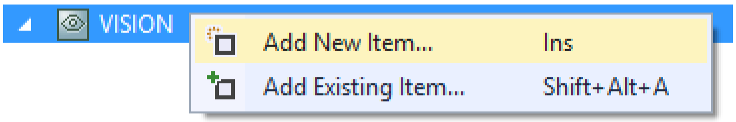
其他日志记录器

Vision Service [▶ 59] 的目标系统上还有一个日志记录器。此外，还有一个用于 TwinCAT Vision 安装的日志文件。所有相关日志记录路径均列于 见重要路径 [▶ 2736] 部分。

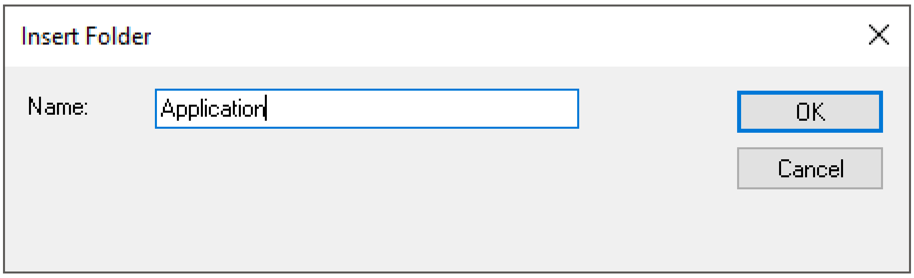
5.3 应用节点

一个 TwinCAT Vision 配置可以包含多个应用节点，所有这些节点都位于 VISION 节点内。创建应用节点的步骤如下：

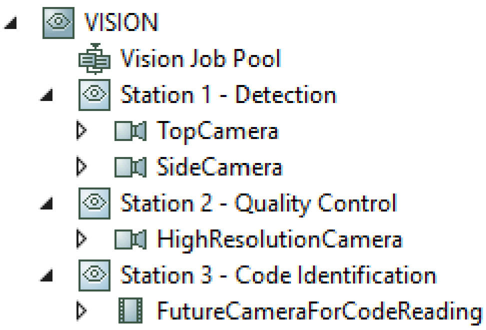
1. 右键点击 VISION node（VISION 节点）→Add New Item…（**添加新项目……**）



2. 输入应用节点的名称，然后点击 OK（**确定**）。



GigE Vision 相机节点 [▶ 63]和文件源节点 [▶ 110]在应用节点下创建。应用节点的目的是整合与主题相关的 Vision 设备。下图显示了一个包含多个 Vision 应用的结构示例：



此外，应用节点还支持单个 XTI 文件中多个 Vision 设备配置的一般导出/导入 [▶ 118]。

5.4 GigE Vision 相机对象

每个 GigE Vision 相机对象代表一个项目中的一个相机。一个项目可以包含多个相机对象。

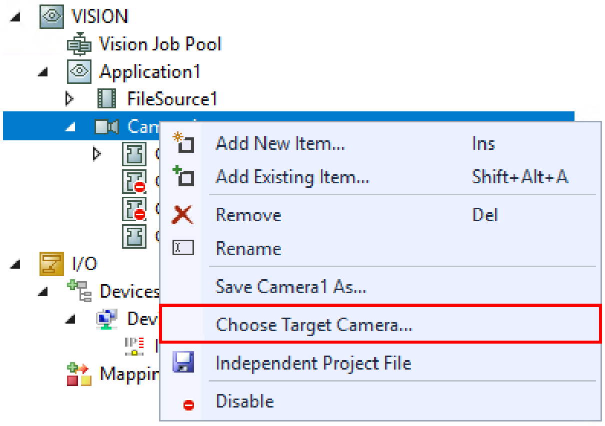
创建一个 GigE Vision 相机对象

创建相机在创建 GigE Vision 相机 [▶ 34]下第一步 [▶ 23]进行说明。此处请参考关于配置网络适配器 [▶ 50]和设备任务 [▶ 55]的说明。

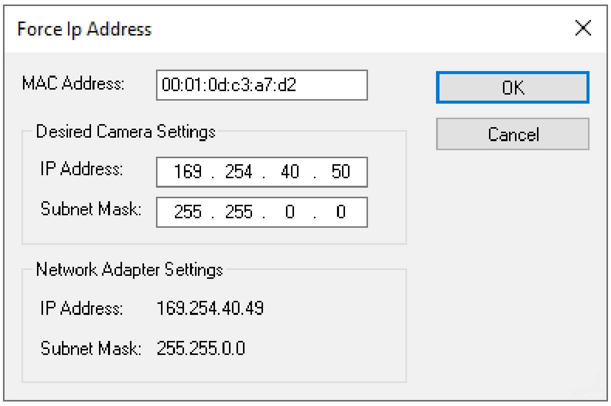
创建相机后，系统会自动尝试从相机中读取 GenAPI 文件。此文件包含相机参数树的完整描述（见GigE Vision [▶ 107]）。在此描述的基础上，配置树就可以在配置助手 [▶ 67]提供，并由相机读取当前的参数值。

选择一个已连接的相机

如需将连接到目标系统的相机与 GigE Vision 相机对象连接起来，可通过右键单击相应的相机对象并选择**选择目标相机**打开**相机初始化助手**。另外，这个对话框也可以通过常规 [▶ 64]选项卡上的连接选项打开。



相机名称	显示所选相机对象的名称。
相机 Ip 地址	保存在 TcCOM 参数中的 IP 地址。在选择相机时，该条目将被覆盖，同时在模块中通过**确定**更新。
发现超时	发现超时描述了连接到相应网络端口的相机的响应所允许的时间。 单位：毫秒（ms） 默认值：1000
为设备设置 Ip 地址	打开以下强制 IP 地址对话框，以设置相机的 IP 地址和子网掩码。指定的 IP 地址和子网掩码被保存在图像采集对象 [▶ 87]的 TcCOM 参数 CameraIPAddress中。
发现设备	搜索程序通过**发现设备**开始。
发现的相机连接结果表	找到相机列表，并显示读出的以下相机信息： 型号名称、制造商、IP 地址、子网掩码、MAC 地址、版本、序列号和用户标识



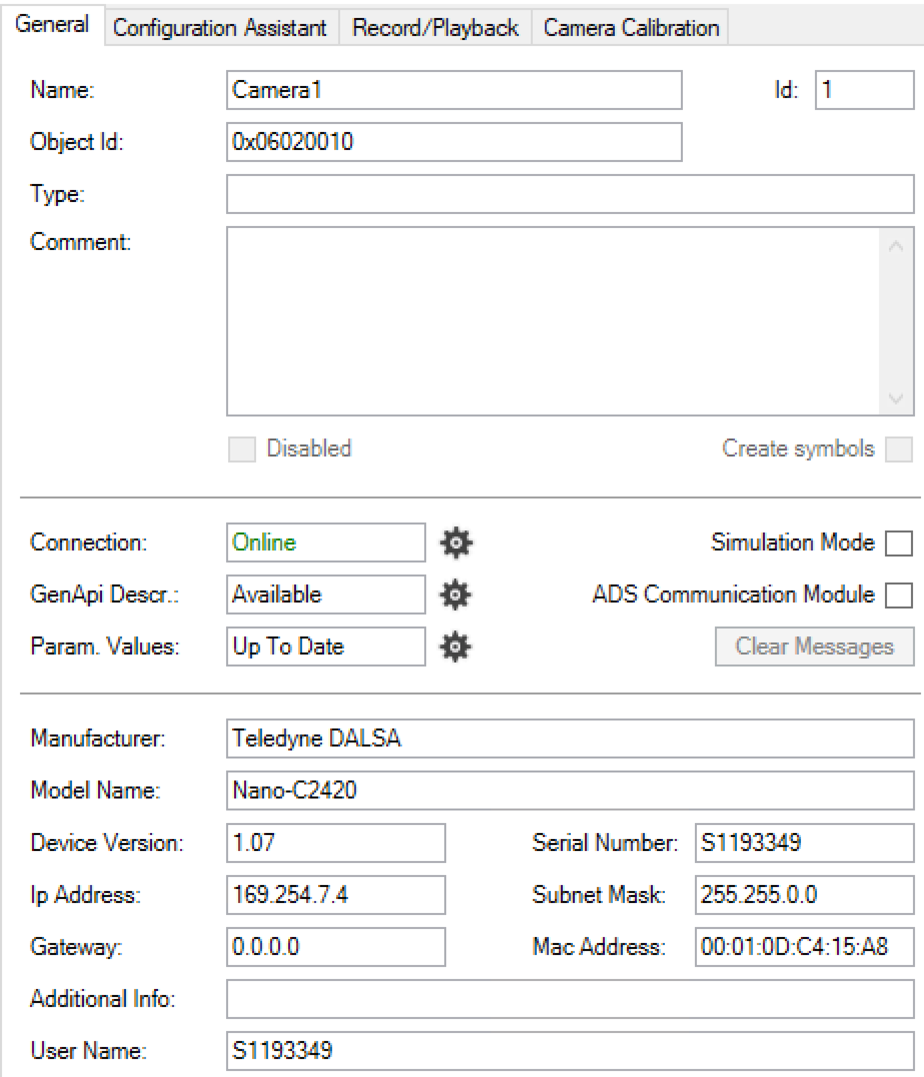
MAC 地址	所选相机的 MAC 地址。
所需的相机设置	输入相机待设置的目标 IP 地址和子网掩码。
网络适配器设置	相机所连接网络适配器的当前 IP 地址和子网掩码。

导出/导入 GigE Vision 相机对象

如果想要保存相机的配置或将其导入项目，请按照导出/导入配置 [▸ 118]一章中的说明进行操作。

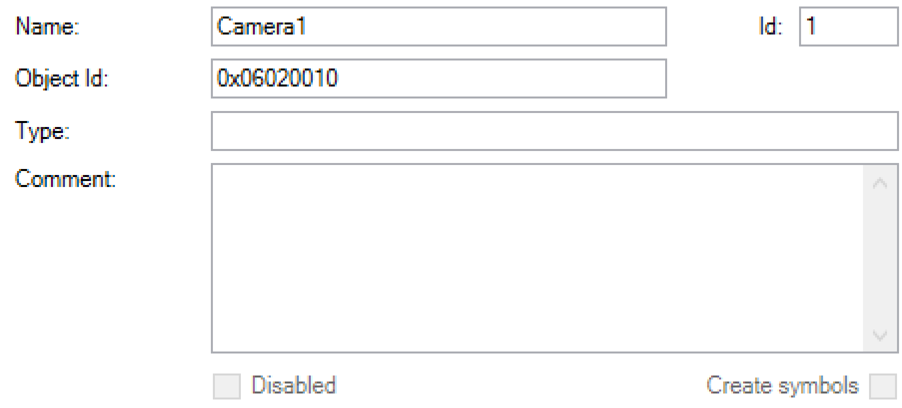
5.4.1 常规

常规选项卡提供了 TwinCAT 中相机对象和所连接相机的概况。



一般对象信息

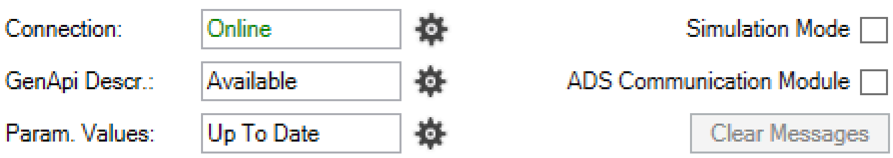
上部包含关于相机对象的一般信息：



名称	选择对象名称。 **请勿多次指定一个对象的名称！**
对象 Id	32 位识别码，在项目中自动分配且唯一。
类型	对象类型。
注释	可自由编辑的用户备注字段。
Id	相机模块的识别号，连续分配。如果删除某个相机模块，识别号将重新空置，且可以分配给新的相机模块。

相机实例的状态

中间部分包含相机实例的状态信息：



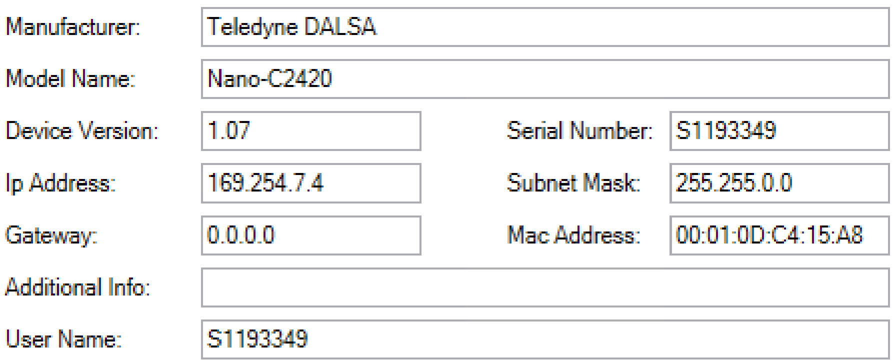
下表解释了各个元素：

连接	显示与相机的连接状态。 通过按钮 ，可以通过**刷新相机状态**更新显示。此外，还可以通过**选择目标相机**打开选择相机的对话框 可能的状态： • **未知**：无法检查与相机的连接。相机模块可能不在工作状态，或者相机模块的创建可能不完整。 • **在线**：与相机的连接已经建立。 • **离线**：没有连接到相机。 • **已禁用**：相机节点被禁用。与相机没有通信。 • **不在配置模式下**：连接状态只能在 TwinCAT 配置模式下显示在这里。 • **加载**：相机模块已被初始化。
GenApi **描述**	这表明 GenApi 结构（配置助手 [▸ 67]中的特征树）是否成功创建。状态选项： • **可用**：GenApi 结构已成功创建。 • **不可用**：未创建 GenApi 结构。不能显示任何特征。 按钮 还可以进行以下操作： • **从相机读取 GenApi 描述**：读取相机的 GenApi 描述。同时，配置树 [▸ 70]中的所有功能都被重置。 • **导出 GenApi 描述**：相机的 GenApi 描述被保存在本地。
参数 值	状态选项： • **最新**：相机的所有功能都有数值，且最新（相机在线）。 • **完整**：相机的所有功能都有数值，但这些值可能已经过时（相机离线）。 • **不完整**：部分相机功能没有数值。 按钮 还可以进行以下操作： • **写入本地更改**：所有偏离相机值的本地参数值（在配置树 [▸ 70]中以橙色标记）都会写入相机。 • **恢复本地更改**：特征树中任何尚未写入相机的本地修改都将被重置。
模拟模式	在模拟模式下，可以将记录的相机流的图像发送到 PLC，而不是由真实的相机提供图像（见记录/回放 [▸ 89]）。 如果模拟模式启用，图像采集对象会自动停用，且图像采集模拟对象会启用。 如果模拟模式停用，图像采集对象会自动激活，而图像采集模拟对象则被停用。 相应的对象链接（见相机详情 [▸ 108]）将分别自动调整。 默认：禁用。

ADS 通信模块	相机流的记录需要，见记录/回放 [▶ 89]。 这个复选框只是激活或停用相机对象的 Ads 通信器对象。 **请注意，如果 Ads 通信器对象被激活，相机拍摄的图像仍将在短时间内被内部引用，因此无法将其转化为可显示的图像。** 更多详情请参见可显示的图像 [▶ 131]。 默认：禁用。
清除信息	使用此按钮可删除此相机对象所写的信息。 该按钮只有在有信息存在时才会激活。

相机信息

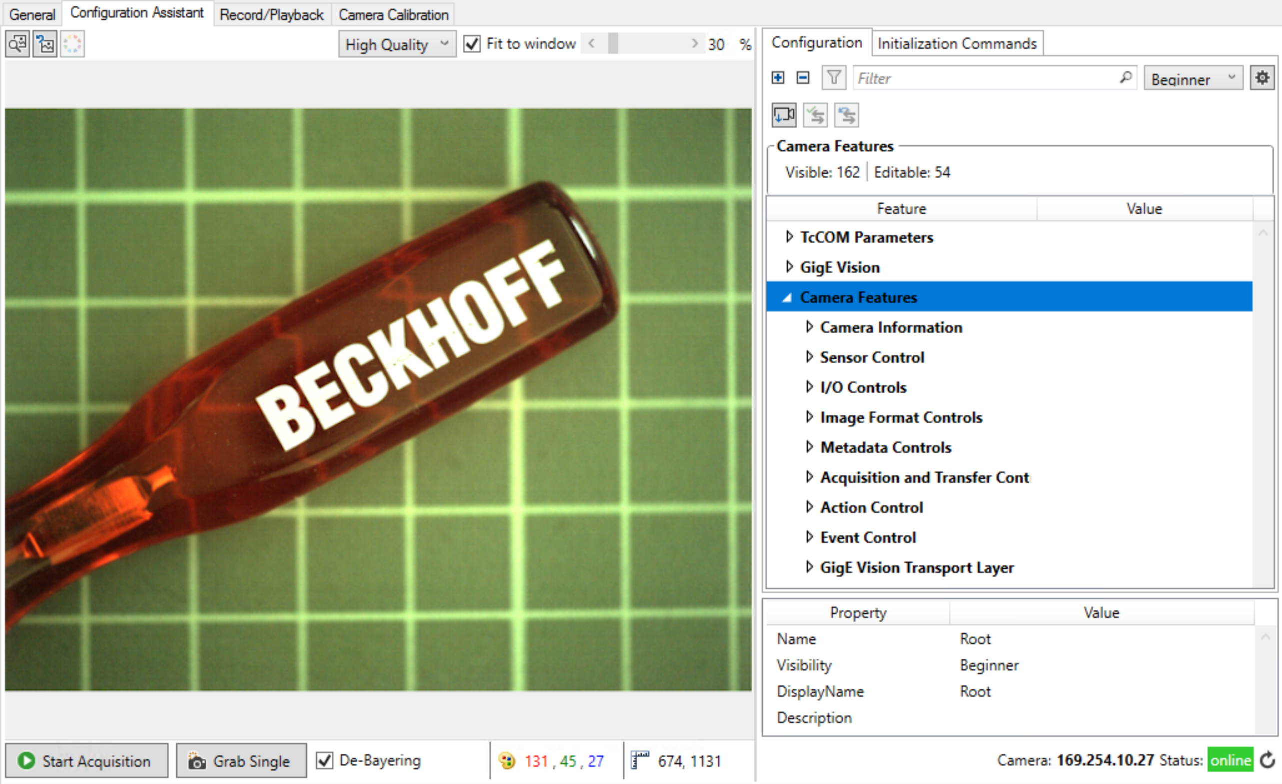
在下半部分，显示的是可以从相机中读取的一般相机信息（如相机的制造商和型号名称）。这些信息也可以在配置树 [▶ 70]中找到。



5.4.2 配置助手

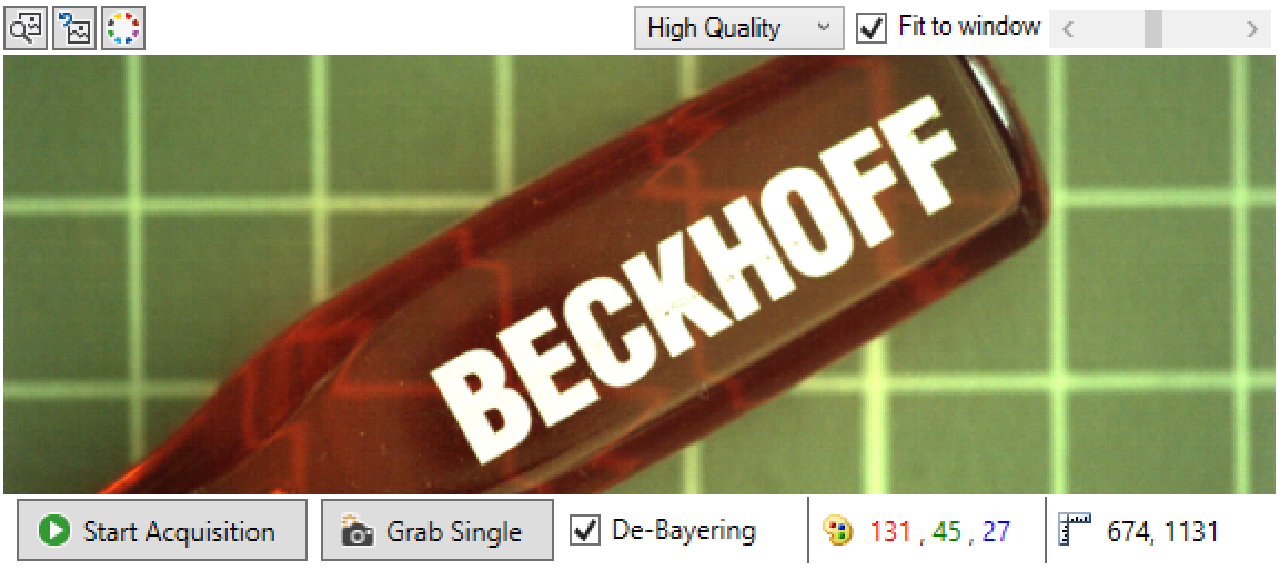
相机参数和通信行为都可以在**配置助手**选项卡上进行配置。配置助手有四个部分：

- 图片预览 [▶ 68]
- 配置树 [▶ 70]
- 初始化命令 [▶ 77]
- 行动和设置 [▶ 74]
- TcCOM 参数 [▶ 87]



5.4.2.1 图片预览

配置助手中左边的区域配有一个预览窗口和一些操作元素。



在这个图像预览中，只有 8 位整数的图像可以正确显示。

下栏

在预览窗口的下面设有开始/停止预览的控制，以及关于图像上光标位置的信息：

开始/停止采集	用于开始或停止图像采集的按钮。
抓取单图	用于触发单个图像的按钮。
De-Bayering	Bayer 格式的图像也可以作为 RGB 图像进行预览。 **如果这个选项启用，来自相机的下一个图像将被转换，但不是目前的图像。**
像素值	这里表示图像上鼠标光标位置的像素的颜色或强度值。
光标位置	显示鼠标光标在图像上的位置。请注意，图像的零点总是在左上方。

图像采集和触发设置

请确保相机的图像采集和触发设置是以相机对命令作出反应的方式选择的。
，例如，如果相机的触发模式处于活动状态，则相机不会在图像采集开始时自动发送图像。关于更多详细信息，请参见实例图像采集和触发 [▶ 2729]。

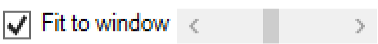
上栏

在预览图像上方设有一些控制按钮，以方便相机配置和调整图像显示。

相机配置

设置 ROI	点击这个按钮后，可以通过按住鼠标左键直接在预览图中绘制一个 ROI。这个 ROI 将被写入相机参数中，这样下一幅图像就会以设定的 ROI 发送。
重置 ROI	单击此按钮后，先前设置的 ROI 将被删除，即图像大小和偏移量将被设置为相机的默认值。
白平衡	点击之后将在彩色相机上执行白平衡。相机必须支持以下 GenAPI 参数才行：*BalanceRatioSelector*和*BalanceRatio*（或*BalanceRatioAbs*，用于老式相机）。此外，相机图像必须包含一个白色（不过度曝光或反射）物体，如一张纸，以实现白平衡。 在白平衡之后，必要的值被保存在名为 GenAPI 参数中，并且可以被采用到初始化命令 [▶ 77]或用户设置中。相机上的自动持久性存储可能是有关相机所特有的，但不是标准功能。 这个按钮只有在有有效的颜色格式（3 通道图像为 RGB 或 1 通道图像为 Bayer 格式）时才会激活。

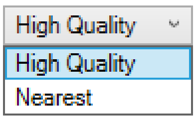
图像大小



图像显示尺寸的调整选项。图像可以在原始尺寸的 20% 到 6400% 之间进行缩放。点击**适应窗口**，以自动设置显示尺寸，使预览能最好地利用屏幕上的可用空间。

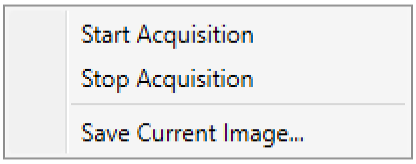
插值

在这里，你可以选择不同的插值方法，将图像从原始尺寸扩展到显示尺寸。



高质量	高质量的插值。
最近	通过采用最近的像素值进行插值。

右键单击图像

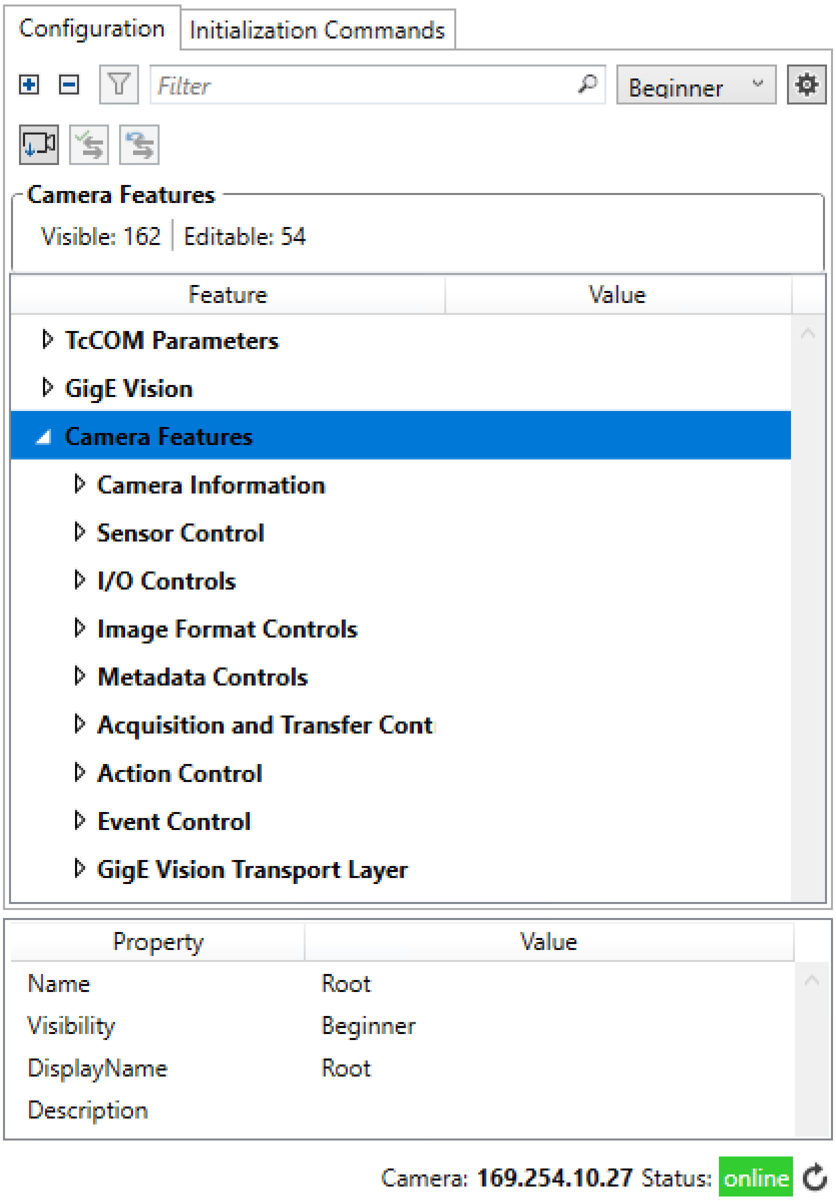


开始采集	开始图像采集。
停止采集	停止图像采集。
保存当前图像	将图像保存为位图、JPEG 或 PNG。

5.4.2.2 配置

Configuration（配置）选项卡由 3 个区域组成，分别如下所述：

- 滤波器和控制器
- 配置树
- 属性窗口

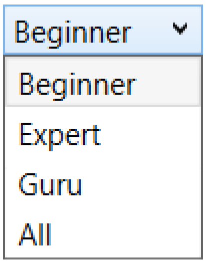


滤波器和控制器

滤波器元件

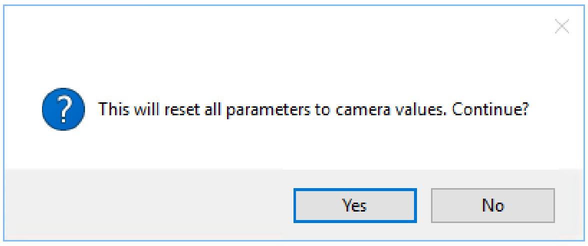
借助滤波器元件，可以在配置树和初始化命令列表中快速轻松的找到各功能：

- 点击 符号可展开所有树元素。所有参数均可见。
- 点击 符号可折叠所有树元素。只显示主要组别。
- 点击 图标将对树项目进行滤波，只显示有变化的行。该按钮只有在显示更改时才会启用。
- 搜索行按特定字符串或（十六进制）地址对参数进行滤波。
- 根据 GenICam 标准，可选择 Beginner（初级）、Expert（专家）、Guru（大师）和 All（全部）四个可视级别。每个特征的可视级别都存储在相机的 GenApi 文件中。



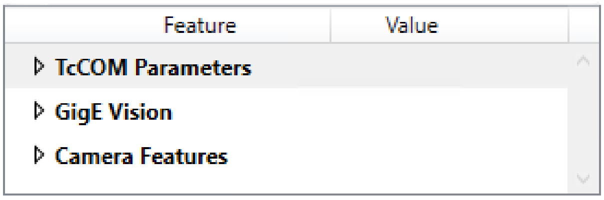
控制器

这些按钮的功能从左到右依次为：

读取相机参数	所有参数值均从相机读取并写入配置树。这必须首先通过以下对话框确认： 
写入本地更改	所有偏离相机值（在配置树中以橙色标记）的本地参数值都会写入相机。
还原本地更改	特征树中任何尚未写入相机的本地更改将被重置。

配置树

配置树包含三个主要组别：



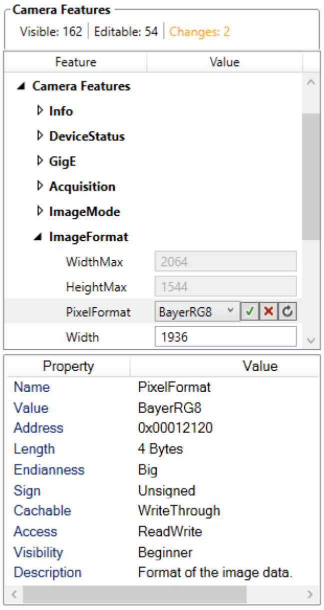
- **TcCOM 参数**：对 TwinCAT 与所连接相机之间的通信行为进行参数设置。有关这些参数的解释，请参见 TcCOM 参数 [▶ 87] 章节。
- **GigE Vision**：根据 GigE Vision 标准得出的相机相关信息，以及 IP 配置选项。根据相机的不同，这些设置也可能在相机特征树中提供。

- **相机特性**：相机参数设置基于 GenICam 标准。所有相机特性的描述都由相机自动从 GenAPI 文件中读取。特性可以是一个可执行命令、一个（可改变）参数或一个组织元件。有关某些设置（如图像采集/触发、合并读出、曝光时间等），请参见 相机配置样本 [▶ 2726]。
  有关特定相机参数的详细信息，请参见相应的供应商文档。此外，GenICam 标准 [▶ 108]还描述了相机特征树的结构方式。

相机特征信息

配置表上方显示有关相机功能的三项信息：

- 可见：所有当前可见或显示特征的数量，具体取决于活动滤波器设置。
- 可编辑：可编辑的“可见”特征数。这个数字一般取决于当前相机的参数设置，因此会有所不同。
- 更改：相机特征树中的更改总数。只有存在更改时才会显示该数字，与滤波器设置是否激活无关。



如果点击配置树中的某个特征，该特征的属性会显示在树下方。

参数设置：

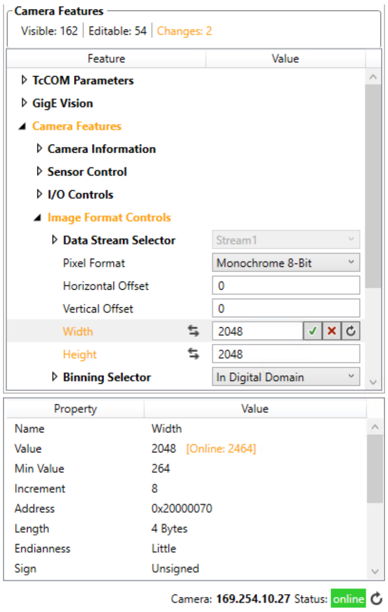
可更改特征可通过文本框、滑块、组合框或复选框进行更改。只读特征显示为灰色。每个可更改配置元件都有以下按钮：

 **写入值**
将本地更改写入相机。

 **重置值**
如果本地值与相机值之间存在差异，则删除本地更改。

 **加载值**
从相机加载参数的当前值。

如果设置的参数值与相机上的实际值有偏差，配置树中的参数和更高级别元件将被标记为橙色。此外，有更改的行会用图标标出，偏差也会显示在属性窗口中。表格上方橙色的“更改”注释一般表示列表中的更改。

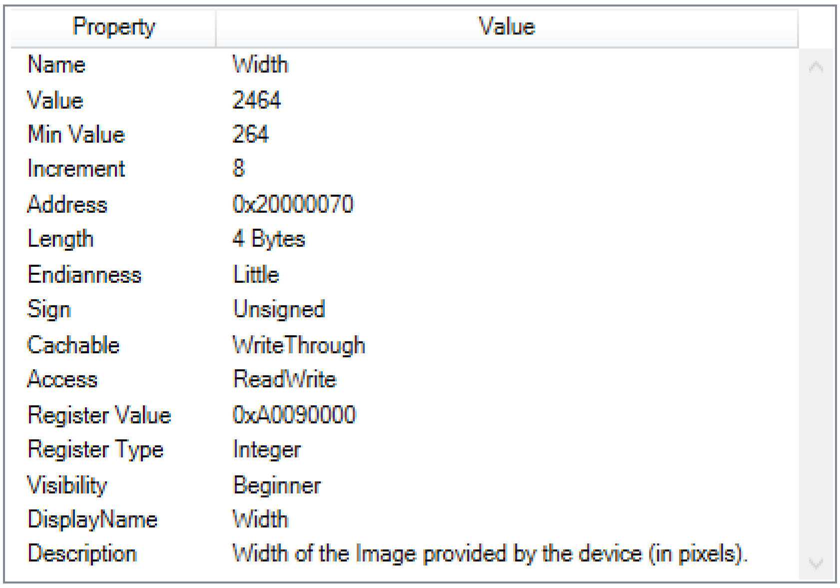


离线更改

如果相机处于脱机状态或目标处于运行模式，也可以更改配置树中的参数。在这种情况下，参数不会直接写入相机，而只会保存在项目中。因此，可以先离线更改参数，之后再与相机同步。

特征属性

单独特征的属性显示在配置树下方。其中包括各相机寄存器的大小和地址、访问权限和说明。



Property	Value
Name	Width
Value	2464
Min Value	264
Increment	8
Address	0x20000070
Length	4 Bytes
Endianness	Little
Sign	Unsigned
Cachable	WriteThrough
Access	ReadWrite
Register Value	0xA0090000
Register Type	Integer
Visibility	Beginner
DisplayName	Width
Description	Width of the Image provided by the device (in pixels).

属性	描述
名称	参数名称。
值	参数的本地值。如果本地值与相机值有偏差，相机值还会显示为橙色。
地址	相机上参数的十六进制地址
长度	参数的大小（以字节为单位）
字节顺序	参数的字节序列。使用功能块 FB_VN_ReadRegister [▶ 1506] 和 FB_VN_WriteRegister [▶ 1513] 时必须遵守该属性。
签名	• **已签名**：参数已签名 • **未签名**：参数未签名
可缓存	指定未连接相机时是否缓存参数值。 • **写透**：值既写入相机，也写入项目缓存和项目内存。 • **无缓存**：根据相机的 GenAPI 描述，此相机参数的值不应保存在本地。因此，如果没有相机连接，此参数可能为空。
访问	在相机上读写此参数的访问权限。
寄存器值	参数的寄存器值。如果是已知类型，则这可能与显示值有偏差。示例： 名称：PixelFormat 值：BayerRG8 寄存器值：13
寄存器类型	寄存器的数据类型。该类型表示使用哪个读/写功能块。
可见性	参数的可见级别。可能值：**初级**、**专家**、**大师**和**全部** 参数的可见性可以通过 滤波器元件 [▶ 71]进行设置。
显示名称	要显示的特征名称。
描述	参数的含义

例如，如果要从 PLC 读取或更改参数，就需要这些属性。基本上，必须注意地址、长度、字节顺序和寄存器值。相应的功能块可在相机寄存器访问 [▶ 1503]下的 API 参考中找到。

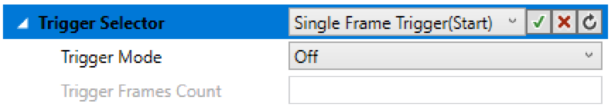
根据寄存器的数据类型，可以添加更多属性，如最小值/最大值/增量。

选择器

选择器是用于分组和选择多个子特征的参数。要使用子特征，必须首先将选择器参数设置为适合的参数（在配置助手和 PLC 中）。

如果选择器参数没有地址属性，则不指向相机寄存器。它的作用只是在配置助手中提供更清晰的显示。在这种情况下，可以直接通过 PLC 更改子参数。

为了在创建初始化命令时考虑选择器的所有子参数，会自动对所有子特征进行迭代。相应的设置载于 操作和设置 [▶ 74]。选择器的一个示例是触发选择器：



5.4.2.3 操作和设置

操作

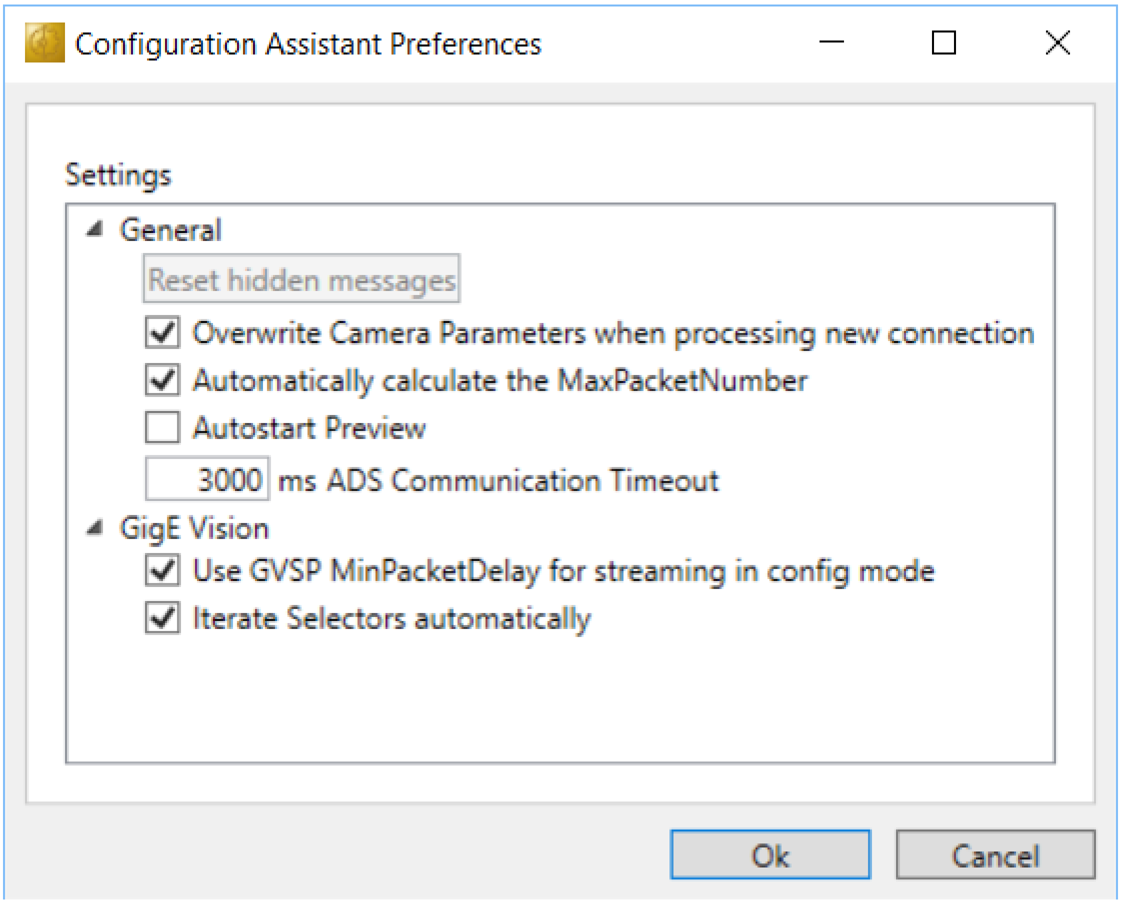
右上方是操作按钮 ，可用于进一步操作或进行附加设置：

导出 GenApi 描述……	相机的 GenAPI 描述本地存储在指定路径中。
偏好……	下述设置窗口打开。

无法加载已保存 GenAPI 文件

无法加载已保存 GenAPI 文件，因为需要与相机连接才能读取某些相机特征。

设置



重置隐藏信息	在某些对话框中，可以使用 Remember my decision（**记住我的决定**）来记住所选选项，例如在初始化命令 [▶ 77]的查询对话框中。 如果选择了 Remember my decision（**记住我的决定**），下次将不会出现相应的对话框，所选选项将自动执行。 在这种情况下，可以使用 Reset hidden messages（**重置隐藏信息**）按钮取消设置。
加载项目时覆盖相机参数	此设置控制在加载项目时是否应将保存的相机值直接写入相机。这样可以确保相机自动处于配置项目时最后保存的状态。 这只有在加载项目后首次初始化相机对象时才会起作用，前提是当时相机已连接且 TwinCAT 处于配置模式。 默认：已激活
自动计算 MaxPacketNumber	如果激活此选项，则会在读出所有相机参数后自动确定图像所需的（和允许的）最大数据包数。计算基于图像大小和像素格式。只有当计算值大于当前值时，才会将其写入负责的 GVSP 对象。该功能可根据实际图像数据量自动调整为待接收图像预留的空间。 默认：已激活。
自动启动预览	如果激活此选项，则在可能情况下，打开配置助手 [▶ 67]时会自动显示相机的实时图像。 默认：禁用。
ADS 通信超时	通信超时与 TwinCAT 中相机实例内部的通信有关。只有当 GVCP 或 GVSP 模块中的其他超时也需要增加时才需要对其进行增加。 单位：毫秒（ms） 默认：3,000（与相机相适应）
在配置模式下为实现流式传输使用 GVSP MinPacketDelay	如果激活该选项，来自 GVSP 模块的 MinPacketDelay 将用于减少配置模式下的带宽。如果关闭此选项，相机的帧率反而会降低。 默认：已激活。
自动迭代选择器	激活此选项后，作为选择器的相机枚举参数会自动迭代。 选择器参数的值用作寻址其他参数的索引。通过改变一个选择器值，可以相应改变从属参数，从而实现相机参数设置的一致性。 如果没有这种迭代，就只能描述当前所选选择器的参数。 这既适用于配置树 [▶ 70]中的设置，也适用于创建初始化命令 [▶ 77]的情况。 默认：已激活。

注意

对初始化命令的影响

如果更改了 Iterate Selectors automatically（**自动迭代选择器**）选项，则必须重新创建初始化命令 [▶ 77]。点击 Initialization Commands （初始化命令）选项卡上的 Create（创建）按钮可重新创建初始化命令。

5.4.2.4 初始化命令

初始化命令的目的是为相机保存特定的参数配置，且如果需要，还可以将相机设置为这种配置。这个功能十分必要，因为相机在重启后总是反映出交付状态。初始化可以在启动 TwinCAT 时自动进行，或者也可以通过 PLC 中的方法调用来手动进行。

初始化命令与 EtherCAT 设备的启动列表或用户设置相类似。用户设置是 GenAPI 的一个功能，可以保存和自动加载相机中的配置。用户设置并非 GigE Vision 标准所要求，因此并非所有相机都提供。

为了能够为所有 GigE Vision 相机统一提供这种功能，TwinCAT Vision 通过初始化命令提供了可能性。当然，如果可用，也可以用用户设置作为替代。

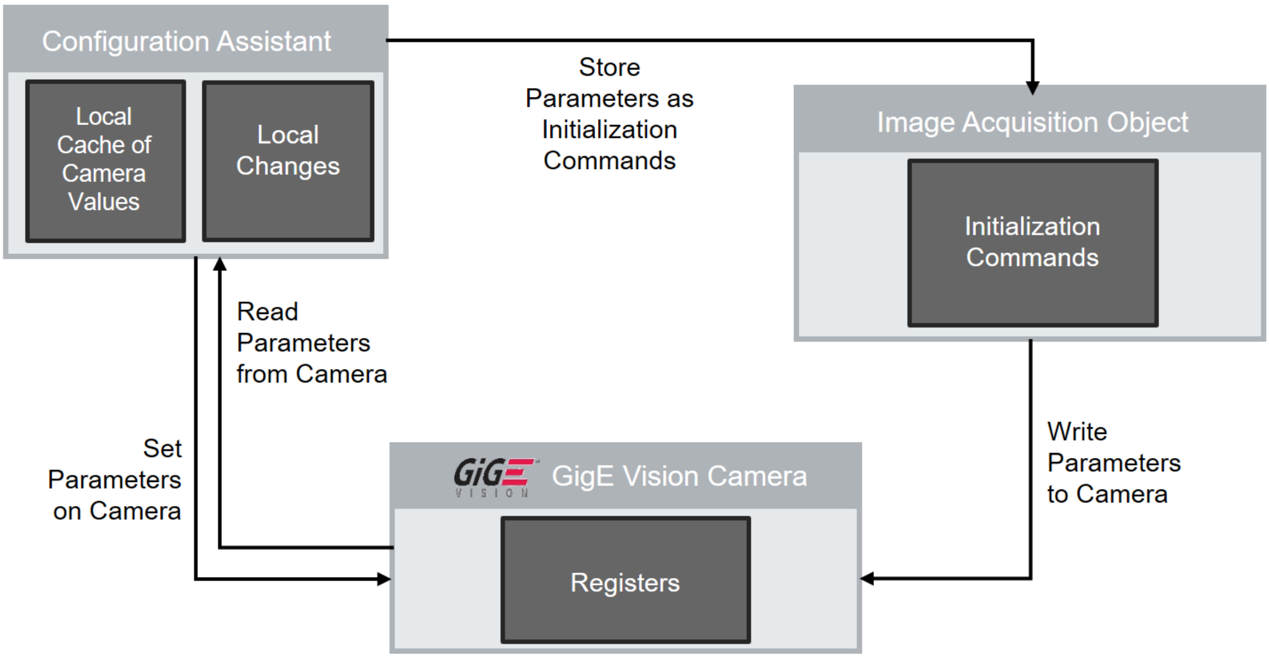
除了这些一般信息外，初始化命令可以按主题划分为以下几个方面：

- 配置 [▶ 78]
- 创建 [▶ 82]
- 故障排除 [▶ 83]

启动过程后的相机状态

在 GigE Vision 相机启动后，它的配置中默认包含出厂设置。为了恢复以前的配置状态，最初必须通过用户设置、初始化命令或在后台由相机助手将相应的参数写入相机。

初始化命令的作用可以描述如下：



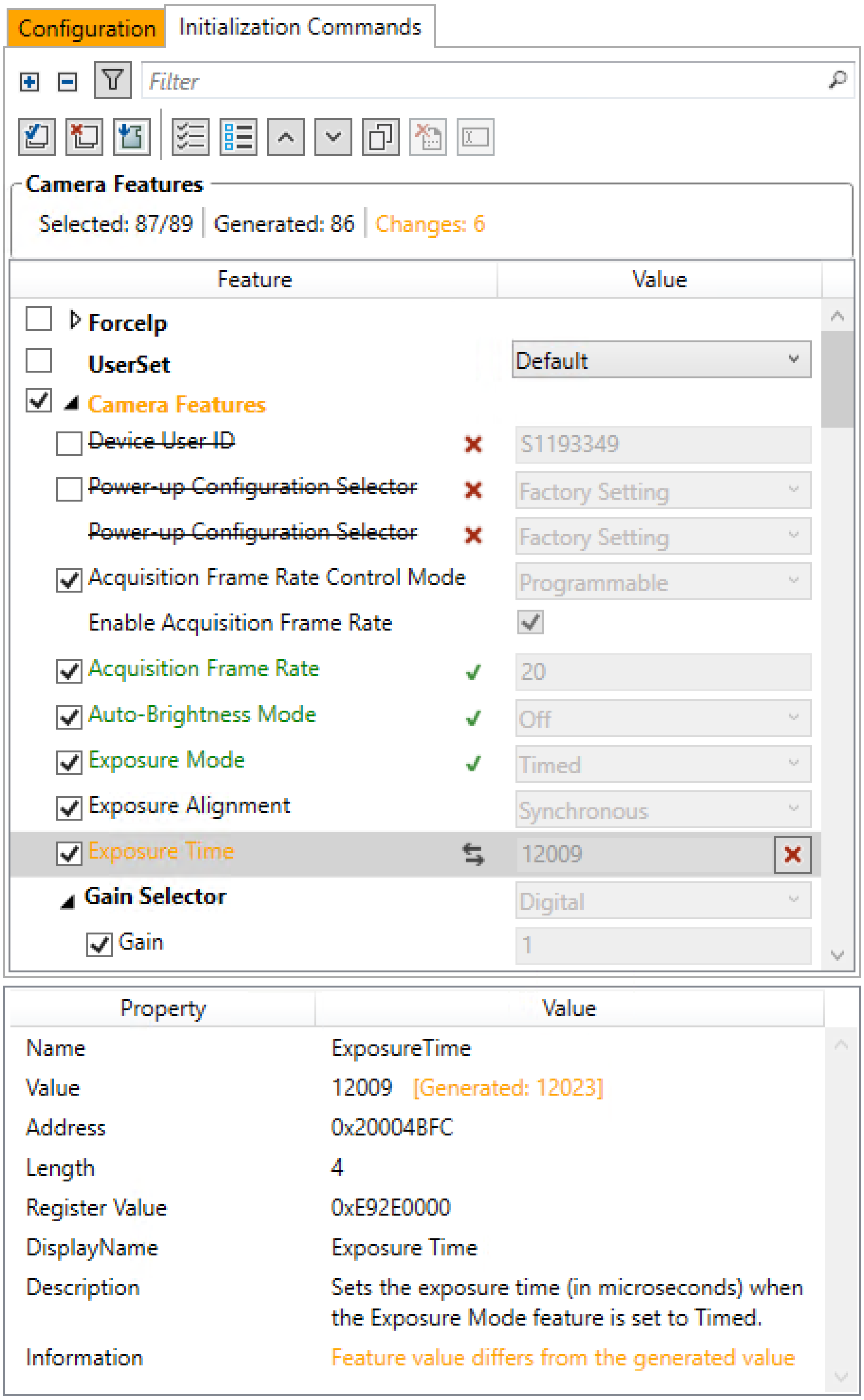
读写操作的触发方式如下：

操作	触发器
从相机中读取参数	• 设置相机后的初始读数 • 重启 TcCOM 对象时读取（例如，当点击**加载设备**时） • 点击**读取相机参数**
向相机写入个别参数	• 在配置树 [▸ 70]中，通过点击**写入值**，对参数进行单独更改和写入。 • 点击**写入本地更改** • 如果在设置 [▸ 74]中**加载项目时覆盖相机参数**激活，在加载 TwinCAT 项目后首次连接相机时
根据当前参数值在相机对象中创建初始化命令	• 在初始化命令选项卡中点击**创建并测试所选的初始化命令**
将初始化命令中的参数写入相机	• 如果图像采集 TcCOM 对象被切换到 OP 状态（启动 TwinCAT），取决于图像采集 [▸ 87]对象中的参数 InitializationAutoMode。 • 通过功能块FB_VN_GevCameraControl [▸ 1550]的一个方法调用

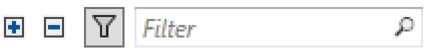
5.4.2.4.1 配置

Initialization Commands（初始化命令）选项卡由 3 个区域组成，分别如下所述：

- 滤波器和控制器
- 配置列表
- 属性窗口



滤波器和控制器



滤波器元件

借助滤波器元件，可以在配置树和初始化命令列表中快速轻松的找到各功能：

- 点击 符号可展开所有树元素。所有参数均可见。
- 点击 符号可折叠所有树元素。只显示主要组别。

- 点击 图标将对树项目进行滤波，只显示有变化的行。该按钮只有在显示更改时才会启用。
- 搜索行按特定字符串或（十六进制）地址对参数进行滤波。

控制器

左侧一组按钮提供基本功能，并始终处于激活状态。从左到右的功能依次为：

按钮	描述
创建并测试选定初始化命令	初始化命令使用配置列表的当前值创建。 **参数也会写入相机并回读以供测试。**
删除更改和初始化命令	在 TcCOM 对象中创建和存储的初始化命令将被删除。这也适用于对列表所做的所有其他更改。 **这会将整个列表重置为初始状态。**
选择相机寄存器类别的所有特征	相机寄存器类别下的所有条目均会被选中。
取消选中相机寄存器类别的所有特征	取消选择相机寄存器类别下的所有条目。

右侧一组按钮提供特殊功能，仅对相机寄存器节点下的条目有效。从左到右的功能依次为：

按钮	描述
向上移动元素	选中元素将上移一个列表位置。
向下移动元素	选中元素将下移一个列表位置。
复制元素	创建所选元素的副本。设定值也会被复制。 **复制条目的值以后不能更改。**
删除复制元素	删除先前复制元素。
重命名复制元素	打开一个窗口，以重命名复制项目。

配置列表

配置列表包含三个主要组别：

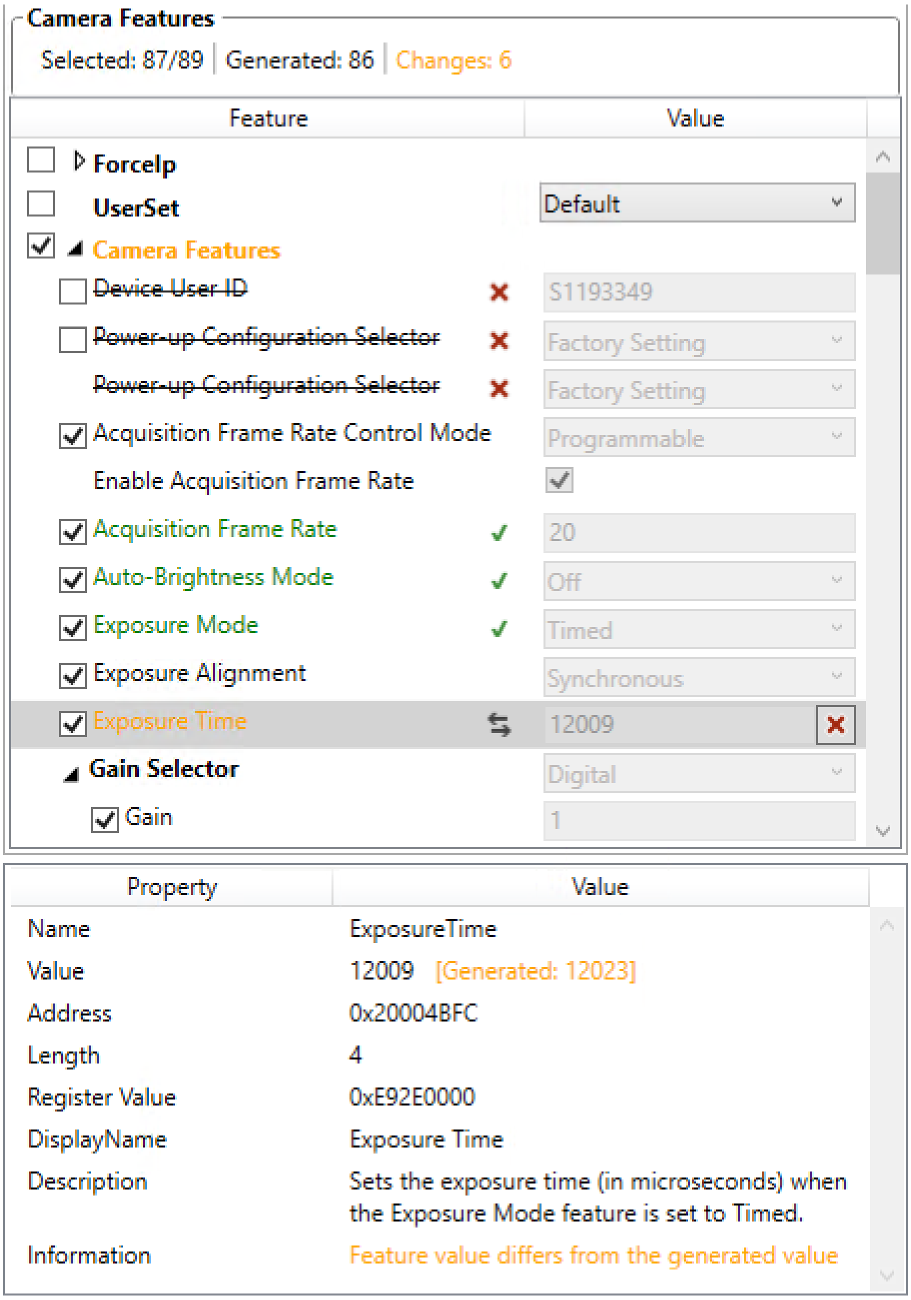
- ForceIP：如果此选项处于激活状态，则在运行时初始化相机时会执行 ForceIp。使用来自图像采集模块 [▶ 87]的 CameraIPAddress。由于命令会发送到相机相应的 Mac 地址，因此在更换相机时，必须使用更改后的地址重新生成初始化命令。使用的数值仅在此处显示。默认：禁用。
- **用户设置**：某些相机具有用户设置功能，可加载存储在相机中的参数图像，以便对相机进行初始化。如果相机提供此功能，则可在此处选择所需的用户设置。如果该选项处于激活状态，则选择和加载相应用户设置的命令会写入初始化命令。默认：禁用。
- **相机特征**：该组别列出在读取 GenApi 文件中声明为可写入并且在当前相机配置中可用的所有寄存器。这与存储的可见性无关，因此也会列出不可见的寄存器，这些寄存器在配置树中只在“All（全部）”可见级别下显示。初始顺序与 GenApi 文件中的顺序一致。如果相机特征选项处于激活状态，则在其下面选择的所有寄存器都将用于创建初始化命令。可以选择或取消选择单个寄存器，并通过相应的操作元件进行更改。地址相同且仅掩码不同的寄存器仅用复选框显示，因为它们不能独立写入。对已创建初始化命令的更改会用颜色和图形标识。默认：已激活。

当选择或取消选择其中一个组别或更改组别下的设置时，必须重新创建初始化命令。使用用户设置时，初始化命令包含必要的用户设置命令，而非参数。用户设置和相机寄存器选项互斥。

相机特征信息

配置列表上方显示有关相机特征的三项信息：

- 已选择：从最大可用特征数中选择的所有当前相机特征数。这通常取决于当前相机的参数设置，因此会有所不同。
- 已生成：项目中存储的最近创建初始化命令的数量。
- 更改：相机特征列表中的更改总数。只有存在更改时才会显示该数字，与滤波器设置是否激活无关。

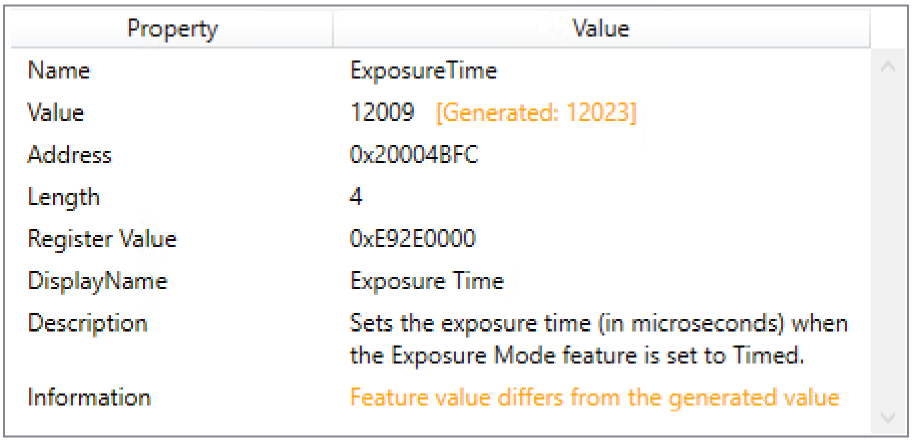


选择器功能

如果在设置 [▸ 74]中激活了**自动迭代选择器**，则“选择器”特征的每个值都有一个单独的组别。在该组别中，会显示所有取决于选择器值且可写入的寄存器。因此，组别和寄存器名称可能出现不止一次，在这种情况下，值或寄存器地址会有所不同。如果设置未启用，则每个选择器只有一个条目。

属性窗口

“属性”窗口显示所选特征的以下信息：



属性	描述
名称	参数名称。
值	参数的本地值。如果本地值与创建的初始化命令值不同，则该特征显示为橙色。
地址	相机上参数的十六进制地址。
长度	参数的大小（以字节为单位）。
寄存器值	参数的寄存器值。
显示名称	特征显示名称。
描述	参数的含义。
信息	有关值是否不同（橙色），特征已添加（绿色）或删除（黑色，删除线）的信息。

5.4.2.4.2 创建

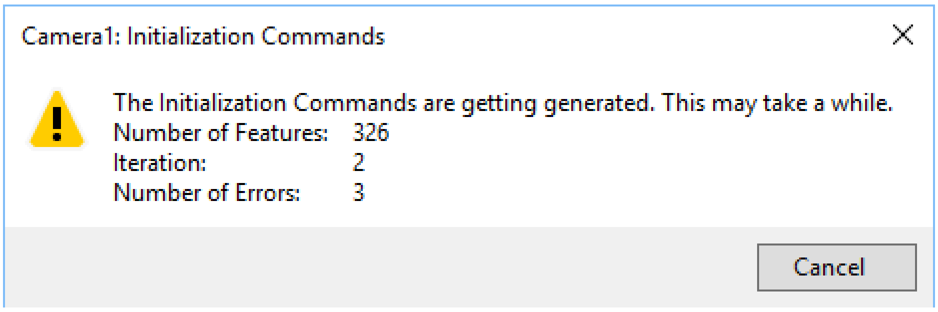
初始化命令的创建只能通过初始化命令选项卡上的按钮**创建并测试选定的初始化命令**启动。在此之前，应该检查设置或更改，以确保所有需要的寄存器都已选中或者不需要的寄存器已取消选择并且寄存器的值与操作所需的设置相对应。

创建步骤

初始化命令的创建包括几个在后台执行的动作。

- 测试向所连接的相机写入初始化命令。
- 测试从所连接的相机中读取先前写入的初始化命令。
- 如果检测到数值差异或错误，会出现一个包含详细信息的列表。如果没有发生关键错误，用户仍然可以接受无错误的初始化命令。
- 将初始化命令数组写入相机的图像采集 TcCOM 对象中。

通过测试写入和随后读取，可以验证相机是否接受初始化命令。在生成过程中会显示以下窗口之一：



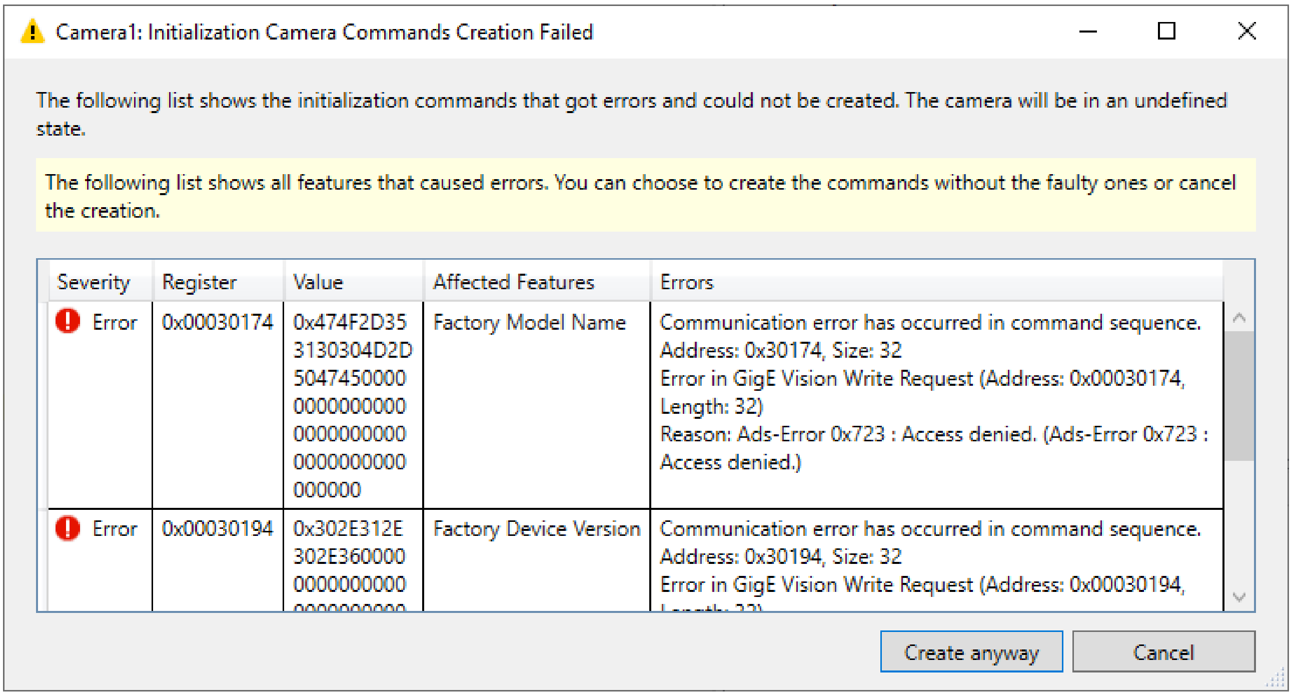
显示以下状态信息：

特征数量	创建初始化命令的参数数量。
迭代	以前尝试向相机写入初始化命令的次数。在成功的情况下，该数值为“1”。如果失败，这个值会增加，同时在没有错误参数的情况下再次尝试将初始化命令成功地写入相机。通常情况下，不应发生超过两次的迭代；然而，如果发生，必须停用图像采集的 TcCOM 参数中的串联。
错误数量	已发生的错误数量。这些在下面的窗口中会有更详细的描述。

如果向相机写入初始化命令的工作非常迅速和成功，该窗口又立即消失且不可见。

消息

在成功创建初始化命令后，相机（由于测试写入）完全具有以后在操作中写入相机的配置。如果在测试向相机写入初始化命令时出现错误，这些错误将显示在以下窗口中：



附图 1：创建初始化命令时的非关键性错误

只要没有发生重大错误，就可以通过点击**仍然创建**创建并保存初始化命令。在这种情况下，错误的参数将被排除在初始化命令之外，因此以后不会写入相机。

在发生重大错误的情况下，初始化命令无法保存。因此，无法点击**仍然创建**。首先，必须纠正错误的原因。

这些错误情况及其纠正方法如下：故障排除 [▶ 83]。

创建初始化命令时的警告

除了错误之外，还可能出现警告。这些大多只是表明相机没有接受和纠正某个数值。

5.4.2.4.3 故障排除

对于基本相机通信和参数化以及初始化命令创建，主要涉及三个部分：

- GenApi 设备描述文件的内容
- 相机行为
- 以及GevImageAcquisition [▶ 108]对象行为。

GenApi 设备描述文件

GigE Vision [▶ 107]规范包括 GenICam 规范的 GenApi 模块，该模块描述了相机所支持的功能。这种描述采取 XML 设备描述文件的形式，必须按照规范的要求创建。相机提供的 GenApi 描述在收到后会受到检查。如果包含不符合规定的数字，则会报告警告或错误。

- 例如，如果未定义的寄存器位于保留的引导区中，就会发出警告。这些寄存器随后被忽略，但这通常不会导致操作上的限制。在个别情况下，必须检查是否需要所报告的寄存器。
- 例如，如果描述的访问权限与相机的行为不匹配，就会报告错误。也就是说，相机在读取标记为可读的寄存器或写入标记为可写的寄存器时返回一个错误。
- 如果 GenApi 描述包含完全不符合要求的段落，在极少数情况下，相机可能根本无法运行。

相机行为

如前所述，如果相机行为与 GenApi 的描述不同以及如果寄存器包含相互依赖关系或相互影响，相机的行为会产生影响。

- 例如，寄存器可能被其他寄存器锁定，因此这些寄存器必须首先被激活或停用。
- 寄存器的值可能会受到其他寄存器或公式的影响，这会造成限制最大值等情况。
- 当其他寄存器被写入时，相机可以改变寄存器的值。例如，相机可能在写入像素格式时自动重置 ROI。

参考制造商的文件

如果相机行为存在问题，请与制造商的文件进行精确比较。例如，个别读写寄存器的写入可能被相机错误拒绝。

GevImageAcquisition 对象的行为

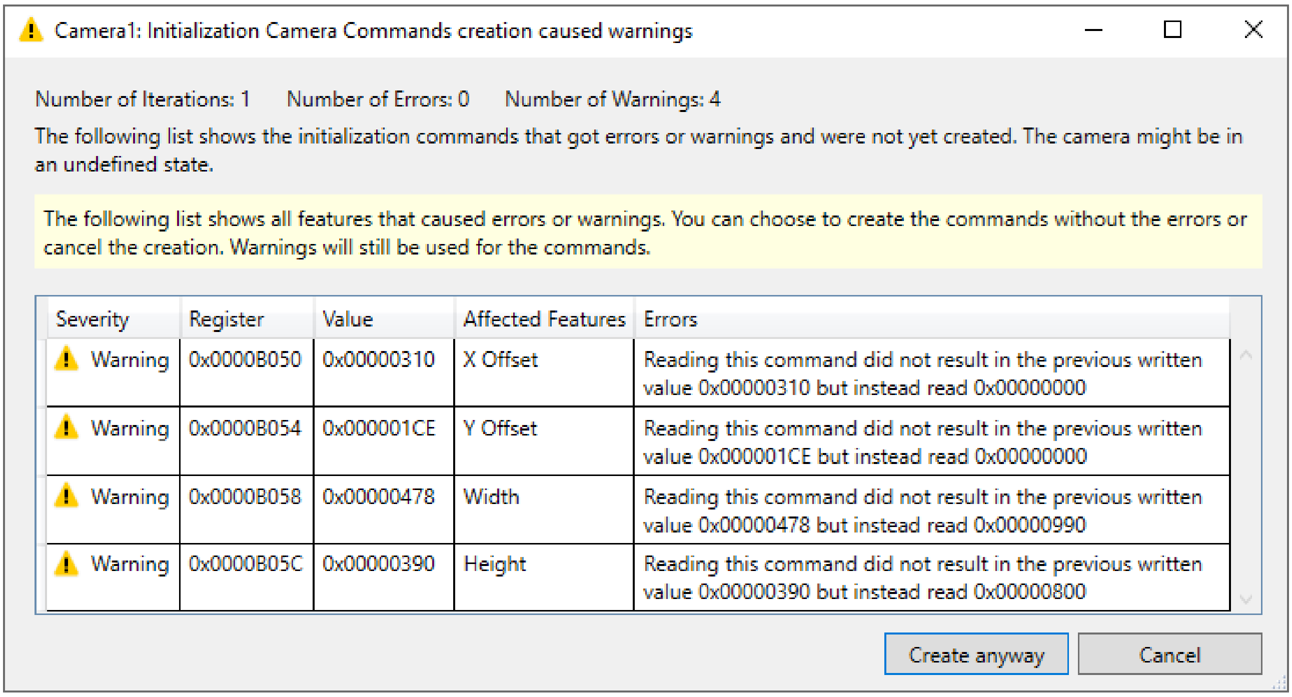
GevImageAcquisition 对象作为 TC3 GigE Vision 连接器产品的一部分通过了 GigE Vision 认证，因此其行为符合标准。串联命令（如果相机支持）和迭代选择器在设置中默认为激活。这意味着，在与相机通信时，所有可能的寄存器组合都被读取或写入。这偶尔会因为命令的数量或时间而导致问题。为了补救这个问题，你可以在图像采集模块中禁用**启用串联命令**或者在设置中禁用**自动迭代选择器**。

消息

上面描述的组件及其行为对初始化命令的创建或可能的消息有不同的影响。

警告

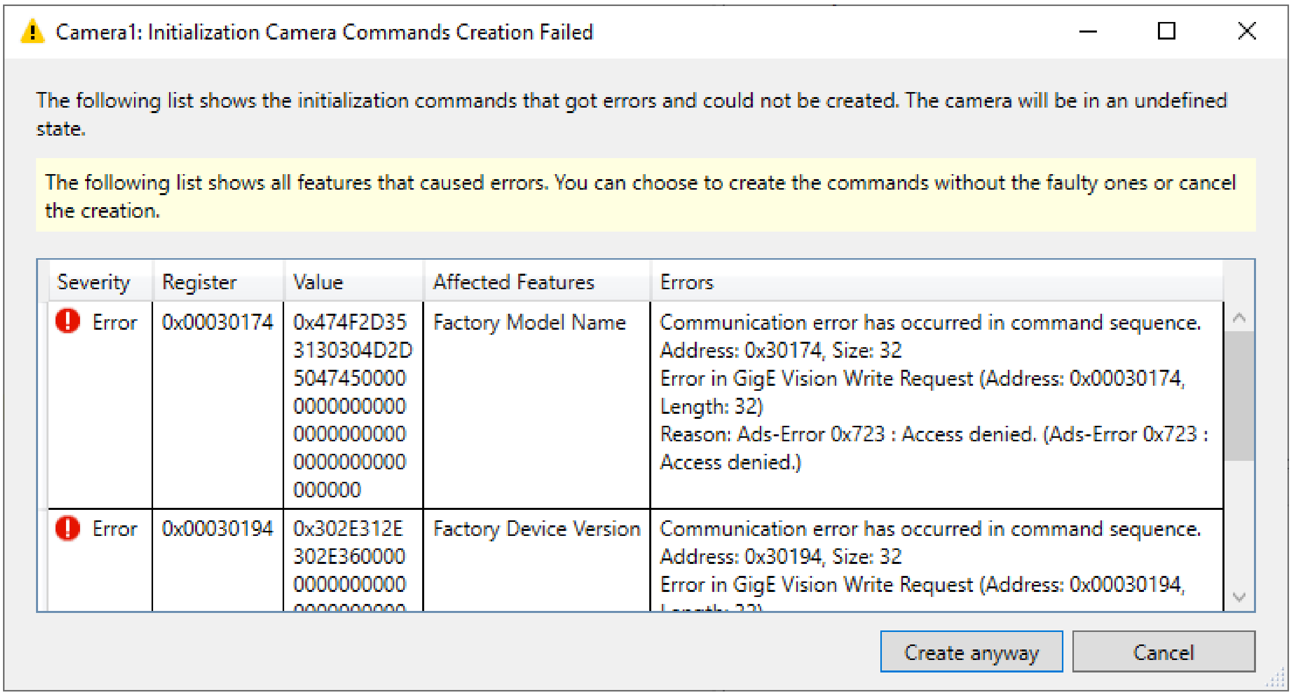
如果读回的数值与之前写入的数值不同，则显示警告。这些差异可能由不同的相机行为或寄存器之间的依赖关系造成。这并不意味着相机不能用初始化命令工作。因此，警告对通过**仍然创建**进行创建没有影响。之后，你应该测试报告的寄存器是否有正确的值以及相机的行为是否符合要求。



一个可能出现警告信息的例子是当设置了一个 ROI 时。根据相机中的当前值，警告会发生，因为根据 GenApi 的顺序，偏移量首先写入。如果需要写入大于 0 的偏移量，宽度或高度必须相应地小于最大值。在这种情况下，相机就会重新设置所有的 ROI 值，造成差异。在这种情况下，可以改变配置列表中 ROI 寄存器的顺序，使宽度和高度在两个偏移量之前写入。在特殊情况下，如果需要，也可以复制一个寄存器，移动顺序，从而多次写入。

错误

如果个别寄存器数值无法写入，则报告错误。如上所述，这可能有各种原因，如极限值、地址、写入权限或写入时公式计算不正确或更改。在通过**仍然创建**进行创建时，受影响的寄存器被省略，并在配置列表中经过核对。随后，应测试被省略的寄存器是否对相机功能有影响。



例如，从截图中可以看出，如果只是型号名称更改，或者寄存器没有更改且设有合适的默认值，那么尽管存在这个错误信息，相机还是可以不受限制地使用。

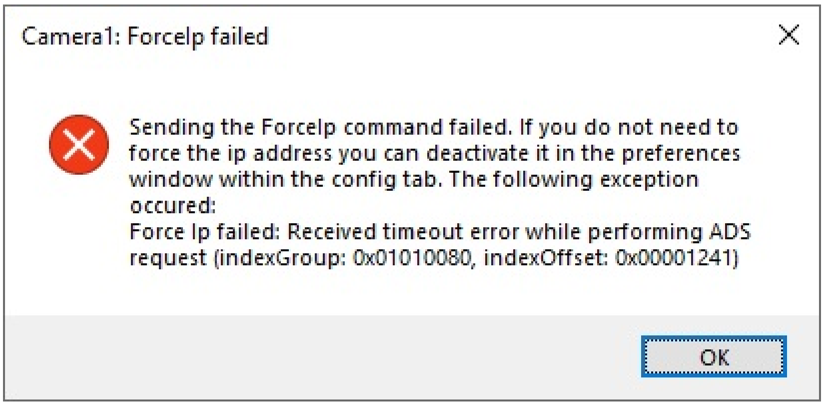
重大错误

只有在相机在一定时间内没有反应时才会发生重大错误。然后该过程被中止，并出现以下信息之一。在这些情况下，必须调整相应的超时。

超时

强制 IP 或 GVCP 超时

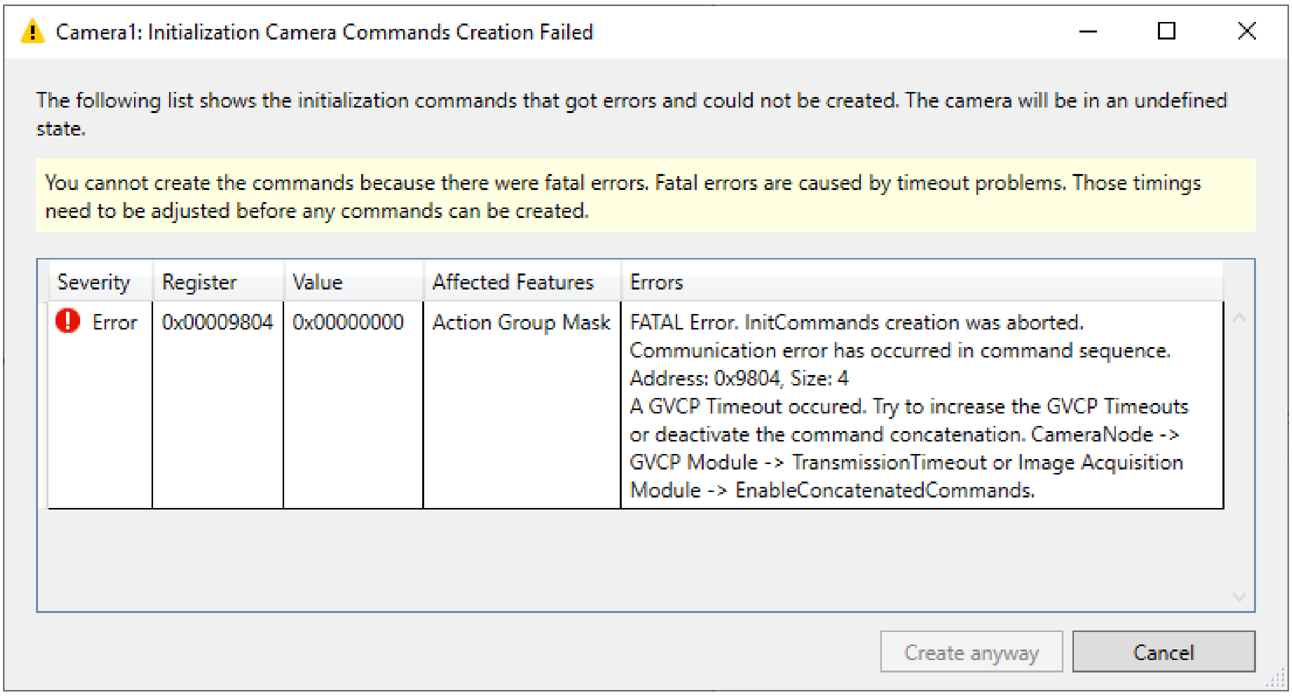
在写入初始化命令的过程中，如果配置了强制 IP 命令，也会发送该命令。如果各相机用此命令设置 IP 的时间很长，可能会发生以下超时错误：



附图 2: 强制 IP 过程中的错误

> **永久 IP 地址**
>
> 如果不使用强制 IP，建议设置一个永久 IP 地址。

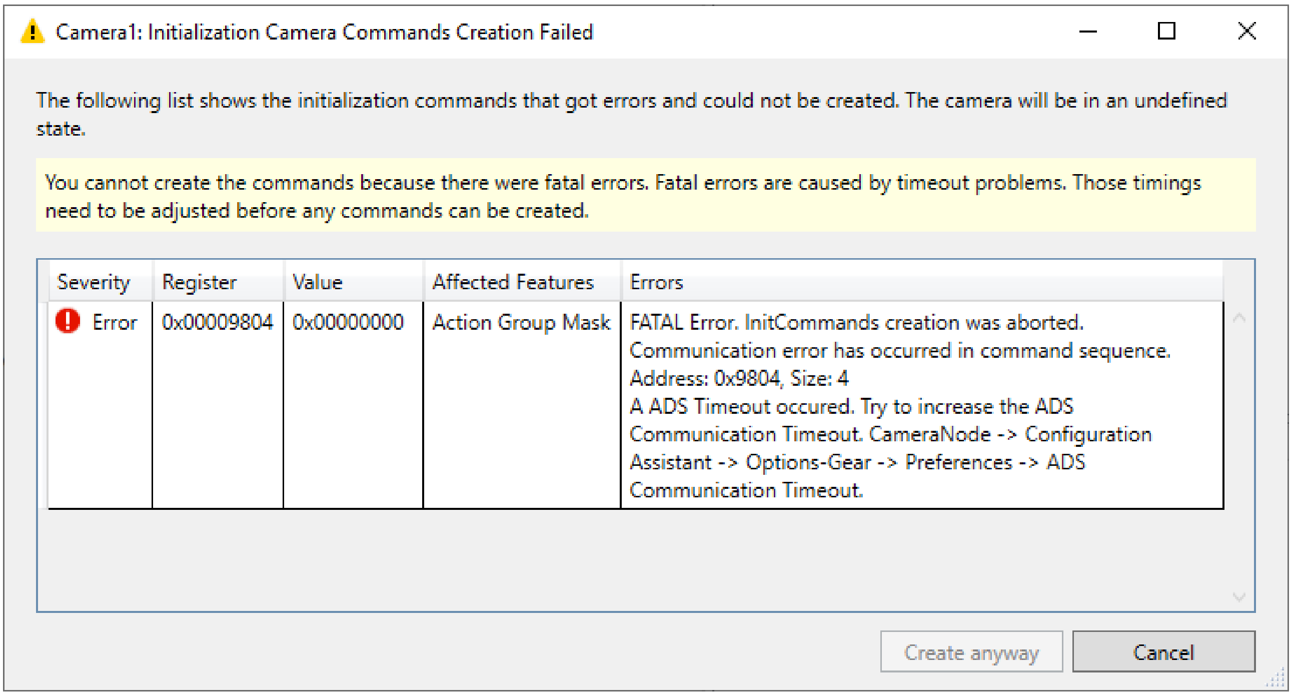
在这种情况下，**强制** IP可以在配置列表 [▶ 78]中停用。另外，在 GVCP 超时的情况下，TransmissionTimeout和/或MaxTimeouts可以从GVCP 模块 [▶ 88]在 TcCOM 参数中增加。请注意，设置窗口 [▶ 74]中的ADS Communication Timeout应始终至少等于前面两个参数的乘积。



附图 3: 创建初始化命令时出现严重的 GVCP 超时错误

ADS 超时

在 ADS 超时的情况下，在设置窗口 [▶ 74]增加参数ADS Communication Timeout。



附图 4: 创建初始化命令时出现严重 ADS 超时错误

5.4.2.5 TcCOM 参数

三个 TcCOM 模块图像采集、GVCP 和 GVSP 的参数描述如下。调整这些参数可以帮助某些相机建立一个稳定的连接。应使用配置树 [▶ 70]以改变参数。

最初，此处仅涉及参数InitializationAutoMode。然而，在出现连接或初始化问题时，可能需要设置其余参数。

在参数更改的情况下重新加载

为了使参数更改生效，必须重新加载相应的对象。

图像采集模块

特征	描述
相机 IP 地址	相机的 IP 地址。这个 IP 地址与强制 IP 命令一起使用。如果这个 IP 地址改变，初始化命令 [▶ 77]必须重新创建，以使更改生效。此外，配置 [▶ 78]中的相应选项必须为强制 IP 命令激活。
初始化超时	向相机传输初始化参数的最长时间。根据所使用的相机，传输时间可能更长，在这种情况下，必须对参数进行相应调整。 单位：毫秒（ms） 默认值：2,000
初始化自动模式	选择图像采集 TcCOM 对象自动执行相机初始化的方式。如果相机没有被初始化，就无法与之通信。 AUTOINIT_SO：相机在 TcCOM 对象的启动阶段被初始化。在这个阶段，仅有限时间可用。如果使用了初始化参数，由于潜在的超时，初始化通常不能在这个启动阶段进行。在这种情况下，即使选择了 AUTOINIT_SO，也会应用 AUTOINIT_AFTER_SO 的行为。 AUTOINIT_AFTER_SO：TcCOM 对象在没有初始化相机的情况下进入 OP 状态。只有当从 PLC 打开与相机的连接时（例如用 FB_VN_SimpleCameraControl.StartAcquisition()或用 FB_VN_GevCameraControl.OpenCamera()），相机才会自动初始化。 NO_AUTOINIT：相机只有在调用 GevCameraController.InitializeCamera()时才会被初始化。该选项旨在针对需要完全控制初始化以便在某个时间执行的情况。 标准：AUTOINIT_AFTER_SO.
启用串联命令	选择在向相机写入多个命令（例如初始化参数）时，是将相关命令链接起来还是单独发送。 命令的链接使传输更快，但可能会导致某些相机出现问题。 默认：激活。

GVCP 模块

特征	描述
本地端口	本地 UDP 端口，GVCP 模块通过该端口进行通信。如果该值为零，则使用自动生成的端口。 默认值：0
最大超时量	在 GVCP 数据包被丢弃之前可能发生的最大传输超时数。每次超时后，GVCP 数据包会被重新发送。如果达到最大的超时数，就会输出一个错误。 默认值：3（适应于相机）
传输超时	在发送一个 GVCP 数据包后，等待响应。如果响应时间长于传输超时时间，则认为数据包没有到达，在这种情况下，数据包会被重新传输。 单位：毫秒（ms） 默认值：600（适应于相机） **传输超时应小于心跳速率。**
心跳频率	应用程序的控制通道向相机发送一个心跳信号，以检查连接是否仍然有效。心跳频率描述了距离下一个信号发送的时间。 单位：毫秒（ms） 默认值：1,000 **心跳速率不应被任意设置得过高，因为许多相机在几秒钟没有心跳后就会触发超时。**
发现超时	发现超时描述了等待发现命令响应所需的时间。在网络中搜索相机 [▶ 63]，就会发送发现命令。 单位：毫秒（ms） 默认值：1,000

GVSP 模块

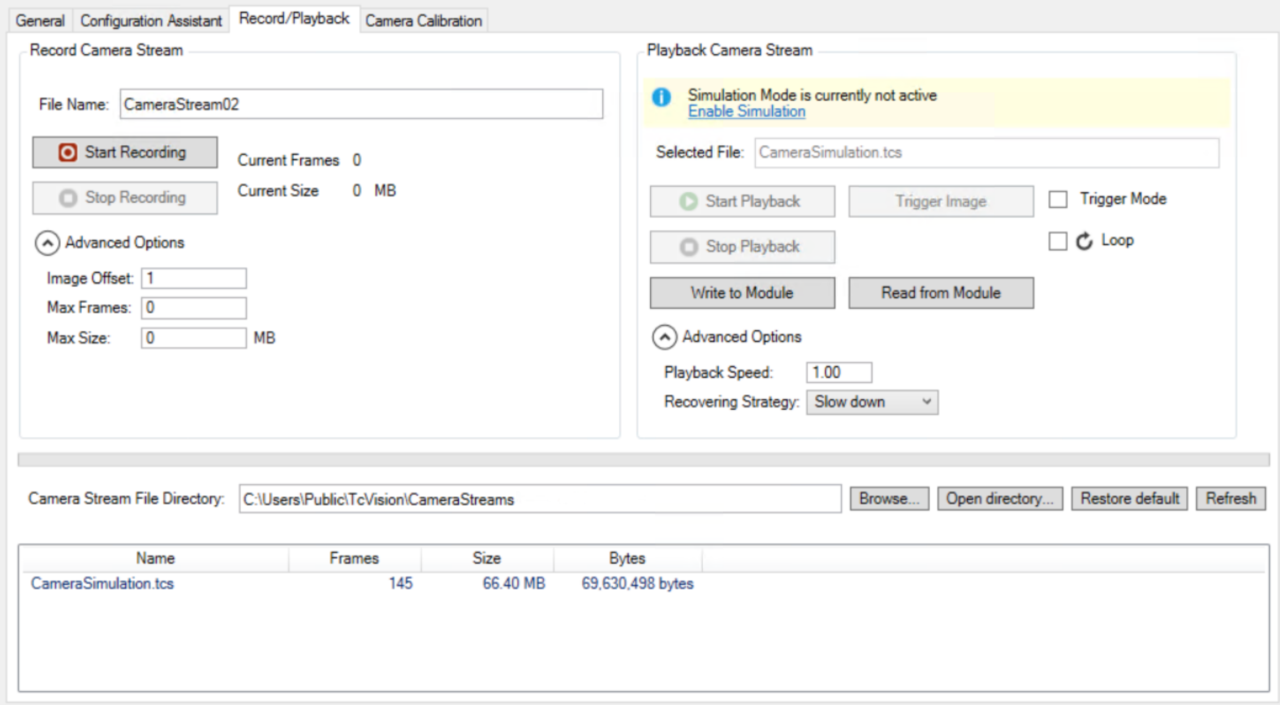
特征	描述
本地端口	本地 UDP 端口，GVCP 模块通过该端口进行通信。如果该值为零，则使用自动生成的端口。 默认值：0
最大数据包尺寸	**最大数据包尺寸**描述了一个 GVSP 数据包的最大数据包大小。以太网帧的标题不包括在这个尺寸中。 单位：字节 默认值：1,500 **目前，1,500 是这个参数的极限。不可能有巨量帧。**
最小数据包延迟	如果**最小数据包延迟**增加，GVSP 数据包会被人为地延迟。这使得带宽可以减少。 单位：时间戳计数单位 默认值：0
最大重发请求数	单个 GVSP 数据包的最大重发请求数，该数据包有问题或耗时过长。一幅图像在一个由几个数据包组成的 GVSP 块中传输。如果一个区块的数据包无法成功传输，则该图像无效。 默认值：1
最大区块数	可以同时处理的 GVSP 块的最大数量。如果该值为零，则该数字无限制。 默认值：5
最大包数	一个区块中 GVSP 数据包的最大数量。包含更多数据包的区块不能解释为有效图像。 默认值：25,000 **如果偏好设置中的“自动计算最大包数”选项处于启用状态，该数字将自动计算。**
区块超时	**区块超时**描述了一个块的单个数据包之间可能经过的时间。 单位：微秒（µs） 默认值：5,000

5.4.3 记录/回放

在**记录/回放**选项卡上可以记录相机流，并输入到 PLC。这对开发很有帮助，以便在桌面上记录相机流，并在以后的时间里反复进行离线回放。因此，相机硬件的真实行为通过原始图像进行模拟。

要么激活模拟模式，在这种情况下可以回放相机流，要么停用模拟模式，在这种情况下可以记录相机流。不可能同时进行这两项工作。

相机流由单个图像组成，这些图像与相应的时间标记一起以自己的格式（.tcs）保存。通过额外保存的时间标记，机器上的相机的时间行为可以模拟出应用的真实性。图像数据正是以相机发送的格式保存的，不需要进一步转换。



相机要求

相机必须支持时间戳并在图像标题中输入有效值。每个图像都需要时间戳，以便能够模拟时间行为。

系统要求

使用快速的固态硬盘和至少 8GB 的内存来记录和回放相机流。如果使用较慢的存储介质，必要时可通过参数**图像偏移**降低图像速率。

禁用/激活 TcCOM 对象

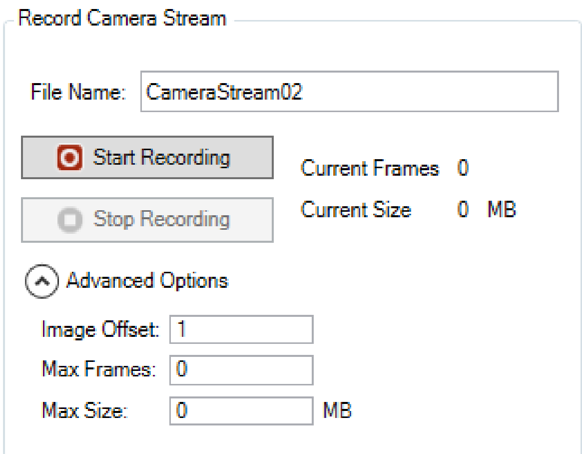
相机 TcCOM 对象的激活/停用不应手动进行，因为在后台从模拟模式和 Ads 通信更改时，必须调整更多链接。

记录/回放助手的三个区域（记录、回放、流文件）解释如下。

记录相机流：

相机流的记录只有在相机连接激活的情况下才能进行。因此，相机实例 [▶ 63]的模拟模式必须禁用，并选择标准采集。另外，还需要 Ads 通信器 TcCOM 模块。

记录相机流的设置位于记录相机流区域：



文件名称	相机流文件名的文本框。用于保存文件的路径是在选择文件路径和流 [▶ 93]中定义。
	用于开始记录的按钮。只有在可以记录的情况下才激活。
	用于停止记录的按钮。仅在记录期间激活。
当前帧数	当前记录的相机流中的图像数量。
当前尺寸	当前记录的相机流的文件大小。
高级选项	显示和隐藏高级选项的选项。
图像偏移	指定图像记录顺序。根据存储介质的写入速度、图像大小和每秒帧数的不同，可能需要跳过图像。 默认值：1（即尝试记录每张图像）。
最大帧数	待记录相机流的最大允许图像数。一旦达到这个数量的图像，记录就会自动结束。 默认：0（即没有限制）。
最大尺寸	待记录相机流的最大允许文件大小，单位：MB。一旦达到这个文件大小，就会自动停止记录。 默认：0（即没有限制）。

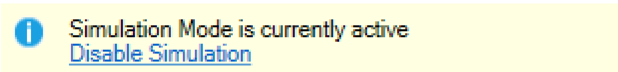
是否存在足够的自由存储空间？

在记录数据流之前，确保你的目标系统有足够的存储空间，或者必要时通过**最大帧数**和**最大尺寸**限制数据流大小。

必须满足以下条件，才能记录数据流：

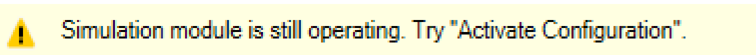
- 模拟模式已禁用。
- Ads 通信对象存在并已激活。
- TwinCAT 处于运行模式。
- 相机必须处于ACQUIRING状态。

如果不符合这些条件，可能会出现以下说明和建议解决方案：

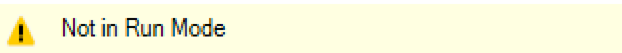


- 模拟模式是指可以回放数据流但不记录的一种状态。该状态可以通过注释中的**禁用模拟**链接进行更改。

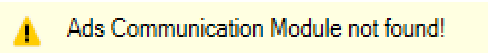
  这也需要重新激活配置  。

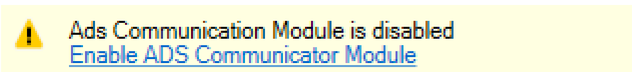


- 模拟模式已被关闭，但该更改只有在配置被激活后才会应用。

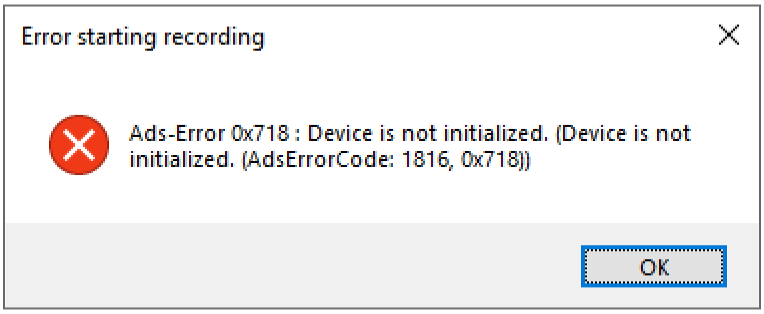


- 只有在 TwinCAT 运行模式下，才能对相机流进行记录。在所有其他 TwinCAT 状态下都会显示该警告。





- 只有在与相机有开放连接的情况下，才有可能记录相机流，因为在记录之前会从相机中读取数据。因此，相机此时应该已经处于ACQUIRING状态。



- Ads 通信模块是记录相机流所必需的。它可以通过说明中的链接激活。如果尚不存在 ADS 通信器对象，可以通过相机对象上的**添加新项目...**和常规选项卡 [▶ 64]中的**尝试修复**进行添加。激活配置后，该更改被接受 。

提前激活 ADS 通信器模块

如果计划以后在不首先停止 TwinCAT 的情况下记录相机流（例如在机器运行期间），必须提前激活 ADS 通信器模块。

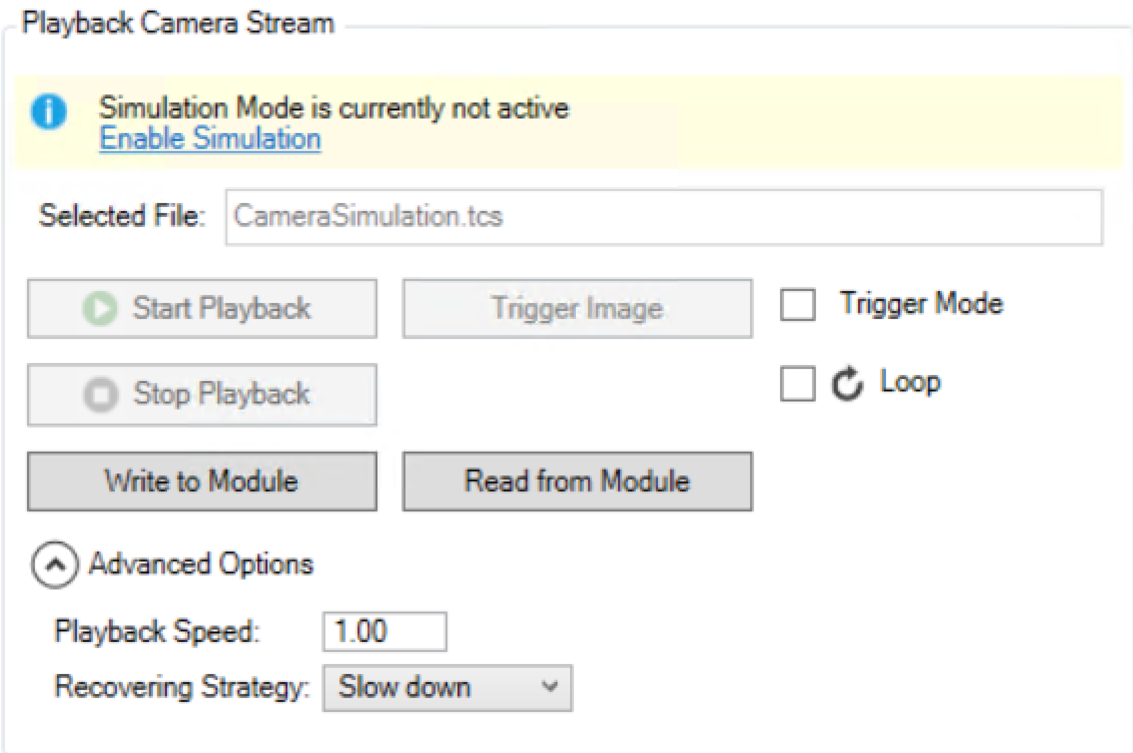
如果 Ads 通信器对象已激活，可能来自相机的图像在短时间内仍被对象内部引用，且因此不能直接转化为可显示的图像。关于更多详细信息，请参见可显示的图像 [▶ 131]。

回放相机流：

为了回放相机流，必须激活相机的模拟模式；这可以通过相机对象的常规 [▶ 64]选项卡或下面提到的信息来实现。

如果通过写入模块将配置写入 TcCom 对象并保存在项目中，那么这些数值在项目激活后就已经可用，且模拟会随着 PLC 的开始采集命令直接开始。这样做的好处是，不必每次都手动重置设置和启动数据流。

回放相机流的设置位于**回放相机流**区域：

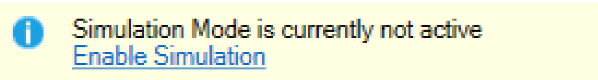


选定的文件	用于显示所选数据流文件的文本框。从可用数据流的列表中进行选择。
开始回放	用于开始回放的按钮。只有在可以回放的情况下才启用。回放总是从图像 1 开始，无论数据流之前在哪里停止。
停止回放	用于停止回放的按钮。只在回放数据流时启用。
触发模式	选择触发模式，在这种模式下，数据流中的帧可以被单独触发。
触发图像	用于触发图像的按钮。仅在选择了触发模式且回放数据流时有效。
循环	选择数据流回放一次或无限循环的选项。
写入模块	将当前设置写入 TcCom 模块。存储内容有回放模式、回放速度、回放策略、循环播放和所选数据流文件的路径。
从模块中读取	从模块中读取存储的值并设置这些设置。
高级选项	显示/隐藏高级设置。
回放速度	回放速度的加速/减速系数。（>1：加速；<1：减速） 默认值：1.00 = 速度与记录相符
恢复策略	在无法以指定速度回放的情况下的策略： • 放慢速度：回放速度延迟（默认设置） • 跳过图像：跳过图像 • 滞后时中止：终止回放（不建议）

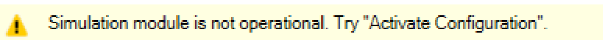
必须满足以下条件，才能回放数据流：

- 模拟模式启用。
- TwinCAT 处于运行模式。

以下说明和警告可能会出现在选定文件名的文本字段上方：



- 模拟模式是数据流可以回放的状态。这个状态可以通过**启用模拟**链接和随后激活配置  达到。

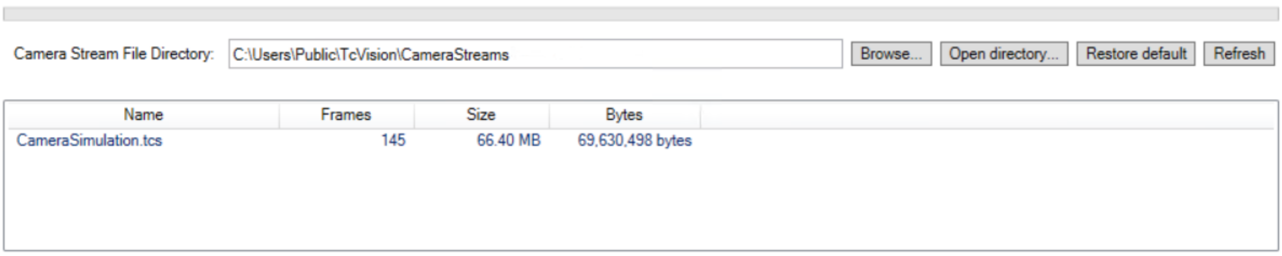


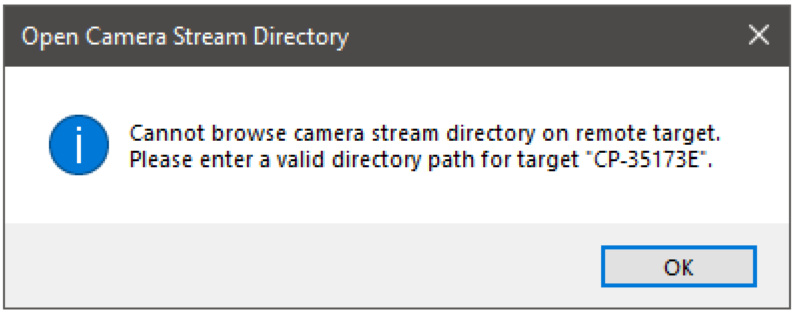
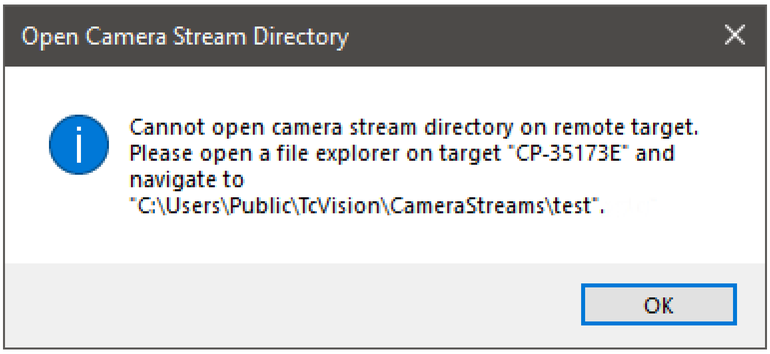
- 模拟模式已启用，但更改仅在配置启用后生效。

- 仅在 TwinCAT 运行模式下才可以回放相机流。在所有其他 TwinCAT 状态下都会显示该警告。

选择文件路径和数据流

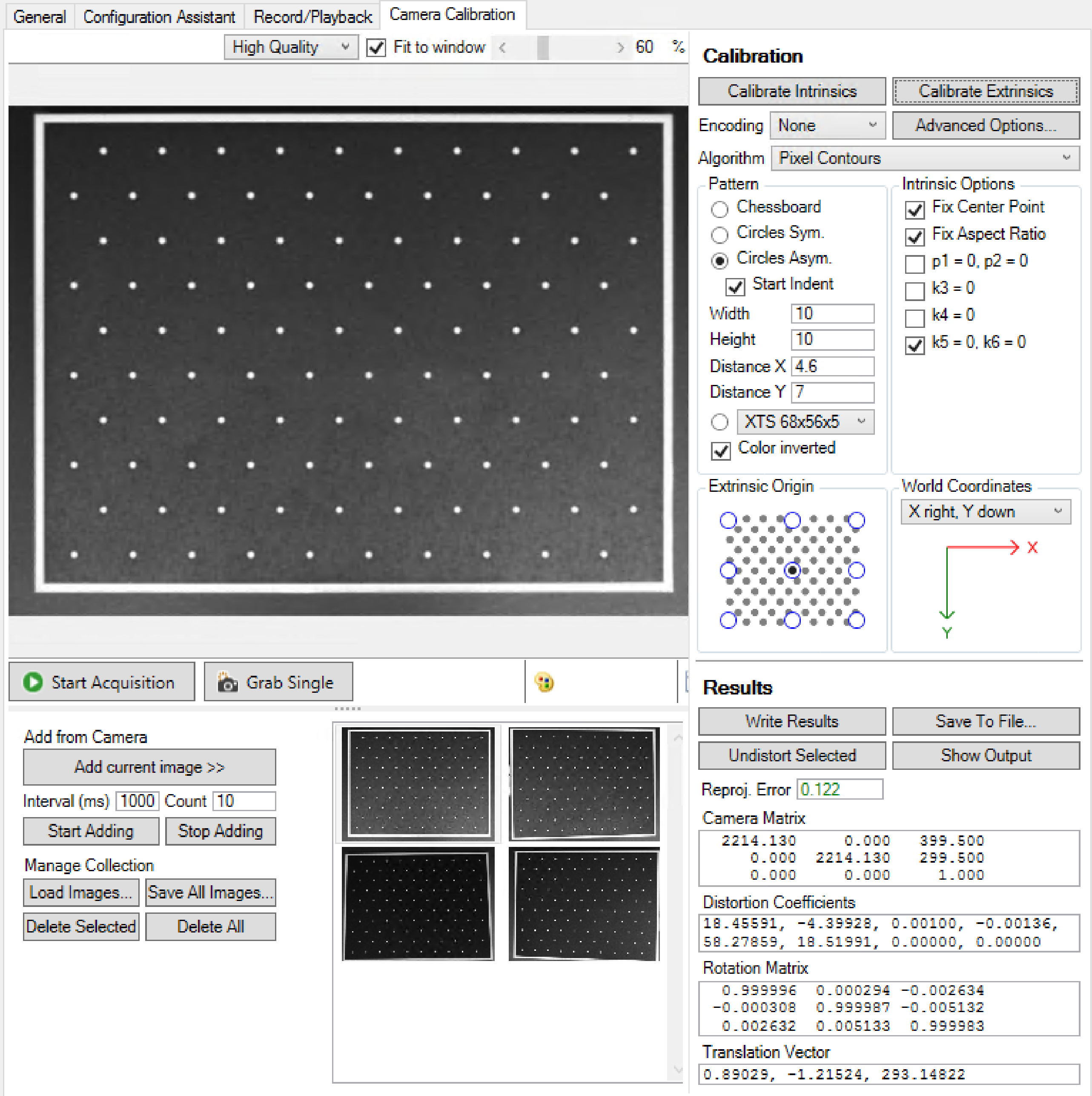
文件路径和数据流的选择如下所示：



	录制或回放的状态栏
相机流文件目录	目前设置的文件路径。这个路径既用于读取已经记录的数据流，也用于写入新的数据流。子文件夹也会被搜索数据流文件。 **文件路径始终指的是目标系统，而不是开发系统。** **确保这个文件路径不受写入过滤器或类似元素的影响。** 见重要路径 [▶ 2736]。
浏览...	该按钮可打开选择窗口，用于浏览目标文件夹。 **仅在开发系统也是目标系统时，才有可能。** 否则会出现以下信息： 
打开目录...	用于在本地文件资源管理器中打开指定文件路径的按钮。这也可以通过在文件列表中的自由区域内双击来完成。 **仅在开发系统也是目标系统时，才有可能。** 否则会出现以下信息： 
恢复默认	用于设置默认文件路径的按钮。
刷新	用于刷新数据流列表的按钮。
名称	所选文件路径中所有相机流的列表，也用于回放时的数据流选择。
帧	各个相机流所包含的图像数量。
大小	各个相机流的文件大小，单位：MB。
字节	大小规格（单位：字节）。

5.4.4 相机校准

区域扫描相机可以在相机校准选项卡上校准模式的帮助下进行几何校准。



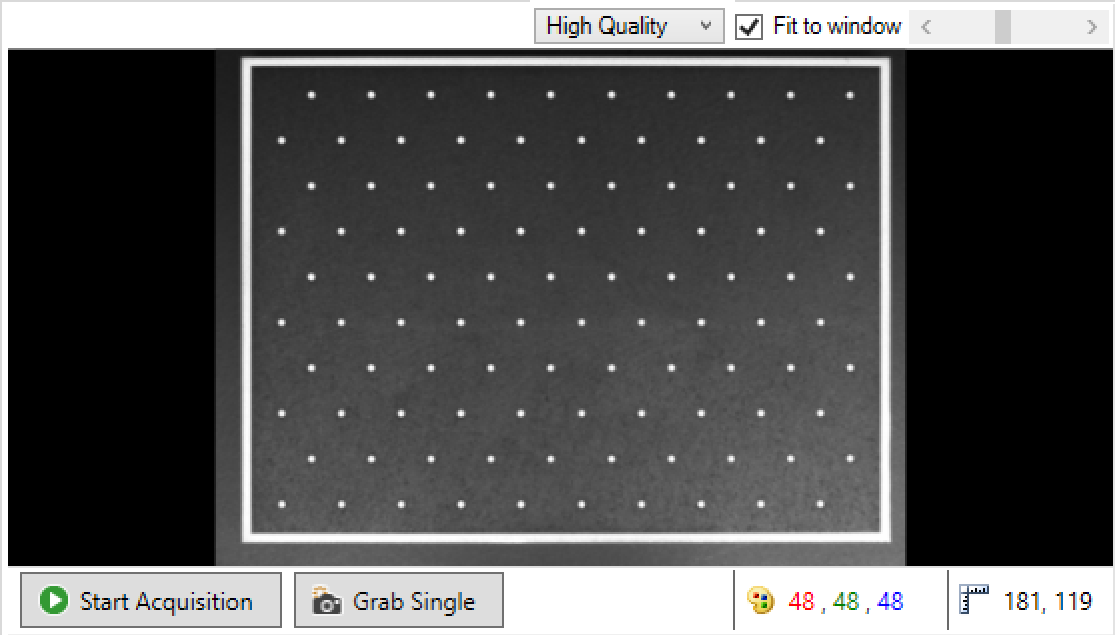
在区域扫描相机的几何相机校准过程中，成像过程由图像重建。为此，图像中的点需要在现实世界中的位置是已知的。具有定义点的校准模式可用，可从中提取图像。

支持的校准模式：

- 棋盘图案 [▶ 102]
- 对称圆形图案 [▶ 103]
- 不对称的圆圈图案 [▶ 104]
- 单个圆圈图案 [▶ 105]

图像采集/选择

如同在配置助手 [▶ 67]中一样，配有一个图像采集的预览图像，上面设有显示选项，下面设有图像采集的控制按钮：



显示预览图像

图像总是按照从相机接收的方式显示。因此，Bayer 格式的图像将被显示为未编码的灰度图像。在这一点上，设置编码还没有应用到图像上。编码只与校准过程有关，但与预览图像无关。

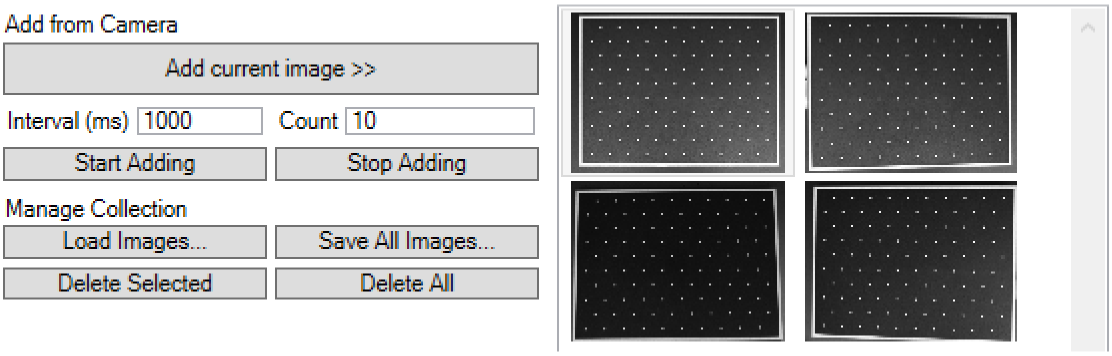
在预览窗口上方：

- **预览图像的插值：**
  组合框，用于选择是否要对预览图像进行高质量或最近邻的插值。该选择仅指预览。校准基于原始图像数据。
- **预览图像的大小**
  如果选择了**适应窗口**，图像会被缩放到预览窗口的大小。另外，也可以通过滑块指定显示尺寸。

在预览窗口下面：

- **开始采集/停止采集**
  按钮用于开始或停止图像采集。相机的数据流模式必须处于激活状态。
- **抓取单图**
  按钮，用于触发单个图像。
- **像素值**
  显示图像上鼠标光标位置的像素的颜色值。
- **光标位置**
  显示鼠标光标在图像上的位置。请注意，图像的零点总是在左上方。

如果校准模式在相机的视场内，可以选择相应的图像进行校准。控制按钮位于左边的预览图像下方，而校准图像集合位于右边。

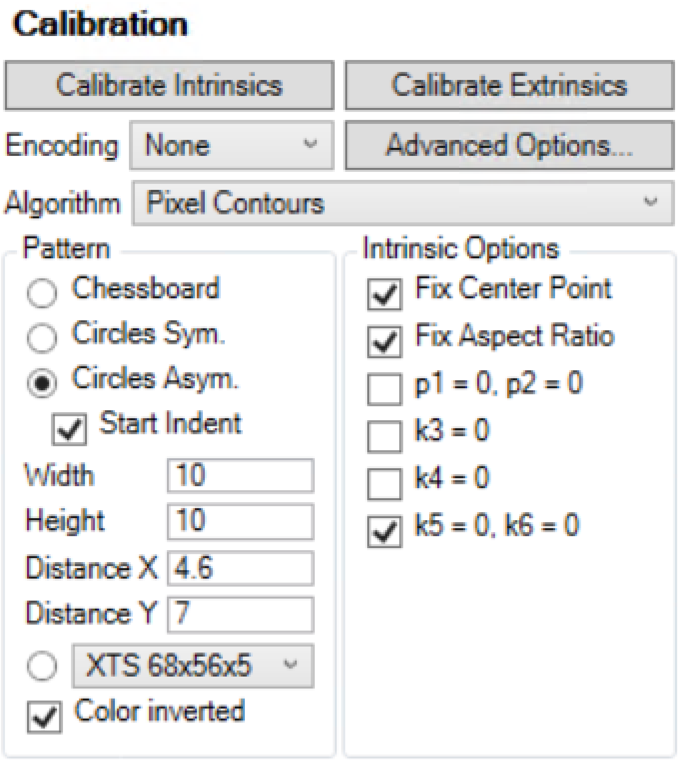


- **添加当前图像 >>**
  添加当前图像将当前显示的相机图像添加到集合中进行校准。

- 开始/停止添加
  一系列图像可以被添加到集合，而不是单个图像。图像系列通过**开始添加**启动。随后，按照**间隔**设定的时间间隔添加图像，直到达到**计数**输入的数字或通过**停止添加**终止图像的添加。
- 加载图像...
  作为图像采集的替代方法，可以通过**加载图像**加载保存的图像。请注意，这些都是通过相同的相机系统、相同的相机设置、相同的镜头设置和相同的校准模式记录。
- 保存所有图片...
  保存所有图片可以用来保存整个图片集。
- 删除所选
  删除所选将所选的图像从集合中删除。
- 删除所有
  删除所有将所有图像从集合中删除。

> **图片格式**
> 目前，仅可使用 Mono8、RGB24 和 Bayer8 图像。

设置和计算



编码

对于测量任务，建议使用单色相机。然而，如果需要通过 Bayer 模式校准彩色相机，必须通过**编码**指定相应的 Bayer 模式。这在默认情况下自动选择。

算法

对于圆形图案检测，可以选择不同的边缘定位功能。根据图像质量，可以使用子像素功能来改善结果。然而，这也意味着整个过程需要更多时间。关于子像素功能的更多信息，请参见测量 [▶ 1428]一章。

- 像素轮廓
  基于轮廓
- 子像素插值
  用子像素进行插值
- 子像素误差近似
  误差近似后的子像素

标定模板的定义

在**模板**下选择和定义使用的标定模版：

- **棋盘**
  使用棋盘模板
- **对称圆圈**
  使用对称圆圈模板
- **不对称圆圈**
  使用不对称圆圈模板
  - **开始缩进**
    用于指定圆圈模板首行缩进的复选框。在没有复选标记的标准系统中，预计不缩进行。
- **宽度**
  水平方向的特征点数量
- **高度**
  垂直方向的特征点数量
- **距离 X**
  水平方向特征点在世界坐标系中的距离
- **距离 Y**
  垂直方向特征点在世界坐标系中的距离
- **单个圆圈模板的下拉菜单**
  选择单个圆圈模板（只有在指定文件夹中找到相应文件时才会显示条目）
- **颜色反转**
  用于选择图像颜色反转的复选框。默认情况下（无复选标记），白色背景上显示黑色对象。

对称模板

对于完全对称的模板，找到点的分配可能与定义点有偏差，或者坐标原点可能位于不同的象限。如果所选图像的模板定向与定义不同，也会出现这种情况。

因此，在完成标定程序后，应检查坐标系是否与您的规格相对应。

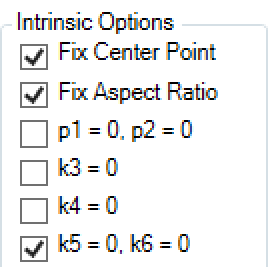
检查填充圆圈

如果是圆圈模板，还要检查圆圈是否填充了相应的颜色。如果在这一点上出现偏差，相应点就不会被识别为模板的一部分。

有关模板和参数设置的信息，请参见支持的标定模板类型：

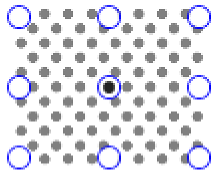
- 棋盘图案 [▶ 102]
- 对称圆形图案 [▶ 103]
- 不对称的圆圈图案 [▶ 104]
- 单个圆圈图案 [▶ 105]

内在选项



- **固定中心点**：主点对应于图像的中心。
- **固定长宽比**：假设相机传感器有方形像素，因此像素的长宽比在 x 和 y 方向上相同。
- p1=0，p2=0：假定不存在切向畸变，相应的系数设置为 0。
- k3=0：径向畸变系数 k_3假定为 0
- k4=0：径向畸变系数 k_4假定为 0
- k5=0，k6=0：径向畸变系数 k_5和 k_6 假定为 0

外在原点

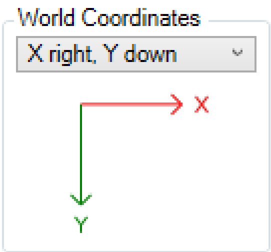


外在原点指定校准模式中全局坐标系的原点。

根据所选择的校准模式，会显示相应的背景图像。

> **原点**
>
> 选定的原点是指校准模式的对应点，而不是指整个图像。

全局坐标



全局坐标设置全局坐标系的方向。这可以用来设置一个与图像坐标系相比不同的方向。在所示的默认设置下，两个坐标系相匹配。

操作

校准内在

校准内在计算相机矩阵和畸变系数。所有选择的图像都用于此目的。这些都必须包含所定义的校准模式的全部内容。如果不是这种情况，就会出现负的**再投影。显示错误**。

校准外在

校准外在计算旋转矩阵和平移矢量。这方面的一个先决条件是内在校准（相机矩阵和畸变系数）和选择图像，其中二维校准图案的点位于随后的测量平面内。

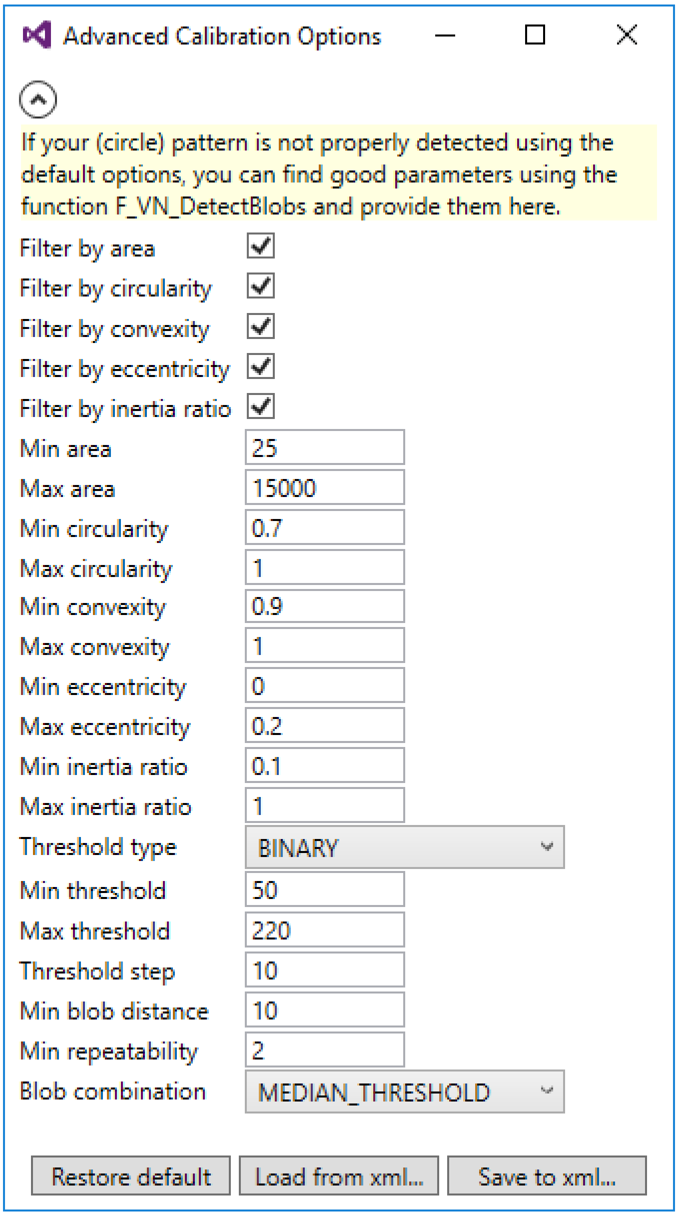
> **校准的持续时间**
>
> 校准的时间随着校准模式中参考点的数量和校准图像的数量而增加。因此，在选择校准模式时，确保持续时间保持在可接受的范围内。

> **结果的偏差**
>
> 由于所使用的算法，校准的结果在不同的调用中在小数位范围内可能彼此略有不同。这涉及到校准助手中的校准结果和 PLC 中的校准结果，以及两个校准变体之间的校准结果。

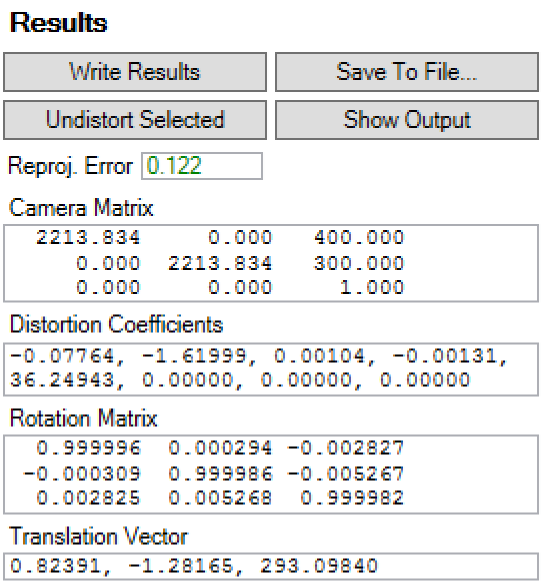
> **将设置转移到 TwinCAT 项目中**
>
> 仅在执行**校准内在**或**校准外在**功能后，设置的值和选项才会应用于 TwinCAT 项目。如果事后保存项目，这些值将在再次打开项目时被重新加载。否则，将始终使用默认设置。

高级选项

圆形图案识别可以通过**高级校准选项**重新进行参数化。仅在默认设置存在困难的情况下才可以这样做。为了更好地理解设置参数，请参见函数的参数化F_VN_DetectBlobs [▸ 1139]。


Advanced Calibration Options
If your (circle) pattern is not properly detected using the default options, you can find good parameters using the function F_VN_DetectBlobs and provide them here.
Filter by area
Filter by circularity
Filter by convexity
Filter by eccentricity
Filter by inertia ratio
Min area 25
Max area 15000
Min circularity 0.7
Max circularity 1
Min convexity 0.9
Max convexity 1
Min eccentricity 0
Max eccentricity 0.2
Min inertia ratio 0.1
Max inertia ratio 1
Threshold type BINARY
Min threshold 50
Max threshold 220
Threshold step 10
Min blob distance 10
Min repeatability 2
Blob combination MEDIAN_THRESHOLD
Restore default
Load from xml...
Save to xml...

结果


Results
Write Results
Save To File...
Undistort Selected
Show Output
Reproj. Error 0.122
Camera Matrix
2213.834 0.000 400.000
0.000 2213.834 300.000
0.000 0.000 1.000
Distortion Coefficients
-0.07764, -1.61999, 0.00104, -0.00131,
36.24943, 0.00000, 0.00000, 0.00000
Rotation Matrix
0.999996 0.000294 -0.002827
-0.000309 0.999986 -0.005267
0.002825 0.005268 0.999982
Translation Vector
0.82391, -1.28165, 293.09840

再投影 误差
再投影误差表示内部标定的结果：

- >0：有效结果
  - <1：良好结果
  - >2：不准确结果
- <0：无结果和具体错误编号：
  - -100：未找到图像
    预览窗口下方的图像集合用于内部标定。不考虑相机的实时图像。
  - -200：至少在一个图像中找不到模板。
    预览窗口下方集合中的所有图像必须显示完整模板，包括所有特征点。为了更好地进行分析，会在 *C:\ProgramData\Beckhoff\Vision_CalibrationAssistantOutput* 下创建结果图像，显示找到的特征点。
  - -300：现有图像中均不包含所选模板。
    请参见适用于 -200 的描述，此外，建议检查模板配置。
  - -400：单个标定模板的用户定义描述文件无效。
  - -500：发生了一个错误。

相机矩阵

相机矩阵描述从三维到二维的图像过程。

$$CameraMatrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

- 参数 f_x 和 f_y 表示 x 和 y 方向上的焦距。
- 参数 c_x 和 c_y 表示光轴与图像的交点，称为主点。理想情况下，其应位于图像中心。如果在 Intrinisic Options（**内部选项**）下选择了 Fix Center Point（**固定中心点**），参数 c_x 和 c_y 将自动设置为图像的中心。

畸变系数

光学系统的畸变会导致图像中发生几何畸变。这些问题可以在坐标转换过程中通过**几何畸变系数**得到部分补偿。

$$DistortionCoefficients = [k_1, k_2, p_1, p_2, k_3, k_4, k_5, k_6]$$

- 系数 k_1 至 k_6 描述图像中的径向几何畸变。
- 系数 p_1 和 p_2 描述图像中的切向几何畸变。

旋转矩阵

旋转矩阵描述相机坐标系相对于世界坐标系的定向，反之亦然。

$$RotationMatrix = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

平移向量

平移向量描述相机坐标系的坐标原点相对于世界坐标系的位移，反之亦然。

$$TranslationVector = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

写入结果

结果将写入相机的图像提供程序 TcCOM 对象。这意味着可以通过功能块 F_VN_GevCameraControl [▶ 1550] 从 PLC 调用它们。

即使尚未进行标定，也会写入设定标定模板的描述。这可通过 GetCalibPatternRef [▶ 1558] 方法在 PLC 中进行检索。另外，也可通过 F_VN_GenerateCalibrationPatternReferencePoints [▶ 1558] 直接在 PLC 中生成标定模板的描述。

保存到文件……

结果可以保存在 json 文件中。此外，还可通过 FB_VN_ReadCalibrationResult [▶ 1526] 将结果读入 PLC。

已选择去几何畸变

几何畸变系数应用于所选图像，结果显示在单独窗口中。

显示输出

打开目录 *C:\ProgramData\Beckhoff\Vision_CalibrationAssistantOutput*。创建结果图像是为了能够分析发现了哪些特征点以及如何分配这些特征点。

PLC 中的替代方案

如果必须在机器运行时对相机进行校准，在 PLC 中可使用函数F_VN_CalibrateCamera [▶ 1032]进行校准。

在 PLC 中进行校准

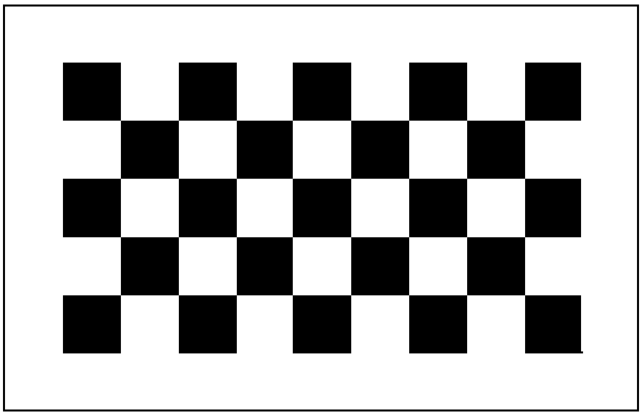
校准的持续时间可能比一个 PLC 周期长得多。因此，在 PLC 中进行校准时，确保在你的机器序列中不会出现问题。

校准助手样本

校准助手 [▶ 2720]

5.4.4.1 棋盘图案

棋盘图案由浅色和深色的方块交替组成，它们共同构成一个矩形图案。参见下面包含 4 x 8 交叉图案的棋盘示例：



对该图案的要求

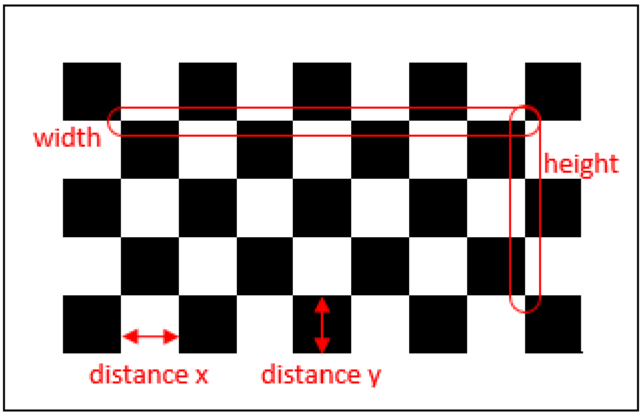
- 图案应用于尽可能平坦的表面（平面），例如玻璃板。
- 浅色和深色的方块必须
  - 以交替的方式排列，
  - 尺寸相同
  - 并且是方形。
- 一个图案必须包含至少 16 个完全可见的方块，且这些方块共同构成至少 3×3 的交叉图案。
- 图案周围的背景色必须形成一个‘安静区’。

对图像采集的要求

- 曝光
  - 图案照度均匀
  - 画面中的明暗方块对比良好
  - 在交叉口没有过度曝光导致的开口
- 数字
  - 一个方块必须至少由 15x15 像素表示。

相机校准助手 [▶ 94]**中的配置**

- **宽度**
  对应于水平方向的交叉点的数量。
- **高度**
  对应于垂直方向上的交点数量。
- **距离 x**
  对应于水平方向上两个交叉点之间的距离。
- **距离 y**
  对应于垂直方向上两个交叉点之间的距离。



5.4.4.2 对称圆形图案

对称圆形图案由均匀地排列在行和列中的圆形组成。

其他支持的圆形图案有：不对称圆形图案 [▶ 104]和独立圆形图案 [▶ 105]。

对该图案的要求

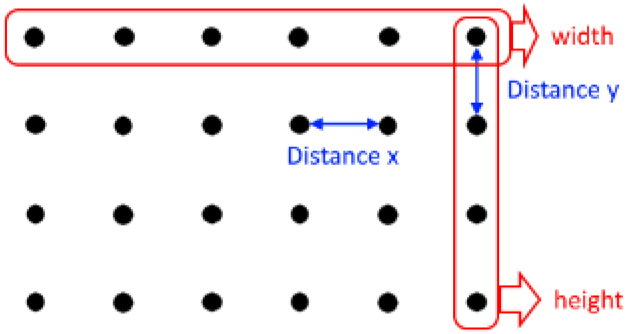
- 图案应用于尽可能平坦的表面（平面），例如玻璃板。
- 圆圈的大小应尽可能相等。
- 一个样本必须由至少 9 个点组成。
- 圆心必须在水平和垂直方向的相同距离上应用。这个距离必须足够大，以确保圆圈不会相互接触，并被识别为独立的对象。
- 图案周围的背景色必须形成一个‘安静区’。

对图像采集的要求

- 曝光
  - 图案照度均匀。
  - 圆圈和背景之间的对比良好。
- 数字
  - 透视畸变应保持在最低限度。
  - 必须遵守或适应**高级校准选项**的标准规格。默认情况下，一个圆必须由至少 25 个像素且最大 15000 像素的区域来表示。这个区域不能太小，否则就会接收到来自背景的干扰。

相机校准助手 [▶ 94]中的配置

- **宽度**
  对应于每行的圆圈数量。
- **高度**
  对应于每列的圆圈数量。
- **距离 x**
  对应于水平方向上两个圆心之间的距离。
- **距离 y**
  对应于垂直方向上两个圆心的距离。



- **颜色反转**
  默认情况下，假定深色圆圈搭配浅色背景。如果图案由浅色圆圈搭配深色背景组成，则可以通过**颜色反转**选项反转颜色。

5.4.4.3 不对称的圆圈图案

不对称圆圈图案类似于对称圆圈图案 [▶ 103]，只是每隔一行就偏移半列。这使得两行之间的距离更小，允许在同一区域设置更多的线条。

图案要求

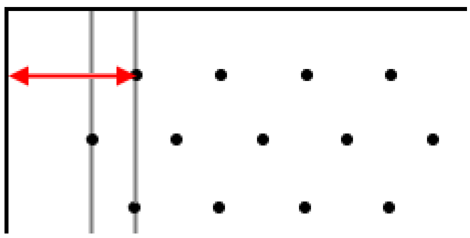
- 图案应用于尽可能平坦的表面（平面），例如玻璃板。
- 圆圈的大小应尽可能相等。
- 一个样本必须由至少 9 个点组成。
- 圆心必须在水平和垂直方向的相同距离上应用。这个距离必须足够大，以确保圆圈不会相互接触，并被识别为独立的对象。
- 图案周围的背景色必须形成一个‘安静区’。

对图像采集的要求

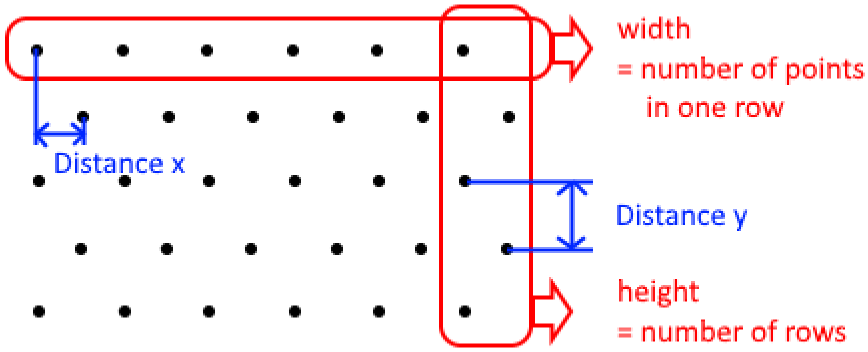
- 曝光
  - 图案照度均匀。
  - 圆圈和背景之间的对比良好。
- 数字
  - 透视畸变应保持在最低限度。
  - 必须遵守或适应**高级校准选项**的标准规格。默认情况下，一个圆必须由至少 25 个像素且最大 15000 像素的区域来表示。这个区域不能太小，否则就会接收到来自背景的干扰。

相机校准助手 [▶ 94]中的配置

- **开始缩进**
  第一行相对于第二行缩进。



- **宽度**
  对应于每行的圆圈数量。
- **高度**
  对应于每列的圆圈数量。
- **距离 x**
  对应于水平方向（跨行）上到最近圆心的距离。
- **距离 y**
  对应于垂直方向上到最近圆心的距离（在一列内）。



- **颜色反转**
  在在深色背景上使用浅色图案时。

5.4.4.4 单个圆圈图案

单个圆圈模板由多个圆圈组成，用户可以选择自定义位置，从而对单个圆圈组件进行排列。这也支持使用复杂的定制 3D 模板进行标定。

模板要求

- 模板应尽可能应用于平坦（平面）表面
- 圆圈大小应尽可能相等。
- 一个模板必须由至少 9 个点组成。
- 距离必须足够大，以确保圆圈不会相互接触，并会被识别为独立物体。
- 模板周围的背景颜色必须形成一个“安静区”。

图像采集要求

- 曝光
  - 模板的均匀照明。
  - 圆圈与背景对比鲜明。
- 图
  - 透视失真应保持在最低限度。
  - 必须遵守或调整**高级标定选项**的标准规格。默认情况下，一个圆圈必须用至少 25 像素、最多 15000 像素的区域来表示。区域不能太小，否则会接收到背景干扰。

描述文件要求

- 模板必须以 XML 文件的形式描述，并存储在开发计算机的以下文件夹中：*C:\Users\Public\TcVision\CalibrationPattern*
- XML 文件必须根据 *%TwinCAT3DIR%\Components\Vision\TcVision* 文件夹中的 TcVnCalibrationPattern.xsd 模式进行验证。
- 描述文件必须至少包含显示名称、原点和圆点的二维或三维坐标。

中心点

由于中心点是由点的描述自动给出，因此原点规格只是充当补充信息，在计算中不进行评估。

二维模板

当指定二维模板时，在内部使用 0 的 Z 坐标以进行计算。

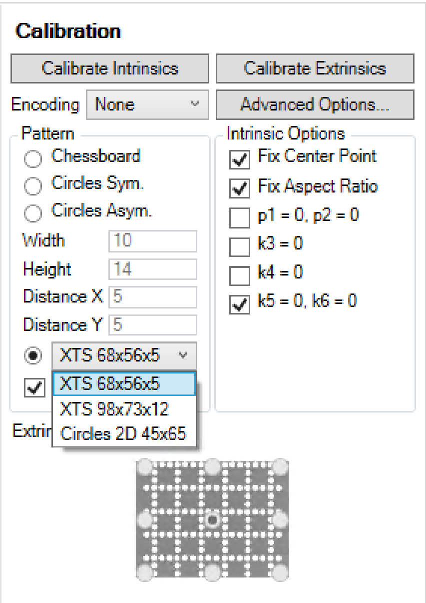
常量偏移量

常量偏移量可以作为具有常量 Z 值的三维模板来实现。

相机标定助手 [▶ 94]中的配置

如果在指定文件夹中找到有效 XML 描述文件，则会用从该文件中读取的显示名称显示这些文件。

- **单个圆圈模板的下拉菜单**
  选择单个圆圈模板。
- **颜色反转**
  用于选择图像颜色反转的复选框。默认情况下（无复选标记），白色背景上显示黑色对象。
- **外部原点**
  该值将自动设置为针对所选模板存储在文件中的值，无法通过用户界面更改。
  - 如果在描述文件中指定了包括文件扩展名在内的图像名称，且图像文件位于同一文件夹中，则会相应显示标定模板示意图的可选背景图像。



单个圆圈模板描述文件的样本示例

以下代码块可用作创建您自己描述文件的模板，并包含已完成元数据和二维点描述的样本。三维点描述也包括在内，但已被注释掉。如果需要三维，则必须用 Pattern3D 块完全取代 Pattern2D 块，即不允许混合。

点分配

如果点按行或列排序，并与标定图像中点的位置直接匹配，这会很有帮助。这样可以更快且更可靠地分配点。

```
<?xml version="1.0" encoding="UTF-8"?>
<TcVnCalibrationPattern xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="file:///C:/TwinCAT/3.1/Components/Vision/TcVision/
TcVnCalibrationPattern.xsd">
    <MetaData>
        <DisplayName>12345</DisplayName>
        <Origin>Center</Origin>
        <PreviewImage>PreviewImageSample.png</PreviewImage>
        <Description>This is a calibration pattern</Description>
        <Revision>V1.0</Revision>
        <Width>10</Width>
        <Height>10</Height>
        <Depth>4</Depth>
        <NumberOfPoints>9</NumberOfPoints>
        <PointDiameter>1</PointDiameter>
    </MetaData>
    <Pattern2D>
        <!-- 1st row -->
        <Point2D x="-5" y="-5"/>
        <Point2D x="-0" y="-5"/>
        <Point2D x="5" y="-5"/>
        <!-- 2nd row -->
        <Point2D x="-5" y="0"/>
        <Point2D x="0" y="0"/>
        <Point2D x="5" y="0"/>
        <!-- 3rd row -->
        <Point2D x="-5" y="5"/>
        <Point2D x="0" y="5"/>
        <Point2D x="5" y="5"/>
    </Pattern2D>
    <!-- <Pattern3D> -->
        <!-- 1st row -->
        <!-- <Point3D x="-5" y="-5" z="-2"/> -->
        <!-- <Point3D x="-0" y="-5" z="-2"/> -->
        <!-- <Point3D x="5" y="-5" z="-2"/> -->
        <!-- 2nd row -->
        <!-- <Point3D x="-5" y="0" z="0"/> -->
        <!-- <Point3D x="0" y="0" z="0"/> -->
        <!-- <Point3D x="5" y="0" z="0"/> -->
        <!-- 3rd row -->
        <!-- <Point3D x="-5" y="5" z="2"/> -->
        <!-- <Point3D x="0" y="5" z="2"/> -->
        <!-- <Point3D x="5" y="5" z="2"/> -->
    <!-- </Pattern3D> -->
</TcVnCalibrationPattern>
```

5.4.5 GigE Vision

TwinCAT Vision 或 TC3 GigE Vision 连接器产品已获得 GigE Vision（2.0.3 版）认证。因此，TwinCAT Vision 可以与所有兼容 GigE Vision 的相机一起使用。GigE Vision 标准在每个版本中都细分为必备的和可选功能，因此认证并不意味着该版本的所有功能都受到支持。由于相机本身也可能有不符合 GigE Vision 标准的额外功能，例如数据压缩，因此不能声明完全支持。因此，建议单独比较和测试所需的功能。

GigE Vision 标准

GigE Vision 是一个工业图像处理的接口标准。它为工业相机的配置和操作提供了便利。GigE 是指千兆位以太网。通过使用千兆位以太网通信协议，GigE Vision 受益于以下特点：

- 默认传输速率为 1 千兆/秒（可扩展为 2.5 千兆、5 千兆和 10 千兆）。
- 电缆长度可达 100 米，无需放大。
- 使用千兆以太网的标准硬件和软件接口。

千兆以太网视觉标准定义了通过 UDP/IP 与兼容设备的通信，且由以下四个元素组成：

- 千兆以太网视觉控制协议（GVCP）定义了如何对 GigE Vision 设备进行寻址。它规定了 PC 和相机之间传输图像和配置数据的数据通道和机制。
- 千兆以太网视觉流协议（GVSP）规定了用于将图像从相机传输到 PC 的不同数据类型和传输方法。另外，提供了一个数据包重发选项，可用于修复传输错误。
- 通过 GigE 设备发现机制，可使用搜索查询寻找网络中的相机。
- 一个 XML 文件包含 GenAPI 描述，其中定义了相机的所有公共功能。这种描述基于 GenICam 标准。

GenICam 标准

关于相机 GenICam 标准和 GenAPI 描述的信息，可查看EVMA 网站。

兼容性

如何了解一台相机是否与 TwinCAT Vision 兼容：查看产品描述中是否明确提到了“GigE Vision”这一名称，或者是否显示了 GigE Vision 标识：

“GigE”与“GigE Vision”的区别

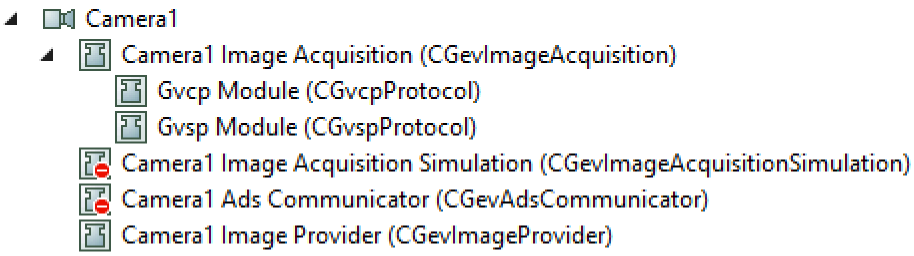
请注意“GigE”和“GigE Vision”之间的区别！仅支持符合 GigE Vision 要求的相机。

不支持与其他设备的连接

此外，GigE Vision 标准允许与其他设备（如频闪控制器）连接。TwinCAT Vision 目前不支持此功能。

5.4.6 TcCom 对象

在 TwinCAT 项目中，每个 GigE Vision 相机实例均包含若干 TcCOM 对象：



由 TwinCAT Vision 自动管理的 TcCOM 对象

TcCOM 对象始终由 TwinCAT Vision 自动管理，无需手动调整或配置。因此，以下信息主要用于提供更全面的技术理解。

循环任务

如果 TcCOM 对象 GevImageAcquisition、GevImageAcquisitionSimulation 和GevAdsCommunicator 被激活，则必须将它们与循环任务相链接。相应配置载于对象的 **Context**（上下文） 选项卡中。通常情况下，无需手动链接任务，因为创建 Vision 设备时所需的任务会自动创建和链接。有关创建和链接任务的详细信息，请参见 CPU 内核和任务 [▶ 55] 章节。

GigE Vision 图像采集

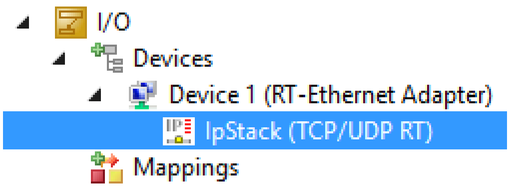
GevImageAcquisition 对象协调与 GigE Vision 相机的通信。它整合并管理实现 GigE Vision [▶ 107] 标准的两个协议的两个子对象：

- GigE Vision 控制协议（GVCP）
- GigE Vision 流协议 (GVSP)

有关相机通信的诊断信息载于两个子对象的 **Parameter (Online)**（**参数，联机**）选项卡。

它还在适当的网络适配器上使用 UDP/IP 栈与 GigE Vision 相机通信。诊断信息也可以在这些对象中检索到。有关这两个诊断选项的详细信息，请参见位于相机通信 [▶ 2751]的“附录/故障排除”部分。

在 TwinCAT 项目的 I/O 节点下可以找到 RT 以太网适配器和 IP 栈：



图像采集和两个协议（GVCP 和 GVSP）的参数设置在配置树 [▶ 70]中的配置助手 [▶ 67]中进行。有关参数的描述，请参见 TcCOM 参数 [▶ 87] 章节。相机仿真激活时，此对象将被停用。

GigE Vision 图像采集模拟

GevImageAcquisitionSimulation 对象模拟了 GigE Vision 相机数据流行为，且默认情况下停用。如果模拟模式处于激活状态，GevImageAcquisition 对象反而会被停用。

启用块 ID 校正	该设置可用于明确指出应使用人工生成的连续块 ID，而不是原始 ID。 默认：禁用

GigE Vision Ads 通信器

GevAdsCommunicator 对象在记录相机流 [▶ 89]时需要。它通过 ADS 将相应的图像从路由器内存发送到 TwinCAT Vision 服务。相比通过功能块 FB_VN_WriteImage 保存图像，记录功能需要单独的 TcCOM 对象，因为记录是独立于 PLC 处理而运行。

在配置模式下，再使用一个 GevAdsCommunicator 模块的实例，以便所有相机助手能够与相机进行通信。

发布图片	通过该设置，Ads 通信器每隔 n 个图像就会向 TwinCAT Vision 服务发送一次。该设置在激活数据流记录时自动设置，并取决于记录/回放选项卡 [▶ 89]的**图像偏移**参数。它不应手动设置。 默认值：0

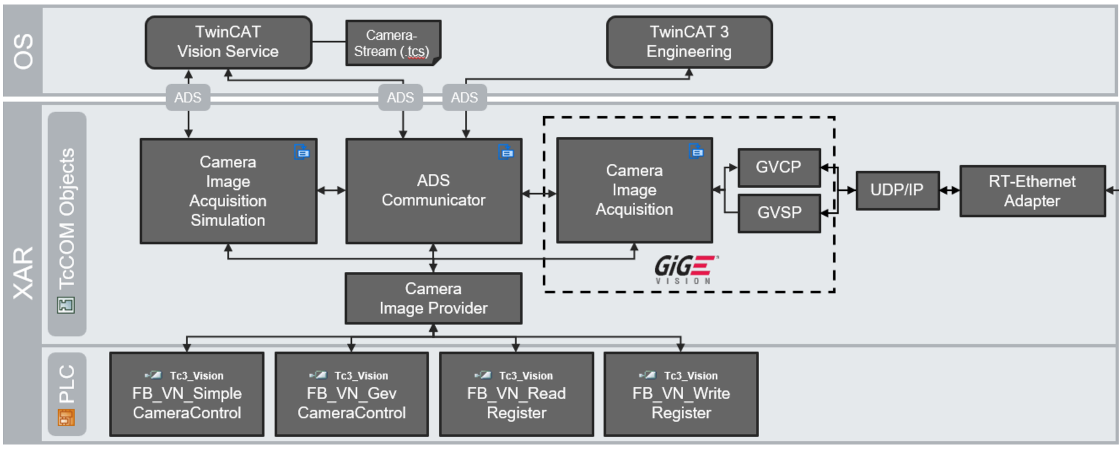
GigE Vision 图像提供者

GevImageProvider 对象代表相机和 PLC 中相应功能块之间的接口。其中包括FB_VN_GevCameraControl、FB_VN_SimpleCameraControl、FB_VN_ReadRegister和FB_VN_WriteRegister以及FB_VN_ReadMemory和FB_VN_WriteMemory。为此，项目实例的符号初始化选项卡中的对象必须与所需功能块的 PLC 符号相连接。如果项目中存在多个相机实例，确保根据名称选择正确的 GevImageProvider 模块。

图像队列大小	如果 PLC 中的功能块不直接接受所采集图像，需要进行缓冲的采集图像数量。 默认值：1

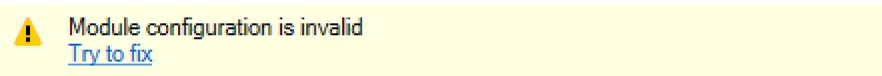
链接 TcCOM 对象

下面的概览图显示了相互通信或检索数据的不同对象之间的简化关系。



附图 5: GigE Vision 相机功能的架构

TcCOM 对象的链接根据模拟模式和 Ads 通信器对象的设置而有所不同（见常规选项卡 [▸ 64]）。关于相应的链接设置，可查看各个对象的**接口指针**选项卡。默认情况下，该链接根据相机助手中的设置自动创建，而不应手动更改。如果链接被错误更改或者 TcCOM 对象后来被添加到相机中，则可以在**常规**选项卡中单击**尝试修复**修复链接。



附图 6: 关于 TcCOM 对象的无效链接的说明

5.5

还请参阅有关此

- GigE Vision 相机对象 [▸ 63]

5.6 文件源对象

文件源代表相机的另一个图像源。它使存储的图像可以从文件系统中加载到 TwinCAT 实时系统中，而不是实时图像。请注意，与 GigE Vision 相机对象的记录/回放 [▸ 89]功能相比，文件源不是用于加载连续的相机流，而是用于加载单个图像。

只要图片是以标准的图片格式保存，如 bmp、jpg、png，就可以使用。

压缩图像格式

当使用压缩图像格式时，这取决于所应用的方法是否有损，因为图像数据之后将不再对应于相机的原始图像数据。因此，建议使用 BMP 作为存储格式。

灰度图像可以是彩色图像

看起来像灰度图像的图像可以用几个通道保存。在调用转换功能时注意这一点。

功能块FB_VN_FileSourceControl [▸ 1541]和FB_VN_SimpleCameraControl [▸ 1575]的存在是为了能在 PLC 中接收图像。与文件源控制的通信通过这些功能块进行，类似于与相机的通信方式。另外，将单个图像加载到 TwinCAT 实时中的另一个选择功能块FB_VN_ReadImage [▸ 1529]。

图像传输模式

文件源控制有两种提供图像的模式：

1. 数据流模式：像流式相机一样，以指定周期提供新的图像。
2. 触发模式：通过触发信号提供新的图像。这可以通过软件触发器或手动启动按钮触发。

设置和图像选择在文件源控制 [▸ 111]下完成。

创建文件源节点

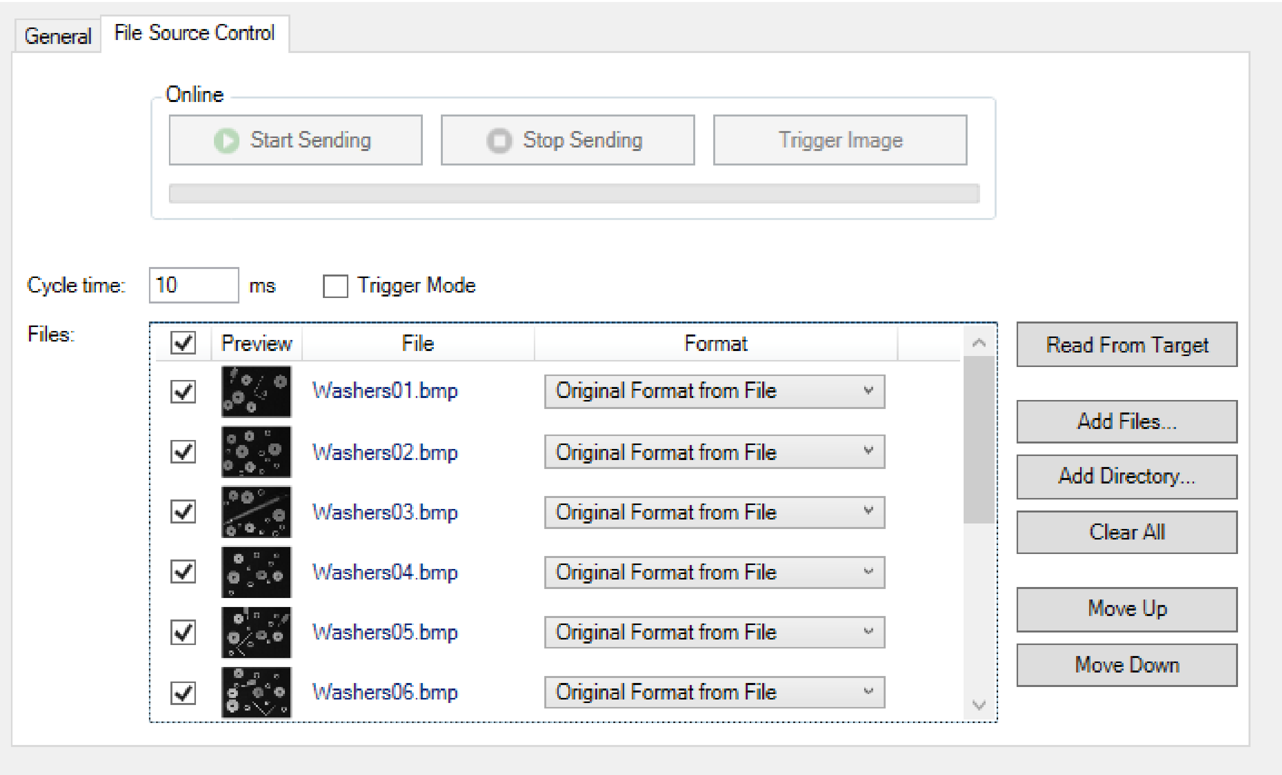
关于创建文件源节点的信息，可参见创建文件源控制 [▶ 31]下第一步 [▶ 23]。

文件源节点的导出/导入

如果想要保存文件源的配置或将其导入项目，请按照导出/导入配置 [▶ 118]一章中的说明进行操作。

5.6.1 文件源控制

文件源控制的配置在同名的选项卡上进行：
视觉节点 > 文件源节点 > 文件源控制选项卡。

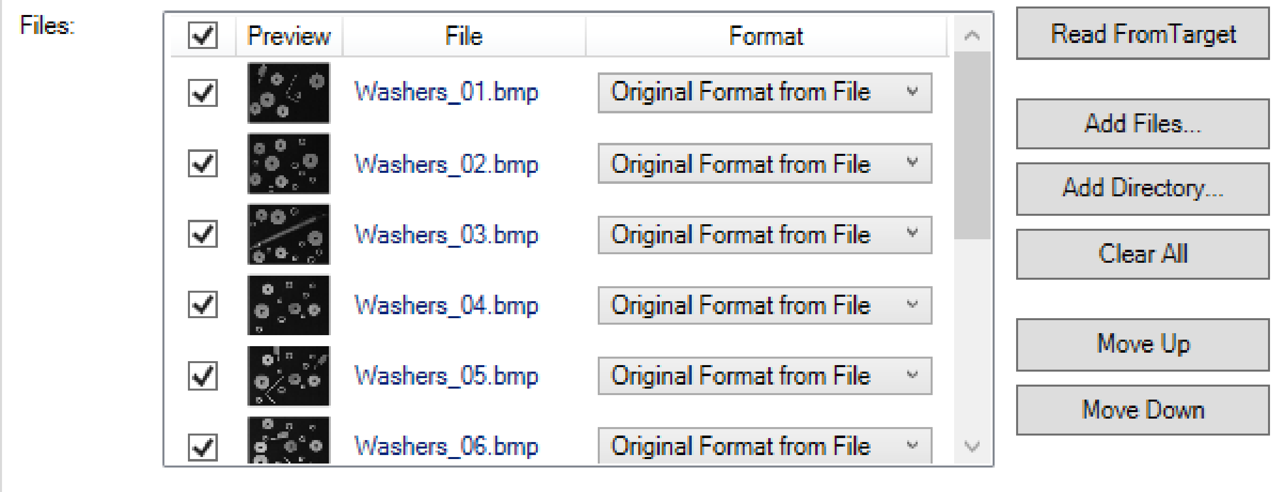


附图 7: 文件源控制

管理图像选择

文件源控制待使用的图像通过文件列表和相关控制进行管理：

> **从目标读取**
>
> 在创建一个文件源后，必须按一次**从目标读取**，并激活配置，以便可以添加图像。

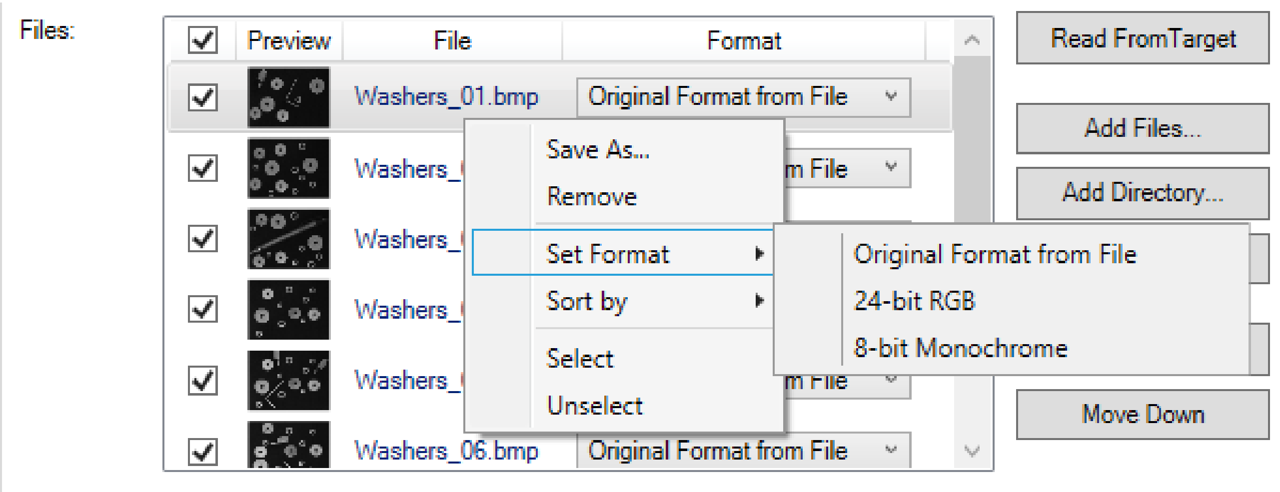


附图 8: 文件源控制中的图像选择

复选框	第一栏中的复选框指示是否使用了图像。
预览	预览图像的大小取决于列的宽度。
文件	图像的文件名，包括数据类型。
格式	默认情况下，图像会以保存的格式发送。另外，也可以指定事先将其转换为 1 通道灰度图像或 3 通道 RGB 图像，然后再发送。 **建议明确指定每张图片的格式。**
从目标读取	读取相关文件源控制在目标系统上存储的图像信息。
添加文件	通过文件资源管理器，可以选择来自开发系统的图像。然后，在后台将这些内容复制到目标系统，这样它们就可以独立于开发系统在那里使用。请注意，仅可使用具有独特名称的文件。此外，此处仅可添加图像格式，即不能添加视频格式。如果想要把相机流（*.tcs）载入 PLC，请使用记录/回放 [▶ 89]功能。 **除非另有配置，否则图像文件将存储在目标系统的路径“C:\Users\Public\TcVision\FileSources”中。确保相应的路径不受其他软件（如WriteFilter）的影响。**
添加目录	通过文件资源管理器可以选择开发系统的一个目录，所有图像都要从这个目录中使用。
清除所有	所有图像将从文件列表中删除。
上移/下移	选定图像在文件列表中向上/向下移动一个位置。

上下文菜单

使用CTRL+点击，在文件列表中选择几个图像。通过右键点击其中一个选定的图像，然后可以调整所有选定图像的设置：



附图 9: 文件源控制中的图像选择的上下文菜单

另存为...	打开一个文件资源管理器对话框以保存所选图像。
删除	删除选定的图像。
设置格式	为所有选定的图像设置图像格式： • **来自文件的原始格式：**使用存储图像的格式。 • **24 位 RGB：**图像被文件源作为 3 通道 RGB 图像发送。 • **8 位单色：**图像被文件源作为单通道灰度图像发送。
排序方式	按文件列表对所有图像进行排序： • **名称（升序）：**按名称升序 • **名称（降序）：**按名称降序
选择	选择突出显示的图像，以便它们可以通过文件源控制发送。
取消选择	取消选择突出显示的图像，以便不通过文件源控制发送。

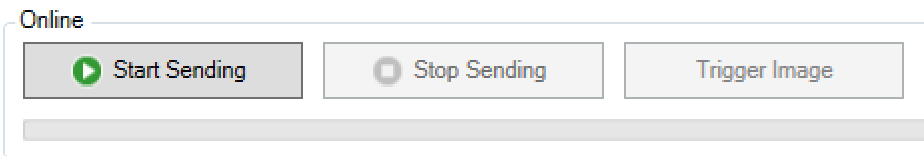
图像传输设置

像流式相机一样，文件源控制可以在指定的时间周期或调用软件触发器后提供新的图像。

附图 10: 在文件源控制中设置选项

周期时间	如果没有选择触发模式，图像将在周期时间的间隔内被发送到实时环境中。根据文件列表，图像的顺序被考虑在内。这个参数与相机的帧率相当。 **加载图像的最大速度取决于系统和图像的情况。周期时间只是期望值。**
触发模式	如果选择了触发模式，则在调用相应的软件触发器后会发送图像。通过标准的软件触发器，遵守根据文件列表的图像顺序。另外，也可以根据文件名触发特定的图像。

在线



附图 11: 文件源控制中的操作

开始发送	PLC 中的**开始发送**对应于功能块FB_VN_FileSourceControl [▶ 1541]的开始采集 [▶ 1546]方法。如果没有选择触发器，则图像将在**周期时间**间隔通过文件源控制发送。如果选择了触发器，文件源控制会对软件触发器做出反应，并根据触发信号发送图像。
停止发送	PLC 中的**停止发送**对应于功能块FB_VN_FileSourceControl [▶ 1541]的停止采集 [▶ 1547]方法，并可终止通过文件源控制发送图像。
触发图像	PLC 中的**触发图像**对应于功能块FB_VN_FileSourceControl [▶ 1541]的触发图像 [▶ 1549]方法。这可以用来手动触发图像。这个按钮仅在触发模式启用且通过文件源传输图像功能激活的情况下才可用。

5.6.2 TcCom 对象

在 TwinCAT 项目中，每个 FileSource 实例均包含两个 TcCOM 对象：

TwinCAT Vision 自动管理 TcCOM 对象

TcCOM 对象始终由 TwinCAT Vision 自动管理，无需手动调整或配置。因此，以下信息主要用于提供更全面的技术理解。

文件图像采集

TcIoFileImageAcquisition 对象为 FileSource 实例协调从硬盘加载图像。这个对象包含以下参数（除了CycleTime和TriggerMode，这些参数可以在文件源控制 [▶ 111]中设置）：

基地目录	目标系统上的文件路径，添加到选择列表中的图像将存储在该路径下。 标准：*C:\Users\PublicTcVision\FileSources\<UniqueId>*
服务超时	与TwinCAT Vision Service [▶ 59]通信的超时（单位：毫秒）。 默认值：5000

循环任务

文件图像采集对象必须与循环任务相联系。关于相应配置，可查看对象的**上下文**选项卡。通常情况下，无需手动连接任务，因为创建 Vision 设备时需要的任务会自动创建和连接。关于创建和链接任务的详细信息，可查看CPU 内核和任务 [▶ 55]章节。

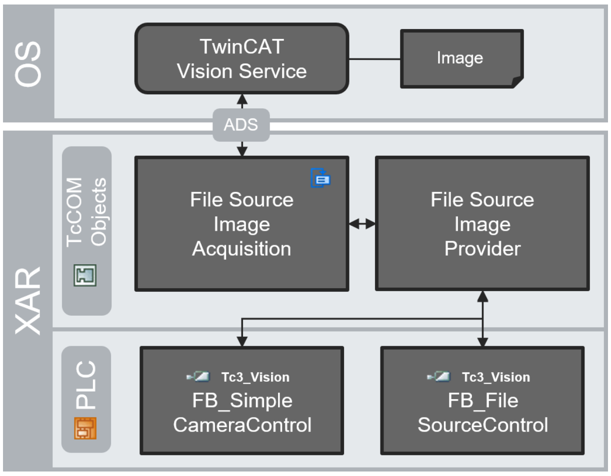
文件图像提供者

这个对象代表了 PLC 中文件源对象和文件源功能块之间的接口。功能块FB_VN_FileSourceControl [▶ 1541]或FB_VN_SimpleCameraControl [▶ 1575]的 ObjectID 变量必须在编译 PLC 代码后在项目实例的**符号初始化**选项卡上与相应的文件图像提供者 TcCOM 对象链接。如果项目中存在几个文件源实例，必须注意根据名称选择正确的对象。这个对象包含以下参数：

图像队列大小	如果 PLC 中的功能块不直接接受已加载的图像，则要对其进行缓冲的数量。 默认值：1

链接 TcCOM 对象

对于文件源设备，仅图像提供者必须与图像采集链接。默认情况下，这个链接会自动创建，且不应手动更改。下面的概览图显示了相互通信或检索数据的不同对象之间的简化关系。

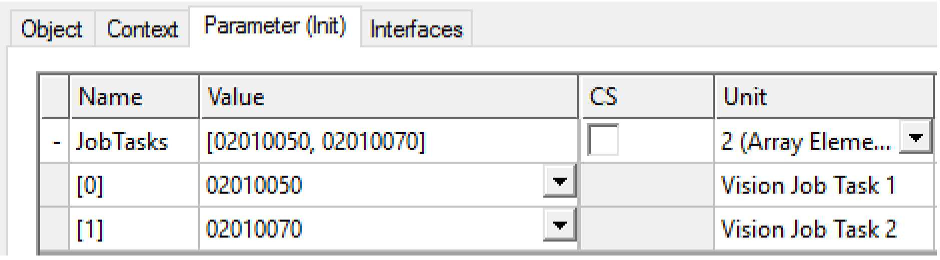


附图 12: 文件源功能的架构

5.7 视觉工作池

视觉工作池用于针对图像处理任务管理工作任务 [▶ 58]。这使得选定的 TwinCAT Vision 算法可以并行执行，而这些算法在API 参考资料 [▶ 119]中由以下补充内容标记：“可使用可用的 TwinCAT 工作任务执行并行代码区域”。

点击**视觉工作池**，打开一个窗口。工作任务的分配在这个窗口的**参数（初始化）**选项卡上进行。在第一行，**单位**栏指示需要分配多少工作任务。然后，**工作任务**下的数组元素可以通过**值**列中的下拉菜单分配一个工作任务。每个工作任务仅可分配给一个元素。



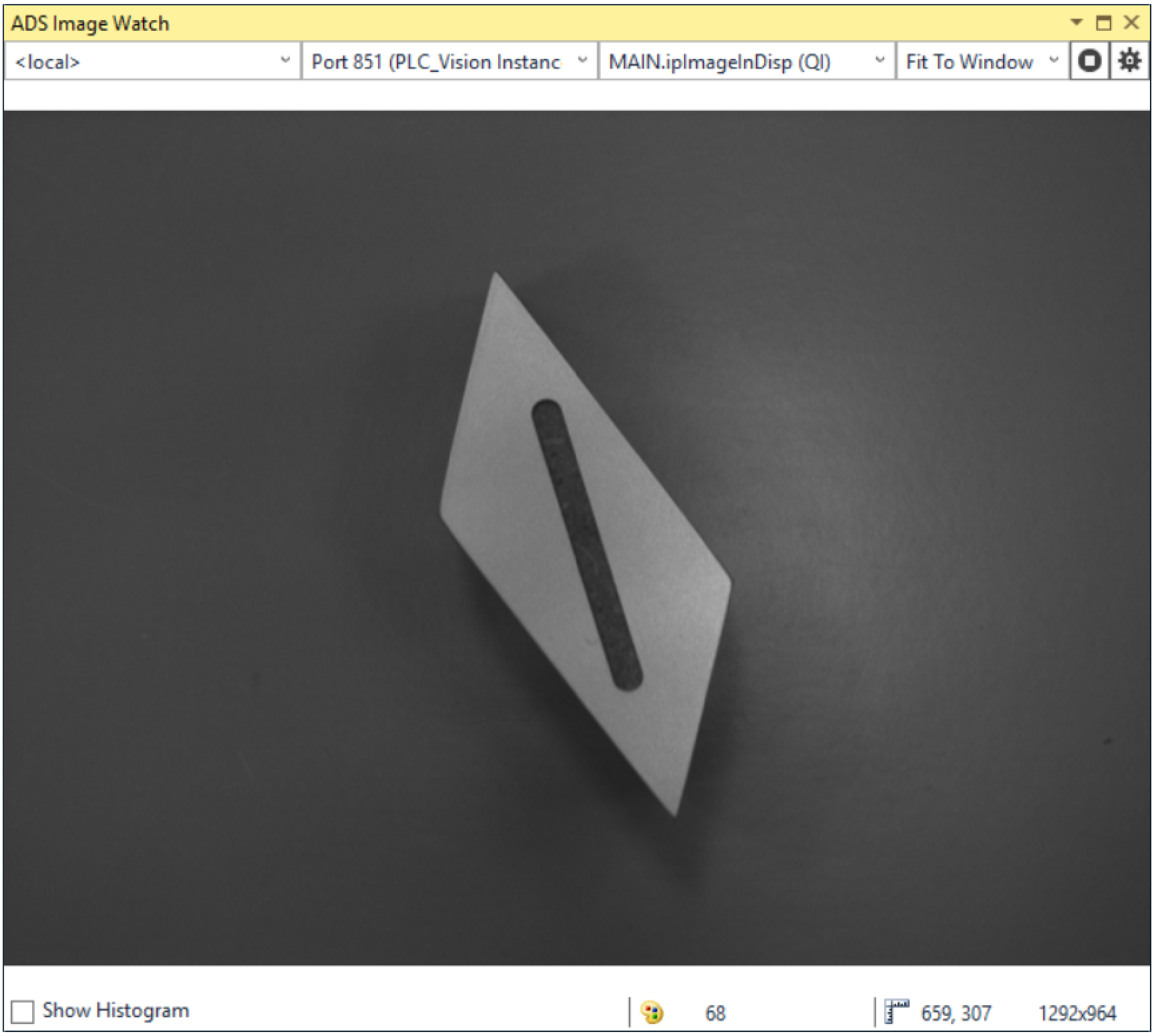
	Name	Value	CS	Unit
-	JobTasks	[02010050, 02010070]		2 (Array Eleme...
	[0]	02010050		Vision Job Task 1
	[1]	02010070		Vision Job Task 2

注意

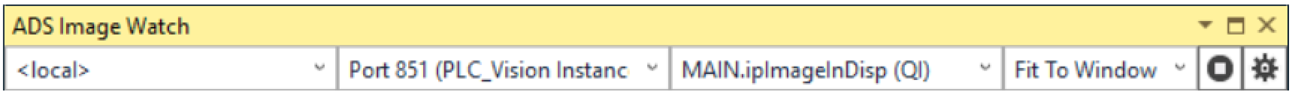
在创建工作任务时，请务必参考限制和建议 [▶ 59]。

5.8 ADS 图像查看

通过 ADS 图像查看，类型ITcVnDisplayableImage [▶ 390]的图像可以从 PLC 中检索到并在开发环境中显示。通过 Visual Studio 菜单**TwinCAT > Windows > ADS 图像查看**打开 ADS 图像查看。



图像选择



图像选择通过选择区域从左到右进行。

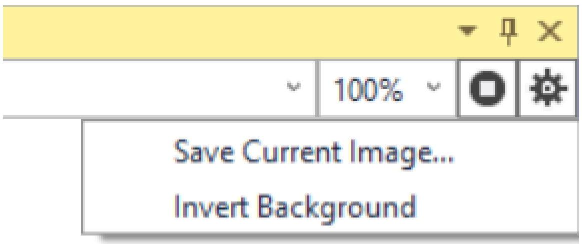
目标系统	ADS 设备通过ADS-AmsNetId 和 ADS 端口寻址。ADS-AmsNetId 明白地确定了必须首先选择的目标系统。默认情况下选择本地系统。然而，可以选择任何已经创建了路线的系统；见选择目标系统 [▶ 25]。
端口	系统中的各个设备由 ADS 端口号识别。为 PLC 运行时间定义了一个固定的端口号范围，因此选择列表已根据这些端口号预先过滤。如果在 ADS 图像查看中选择后目标系统上只存在一个 PLC 端口，则会自动选择 PLC 端口。
图像变量	所有类型为ITcVnDisplayableImage [▶ 390]的变量均显示在此处且可以选中，无论该变量当时是否指向受支持图像格式 [▶ 117]的图像。
大小	如果没有更多设置，图像会被缩放到当前窗口的大小（**适应窗口**）。然而，它可以选择性地被设置为相对于原始图像的尺寸。请注意，如果选择的尺寸小于原始图像，则在通过 ADS 传输之前图像会被实时环境中的*PlcAux 任务*缩小。如果之后保存图像，则图像不不具有原始尺寸。
停止/播放	交互停止/播放按钮可用于停止或重新启动图像显示的更新。如果暂停图标可见，则更新处于活动状态，并持续接收图像。如果播放图标可见，说明更新已停止，并保持最后一张图片的显示。不再接收更多的图像。 这使得分析图像和每个细节成为可能，例如通过放大。

图像缩放

如果选择的尺寸不是 100%，显示的图像将被插值。对于**适应窗口**或缩小，使用线性插值，对于放大，使用最近邻插值。

根据图像内容或像素值的不同，可能会出现偏差。如果需要对个别像素进行详细分析，应因此将尺寸设置为 100％。

选项



通过右上方图像大小旁边的**行动**图标，然后使用**保存当前图像**，可以保存当前显示的图像。

此外，显示图像周围的背景可以通过**反转背景**在明暗之间改变。这对非常明亮或黑暗的图像很有帮助，这样可以清楚地看到图像内容或图像边缘，并通过与背景的高度对比来进行区分。

图像大小

图像被保存为 ADS 图像查看从 PLC 查询的尺寸。为了以原始尺寸保存图像，必须将图像至少设置为100%，或将窗口至少拉到原始图像尺寸。如果图像大小被设置为大于 100%，图像也将以原始尺寸保存。如果设置较小，使用**适应窗口**或窗口太小，图像将以相应的缩小尺寸存储。

支持的图像格式

在 ADS 图像查看中，以下图像格式可以在下列色彩空间中显示：

- 1 **通道图像**被解释为**灰度图像**。支持所有像素类型为枚举ETcVnElementType [▶ 178]的图像。
- 3 **通道图像**被解释为**RGB 彩色图像**。支持每通道 8 位或 16 位的图像；这对应于像素类型 [▶ 178]USINT、SINT、UINT 和 INT。
- 4 **通道图像**被解释为**RGBA 彩色图像**。这里支持每通道 8 位/像素的图像；这对应于像素类型 [▶ 178]USINT 和 SINT。

其他图像格式

其他图像格式和色彩空间可以在 PLC 中作为 ITcVnImage 处理。然而，为了显示它们，它们必须首先通过 F_VN_ConvertColorSpace 转换为可显示的格式之一，然后再转换为 ITcVnDisplayableImage。如果不这样做，图像要么以错误的颜色表示方式显示，要么根本就不显示。

显示 16 位图像

16 位图像作为 8 位图像从 PLC 传输到 ADS 图像查看。因此，像素值只在状态栏中以 256 的增量显示（0、256、512 等），但还是与 16 位有关：（0：最小，65.280：最大）。

显示（L）REAL 图像

（L）REAL 图像作为 8 位图像从 PLC 传输到 ADS 图像查看。为此，数值范围[-1，+1]被缩放为[-127，+127]并以整数传输。在这些限制之外的浮点值将被解释为 -1 或 +1。在 ADS 图像查看中，待传输的像素值再次除以 127，以便在[-1，+1]范围内显示浮点值。

状态栏

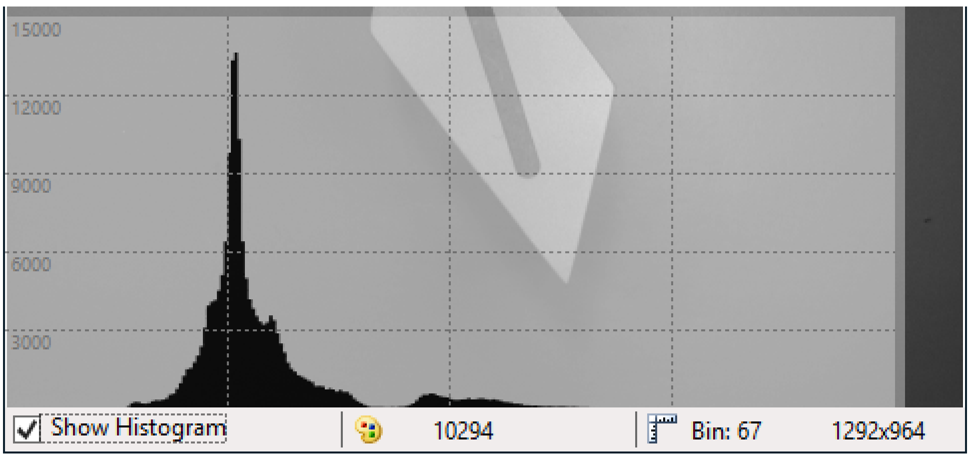
图像下方的状态栏显示以下信息（从左到右）：

附图 13: ADS 图像查看中的状态栏

- 图像上鼠标指针位置的颜色值/强度值
  或直方图中鼠标指针位置的相应容器中的像素数
- 在图像上鼠标指针位置的图像坐标
  或直方图中鼠标指针位置的容器
- 图像分辨率

直方图

当**显示直方图**复选框被选中时，图像的直方图在图像的左下方显示。这可以有几个通道叠加显示，具体取决于图像类型。



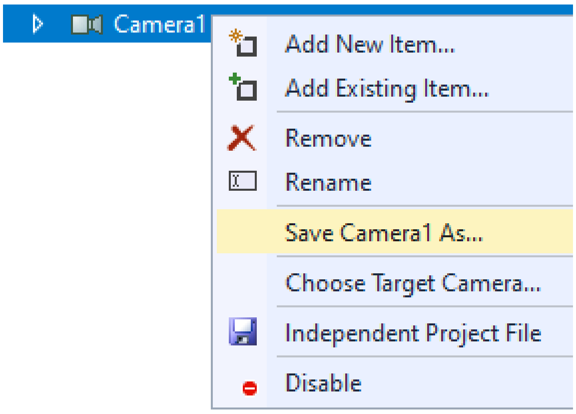
附图 14: ADS 图像查看中的直方图

5.9 导出/导入节点

TwinCAT 提供将 I/O 或 MOTION 元件等单独配置节点保存在单独的 XTI 文件中或从该等文件中导入的选项。使用应用节点以及 GigE Vision 相机和文件源对象也可以实现这一点。这样，Vision 设备的配置可以独立于项目保存，并在其他地方重新导入。

导出

1. 右键点击要导出的节点，并选择 Save <Name> As…（将<名称>另存为……）

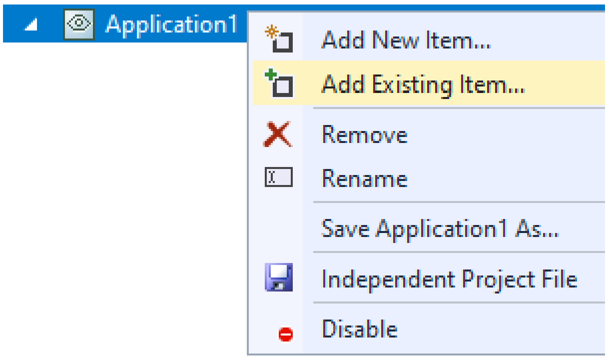


2. 为 XTI 文件选择一个位置。

导入

根据要导入整个应用节点还是只导入单个 Vision 设备，应在 VISION 节点或相应的应用节点上执行以下步骤。

3. 右键点击要导入配置的节点，然后点击 Add Existing Item…（添加现有项……）



4. 选择要导入的 XTI 文件。
5. 如有必要，为导入的 Vision 设备应用特定于系统的设置（如网络适配器）。

复制相机配置

请注意，只有在相机型号和固件相同的情况下才能保证为多个相机成功传输相同的配置。否则可能会出现问题，例如相机寄存器地址改变。

独立项目文件

也可以将配置节点保存在项目中的一个独立文件中，而不是完全导出。在这种情况下，相应节点的配置不会保存在 TwinCAT 项目文件中，而是保存在一个单独的 XTI 文件中。例如，这有助于版本控制系统，当修改配置时，不会更改整个项目文件。

6 API 参考

本章介绍 TwinCAT Vision API，它为 TwinCAT 3 Vision 库的编程使用提供了选项。由于 TwinCAT Vision 可以在 TwinCAT 3 PLC 和 TwinCAT 3 C++ 中使用，因此这两个文档构成了基础。因此，所有特定于 TwinCAT Vision 的描述均分为以下两个子章节。

6.1 PLC

本章包含有关在 PLC 中编程使用库的 TwinCAT Vision API 描述。由于 TwinCAT Vision 带来了一些创新，因此对软件概念进行了明确说明：

- 软件概念 [▸ 119]
- 数据类型 [▸ 139]
  （别名、常量、数组、枚举、结构）
- 接口 [▸ 226]
  （图像、容器等）
- 函数 [▸ 408]
  （图像处理）
- 功能块 [▸ 1502]
  （与相机和文件系统通信、图像处理）

6.1.1 软件概念

本章介绍 TwinCAT Vision 带来的创新或特殊特征。TwinCAT 3 中的PLC 编程原理以及参考编程载于一般 TwinCAT 3 文档。在 PLC 的描述中，有一章介绍TwinCAT 3 编程约定。其中包含的建议和说明以及优缺点可支持实现程序结构标准化，对象、变量、实例命名一致性，以及代码易读性和易懂性，从而简化程序的开发、使用和维护。

对库模块的明确访问

如果 Tc3_Vision 库要包含（引用）到另一个库中，并因此在一个项目中出现多次，则必须通过命名空间作为标识符的前缀进行明确的访问。默认，例如 hr :=
Tc3_Vision.F_VN_TransformIntoDisplayableImage(ipImageIn, ipImageInDisp, hr);

有关更多信息和设置选项，请参见 PLC 项目设置和库属性。

⚠ 警告

使用多个任务时访问相同数据

例如，如果多个任务要访问同一图像，则必须根据这些准则同步这些访问。否则，就有可能出现数据集不一致的情况，从而导致不可预见的后果和程序故障。

6.1.1.1 接口指针

TwinCAT Vision 需要动态内存区域，以保存图像和容器等。这些图像和容器由类型为ITcVnImage [▸ 390]和ITcVnContainer [▸ 349]的变量表示。由于它们是指向动态内存区域的接口，这些变量被称为接口指针。

通过 TwinCAT Vision API 功能，相关的内存管理在功能中被自动控制，且此处无需注意。

例如，创建一个新的图像（800 x 600 像素的类型TCVN_ET_USINT，包含一个通道）：

```
VAR
    hr          : HRESULT;
    ipImage     : ITcVnImage;
END_VAR

hr := F_VN_CreateImage(ipImage, 800, 600, TCVN_ET_USINT, 1, hr);
```

这个图像现在要被转换成 3 个通道的 RGB 图像。因此需要 3 倍的内存。下面的函数调用足以实现这一点：

```
hr := F_VN_ConvertColorSpace(ipImage, ipImage, TCVN_CST_GRAY_TO_RGB, hr);
```

在内部，它释放了先前的内存区域，并以合适的大小分配了新的内存。因此，接口指针是管理 PLC 中动态需要内存的对象的一种简单方法。

⚠ 警告

误用的后果

如不遵循以下关于使用接口指针的说明，可能会导致**系统完全崩溃**和**内存泄漏**！

参考计数器

然而，一旦变量不再专门作为 TwinCAT Vision API 功能的功能参数使用，则需要更准确地熟悉接口指针的概念。

务必了解，接口指针变量只是一个指向内存中的对象的指针。如果你把一个变量赋值给另一个变量，只有这个内存区域的指针被复制，而不是数据本身。

这意味着，如果两个变量中一个变量的数据被更改，另一个也会自动更改。
如果其中一个变量中的内存被释放（如上面的例子F_VN_ConvertColorSpace [▶ 1197]），而第二个变量仍然有其指向这个内存区域的指针，访问这个已经释放的内存往往会导致系统崩溃。

由于这个原因，内存中的对象都有一个参考计数器。只要一个变量被分配给另一个变量，这个引用计数器就必须增加。如果一个新的指针被分配给一个变量，那么必须再次减少参考计数器。

当参考计数器达到 0 时，内存被释放。因此，相应的数据被完全删除。

为了表明一个指针变量是否指向一个有效的内存区域，如果这个指针不再使用（参考计数器因此被减少），这个指针必须始终被设置为 0。默认情况下，当程序启动时，PLC 中所有创建的变量（因此也包括指针）最初被设置为 0。

i 接口指针的管理

在 TwinCAT Vision API 功能中（用F_VN_...标记），参考计数器由接口指针自动管理。因此，在使用它们时不需要特别注意。然而，如果接口指针被手动复制或使用，例如，作为功能参数，则需要手动管理（见下文）。

检查接口指针的有效性

由于接口指针值 0 基本上只对应一个无效的内存区域，而且对它的访问也会导致严重的系统错误，所以务必在每次使用前检查该值是否为非零。

在 TwinCAT Vision API 功能（F_VN_...）中，这种检查和相应的处理在内部自动进行。然而，对于下面示例中的TcRelease()等接口方法不可行，因为如果变量为 0，调用这些方法已经导致了错误。因此，必须事先明确检查变量。

⚠ 警告

对无效内存区域的方法访问导致系统崩溃

对于接口方法，内部检查不可行，因为相应的对象必须存在才能调用该方法。

即使在方法调用之外，也要确保相应的对象存在（接口指针应该是非零）。

此外，系统崩溃不一定总是发生。如果无效的内存区域碰巧保持未使用状态，系统崩溃可能不会发生。

启用接口指针

接口指针变量的启用包括检查该变量是否已为 0，否则将减少引用计数器并将该变量设置为 0。由于这是一个经常用例，因此提供了一个函数 FW_SafeRelease，只需一次调用即可执行所有必要步骤：

```
hr := FW_SafeRelease(ADR(ipImage));
```

FW_SafeRelease 函数的内部运行模式通过以下代码片段进行说明：

```
IF ipImage <> 0 THEN        // Check if interface pointer is already = 0
                            // If not, call .TcRelease() and set it to 0
    ipImage.TcRelease();    // decrement reference count
    ipImage := 0;           // set to 0
END_IF
```

复制接口指针

复制接口指针时，必须手动增加其引用计数器。通过 := 操作符进行直接分配，以及将接口指针作为输入、输出或返回值传输的函数/方法/功能块调用，都属于复制程序。通过 I/O 图像进行的传输也包括在内。可以使用以下方法增加引用计数器：

```
ipImage.TcAddRef(); // increment reference count
```

此外，复制前必须确保要写入的接口指针已启用。否则会出现内存泄漏。例如，复制接口指针的手动程序如下所示：

```
IF ipSrcImage <> 0 THEN                 // check if source pointer is valid
    FW_SafeRelease(ADR(ipDestImage));   // release destination pointer
    ipDestImage := ipSrcImage;          // assign source pointer to destination pointer
    ipSrcImage.TcAddRef();              // increment reference count
END_IF
```

请注意，只复制接口指针，而非底层图像。

经验法则

在处理接口指针时，请遵循以下经验法则。

接口指针是否是（可显示）图像、容器、迭代器等并不重要。— 它们在样本中可以相互替换。

如果调用接口指针上的方法

必须确保接口指针有效
（检查 <> 0）：

```
IF ipImage <> 0 THEN
    ipImage.GetImageInfo(stImageInfo);
END_IF
```

一般情况下，建议使用替代 F_VN_ 函数（如果存在的话），而不是接口方法，因为内部会自动执行检查：

```
hr := F_VN_GetImageInfo(ipImage, stImageInfo, hr);
```

如果在同一个表达式中执行检查和调用接口方法，则必须使用 AND_THEN 而非 AND。只有在这种情况下，如果前部返回 TRUE，后部才会被执行：

```
IF ipIterator <> 0 AND_THEN ipIterator.CheckIfEnd() <> S_OK THEN
    ipIterator.GetContainer(ADR(ipElement));
END_IF
```

手动复制接口指针

事先确保要写入的接口指针已启用。复制后增加接口指针的引用计数器：

```
FW_SafeRelease(ADR(ipDestImage));
IF ipSrcImage <> 0 THEN
    ipDestImage := ipSrcImage;
    ipSrcImage.TcAddRef();
END_IF
```

在 VAR_INPUT 中声明接口指针

对于函数和方法，什么都不需要观察。对于程序和功能块，变量中的指针会在调用后保留，因此应在 POU 结束时设置为 0：

```
VAR_INPUT
    ipSrcImage: ITcVnImage;
END_VAR
//… functional code
ipSrcImage := 0;
```

这样就能确保下次调用时无法再访问内存区域，因为它可能已经被释放出去。在 POU 中增加引用计数器并不是一种解决方案，因为新分配会覆盖先前指针，因此无法释放，从而导致内存泄漏。如果要保留接口指针，则应将其复制到本地变量。

将接口指针声明为 VAR 中的引用

在函数和方法结束时释放接口指针。结束 POU 后，接口指针不再存在，因此必须先减少引用计数器。对于程序和功能块，一旦不再需要接口指针，就应立即将其释放：

```
VAR
    ipImageWork: ITcVnImage;
END_VAR
//… functional code
hr := FW_SafeRelease(ADR(ipImageWork));
```

不建议 — 在 VAR_OUTPUT 中声明接口指针

不建议在 VAR_OUTPUT 中声明接口指针，因为输出的使用可选，因此可能会发生内存泄漏。
在程序和功能块中，变量中的指针即使在调用后也会保留，这样在外部发生变化时，也可以在功能块中访问已释放内存区域。
为了防止出现这种情况，必须始终确保使用输出，如果是程序和功能块，则必须在复制输出变量后增加引用计数器或将输出变量设置为 0。
这样做非常费力，而且容易出错，因为您必须要注意在代码的每一个点上正确实现。因此，建议将其作为 VAR_INPUT 中的引用进行转移。

注意
内存泄漏 不建议在 VAR_OUTPUT 中使用接口指针。未观察到上述点可能会导致内存泄漏和系统崩溃。

将接口指针声明为 VAR_INPUT 中的引用

如果要在 POU 内将对接口指针的更改（如分配新图像/容器）传递到外部，以便在那里进一步使用，则应使用作为引用的传输。

```
VAR_INPUT
    ipDestImage: REFERENCE TO ITcVnImage;
END_VAR
```

例程 自行编写的函数 [▶ 2712] 解释了在自写函数和方法中接口指针的正确使用。

误用的后果

如果一个接口指针变量被复制，但参考计数器没有同时调整TcAddRef()，就会有两个指针；但是，只有其中一个被程序所知。因此，一旦两个接口指针变量中的一个被释放，那么这两个接口指针变量所指向的数据就会被删除。释放可以通过手动执行FW_SafeRelease，也可以通过改写变量的 TwinCAT Vision API 功能来实现。如果剩下的接口指针变量再被方法调用访问，系统将**崩溃**，如上所述。

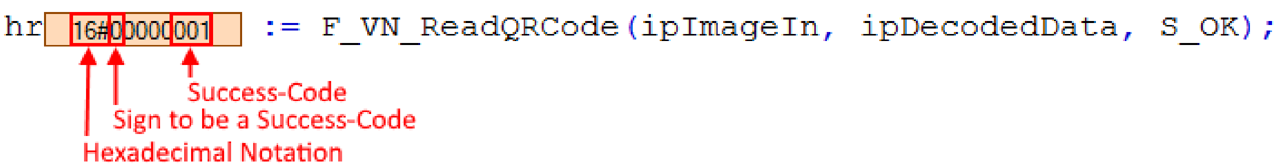
如果与此相反，接口指针变量在没有事先释放的情况下被改写（例如通过:= 赋值），这将导致**内存泄漏**。例如，如果接口指针被用作方法变量，并且在结束方法之前没有释放，就会发生这种情况。在 AMS 路由器的信息窗口中可以找到内存泄漏的指示（见路由器内存 [▶ 49]）。如果那里标记为可用的内存随着时间的推移而减少，这表明有内存泄漏。

6.1.1.2 HRESULT

所有 TwinCAT Vision 功能在执行后都会返回一个 HRESULT。它的值表示执行成功与否。

成功代码

成功的执行由正数代码表示。在十六进制的符号中，第一个数字在 0 和 7 之间。



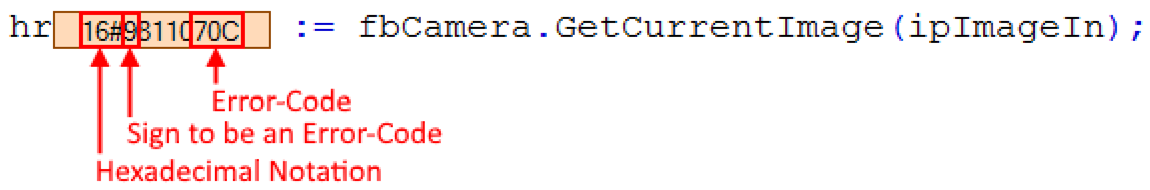
经常出现的成功代码是：

代码	名称	描述
16#000	S_OK	功能已成功执行
16#001	S_FALSE	功能已成功执行，但没有达到完整的结果。(例如，在图像中没有发现代码时，代码读取功能会出现这种情况。)
16#203	S_PENDING	异步方法已经启动，但还没有结果（例如，在第一次调用 fbCameraControl.StartAcquisition() 时出现；例如，在第二次调用 S_OK 返回时出现）
16#256	S_WATCHDOGTIMEOUT	函数被看门狗中止。

错误代码

失败的执行由负数代码表示，或者在十六进制符号中，第一个数字>=8。代码的最后几个数字对应于ADS 返回代码 [▶ 2753]。

> **注意**
>
> **错误代码**
>
> 如果返回了错误代码，表示功能的所有返回结果无效，且因此不得使用。



TwinCAT Vision API 元素的常见错误代码：

十六进制	十进制	名称	描述
16#70A	1802	NOMEMORY	内存不足
16#70B	1803	INVALIDPARM	参数值无效
16#70C	1804	NOTFOUND	未找到（文件、图像...）
16#70E	1806	INCOMPATIBLE	对象不匹配
16#712	1810	INVALIDSTATE	一个 FB 的方法在不允许的状态下被调用。
16#719	1817	TIMEOUT	超时
16#71A	1818	NOINTERFACE	接口查询失败
16#71B	1819	INVALIDINTERFACE	请求了错误的接口
16#71D	1821	INVALIDOBJID	对象 ID 无效。
16#734	1844	OUTOFRANGE	超出有效范围。

关于完整列表，请参见ADS 返回代码 [▶ 2753]。

应用

如果出现错误，HRESULT为负（错误代码），否则为正（成功代码）。使用FAILED ()和SUCCEEDED ()功能进行检查。

```
PROGRAM MAIN
VAR
    hr      : HRESULT := S_OK;
END_VAR

IF SUCCEEDED(hr) THEN
    //code
END_IF
```

> **限制性查询**
>
> 请注意，上述查询仅检查功能是否成功执行，而不是检查是否获得正确或期望的结果。

例如，如果在图像中没有发现或读到代码，则代码读取功能 [▶ 796]返回成功代码S_FALSE（16#001）。为了检测这种情况，必须将hr与S_OK或S_FALSE直接进行比较。相反，如果待测量的对象从图像中丢失，则测量功能 [▶ 1428]返回NOTFOUND（16#70C）。

大多数 TwinCAT Vision 功能都希望HRESULT作为最后一个输入参数，并且只有这是一个成功代码且从而确认到目前为止没有发生过错误时才会被执行。否则，相应的功能不会被执行，而是返回之前的HRESULT。这确保了，在HRESULT从一个功能传递到另一个功能时，输出的是第一个发生的错误代码，而不是后续的错误。

成功代码

成功代码（如S_FALSE）不被被下一个功能转发和覆盖。如有必要，必须明确检查各功能的 HRESULT 值。

访问接口方法

如果一个功能要创建一个接口指针，该功能的 HRESULT 指示接口指针现在是否存在。此外，必须检查接口指针以确定其是否为零。

⚠ 警告

对无效内存区域的方法访问导致系统崩溃

对于接口方法，内部检查不可行，因为相应的对象必须存在才能调用该方法。

即使在方法调用之外，也要确保相应的对象存在（接口指针不为 0），并且程序还没有返回错误代码。

这是一个安全的查询，以使用一个新创建的接口指针：

```
IF SUCCEEDED(hr) AND ipImage <> 0 THEN
    hr := ipImage.GetWidth(nWidth);
END_IF
```

集成到用户定义的功能块中

HRESULT可以被集成到用户定义的功能块中，如下所示：

```
METHOD MyMethod : HRESULT
VAR_INPUT
    hr : HRESULT
END_VAR

IF FAILED(hr) THEN
    MyMethod := hr; //skip the whole method
ELSE
    (*
        code
        code
        code
    *)
    MyMethod := S_OK; // or some error code if something went wrong
END_IF
```

这在样本自行编写的函数 [▶ 2712]中进行了更详细的解释。

提取 ADS 的返回代码

例如，如果需要在人机界面中显示纯粹的 ADS 返回代码，可以从HRESULT以数字和文本形式提取，如下所示：

```
PROGRAM MAIN
VAR
    hr              :   HRESULT;
    nReturnCode     :   DWORD;
    sReturnCode     :   STRING;
END_VAR

nReturnCode := DINT_TO_DWORD(hr) AND 16#FFF;
sReturnCode := DWORD_TO_HEXSTR(nReturnCode, 3, FALSE);
```

为此需要使用 Tc2_Utilities 库。

6.1.1.3 机器学习

在机器学习中，优化模型以解决一项任务。这个过程被称为培训。在培训模型的帮助下，可以自动进行复杂的数据分析。这意味着手动创建的复杂程序结构可以被取代。机器学习模型在 TwinCAT 3 实时环境中使用所提供的机器学习函数 [▶ 1335]创建、培训和执行。它可以解决的任务多种多样，大致可分为五个任务领域（分类、异常检测、回归分析、聚类分析和特征转换）。根据任务的不同，有不同的模型类型可供选择，每种类型都有一般或特定于任务的优缺点。

在机器学习中，有监督学习和无监督学习之分。监督学习需要额外的信息，如标注（如标记）数据集，以便模型可以学习到已知类别（分类）或所需目标变量的映射。然后，模型可以对未知数据的类别或目标变量做出预测。相比之下，无监督学习无需额外信息，完全基于特征本身。

所有分类和回归模型一般都可归入监督学习。相比之下，所有异常检测和聚类模型都被可归入无监督学习。这两个选项都可用于特征转换，线性判别分析法基于监督学习，主组件分析基于无监督学习。

以下对任务区域进行的模型类型分配有助于选择合适的模型。需要注意的是，此处具体展示的是 TwinCAT Vision 中机器学习的处理，并不具有普遍性。

模型类型	分类	异常检测	回归	聚类	特征转换
K 均值聚类++ (KMPP)		x		x	
K 最近邻算法 (KNN)	x	x	x		
Linde-Buzo-Gray (LBG)		x		x	
线性判别分析 (LDA)					x
普通贝叶斯分类器 (NBC)	x	x			
主成分分析 (PCA)					x
随机森林 (RTrees)	x		x		
简化 TopoART (STA)	x	x	x	x	
支持向量机 (SVM)	x	x	x		
支持向量机 — 随机梯度下降 (SVM-SGD)	x				

只有使用机器学习 [▶ 1335]组的 F_VN_Create 函数之一创建的模型才能使用。可通过功能块 FB_VN_WriteMlModel [▶ 1538] 和 FB_VN_ReadMlModel [▶ 1530] 将以这种方式创建和随后培训的模型保存或加载到硬盘上。不打算导入外部培训模型，如 ONNX 文件。

机器学习的基本程序：

- 分析问题，定义预期结果：
  - 除其他事项外，这还决定了应用范围和可使用的模型类型。
- 收集数据：
  - 培训、验证和测试数据
  - 数据必须包含适合解决问题的信息。
  - 数据必须涵盖以后可能出现的所有变量和特征。
  - 如果是监督学习，还必须对数据进行注释。
- 选择并提取合适的特征：
  - 哪些特征是独特的、特殊的，或可为差异化提供相关信息？
  - 应避免冗余特征。
  - 特征转换在此可能会有所帮助。

    - 最后，应进行特征归一化处理，以便使不同特征的值范围具有可比性，并满足模型的要求。
- 确定合适的模型和超参数（验证和优化）。
  （用确定的超参数在培训和验证数据上培训选定模型）。
- 使用之前未使用的独立数据进行最终测试。

数据必须以样本的形式提供。样本是由特征组成的向量。这些特征既可以是相关图像的有形变量（如颜色值）或轮廓值（如重心形式）等，也可以是抽象变量（如 PCA 的结果），它们以有意义的方式对图像进行统计描述，但用户无法直接解读。对于监督模型的培训，每个样本均需要额外的信息，如类别分配。多个样本的集合称为一个批次。

根据模型类型，可提供用于样本和/或批量培训和预测的函数。有些模型可以重新培训，以便在运行时期间进一步优化模型或使其适应新条件。

特征提取

特征提取涉及从图像数据中提取数字特征，是机器学习解决方案的重要组成部分。必须区分是否可以通过函数直接确定特征，是否需要结合函数，或者是否可以利用对象的逻辑特征（如孔数）进行区分。您可以在 特征提取功能 [▶ 126] 章节中找到一份示例清单。

特征归一化

对特征进行缩放以实现归一化主要是为了确保所有特征对模型学习过程的贡献相同。数值较大的特征会在距离计算中占主导地位，使基于距离的模型对这些特征更加敏感。为了避免这种情况并满足模型的要求，我们通常建议执行特征归一化。

函数 F_VN_GetFeatureScales [▶ 1391] 提供三种缩放选项（ETcVnFeatureScalingType [▶ 181]）。您还可以创建和使用自己的缩放比例。使用函数 F_VN_FeatureScaling [▶ 1379] 应用缩放。

> **相同特征提取和归一化**
>
> 需要注意的是，所有处理步骤都必须在训练集训练期间和测试集测试模型时进行。因此必须在两个位置执行函数，必要时还必须保存参数以供日后使用。

6.1.1.3.1 特征提取功能

该列表包含适合特征提取的函数。该列表仅为示例，并不详尽。

图像特征

- F_VN_ImageAverage [▶ 1178]
- F_VN_ImageAverageStdDev [▶ 1179]
- F_VN_ImageMoments [▶ 1185]
    - F_VN_HuMomentInvariants [▶ 1483]
- F_VN_CountNonZeroPixels [▶ 1172]
- F_VN_HistogramExp [▶ 1204]
- F_VN_ImageCenterOfMass [▶ 1181]
- F_VN_MaxPixelValue [▶ 1186]
- F_VN_MinPixelValue [▶ 1188]
- F_VN_RegionOrientation [▶ 1190]

轮廓特征

- F_VN_ContourArea [▶ 916]
- F_VN_ContourCircularity [▶ 918]
- F_VN_ContourRoundness [▶ 928]
- F_VN_ContourMoments [▶ 924]
    - F_VN_HuMomentInvariants [▶ 1483]
- F_VN_ContourConvexity [▶ 919]
- F_VN_ContourPerimeter [▶ 927]
- F_VN_ContourOrientation [▶ 925]

- F_VN_ContourCenterOfMass [▶ 917]
- F_VN_ContourEccentricity [▶ 920]
- F_VN_ContourElongation [▶ 921]
- F_VN_ContourExtremePoint [▶ 922]
- F_VN_ContourInertiaRatio [▶ 923]
- F_VN_ConvexHullPoints [▶ 929]
- F_VN_ConvexityDefects [▶ 931]
- F_VN_FourierDescriptors [▶ 940]
- F_VN_EnclosingCircle [▶ 934]
- F_VN_FitEllipse [▶ 937]
- F_VN_EnclosingRectangle [▶ 935]

图像纹理

- F_VN_Granulometry [▶ 1499]
- F_VN_HaralickFeatures [▶ 1501]

关键点

- F_VN_KeyPointsAGAST [▶ 1308]
- F_VN_KeyPointsAndDescriptorsAKAZE [▶ 1310]
- F_VN_KeyPointsAndDescriptorsORB [▶ 1315]
- F_VN_KeyPointsFAST [▶ 1317]
- F_VN_KeyPointsGFTT [▶ 1319]
- F_VN_KeyPointsSB [▶ 1323]

6.1.1.4 看门狗

由于许多图像处理算法的运行时取决于图像内容，因此在不利情况下，函数调用的时间会比平时长很多。

例如，光照条件的变化、图像中出现意外物体以及其他因素都可能导致这种情况，从而极大地影响 F_VN_FindContours() [▶ 1147] 等函数的计算时间。例如，如果在正常条件下只能找到 10 个轮廓，那么在其他照明条件下可能会有 100 个或更多。

这可能导致循环超限，必须不惜一切代价避免，因为这会导致未定义行为。

为此，（Watchdog（看门狗）很有帮助。Watchdog（看门狗）可以在一定时间后中止单个 Vision 函数或包含多个函数的整个代码段。某些 TwinCAT Vision 函数可以在执行过程中中止，并返回到那个时候为止计算出的部分结果。如果多个函数受 Watchdog（看门狗）监控，则在指定时间过后，Watchdog（看门狗）范围内的所有剩余函数都会被跳过。不过，随后的函数（Watchdog（看门狗）区域之后）会再次正常执行。

有两种函数可用于定义 Watchdog（看门狗）监控区域的起始，它们在指定的开始计数时间上有所不同：

- F_VN_StartAbsWatchdog [▶ 948] 定义相对于任务循环开始的绝对中止时间
- F_VN_StartRelWatchdog [▶ 949] 定义相对于 Watchdog（看门狗）区域开始的相对终止时间

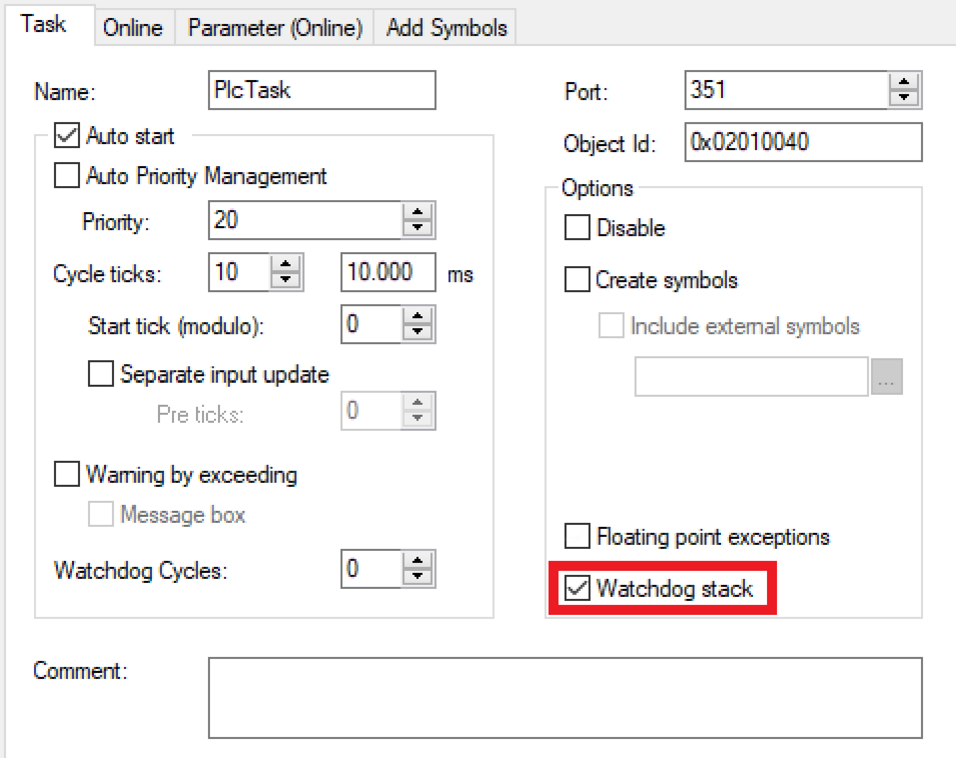
在这两种情况下，区域结束都由函数 F_VN_StopWatchdog [▶ 951] 定义，其可选择返回处理组件和所需时间。

注意

具有相对终止时间的 Watchdog（看门狗）

在使用 F_VN_StartRelWatchdog 时，必须注意中止时间不要设置过长，否则会继续出现循环过冲。因此，要观察相应任务的循环时间、调用函数之前已经耗费的时间以及 F_VN_StopWatchdog 之后到任务结束还需要的时间。

为了让 Watchdog（看门狗）监控函数，必须在相应的执行任务上启用“Watchdog stack（看门狗栈）”选项。



样本

在下方示例中，Watchdog（看门狗）被启动，相对于 Watchdog（看门狗）函数调用的停止时间为 10 ms。例如，如果循环时间为 20 ms，调用 F_VN_StartRelWatchdog [▶ 949] 时当前任务循环时间已过去 4 ms，而 Watchdog（看门狗）启动时的停止时间 tStop 相对于当前时间为 10 ms，则 Watchdog（看门狗）会在当前循环时间过去 14 ms 后停止受监控函数（即 Watchdog（看门狗）启动和停止之间调用的任何函数）。

```
VAR
    ipImage     : ITcVnImage := 0;
    ipContours : ITcVnContainer := 0;

    // watchdog runtime info
    nFunctionsMonitored  : ULINT;
    nFractionProcessed   : UDINT;
    tRest                : DINT;
END_VAR

(* imagine some other functions that use 4ms up to here *)

hr := F_VN_StartRelWatchdog(10000, hr); // 10ms
    hr := F_VN_Threshold(ipImage, ipImage, 120, 255, TCVN_TT_BINARY, hr);
    hr := F_VN_FindContours(ipImage, ipContours, hr);
hr := F_VN_StopWatchdog(nFunctionsMonitored => nFunctionsMonitored, nFractionProcessed => nFractionP
rocessed, tRest => tRest);
```

现在有两种可能情况：

- 两个函数均及时终止 — 这是正常情况。

假设 F_VN_Threshold [▶ 1287] 需要 1 ms，F_VN_FindContours [▶ 1147] 需要 5 ms。当调用 F_VN_StopWatchdog [▶ 951] 时，剩余 4 ms。Watchdog（看门狗）的运行时信息是：

```
nFunctionsMonitored = 2
nFractionProcessed = 100 // in %
tRest = 4000 // in us, equals 4ms
```

- Watchdog（看门狗）必须干预。

情景：函数 F_VN_Threshold [▶ 1287] 与图像内容无关（只与像素数有关），因此只需 1 ms，但照明条件发生了不利变化，因此 F_VN_FindContours 所需的时间将超过 9 ms。因此 Watchdog（看门狗）停止了 F_VN_FindContours [▶ 1147]，但还是返回了目前为止找到的轮廓。Watchdog（看门狗）的运行时信息如下所示：

```
nFunctionsMonitored = 2
nFractionProcessed = 70
tRest = -50
```

在这种情况下，当函数中止时，计算出的估计处理百分比为 70%。剩余时间为负值，即计划的停止时间超出了 50 us，因此在函数 F_VN_StopWatchdog [▶ 951] 之后，从任务循环时间开始已经过去了 14050 us，而不是计划的 14000 us。这种超限是由于要继续使用已经计算过的函数的部分结果。因此，一方面，算法只能在特定点中止，另一方面，必须整理并返回之前的结果。最大超限一般取决于具体函数和图像内容。因此，在程序中，应始终选择终止时间，以便在任务循环结束时保留安全缓冲。

监控函数

以下函数在 Watchdog（看门狗）终止时提供部分结果：

- F_VN_AdaptiveThreshold(Exp) [▶ 1282]
- F_VN_ConvertColorSpace [▶ 1197]
- F_VN_DetectBlobs(Exp) [▶ 1139]
- F_VN_FindContourHierarchyExp [▶ 1143]
- F_VN_FindContours(Exp) [▶ 1147]
- F_VN_HoughCircle(Exp) [▶ 1151]
- F_VN_LocateCircularArc(Exp) [▶ 1435]
- F_VN_LocateEdge(Exp) [▶ 1441]
- F_VN_LocateEdges(Exp) [▶ 1447]
- F_VN_LocateEllipse(Exp) [▶ 1453]
- F_VN_MatchTemplate(Exp) [▶ 1159]
- F_VN_MatchTemplateAndEvaluate(Exp) [▶ 1160]
- F_VN_MeasureAngleBetweenEdges(Exp) [▶ 1458]
- F_VN_MeasureEdgeDistance(Exp) [▶ 1463]
- F_VN_MeasureMinEdgeDistance(Exp) [▶ 1468]
- F_VN_NormalizeImageForDisplay [▶ 1209]
- F_VN_ReadDataMatrixCode(Exp) [▶ 803]
- F_VN_ReferenceColorSimilarity(Exp)_ITcVnColorModel (ITcVnMlModel) [▶ 1210]
- F_VN_ReferenceColorSimilarity(Exp)_TcVnVector3_LREAL [▶ 1212]
- F_VN_ResizeImage [▶ 1120]
- F_VN_Threshold [▶ 1287]
- F_VN_TrainImageColor(Exp(2)) [▶ 1218]
- F_VN_WarpAffine (Exp) [▶ 1125]
- F_VN_WarpPerspective (Exp) [▶ 1131]F_VN_WarpPerspective(Exp)F_VN_WarpPerspective(Exp)

其他函数在执行过程中无法中止，但如果 Watchdog（看门狗）时间已过，则会跳过。

示例

- 带看门狗监控的斑点检测 [▶ 2664]

6.1.1.5 图像处理

在以下各章中，将参照 TwinCAT Vision 对图像处理的一些概念进行一般和具体说明。它们将帮助你了解许多 API 功能 [▶ 408]的操作模式。

6.1.1.5.1 图像

在 TwinCAT Vision 中，图像由图像对象处理，这些对象可以实现各种接口，如ITcVnImage [▶ 390]。通过接口指针可以访问图像对象的实际图像数据，然后在 PLC 中用于处理。这些图像可以由各种像素类型和一些通道组成，并且可以有几乎任意的尺寸。由于处理来自实时和动态的大小，图像数据位于路由器内存中。相机图像在收到后直接存储在那里，而来自硬盘的图像必须首先通过文件源或 PLC 功能块加载，并以这种方式最终进入路由器内存。相机对象中的助手对路由器内存的图像不起作用。

查询一个视觉设备的图像接口

在许多情况下，图像通过相机或通过加载一个图像文件创建。以下功能块可用于此目的：

- FB_VN_SimpleCameraControl [▶ 1575]
  可以连接到 GigE Vision 相机实例或文件源实例。
- FB_VN_GevCameraControl [▶ 1553]
  连接到 GigE Vision 相机实例。
- FB_VN_FileSourceControl [▶ 1542]
  连接到文件源实例。

根据视觉设备的设置，可能需要启动过程或图像触发。此处仅显示 PLC（通过它可以访问内存中的图像）中接口指针的查询：

```
VAR
    hr: HRESULT;
    fbVisionDevice: FB_VN_SimpleCameraControl;
END_VAR

hr := fbVisionDevice.GetCurrentImage(ipImageIn);
IF SUCCEEDED(hr) AND ipImageIn <> 0 THEN
    // Process image…
END_IF
```

图像格式

最常用的两种图像格式是单色和 RGB。在相机中，传输的图像格式在PixelFormat选项卡下设置。为了减少数据量，从而能够实现更高的传输率，也有打包或过滤的图像格式，如 Bayer 格式。由于这些传输的图像格式不能使用，或者由于过滤器的原因与原始图像不一致，所以在收到后应直接转换这些格式。

- 如果是打包的图像，当调用GetCurrentImage()时转换将在内部进行，然后会直接得到解包的图像。
- 在所有其他情况下，就像使用 Bayer 格式一样，应该先用F_VN_ConvertColorSpace [▶ 1197]转换图像。之后，输入的图像不再需要，且可以丢弃。

这一点在使用插值图像信息的函数（如F_VN_CompensateLensDistortion [▶ 1051]、F_VN_WarpAffine [▶ 1125]、F_VN_RotateImageExp [▶ 1123]或诸如受过滤图像干扰的校准函数）时尤其需要注意。因此，一般建议首先将图像读入 PLC，在那里进行转换，然后将所有后续功能应用于原始图像。

创建图像

在有些情况下，需要的图像不是由相机拍摄且也不是从文件系统加载。例如，可以创建一个彩色图像来显示结果。通过指定图像尺寸、像素类型和通道数量，可以为此使用函数F_VN_CreateImage [▶ 755]。这里选择的像素类型TCVN_ET_USINT对应于 8 位/通道的常规图像。请注意，图像所需的内存只被分配，但没有被初始化。因此，可以通过函数F_VN_SetPixels [▶ 778]将图像设置为统一的颜色。

> **最大图像尺寸**
>
> 可以创建、接收或处理的最大图像尺寸限制在行或列的 2^20 像素数或图像的总尺寸 2^30 像素，以先到者为准。

```
PROGRAM MAIN
VAR
    ipImage      :   ITcVnImage;
    aColorBlack :   TcVnVector4_LREAL := [0, 0, 0];
    hr           :   HRESULT;
END_VAR

hr := F_VN_CreateImage(
    ipImage      :=  ipImage,
    nWidth       :=  640,
    nHeight      :=  480,
    ePixelType   :=  TCVN_ET_USINT,
```

```
    nChannelNum :=  3,
    hrPrev      :=  hr
);
hr := F_VN_SetPixels(ipImage, aColorBlack, hr);
```

或者，可以使用F_VN_CreateImageFromArray [▶ 758]函数从现有的图像数据中创建一个图像。

```
PROGRAM MAIN
VAR
    aImageData   :   ARRAY [0..479, 0..639] OF USINT;
    ipImage      :   ITcVnImage;
    hr           :   HRESULT;
END_VAR

hr := F_VN_CreateImageFromArray(
    pData        :=  ADR(aImageData),
    ipImage      :=  ipImage,
    nWidth       :=  640,
    nHeight      :=  480,
    ePixelType   :=  TCVN_ET_USINT,
    nChannelNum  :=  1,
    hrPrev       :=  hr
);
```

访问图像数据

函数F_VN_GetPixel [▶ 771]和F_VN_SetPixel [▶ 777]可用于访问图像的单个像素。这种方法很容易使用。然而，当重复应用于许多像素时，它的运行时间相对较长。在这种情况下，直接访问图像数据阵列比较合适。一个完整的像素行的指针被获取，且数组算子被用来访问各个像素：

```
PROGRAM MAIN
VAR
    ipImage      :   ITcVnImage;
    nHeight      :   UDINT;
    nWidth       :   UDINT;
    y            :   UDINT;
    x            :   UDINT;
    // Pointer type must match pixel type of ipImage!
    pRow         :   POINTER TO USINT;
    nPixelValue  :   USINT;
END_VAR

// Create ipImage here or set to an existing image

// Determine width and height of image
hr := F_VN_GetImageWidth(ipImage, nWidth, hr);
hr := F_VN_GetImageHeight(ipImage, nHeight, hr);

// Iterate every pixel of the image
FOR y := 0 TO nHeight - 1 DO
    IF hr = S_OK THEN
        hr := ipImage.GetRowPointer(y, ADR(pRow));
        IF hr = S_OK THEN
            FOR x := 0 TO nWidth - 1 DO
                // Access pixel via pRow[x]
                nPixelValue := pRow[x]; // e.g.
                pRow[x] := nPixelValue; // e.g.
            END_FOR
        END_IF
        // It’s important to release the pointer. Otherwise there will be memory leaks.
        hr := ipImage.ReleaseRowPointer(ADR(pRow));
    END_IF
END_FOR
```

可显示的图像

为了能够通过 ADS 传输和显示图像（例如在ADS 图像查看 [▶ 115]中），必须首先将其转换为类型为ITcVnDisplayableImage [▶ 390]的可显示图像。

严格区分显示和处理图像是出于以下原因：通过 ADS 的图像传输与 PLC 周期不同步。因此，图像处理功能可以在图像传输的同时进行。为了避免图像传输过程中的内存冲突，因此有必要将内存区域与要显示和处理的图像分开。

两个接口ITcVnImage [▶ 390]和ITcVnDisplayableImage [▶ 390]可防止意外目的的图像访问（无论是处理还是传输）。以下函数可用于转换为可显示图像（ITcVnDisplayableImage [▶ 390]）：

- F_VN_CopyIntoDisplayableImage [▶ 753]

- F_VN_TransformIntoDisplayableImage [▶ 784]
- F_VN_TransformIntoDisplayableImageExp [▶ 785]

F_VN_CopyIntoDisplayableImage [▶ 753]函数创建了图像数据的深度拷贝，并使其成为可显示的图像。如果待显示图像将用于下面的程序序列，务必使用功能：

```
// Processing, e.g.:
hr := F_VN_Threshold(ipImage, ipImage, 128, 255, TCVN_TT_BINARY, hr);

// Show intermediate processing step
hr := F_VN_CopyIntoDisplayableImage(ipImage, ipImageDisp, hr);

// Further processing, e.g.:
hr := F_VN_FindContours(ipImage, ipContours, hr);
```

函数F_VN_TransformIntoDisplayableImage [▶ 784]使用现有的图像数据并释放图像的原始接口指针。这意味着图像数据不能再被处理，且完全可用于图像传输。由于没有创建图像数据的副本，该功能的执行时间要短得多，而且没有使用额外的内存。然而，这个功能仅在没有其他指向图像数据的指针时才起作用。否则，将返回错误代码E_INCOMPATIBLE (16#70E)。如果Ads 通信器对象 [▶ 109]激活（例如数据流记录 [▶ 90]），就会出现这种情况，因为这将在短时间内继续在内部引用该图像。因此，这个功能主要对处理链的末端有用，以（例如）显示已经生成的结果图像。

```
// Put last results on image, e.g.:
hr := F_VN_DrawContours(ipContours, -1, ipImageRes, aColorRed, 5, hr);

// Finally transform result image into a displayable image
hr := F_VN_TransformIntoDisplayableImage(ipImageRes, ipImageResDisp, hr);
```

此外，函数F_VN_TransformIntoDisplayableImageExp [▶ 785]提供了按需创建原始图像深度拷贝的选项。因此，如果可能，会使用节省资源的转换变体，而必要时则使用复制变体。如果GigE Vision 相机 [▶ 63]实例的AdsCommunicator 对象 [▶ 108]被激活以将图像作为数据流保存到文件系统中，就会出现相应的用例。除了 PLC 中的指针外，AdsCommunicator 对象中还有一个图像数据指针。如果在使用该图像时 PLC 中尚未启用该功能，则必须复制该图像进行显示。

```
hr := fbCamera.GetCurrentImage(ipImageIn);
hr := F_VN_CopyImage(ipImageIn, ipImageWork, hr);

// Reliably show original image and try to reuse image data to avoid deep copy
hr := F_VN_TransformIntoDisplayableImageExp(
    ipSrcImage      :=  ipImageIn,
    ipDestImage     :=  ipImageInDisp,
    bAllowDeepCopy  :=  TRUE,
    hrPrev          :=  hr
);

// Manipulate ipImageWork during processing here ...
```

可显示图像的内存限制

在开发环境中显示的图像（ADS 图像查看或相机助手）与各自的 Visual Studio 实例位于同一 RAM 区域。如果现在希望查看非常多的图像或大尺寸的图像，由于每个实例的最大可用内存，可能会出现瓶颈。

由于图像是从 PLC 中检索出来的，以便在 ADS 图像查看（ADS 通信）中显示，而且这种交换通过路由器内存进行，因此在这一点上必须有足够的内存空间。

元信息

图像的元信息可以通过函数F_VN_GetImageInfo [▶ 769]获得。作为使用该函数的结果，会返回一个类型为TcVnImageInfo [▶ 210]的结构。其中包含的信息可以用于（例如）确保图像具有预期格式，以便进行转换：

```
hr := F_VN_GetImageInfo(ipImageIn, stImageInfo, hr);
IF stImageInfo.stPixelFormat.nChannels = 3 THEN
    hr := F_VN_ConvertColorSpace(ipImageIn, ipImageIn, TCVN_CST_RGB_TO_GRAY, hr);
END_IF
```

6.1.1.5.2 容器

图像处理的一个典型任务是寻找图像中的物体并确定其形状、位置和大小。所发现的物体可以通过其轮廓点等进行描述。轮廓点的数量取决于具体的对象，所以变化很大。因此，使用了动态尺寸的容器来存储这些轮廓。

然而，容器并不限于存储轮廓点，也可以包含各种其他数据类型。在一个容器内，不同的数据类型不能混合。下面解释了处理容器的一些基本概念。

创建容器

容器可以通过单个函数调用创建。仅须定义元素类型（见可用类型 [▶ 139]）和初始数量：

```
PROGRAM MAIN
VAR
    hr            : HRESULT;
    ipContainer   : ITcVnContainer;
END_VAR

hr := F_VN_CreateContainer(ipContainer, ContainerType_Vector_REAL, 10, hr);
```

在这个示例中，创建了包含 10 个*REAL*类型元素的新容器，内部组织形式 C++ 矢量。另外，函数 F_VN_CopyContainer [▶ 722]可用于创建现有容器的深度拷贝。不仅复制数据的指针，就像分配给另一个变量一样，而且数据被复制到一个新分配的内存区域。

所有可用的容器类型都存储为全局常量 [▶ 139]，并按照以下模式命名：所有名称都以“ContainerType_”开头。接下来是类型，由实际容器类型的描述（基于 C++ 容器）和元素类型（如：REAL、UDINT等等）组成。容器也可以反过来包含更多的容器作为元素。容器结构的每一层在名称中都由各自的容器类型表示，例如“Vector_”或“String_”。因此，带有REAL元素的容器的结果类型名称是ContainerType_Vector_REAL。

添加元素

由于容器能够动态地扩展分配的内存，因此可以在现有的容器中添加额外的元素。函数 F_VN_AppendToContainer [▶ 521]和F_VN_InsertIntoContainer [▶ 650]可用于此。

然而，由于内部可能需要分配一个全新的内存区域并复制现有的数据，因此建议事先保留所需的最大内存。因此，新的内存只分配一次，且程序的整体性能更高。为此存在一个函数，即F_VN_ReserveContainerMemory [▶ 742]，它只是保留内存，但不改变容器中元素的数量。

```
// Reserve memory for 100 elements
hr := F_VN_ReserveContainerMemory(ipContainer, 100, hr);

// Append the value 1.23 to the end of the container
hr := F_VN_AppendToContainer_REAL(1.23, ipContainer, hr);

// Insert the value 4.56 into the container at index 2 (3rd element)
hr := F_VN_InsertIntoContainer_REAL(4.56, ipContainer, 2, hr);
```

访问元素

如果仅需要访问容器的几个单独元素，建议使用 F_VN_GetAt_... 或 F_VN_SetAt_... 函数。由于在 PLC 中不可能有功能名过载，所以每个支持的容器类型都有一个单独的功能名。容器中的第一个元素的索引是 0。

```
PROGRAM MAIN
VAR
    hr            : HRESULT;
    ipContainer   : ITcVnContainer;
    fElement      : REAL;
END_VAR

hr := F_VN_GetAt_REAL(ipContainer, fElement, 0, hr); // Get first element
hr := F_VN_SetAt_REAL(fElement, ipContainer, 2, hr); // Set third element
```

使用这些现成功能的好处是，只需要一行代码，且所有的检查（例如，容器类型是否正确以及索引是否过大）都在内部进行。如果想要一个接一个地访问容器中的所有元素，特别是对于大型容器，建议通过迭代器和相关访问接口访问容器元素，以便获得更好的性能：

```
PROGRAM MAIN
VAR
    hr            :   HRESULT;
    ipContainer   :   ITcVnContainer;
    ipIterator    :   ITcVnForwardIterator;
    ipAccess      :   ITcVnAccess_REAL;
    fElement      :   REAL;
END_VAR

hr := F_VN_GetForwardIterator(ipContainer, ipIterator, hr);
IF SUCCEEDED(hr) AND ipIterator <> 0 THEN
    hr := ipIterator.TcQueryInterface(IID_ITcVnAccess_REAL, ADR(ipAccess));
    IF SUCCEEDED(hr) AND ipAccess <> 0 THEN
        WHILE SUCCEEDED(hr) AND ipIterator.CheckIfEnd() <> S_OK DO
            hr := ipAccess.Get(fElement);
            IF SUCCEEDED(hr) THEN
                fElement := fElement + 1;
```

```
                hr := ipAccess.Set(fElement);
            END_IF
            IF SUCCEEDED(hr) THEN
                hr := ipIterator.Increment();
            END_IF
        END_WHILE
    END_IF
    hr := FW_SafeRelease(ADR(ipAccess));
END_IF
hr := FW_SafeRelease(ADR(ipIterator));
```

如果容器中的元素也是容器，请使用以下程序代替。

容器的容器

例如，函数F_VN_FindContours [▸ 1147]不仅可以找到一个轮廓，而且可以找到多个轮廓，所以容器可以包含进一步的容器作为元素（在这种情况下，所有内部容器必须包含相同类型的元素）。原则上，这些可以按照上述简单容器完全相同的方式使用。为此，提供了函数F_VN_GetAt_ITcVnContainer [▸ 617]，用于从 PLC 获得任何可使用的内部容器的深度拷贝（从技术上讲，不可能获得原始数据的接口指针）。由于这一层的一般 ITcVnContainer 数据类型，所以无需区分内部容器中包含哪些基本元素类型。这简化了通过迭代器的访问，因为不再需要特殊的访问接口：

```
PROGRAM MAIN
VAR
    hr            : HRESULT;
    ipContainer   : ITcVnContainer;
    ipIterator    : ITcVnForwardIterator;
    ipElement     : ITcVnContainer;
END_VAR

hr := F_VN_GetForwardIterator(ipContainer, ipIterator, hr);
IF SUCCEEDED(hr) AND ipIterator <> 0 THEN
    WHILE SUCCEEDED(hr) AND ipIterator.CheckIfEnd() <> S_OK DO
        // ipElement gets a deep copy!
        hr := F_VN_GetContainer(ipIterator, ipElement, hr);
        IF SUCCEEDED(hr) AND ipElement <> 0 THEN
            // 1. extract information or manipulate ipElement
            // 2. write back changes if ipElement was manipulated
            hr := F_VN_SetContainer(ipIterator, ipElement, hr);
        END_IF
        hr := F_VN_IncrementIterator(ipIterator, hr);
    END_WHILE
    // more efficient to release outside loop, GetContainer handles it inside
    hr := FW_SafeRelease(ADR(ipElement));
END_IF
hr := FW_SafeRelease(ADR(ipIterator));
```

显示容器

目前，如果不采取进一步行动，容器内容不能在 Visual Studio 实时调试中显示。相反，数据可以被导出到一个数组中，并以这种方式查看。

```
VAR
    aArray       :   ARRAY [0..9] OF REAL;
    ipContainer  :   ITcVnContainer;
    nBufferSize  :   ULINT;
END_VAR

hr := F_VN_ExportContainerSize(ipContainer, nBufferSize, hr);
IF nBufferSize = SIZEOF(aArray) THEN
    hr := F_VN_ExportContainer(
        ipContainer :=ipContainer,
        pBuffer     :=ADR(aArray),
        nBufferSize :=nBufferSize,
        hr
    );
END_IF
```

覆写容器

一般来说，具有 API 功能的容器可以简单地覆盖。然而，目前，如果现有容器和待写入容器类型不同，则无法覆盖。这是由错误代码 70E（INCOMPATIBLE）表示。

6.1.1.5.3 轮廓

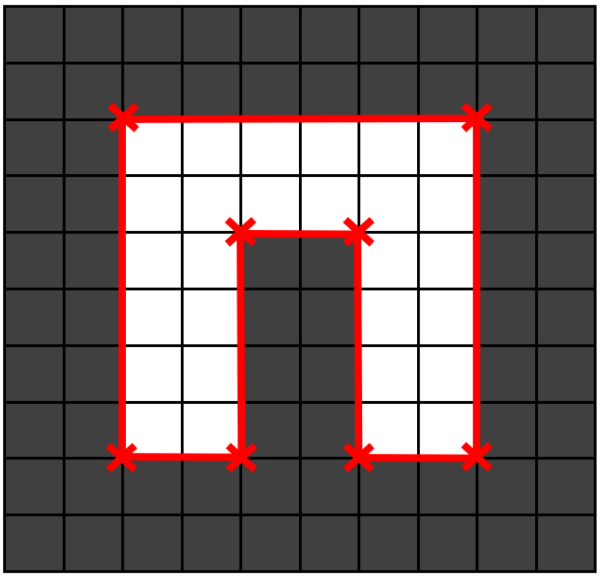
本章介绍了 TwinCAT Vision 中轮廓的使用。

什么是轮廓?

一般来说，轮廓（也叫轮廓线）是一个物体与周围环境的分界线；这在图像处理中是典型的二维处理。轮廓描述了这个物体的形状、大小和位置。在图像处理的范围内，这意味着图像 [▸ 130]中的一个物体；因此，轮廓描述了图像中物体的轮廓线。

轮廓如何显示?

轮廓显示为二维点的集合。如果没有设置 ROI，这些点大多指的是物体被映射的图像的原点。如果把这个领域的所有点都用直线连接起来，就会得到轮廓的图纸。



TwinCAT Vision 中如何显示轮廓?

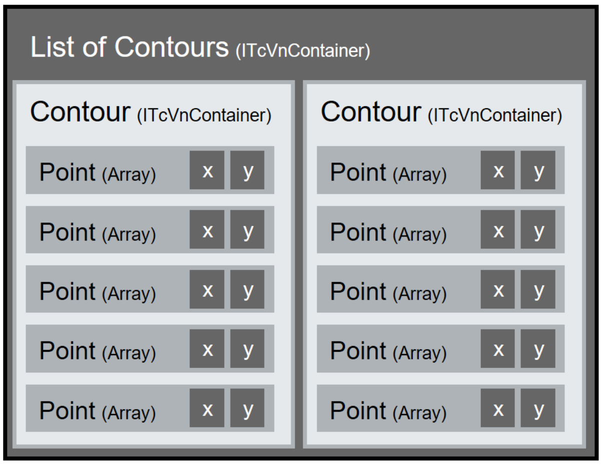
在 TwinCAT Vision 中，动态二维点阵列被处理为点（TcVnPoint2_DINT/REAL）的容器 [▸ 132]（ITcVnContainer）。然后，相应容器的类型名称为ContainerType_Vector_TcVnPoint2_DINT/REAL。

然后，这些点经常直接位于像素网格上（例如在轮廓搜索的情况下）。在这种情况下，DINT元素用于保存这些点。然而，也有一些情况下，精度不够，因此必须使用REAL（例如在测量的情况下）。这样，这些点就不完全位于一个像素的中心点，而是在其他地方。

绘图功能只能在图像中绘制具有整数元素的轮廓。因此，在有疑问的情况下，必须对容器的元素类型进行转换。

轮廓领域

经常出现的情况是需要同时处理几个轮廓（例如，通常通过F_VN_FindContours [▸ 1147]找到几个轮廓）。这在技术上通过使用二维容器器实现。因此，存在轮廓阵列或二维点阵列。然后，相应的类型名称是ContainerType_Vector_Vector_TcVnPoint2_DINT/REAL。



什么是轮廓近似?

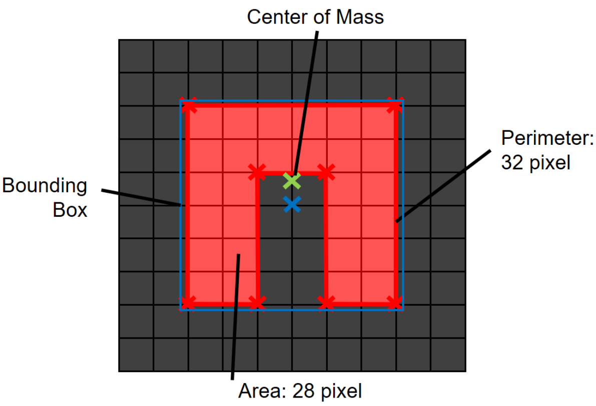
一般来说，在显示一个物体时，总是可以使用非常多的点。然而，这经常并不必要，因为（近似）直线可以通过其起点和终点进行近似。两者之间的点不需要另外保存，因为起点和终点足以定义。

根据所使用的方法和物体的几何形状，近似可能或多或少有所损失。在任何情况下，在不同的近似情况下，几何特征也可以用稍微不同的值来计算。

几何特征

在轮廓上，即从点的阵列中，可以计算出大量单独的特征，而这些特征从几何上描述各自的轮廓。轮廓和相应物体可以被检查并通过这些特征进行分类。这些特征包括：

- 范围
- 表面
- 中心点



封闭和开放的轮廓

轮廓通常封闭。在这种情况下，阵列的第一个和最后一个元素被简单地认为相连。然而，也可能存在开放的轮廓。因此，在技术表现上，开放和封闭轮廓没有区别，而区别在于数据如何解释的相关信息。在 API 的相关点上可相应地查询到各自的轮廓开放或封闭使用。

样本

- 对象检测 [▶ 2664]
- 轮廓分析 [▶ 2655]

6.1.1.5.4 掩码

本章介绍了 TwinCAT Vision 中**掩码**含义。

什么是掩码？

在图像处理方面，掩码是指属于另一个图像 [▸ 130]的掩码图像。掩膜图像具有相同的尺寸，并由一个灰度通道组成。掩膜图像的强度定义了相关图像的哪些像素在某些工作步骤中要被考虑在内。

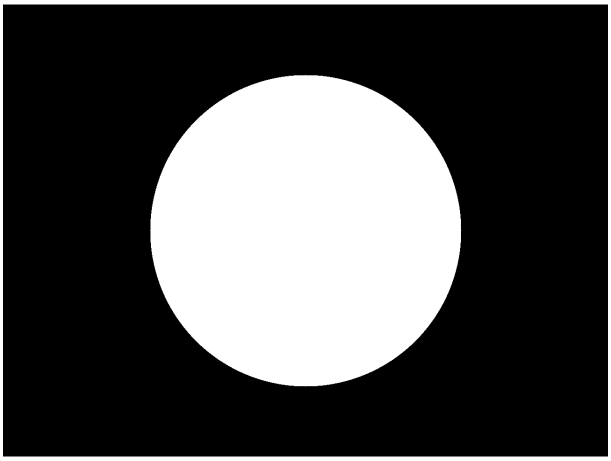
需要掩码做什么？

如果没有进一步的行动，图像将限制为矩形形状。如果在图像操作过程中需要定义矩形以外的形状，可以通过掩码的方式实现。掩码由另一个具有相同形状和大小的图像显示，上面需要考虑的区域（像素）通过不同方式标记出来。

从技术上讲，掩码如何显示？

掩码灰度图像的强度描述了哪些像素需要考虑。原则上，对于强度的解释可能因功能不同而有所偏差。从两方面看，所有数值为 0 的像素通常被解释为不被考虑，而所有不同数值（>=1）的像素被解释为需要考虑。然而，也可以想象到一些情况，其中考虑的程度可以设定得更细。

例如，一个将观察区域限制在图像中的一个圆形区域的掩码如下所示：



当应用该掩码时，相关图像角落里的像素对后续操作没有进一步影响。

样本

举例来说，为图像ipImageWork创建一个掩码，也就是说只考虑图像的一个圆形区域：

```
hr := F_VN_GetImageWidth(ipImageWork, nWidth, hr);
hr := F_VN_GetImageHeight(ipImageWork, nHeight, hr);

hr := F_VN_CreateImage(ipImageMask, nWidth, nHeight, TCVN_ET_USINT, 1, hr);
hr := F_VN_SetPixels(ipImageMask, aColorBlack, hr);
hr := F_VN_DrawCircle(nWidth/2, nHeight/2, MIN(nWidth, nHeight)/2, ipImageMask, aColorWhite, -1,
hr);
```

随后计算出掩码区域内图像的平均强度值：

```
hr := F_VN_ImageAverageExp(ipImageWork, aAverage, ipImageMask, hr);
fAverageIntensityInCircle := aAverage[0];
```

如果没有掩码，就不可能在不采取进一步行动的情况下将平均值限制在圆形区域而不是完整的矩形图像上。

关于完整的样本，可查看：任何类型的形状的平均强度 [▸ 2679]。

6.1.1.5.5 感兴趣区域

本章介绍如何在 TwinCAT Vision 中使用**感兴趣区域**。

什么是感兴趣区域？

感兴趣区域 (ROI) 是一个图像暂时被限制在其中的矩形图像区域。图像本身并没有改变，但需要处理的部分尺寸缩小。可以随时将 ROI 重置为完整图像。

为何需要感兴趣区域？

现有图像往往大于需要处理的区域。在这种情况下，应设定 ROI，以获得以下优势：

1. 更好的性能：如果需要处理的图像区域变小，则图像处理时间更短。
2. 减少干扰：ROI 外的干扰图像对象会被排除在处理过程之外，因此不会影响结果。

通过在图像中的不同点逐个设置 ROI，同样可以通过使用 ROI 分别处理多个图像区域。

注意

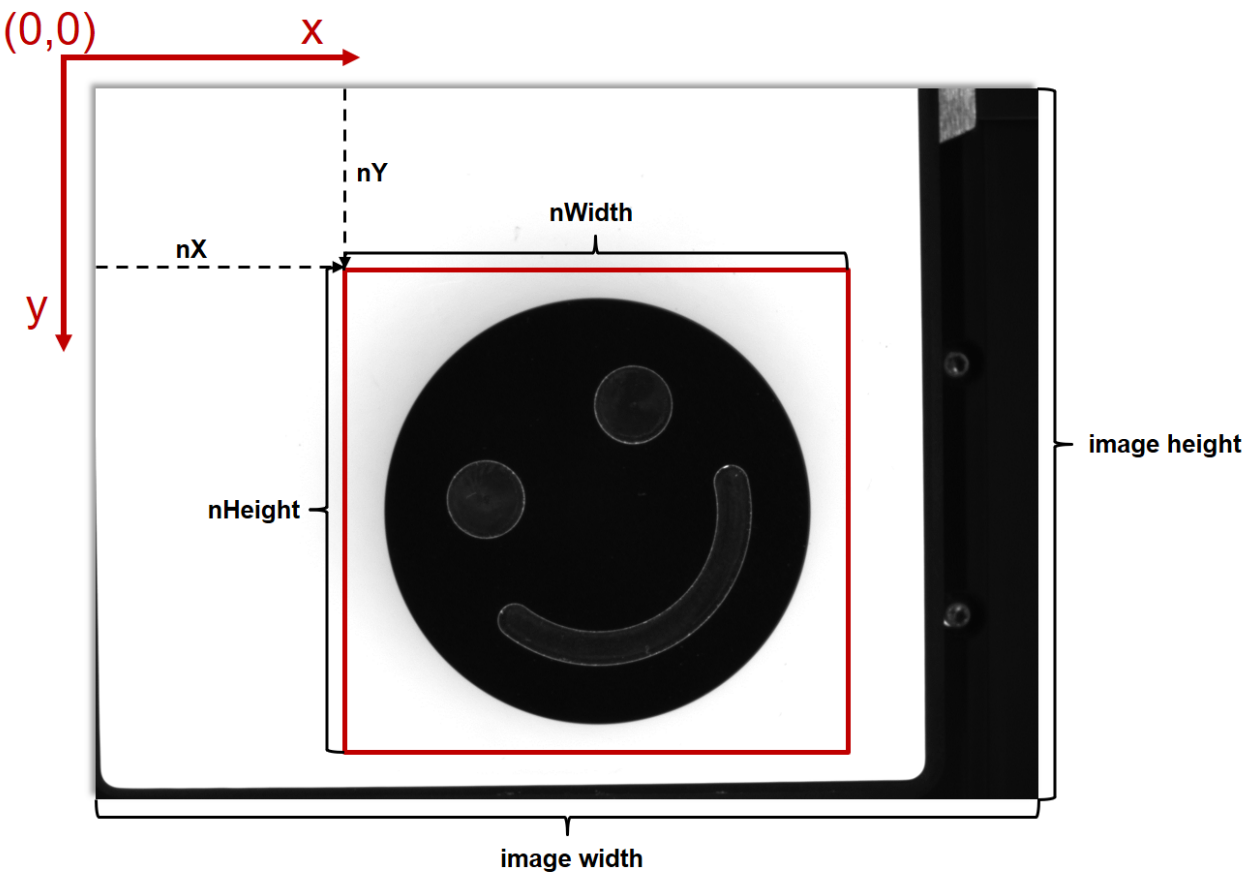
在相机中进行 ROI 配置

如果使用相机作为图像源，最好在其中执行 ROI，以减少需要传输的数据量，并使其尽可能低。根据设置和传感器的不同，可能帧率也会因此提高。请参见相机配置示例：感兴趣区域（ROI） [▶ 2726]

技术上如何显示感兴趣区域？

在 TwinCAT Vision 中，一组 ROI 的相关信息直接存储在相应的图像对象中。ROI 的设置仅包括矩形描述的设置。该区域之外的图像数据仍然存在，但在调用函数时会被忽略。

矩形描述由左上角的位置（nX 和nY）和矩形的大小（nWidth 和 nHeight）组成。确保矩形不超出图片边缘（nWidth ≤ ImageWidth - nX 和 nHeight ≤ ImageHeight - nY）。



要设置 ROI，可以直接传输这些参数，也可以作为 TcVnRectangle_UDINT [▶ 225] 类型的结构进行传输，具体取决于函数：

- F_VN_SetRoi [▶ 780]
- F_VN_SetRoi_TcVnRectangle_UDINT [▶ 782]

函数 F_VN_GetRoi [▶ 773] 可用于检查图像上是否设置了 ROI 以及设置了哪个 ROI。也可以在已经有 ROI 的图像上设置 ROI。如果要将 ROI 放大或重置为完整图像大小，可以使用函数 F_VN_ResetRoi [▶ 775]。

6.1.2 Data Types

共有四种数据类型，每种类型的前缀都是“TcVn”：

- 某些数组类型的别名 [▶ 139]涉及图像处理
- 具有特定数据类型组合和定义数的数组 [▶ 141]经常被图像处理函数使用。
- 枚举 [▶ 141]有助于函数的参数设置，并包含附加前缀“E”
- 结构 [▶ 204]整合多个相关信息片段（如几何描述）

此外，还提供用于描述容器类型的常量 [▶ 139]。

6.1.2.1 Aliases

Name	Type
GVCP_REGISTER_ADDRESS	UDINT
TcVnPoint	TcVnPoint2_DINT [▶ 139]
TcVnPoint2_DINT	TcVnVector2_DINT [▶ 141]
TcVnPoint2_LREAL	TcVnVector2_LREAL [▶ 141]
TcVnPoint2_REAL	TcVnVector2_REAL [▶ 141]
TcVnPoint3_LREAL	TcVnVector3_LREAL [▶ 141]
TcVnPoint3_REAL	TcVnVector3_REAL [▶ 141]
TcVnRectangle	TcVnRectangle_DINT [▶ 224]

6.1.2.2 Constants

以下常量代表了容器 [▶ 132]类型为GUID。对于F_VN_CreateContainer [▶ 725]等函数，需要这些常量。

Name	Type
ContainerType_Vector_SINT	GUID
ContainerType_Vector_USINT	GUID
ContainerType_Vector_INT	GUID
ContainerType_Vector_UINT	GUID
ContainerType_Vector_DINT	GUID
ContainerType_Vector_UDINT	GUID
ContainerType_Vector_ULINT	GUID
ContainerType_Vector_REAL	GUID
ContainerType_Vector_LREAL	GUID
ContainerType_Vector_ITcVnImage	GUID
ContainerType_Vector_TcVnPoint2_DINT	GUID
ContainerType_Vector_TcVnPoint2_REAL	GUID
ContainerType_Vector_TcVnPoint2_LREAL	GUID
ContainerType_Vector_TcVnPoint3_REAL	GUID
ContainerType_Vector_TcVnPoint3_LREAL	GUID
ContainerType_Vector_Vector_TcVnPoint2_DINT	GUID
ContainerType_Vector_Vector_TcVnPoint2_REAL	GUID
ContainerType_Vector_Vector_TcVnPoint3_REAL	GUID
ContainerType_Vector_Vector_UDINT	GUID
ContainerType_Vector_Vector_ULINT	GUID
ContainerType_Vector_Vector_REAL	GUID
ContainerType_Vector_Vector_LREAL	GUID
ContainerType_Vector_TcVnVector2_SINT	GUID
ContainerType_Vector_TcVnVector2_USINT	GUID
ContainerType_Vector_TcVnVector2_INT	GUID
ContainerType_Vector_TcVnVector2_UINT	GUID
ContainerType_Vector_TcVnVector2_DINT	GUID
ContainerType_Vector_TcVnVector2_REAL	GUID
ContainerType_Vector_TcVnVector3_SINT	GUID
ContainerType_Vector_TcVnVector3_USINT	GUID
ContainerType_Vector_TcVnVector3_INT	GUID
ContainerType_Vector_TcVnVector3_UINT	GUID
ContainerType_Vector_TcVnVector3_REAL	GUID
ContainerType_Vector_TcVnVector4_SINT	GUID
ContainerType_Vector_TcVnVector4_USINT	GUID
ContainerType_Vector_TcVnVector4_INT	GUID
ContainerType_Vector_TcVnVector4_UINT	GUID
ContainerType_Vector_TcVnVector4_DINT	GUID
ContainerType_Vector_TcVnVector4_LREAL	GUID
ContainerType_Vector_TcVnRectangle_DINT	GUID
ContainerType_Vector_TcVnKeyPoint	GUID
ContainerType_Vector_TcVnDMatch	GUID
ContainerType_Vector_Vector_TcVnDMatch	GUID
ContainerType_String_SINT	GUID
ContainerType_Vector_String_SINT	GUID

6.1.2.3 Arrays

Name	Type
TcVnArray10_ITcVnImage	array [0..9] of PVOID
TcVnArray3_Point2_REAL	array [0..2] of TcVnPoint2_REAL [▶ 139]
TcVnArray33_UDINT	array [0..32] of UDINT
TcVnArray4_LREAL	array [0..3] of LREAL
TcVnArray4_Point2_REAL	array [0..3] of TcVnPoint2_REAL [▶ 139]
TcVnArray7_LREAL	array [0..6] of LREAL
TcVnArray8_LREAL	array [0..7] of LREAL
TcVnMatrix2x3_LREAL	array [0..1, 0..2] of LREAL
TcVnMatrix3x3_LREAL	array [0..2, 0..2] of LREAL
TcVnVector2_DINT	array [0..1] of DINT
TcVnVector2_INT	array [0..1] of INT
TcVnVector2_LREAL	array [0..1] of LREAL
TcVnVector2_REAL	array [0..1] of REAL
TcVnVector2_SINT	array [0..1] of SINT
TcVnVector2_UINT	array [0..1] of UINT
TcVnVector2_USINT	array [0..1] of USINT
TcVnVector3_INT	array [0..2] of INT
TcVnVector3_LREAL	array [0..2] of LREAL
TcVnVector3_REAL	array [0..2] of REAL
TcVnVector3_SINT	array [0..2] of SINT
TcVnVector3_UINT	array [0..2] of UINT
TcVnVector3_USINT	array [0..2] of USINT
TcVnVector4_DINT	array [0..3] of DINT
TcVnVector4_INT	array [0..3] of INT
TcVnVector4_LREAL	array [0..3] of LREAL
TcVnVector4_SINT	array [0..3] of SINT
TcVnVector4_UINT	array [0..3] of UINT
TcVnVector4_USINT	array [0..3] of USINT

6.1.2.4 Enums

6.1.2.4.1 ETcVn2dCodeSearchStrategy

Offers search strategies for 2d code reading (multiple TCVN_CSS_XXX_INVERTED or multiple TCVN_CSS_XXX_FLIPPED cannot be combined).

Syntax

Definition:

```
TYPE ETcVn2dCodeSearchStrategy :
(
    TCVN_CSS_DEFAULT              := 1,
    TCVN_CSS_ONLY_NOT_INVERTED    := 8,
    TCVN_CSS_FIRST_NOT_INVERTED   := 10,
    TCVN_CSS_ONLY_INVERTED        := 12,
    TCVN_CSS_FIRST_INVERTED       := 14,
    TCVN_CSS_ONLY_NOT_FLIPPED     := 64,
    TCVN_CSS_FIRST_NOT_FLIPPED    := 80,
    TCVN_CSS_ONLY_FLIPPED         := 96,
    TCVN_CSS_FIRST_FLIPPED        := 112
)UDINT;
END_TYPE
```

Values

Name	Description
TCVN_CSS_DEFAULT	The algorithm searches for codes using the default searching strategy, all other flags are ignored (dependant on code type).
TCVN_CSS_ONLY_NOT_INVERTED	The algorithm searches for codes only in the non-inverted image.
TCVN_CSS_FIRST_NOT_INVERTED	The algorithm first searches for codes in the non-inverted and then in the inverted image.
TCVN_CSS_ONLY_INVERTED	The algorithm searches for codes only in the inverted image.
TCVN_CSS_FIRST_INVERTED	The algorithm first searches for codes in the inverted and then in the non-inverted image.
TCVN_CSS_ONLY_NOT_FLIPPED	The algorithm searches for codes only in the non-mirrored image.
TCVN_CSS_FIRST_NOT_FLIPPED	The algorithm first searches for codes in the non-mirrored and then in the mirrored image.
TCVN_CSS_ONLY_FLIPPED	The algorithm searches for codes only in the mirrored image.
TCVN_CSS_FIRST_FLIPPED	The algorithm first searches for codes in the mirrored and then in the non-mirrored image.

更多信息

该枚举为读码函数的搜索策略提供了设置选项。关于更多信息，请参见代码读取搜索策略 [▶ 797]。

相关函数

- F_VN_ReadDataMatrixCodeExp [▶ 805]
- F_VN_ReadQRCodeExp [▶ 815]

6.1.2.4.2 ETcVnAdaptiveThresholdMethod

Offers methods for adaptive threshold.

Syntax

Definition:

```
TYPE ETcVnAdaptiveThresholdMethod :
(
    TCVN_ATM_MEAN      := 0,
    TCVN_ATM_GAUSSIAN  := 1
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_ATM_MEAN	The threshold value is calculated as the mean of the nBlockSize x nBlockSize neighborhood of (x, y) minus fConstant.
TCVN_ATM_GAUSSIAN	The threshold value is the weighted sum (cross-correlation with a Gaussian window) of the nBlockSize x nBlockSize neighborhood of (x, y) minus fConstant.

相关函数

- F_VN_AdaptiveThresholdExp [▶ 1283]

6.1.2.4.3 ETcVnBarcodeSearchDirection

Offers search directions for linear barcodes.

Syntax

Definition:

```
TYPE ETcVnBarcodeSearchDirection :
(
    TCVN_BSD_ANY        := 0,
    TCVN_BSD_HORIZONTAL := 1,
    TCVN_BSD_VERTICAL   := 2
)UDINT;
END_TYPE
```

Values

Name	Description
TCVN_BSD_ANY	The algorithm searches for codes in any supported directions.
TCVN_BSD_HORIZONTAL	The algorithm searches for codes in horizontal direction.
TCVN_BSD_VERTICAL	The algorithm searches for codes in vertical direction.

相关函数

- F_VN_ReadBarcodeExp [▶ 799]

6.1.2.4.4 ETcVnBarcodeType

Offers linear barcode types to search for.

Syntax

Definition:

```
TYPE ETcVnBarcodeType :
(
    TCVN_BT_CODABAR         := 4,
    TCVN_BT_CODE39          := 8,
    TCVN_BT_CODE93          := 16,
    TCVN_BT_CODE128         := 32,
    TCVN_BT_EAN8            := 128,
    TCVN_BT_EAN13           := 256,
    TCVN_BT_ITF             := 512,
    TCVN_BT_UPCA            := 32768,
    TCVN_BT_UPCE            := 65536,
    TCVN_BT_ANY             := 99260,
    TCVN_BT_CODE39EXTENDED := 262144
)UDINT;
END_TYPE
```

Values

Name	Description
TCVN_BT_CODABAR	The algorithm searches for Codabar codes.
TCVN_BT_CODE39	The algorithm searches for Code-39 codes.
TCVN_BT_CODE93	The algorithm searches for Code-93 codes.
TCVN_BT_CODE128	The algorithm searches for Code-128 codes.
TCVN_BT_EAN8	The algorithm searches for EAN-8 codes.
TCVN_BT_EAN13	The algorithm searches for EAN-13 codes.
TCVN_BT_ITF	The algorithm searches for ITF codes.
TCVN_BT_UPCA	The algorithm searches for UPC-A codes.
TCVN_BT_UPCE	The algorithm searches for UPC-E codes.
TCVN_BT_ANY	The algorithm searches for any supported linear barcode. If the type of the code is known, it is recommended to select the specific type directly.
TCVN_BT_CODE39EXTENDED	The algorithm searches for Code-93-Extended codes.

相关函数

- F_VN_ReadBarcode(Exp) [▸ 797]

6.1.2.4.5 ETcVnBlobCombination

Offers multiple blob contours to choose from, which should be returned in a multi-threshold scenario (used in TcVnParamsBlobDetection, which in return is used in F_VN_DetectBlobs).

Syntax

Definition:

```
TYPE ETcVnBlobCombination :
(
    TCVN_BC_SMALLEST         := 0,
    TCVN_BC_LARGEST          := 1,
    TCVN_BC_MIN_THRESHOLD    := 2,
    TCVN_BC_MAX_THRESHOLD    := 3,
    TCVN_BC_MEDIAN_THRESHOLD := 4
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_BC_SMALLEST	Returns the smallest blob contour.
TCVN_BC_LARGEST	Returns the largest blob contour.
TCVN_BC_MIN_THRESHOLD	Returns the blob contour for the minimum threshold.
TCVN_BC_MAX_THRESHOLD	Returns the blob contour for the maximum threshold.
TCVN_BC_MEDIAN_THRESHOLD	Returns the blob contour for the median threshold.

更多信息

枚举ETcVnBlobCombination用于结构TcVnParamsBlobDetection [▸ 213]。

相关函数

- F_VN_CalibrateCamera(Exp) [▸ 1032]

- F_VN_DetectBlobs(Exp) [▶ 1139]

6.1.2.4.6 ETcVnBoostClassifierType

Offers different types of Boost classifiers.

Syntax

Definition:

```
TYPE ETcVnBoostClassifierType :
(
    TCVN_BCT_DISCRETE := 0,
    TCVN_BCT_REAL     := 1,
    TCVN_BCT_LOGIT    := 2,
    TCVN_BCT_GENTLE   := 3
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_BCT_DISCRETE	Discrete AdaBoost
TCVN_BCT_REAL	Real AdaBoost
TCVN_BCT_LOGIT	LogitBoost
TCVN_BCT_GENTLE	Gentle AdaBoost

6.1.2.4.7 ETcVnBorderInterpolationMethod

Offers methods to extrapolate values of non-existing pixels. On the one hand, this is used for filtering functions to enable filtering at the image borders (where the filter mask reaches over the border). On the other hand, this is used to extrapolate the undefined pixels after a geometric image transformation.

Syntax

Definition:

```
TYPE ETcVnBorderInterpolationMethod :
(
    TCVN_BIM_CONSTANT             := 0,
    TCVN_BIM_REPLICATE            := 1,
    TCVN_BIM_REFLECT              := 2,
    TCVN_BIM_WRAP                 := 3,
    TCVN_BIM_REFLECT_101          := 4,
    TCVN_BIM_DEFAULT              := 4,
    TCVN_BIM_TRANSPARENT          := 5,
    TCVN_BIM_ISOLATED_CONSTANT    := 16,
    TCVN_BIM_ISOLATED_REPLICATE   := 17,
    TCVN_BIM_ISOLATED_REFLECT     := 18,
    TCVN_BIM_ISOLATED_WRAP        := 19,
    TCVN_BIM_ISOLATED_REFLECT_101 := 20
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_BIM_CONSTANT	iiiiiii | abcdefgh | iiiiiii (with some specified i)
TCVN_BIM_REPLICATE	aaaaaaa | abcdefgh | hhhhhhh
TCVN_BIM_REFLECT	gfedcba | abcdefgh | hgfedcb
TCVN_BIM_WRAP	bcdefgh | abcdefgh | abcdefg
TCVN_BIM_REFLECT_101	hgfedcb | abcdefgh | gfedcba
TCVN_BIM_DEFAULT	Choose this if you don't know which method to select (same as REFLECT_101).
TCVN_BIM_TRANSPARENT	Corresponding pixels in the destination image will not be modified. Only available for geometric image transformations.
TCVN_BIM_ISOLATED_CONSTANT	Similar to CONSTANT, but ROIs are handled as isolated, which means surrounding image pixels outside the ROI are ignored.
TCVN_BIM_ISOLATED_REPLICATE	Similar to REPLICATE, but ROIs are handled as isolated, which means surrounding image pixels outside the ROI are ignored.
TCVN_BIM_ISOLATED_REFLECT	Similar to REFLECT, but ROIs are handled as isolated, which means surrounding image pixels outside the ROI are ignored.
TCVN_BIM_ISOLATED_WRAP	Similar to WRAP, but ROIs are handled as isolated, which means surrounding image pixels outside the ROI are ignored.
TCVN_BIM_ISOLATED_REFLECT_101	Similar to REFLECT_101, but ROIs are handled as isolated, which means surrounding image pixels outside the ROI are ignored.

更多信息

边界插值用于模拟图像外的像素值。例如，在对图像应用过滤器或形态学算子时，这十分必要。例如，如果一个过滤器在图像上进行迭代，必须指定该过滤器在图像边缘的表现，即过滤器内核的部分位于图像之外。为此，图像外的不存在的像素被模拟出来，以便仍然可以进行过滤操作。进行这种模拟的方法以枚举ETcVnBorderInterpolationMethod定义。数值TCVN_BIM_REFLECT通常是一个很好的选择；它图像边界以为镜像平面反映了外部图像像素。

相关函数

- F_VN_BilateralFilterExp [▸ 1235]
- F_VN_CustomFilterExp [▸ 1245]
- F_VN_GaussianFilterExp [▸ 1253]
- F_VN_LaplacianFilterExp [▸ 1256]
- F_VN_PadImageBorderExp [▸ 1029]
- F_VN_ScharrFilterExp [▸ 1270]
- F_VN_SeparableCustomFilterExp [▸ 1274]
- F_VN_SobelFilterExp [▸ 1278]
- F_VN_WarpAffineExp [▸ 1129]
- F_VN_WarpPerspectiveExp [▸ 1135]

6.1.2.4.8 ETcVnCalibrationPattern

Offers common calibration pattern types.

Syntax

Definition:

```
TYPE ETcVnCalibrationPattern :
(
    TCVN_CP_CIRCLES_SYM         := 0,
    TCVN_CP_CIRCLES_ASYM        := 1,
    TCVN_CP_CIRCLES_ASYM_INDENT := 2
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_CP_CIRCLES_SYM	Symmetric circle pattern
TCVN_CP_CIRCLES_ASYM	Asymmetric circle pattern, i.e. every 2nd row is indented, starting unindented
TCVN_CP_CIRCLES_ASYM_INDENT	Asymmetric circle pattern, i.e. every 2nd row is indented, starting indented

6.1.2.4.9 ETcVnCalibrationPatternOrigin

Offers calibration pattern origins.

Syntax

Definition:

```
TYPE ETcVnCalibrationPatternOrigin :
(
    TCVN_CPO_TOPLEFT      := 0,
    TCVN_CPO_TOPCENTER    := 1,
    TCVN_CPO_TOPRIGHT     := 2,
    TCVN_CPO_CENTERLEFT   := 3,
    TCVN_CPO_CENTER       := 4,
    TCVN_CPO_CENTERRIGHT  := 5,
    TCVN_CPO_BOTTOMLEFT   := 6,
    TCVN_CPO_BOTTOMCENTER := 7,
    TCVN_CPO_BOTTOMRIGHT  := 8
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_CPO_TOPLEFT	Origin is at top left
TCVN_CPO_TOPCENTER	Origin is at top center
TCVN_CPO_TOPRIGHT	Origin is at top right
TCVN_CPO_CENTERLEFT	Origin is at center left
TCVN_CPO_CENTER	Origin is at center
TCVN_CPO_CENTERRIGHT	Origin is at center right
TCVN_CPO_BOTTOMLEFT	Origin is at bottom left
TCVN_CPO_BOTTOMCENTER	Origin is at bottom center
TCVN_CPO_BOTTOMRIGHT	Origin is at bottom right

6.1.2.4.10 ETcVnCameraState

Specifies the state of a camera controller, which controls an attached camera.

Syntax

Definition:

```
TYPE ETcVnCameraState :
(
    TCVN_CS_ERROR                := -1,
    TCVN_CS_INITIAL              := 0,
    TCVN_CS_INITIALIZING         := 1,
    TCVN_CS_INITIALIZED          := 2,
    TCVN_CS_OPENING              := 3,
    TCVN_CS_OPENED               := 4,
    TCVN_CS_STARTACQUISITION     := 5,
    TCVN_CS_ACQUIRING            := 6,
    TCVN_CS_STOPACQUISITION      := 7,
    TCVN_CS_RESETTINGFEATURES := 8,
    TCVN_CS_TRIGGERING           := 9,
    TCVN_CS_CLOSING              := 10
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_CS_ERROR	The camera controller is in an error state (e.g. the connection to the camera was lost).
TCVN_CS_INITIAL	The camera controller is in the initial state and ready to establish the connection to the camera.
TCVN_CS_INITIALIZING	The camera controller is initializing the camera.
TCVN_CS_INITIALIZED	The camera controller has initialized the camera.
TCVN_CS_OPENING	The camera controller is establishing the connection to the camera.
TCVN_CS_OPENED	The connection to the camera has been established and the camera controller is ready to start the image acquisition.
TCVN_CS_STARTACQUISITION	The camera controller is starting the image acquisition.
TCVN_CS_ACQUIRING	The camera is sending images (either streaming or manually triggered, depending on the configuration).
TCVN_CS_STOPACQUISITION	The camera controller is stopping the image acquisition.
TCVN_CS_RESETTINGFEATURES	The camera controller is resetting the features on the camera.
TCVN_CS_TRIGGERING	The camera controller is processing a softwaretrigger.
TCVN_CS_CLOSING	The camera controller is closing the connection to the camera.

更多信息

用于这些相机状态的状态机在各自的图像采集功能块 [▸ 1540]中描述：

- FB_VN_GevCameraControl [▸ 1550]
- FB_VN_FileSourceControl [▸ 1541]
- FB_VN_SimpleCameraControl [▸ 1575]

6.1.2.4.11 ETcVnClusteringAlgorithm

Offers clustering algorithms

Syntax

Definition:

```
TYPE ETcVnClusteringAlgorithm :
(
    TCVN_CA_KMEANSPP := 0,
    TCVN_CA_LBG      := 1
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_CA_KMEANSPP	KMeans++ (fixed number of clusters)
TCVN_CA_LBG	LBG variant (dynamic number of clusters)

6.1.2.4.12 ETcVnColorMap

Offers color maps (similar to GNU Octave/MATLAB types).

Syntax

Definition:

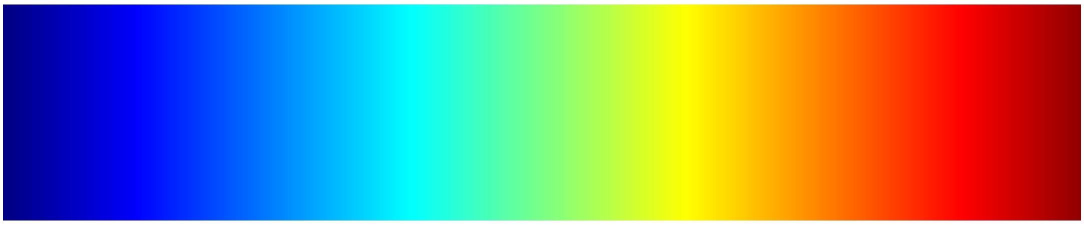
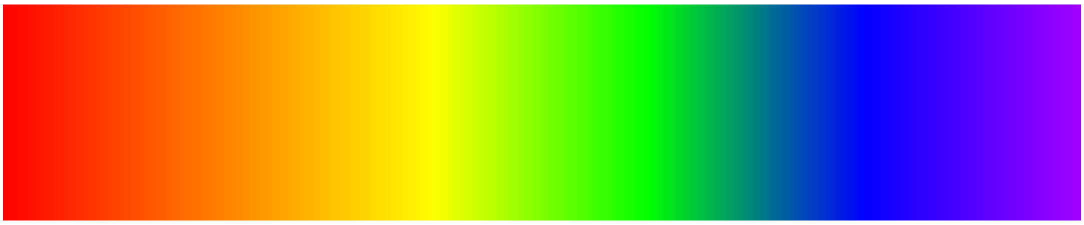
```
TYPE ETcVnColorMap :
(
    TCVN_CM_AUTUMN          := 0,
    TCVN_CM_BONE            := 1,
    TCVN_CM_JET             := 2,
    TCVN_CM_WINTER          := 3,
    TCVN_CM_RAINBOW         := 4,
    TCVN_CM_OCEAN           := 5,
    TCVN_CM_SUMMER          := 6,
    TCVN_CM_SPRING          := 7,
    TCVN_CM_COOL            := 8,
    TCVN_CM_HSV             := 9,
    TCVN_CM_PINK            := 10,
    TCVN_CM_HOT             := 11,
    TCVN_CM_PARULA          := 12,
    TCVN_CM_MAGMA           := 13,
    TCVN_CM_INFERNO         := 14,
    TCVN_CM_PLASMA          := 15,
    TCVN_CM_VIRIDIS         := 16,
    TCVN_CM_CIVIDIS         := 17,
    TCVN_CM_TWILIGHT        := 18,
    TCVN_CM_TWILIGHT_SHIFTED := 19,
    TCVN_CM_TURBO           := 20
)DINT;
END_TYPE
```

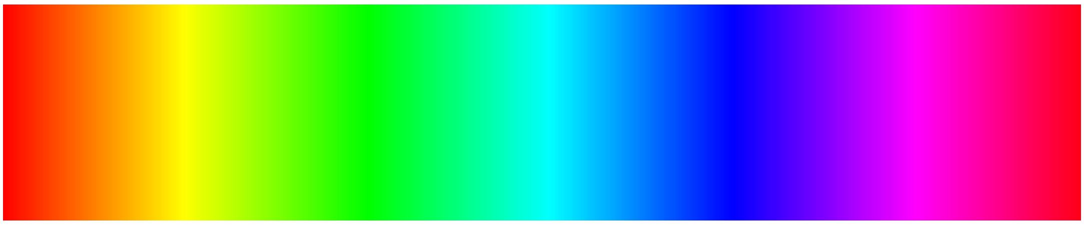
Values

Name	Description
TCVN_CM_AUTUMN	Red - orange - yellow
TCVN_CM_BONE	Black - blueish gray - white
TCVN_CM_JET	Dark blue - green - dark red
TCVN_CM_WINTER	Blue - green
TCVN_CM_RAINBOW	Red - green - purple
TCVN_CM_OCEAN	Black - blue - white
TCVN_CM_SUMMER	Green - yellow
TCVN_CM_SPRING	Pink - yellow
TCVN_CM_COOL	Cyan - magenta
TCVN_CM_HSV	Red - green - blue - red
TCVN_CM_PINK	Black - pink - light yellow - white
TCVN_CM_HOT	Black - red - yellow - white
TCVN_CM_PARULA	Blue - yellow
TCVN_CM_MAGMA	Black - magenta - light yellow
TCVN_CM_INFERNO	Black - magenta - yellow
TCVN_CM_PLASMA	Blue - magenta - yellow
TCVN_CM_VIRIDIS	Purple - yellow
TCVN_CM_CIVIDIS	Blue - gray - yellow
TCVN_CM_TWILIGHT	White - light blue - dark purple - light red - white
TCVN_CM_TWILIGHT_SHIFTED	Dark purple - light blue - white - light red - dark purple
TCVN_CM_TURBO	Dark blue - light green - dark red

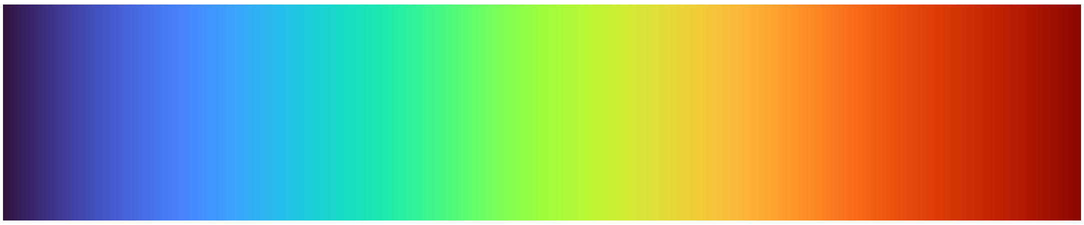
颜色渐变

在函数F_VN_GenerateColorMap [▸ 1198]的帮助下，枚举ETcVnColorMap定义了用于创建颜色表的颜色梯度。以下颜色梯度已预定义：

NONE	
AUTUMN	
BONE	
JET	
WINTER	
RAINBOW	
OCEAN	

SUMMER	
SPRING	
COOL	
HSV	
Pink	
HOT	
PARULA	

MAGMA	
INFERNO	
PLASMA	
VIRIDIS	
CIVIDIS	
TWILIGHT	
TWILIGHT_SHIFTED	

TURBO	

相关函数

- F_VN_GenerateColorMap [▶ 1198]
- F_VN_GenerateCustomColorMap [▶ 1200]

6.1.2.4.13 ETcVnColorMapSize

Offers color map sizes.

Syntax

Definition:

```
TYPE ETcVnColorMapSize :
(
    TCVN_CMS_256   := 256,
    TCVN_CMS_65536 := 65536
)UDINT;
END_TYPE
```

Values

Name	Description
TCVN_CMS_256	256 colors, used for 8 bit images (TCVN_ET_USINT)
TCVN_CMS_65536	65536 colors, used for 16 bit images (TCVN_ET_UINT)

相关函数

- F_VN_GenerateColorMap [▶ 1198]
- F_VN_GenerateCustomColorMap [▶ 1200]

6.1.2.4.14 ETcVnColorSpaceTransform

Offers color space transformations.

Syntax

Definition:

```
TYPE ETcVnColorSpaceTransform :
(
    TCVN_CST_BGR_TO_BGRA                := 0,
    TCVN_CST_RGB_TO_RGBA                := 0,
    TCVN_CST_BGRA_TO_BGR                := 1,
    TCVN_CST_RGBA_TO_RGB                := 1,
    TCVN_CST_BGR_TO_RGBA                := 2,
    TCVN_CST_RGB_TO_BGRA                := 2,
    TCVN_CST_BGRA_TO_RGB                := 3,
    TCVN_CST_RGBA_TO_BGR                := 3,
    TCVN_CST_BGR_TO_RGB                 := 4,
    TCVN_CST_RGB_TO_BGR                 := 4,
    TCVN_CST_BGRA_TO_RGBA               := 5,
    TCVN_CST_RGBA_TO_BGRA               := 5,
    TCVN_CST_BGR_TO_GRAY                := 6,
    TCVN_CST_RGB_TO_GRAY                := 7,
    TCVN_CST_GRAY_TO_BGR                := 8,
```

```
    TCVN_CST_GRAY_TO_RGB                      := 8,
    TCVN_CST_GRAY_TO_BGRA                     := 9,
    TCVN_CST_GRAY_TO_RGBA                     := 9,
    TCVN_CST_BGRA_TO_GRAY                     := 10,
    TCVN_CST_RGBA_TO_GRAY                     := 11,
    TCVN_CST_RGB_TO_BGR_565                   := 12,
    TCVN_CST_BGR_TO_BGR_565                   := 13,
    TCVN_CST_BGR_565_TO_RGB                   := 14,
    TCVN_CST_BGR_565_TO_BGR                   := 15,
    TCVN_CST_RGBA_TO_BGR_565                  := 16,
    TCVN_CST_BGRA_TO_BGR_565                  := 17,
    TCVN_CST_BGR_565_TO_RGBA                  := 18,
    TCVN_CST_BGR_565_TO_BGRA                  := 19,
    TCVN_CST_GRAY_TO_BGR_565                  := 20,
    TCVN_CST_BGR_565_TO_GRAY                  := 21,
    TCVN_CST_RGB_TO_BGR_555                   := 22,
    TCVN_CST_BGR_TO_BGR_555                   := 23,
    TCVN_CST_BGR_555_TO_RGB                   := 24,
    TCVN_CST_BGR_555_TO_BGR                   := 25,
    TCVN_CST_RGBA_TO_BGR_555                  := 26,
    TCVN_CST_BGRA_TO_BGR_555                  := 27,
    TCVN_CST_BGR_555_TO_RGBA                  := 28,
    TCVN_CST_BGR_555_TO_BGRA                  := 29,
    TCVN_CST_GRAY_TO_BGR_555                  := 30,
    TCVN_CST_BGR_555_TO_GRAY                  := 31,
    TCVN_CST_BGR_TO_XYZ                       := 32,
    TCVN_CST_RGB_TO_XYZ                       := 33,
    TCVN_CST_XYZ_TO_BGR                       := 34,
    TCVN_CST_XYZ_TO_RGB                       := 35,
    TCVN_CST_BGR_TO_YCRCB                     := 36,
    TCVN_CST_RGB_TO_YCRCB                     := 37,
    TCVN_CST_YCRCB_TO_BGR                     := 38,
    TCVN_CST_YCRCB_TO_RGB                     := 39,
    TCVN_CST_BGR_TO_HSV                       := 40,
    TCVN_CST_RGB_TO_HSV                       := 41,
    TCVN_CST_BGR_TO_LAB                       := 44,
    TCVN_CST_RGB_TO_LAB                       := 45,
    TCVN_CST_BAYER_RG_TO_BGR                  := 46,
    TCVN_CST_BAYER_BG_TO_RGB                  := 46,
    TCVN_CST_BAYER_GR_TO_BGR                  := 47,
    TCVN_CST_BAYER_GB_TO_RGB                  := 47,
    TCVN_CST_BAYER_BG_TO_BGR                  := 48,
    TCVN_CST_BAYER_RG_TO_RGB                  := 48,
    TCVN_CST_BAYER_GB_TO_BGR                  := 49,
    TCVN_CST_BAYER_GR_TO_RGB                  := 49,
    TCVN_CST_BGR_TO_LUV                       := 50,
    TCVN_CST_RGB_TO_LUV                       := 51,
    TCVN_CST_BGR_TO_HLS                       := 52,
    TCVN_CST_RGB_TO_HLS                       := 53,
    TCVN_CST_HSV_TO_BGR                       := 54,
    TCVN_CST_HSV_TO_RGB                       := 55,
    TCVN_CST_LAB_TO_BGR                       := 56,
    TCVN_CST_LAB_TO_RGB                       := 57,
    TCVN_CST_LUV_TO_BGR                       := 58,
    TCVN_CST_LUV_TO_RGB                       := 59,
    TCVN_CST_HLS_TO_BGR                       := 60,
    TCVN_CST_HLS_TO_RGB                       := 61,
    TCVN_CST_BAYER_RG_TO_BGR_VNG              := 62,
    TCVN_CST_BAYER_GR_TO_BGR_VNG              := 63,
    TCVN_CST_BAYER_BG_TO_BGR_VNG              := 64,
    TCVN_CST_BAYER_GB_TO_BGR_VNG              := 65,
    TCVN_CST_BAYER_RG_TO_RGB_VNG              := 64,
    TCVN_CST_BAYER_GR_TO_RGB_VNG              := 65,
    TCVN_CST_BAYER_BG_TO_RGB_VNG              := 62,
    TCVN_CST_BAYER_GB_TO_RGB_VNG              := 63,
    TCVN_CST_BGR_TO_HSV_FULL                  := 66,
    TCVN_CST_RGB_TO_HSV_FULL                  := 67,
    TCVN_CST_BGR_TO_HLS_FULL                  := 68,
    TCVN_CST_RGB_TO_HLS_FULL                  := 69,
    TCVN_CST_HSV_TO_BGR_FULL                  := 70,
    TCVN_CST_HSV_TO_RGB_FULL                  := 71,
    TCVN_CST_HLS_TO_BGR_FULL                  := 72,
    TCVN_CST_HLS_TO_RGB_FULL                  := 73,
    TCVN_CST_LBGR_TO_LAB                      := 74,
    TCVN_CST_LRGB_TO_LAB                      := 75,
    TCVN_CST_LBGR_TO_LUV                      := 76,
    TCVN_CST_LRGB_TO_LUV                      := 77,
    TCVN_CST_LAB_TO_LBGR                      := 78,
    TCVN_CST_LAB_TO_LRGB                      := 79,
    TCVN_CST_LUV_TO_LBGR                      := 80,
```

```
    TCVN_CST_LUV_TO_LRGB                        := 81,
    TCVN_CST_BGR_TO_YUV                         := 82,
    TCVN_CST_RGB_TO_YUV                         := 83,
    TCVN_CST_YUV_TO_BGR                         := 84,
    TCVN_CST_YUV_TO_RGB                         := 85,
    TCVN_CST_BAYER_RG_TO_GRAY                   := 86,
    TCVN_CST_BAYER_GR_TO_GRAY                   := 87,
    TCVN_CST_BAYER_BG_TO_GRAY                   := 88,
    TCVN_CST_BAYER_GB_TO_GRAY                   := 89,
    TCVN_CST_YUV_420_NV12_TO_RGB                := 90,
    TCVN_CST_YUV_420_NV12_TO_BGR                := 91,
    TCVN_CST_YUV_420_NV21_TO_RGB                := 92,
    TCVN_CST_YUV_420_NV21_TO_BGR                := 93,
    TCVN_CST_YUV_420_SP_TO_RGB                  := 92,
    TCVN_CST_YUV_420_SP_TO_BGR                  := 93,
    TCVN_CST_YUV_420_NV12_TO_RGBA               := 94,
    TCVN_CST_YUV_420_NV12_TO_BGRA               := 95,
    TCVN_CST_YUV_420_NV21_TO_RGBA               := 96,
    TCVN_CST_YUV_420_NV21_TO_BGRA               := 97,
    TCVN_CST_YUV_420_SP_TO_RGBA                 := 96,
    TCVN_CST_YUV_420_SP_TO_BGRA                 := 97,
    TCVN_CST_YUV_420_YV12_TO_RGB                := 98,
    TCVN_CST_YUV_420_YV12_TO_BGR                := 99,
    TCVN_CST_YUV_420_IYUV_TO_RGB                := 100,
    TCVN_CST_YUV_420_IYUV_TO_BGR                := 101,
    TCVN_CST_YUV_420_I420_TO_RGB                := 100,
    TCVN_CST_YUV_420_I420_TO_BGR                := 101,
    TCVN_CST_YUV_420_P_TO_RGB                   := 98,
    TCVN_CST_YUV_420_P_TO_BGR                   := 99,
    TCVN_CST_YUV_420_YV12_TO_RGBA               := 102,
    TCVN_CST_YUV_420_YV12_TO_BGRA               := 103,
    TCVN_CST_YUV_420_IYUV_TO_RGBA               := 104,
    TCVN_CST_YUV_420_IYUV_TO_BGRA               := 105,
    TCVN_CST_YUV_420_I420_TO_RGBA               := 104,
    TCVN_CST_YUV_420_I420_TO_BGRA               := 105,
    TCVN_CST_YUV_420_P_TO_RGBA                  := 102,
    TCVN_CST_YUV_420_P_TO_BGRA                  := 103,
    TCVN_CST_YUV_420_TO_GRAY                    := 106,
    TCVN_CST_YUV_420_NV21_TO_GRAY               := 106,
    TCVN_CST_YUV_420_NV12_TO_GRAY               := 106,
    TCVN_CST_YUV_420_YV12_TO_GRAY               := 106,
    TCVN_CST_YUV_420_IYUV_TO_GRAY               := 106,
    TCVN_CST_YUV_420_I420_TO_GRAY               := 106,
    TCVN_CST_YUV_420_SP_TO_GRAY                 := 106,
    TCVN_CST_YUV_420_P_TO_GRAY                  := 106,
    TCVN_CST_YUV_422_UYVY_TO_RGB                := 107,
    TCVN_CST_YUV_422_UYVY_TO_BGR                := 108,
    TCVN_CST_YUV_422_Y422_TO_RGB                := 107,
    TCVN_CST_YUV_422_Y422_TO_BGR                := 108,
    TCVN_CST_YUV_422_UYNV_TO_RGB                := 107,
    TCVN_CST_YUV_422_UYNV_TO_BGR                := 108,
    TCVN_CST_YUV_422_UYVY_TO_RGBA               := 111,
    TCVN_CST_YUV_422_UYVY_TO_BGRA               := 112,
    TCVN_CST_YUV_422_Y422_TO_RGBA               := 111,
    TCVN_CST_YUV_422_Y422_TO_BGRA               := 112,
    TCVN_CST_YUV_422_UYNV_TO_RGBA               := 111,
    TCVN_CST_YUV_422_UYNV_TO_BGRA               := 112,
    TCVN_CST_YUV_422_YUY2_TO_RGB                := 115,
    TCVN_CST_YUV_422_YUY2_TO_BGR                := 116,
    TCVN_CST_YUV_422_YVYU_TO_RGB                := 117,
    TCVN_CST_YUV_422_YVYU_TO_BGR                := 118,
    TCVN_CST_YUV_422_YUYV_TO_RGB                := 115,
    TCVN_CST_YUV_422_YUYV_TO_BGR                := 116,
    TCVN_CST_YUV_422_YUNV_TO_RGB                := 115,
    TCVN_CST_YUV_422_YUNV_TO_BGR                := 116,
    TCVN_CST_YUV_422_YUY2_TO_RGBA               := 119,
    TCVN_CST_YUV_422_YUY2_TO_BGRA               := 120,
    TCVN_CST_YUV_422_YVYU_TO_RGBA               := 121,
    TCVN_CST_YUV_422_YVYU_TO_BGRA               := 122,
    TCVN_CST_YUV_422_YUYV_TO_RGBA               := 119,
    TCVN_CST_YUV_422_YUYV_TO_BGRA               := 120,
    TCVN_CST_YUV_422_YUNV_TO_RGBA               := 119,
    TCVN_CST_YUV_422_YUNV_TO_BGRA               := 120,
    TCVN_CST_YUV_422_UYVY_TO_GRAY               := 123,
    TCVN_CST_YUV_422_YUY2_TO_GRAY               := 124,
    TCVN_CST_YUV_422_Y422_TO_GRAY               := 123,
    TCVN_CST_YUV_422_UYNV_TO_GRAY               := 123,
    TCVN_CST_YUV_422_YVYU_TO_GRAY               := 124,
    TCVN_CST_YUV_422_YUYV_TO_GRAY               := 124,
    TCVN_CST_YUV_422_YUNV_TO_GRAY               := 124,
```

```
    TCVN_CST_RGBA_TO_PREMULTIPLICATED_RGBA := 125,
    TCVN_CST_PREMULTIPLICATED_RGBA_TO_RGBA := 126,
    TCVN_CST_RGB_TO_YUV_420_I420           := 127,
    TCVN_CST_BGR_TO_YUV_420_I420           := 128,
    TCVN_CST_RGB_TO_YUV_420_IYUV           := 127,
    TCVN_CST_BGR_TO_YUV_420_IYUV           := 128,
    TCVN_CST_RGBA_TO_YUV_420_I420          := 129,
    TCVN_CST_BGRA_TO_YUV_420_I420          := 130,
    TCVN_CST_RGBA_TO_YUV_420_IYUV          := 129,
    TCVN_CST_BGRA_TO_YUV_420_IYUV          := 130,
    TCVN_CST_RGB_TO_YUV_420_YV12           := 131,
    TCVN_CST_BGR_TO_YUV_420_YV12           := 132,
    TCVN_CST_RGBA_TO_YUV_420_YV12          := 133,
    TCVN_CST_BGRA_TO_YUV_420_YV12          := 134,
    TCVN_CST_BAYER_RG_TO_BGR_EA            := 135,
    TCVN_CST_BAYER_GR_TO_BGR_EA            := 136,
    TCVN_CST_BAYER_BG_TO_BGR_EA            := 137,
    TCVN_CST_BAYER_GB_TO_BGR_EA            := 138,
    TCVN_CST_BAYER_RG_TO_RGB_EA            := 137,
    TCVN_CST_BAYER_GR_TO_RGB_EA            := 138,
    TCVN_CST_BAYER_BG_TO_RGB_EA            := 135,
    TCVN_CST_BAYER_GB_TO_RGB_EA            := 136,
    TCVN_CST_BAYER_RG_TO_BGRA              := 139,
    TCVN_CST_BAYER_GR_TO_BGRA              := 140,
    TCVN_CST_BAYER_BG_TO_BGRA              := 141,
    TCVN_CST_BAYER_GB_TO_BGRA              := 142,
    TCVN_CST_BAYER_RG_TO_RGBA              := 141,
    TCVN_CST_BAYER_GR_TO_RGBA              := 142,
    TCVN_CST_BAYER_BG_TO_RGBA              := 139,
    TCVN_CST_BAYER_GB_TO_RGBA              := 140,
    TCVN_CST_MAX                           := 143
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_CST_BGR_TO_BGRA	Transform the color space from BGR to BGRA.
TCVN_CST_RGB_TO_RGBA	Transform the color space from RGB to RGBA.
TCVN_CST_BGRA_TO_BGR	Transform the color space from BGRA to BGR.
TCVN_CST_RGBA_TO_RGB	Transform the color space from RGBA to RGB.
TCVN_CST_BGR_TO_RGBA	Transform the color space from BGR to RGBA.
TCVN_CST_RGB_TO_BGRA	Transform the color space from RGB to BGRA.
TCVN_CST_BGRA_TO_RGB	Transform the color space from BGRA to RGB.
TCVN_CST_RGBA_TO_BGR	Transform the color space from RGBA to BGR.
TCVN_CST_BGR_TO_RGB	Transform the color space from BGR to RGB.
TCVN_CST_RGB_TO_BGR	Transform the color space from RGB to BGR.
TCVN_CST_BGRA_TO_RGBA	Transform the color space from BGRA to RGBA.
TCVN_CST_RGBA_TO_BGRA	Transform the color space from RGBA to BGRA.
TCVN_CST_BGR_TO_GRAY	Transform the color space from BGR to Gray.
TCVN_CST_RGB_TO_GRAY	Transform the color space from RGB to Gray.
TCVN_CST_GRAY_TO_BGR	Transform the color space from Gray to BGR.
TCVN_CST_GRAY_TO_RGB	Transform the color space from Gray to RGB.
TCVN_CST_GRAY_TO_BGRA	Transform the color space from Gray to BGRA.
TCVN_CST_GRAY_TO_RGBA	Transform the color space from Gray to RGBA.
TCVN_CST_BGRA_TO_GRAY	Transform the color space from BGRA to Gray.
TCVN_CST_RGBA_TO_GRAY	Transform the color space from RGBA to Gray.
TCVN_CST_RGB_TO_BGR_565	Transform the color space from RGB(24 bit 888) to BGR(16 bit 565).
TCVN_CST_BGR_TO_BGR_565	Transform the color space from BGR(24 bit 888) to BGR(16 bit 565).
TCVN_CST_BGR_565_TO_RGB	Transform the color space from BGR(16 bit 565) to RGB(24 bit 888).
TCVN_CST_BGR_565_TO_BGR	Transform the color space from BGR(16 bit 565) to BGR(24 bit 888).
TCVN_CST_RGBA_TO_BGR_565	Transform the color space from RGBA(32 bit 8888) to BGR(16 bit 565).
TCVN_CST_BGRA_TO_BGR_565	Transform the color space from BGRA(32 bit 8888) to BGR(16 bit 565).
TCVN_CST_BGR_565_TO_RGBA	Transform the color space from BGR(16 bit 565) to RGBA(32 bit 8888).
TCVN_CST_BGR_565_TO_BGRA	Transform the color space from BGR(16 bit 565) to BGRA(32 bit 8888).
TCVN_CST_GRAY_TO_BGR_565	Transform the color space from Gray to BGR(16 bit 565).
TCVN_CST_BGR_565_TO_GRAY	Transform the color space from BGR(16 bit 565) to Gray.
TCVN_CST_RGB_TO_BGR_555	Transform the color space from RGB(24 bit 888) to BGR(16 bit 555).
TCVN_CST_BGR_TO_BGR_555	Transform the color space from BGR(24 bit 888) to BGR(16 bit 555).
TCVN_CST_BGR_555_TO_RGB	Transform the color space from BGR(16 bit 555) to RGB(24 bit 888).
TCVN_CST_BGR_555_TO_BGR	Transform the color space from BGR(16 bit 555) to BGR(24 bit 888).
TCVN_CST_RGBA_TO_BGR_555	Transform the color space from RGBA(32 bit 8888) to BGR(16 bit 555).
TCVN_CST_BGRA_TO_BGR_555	Transform the color space from BGRA(32 bit 8888) to BGR(16 bit 555).

Name	Description
TCVN_CST_BGR_555_TO_RGBA	Transform the color space from BGR(16 bit 555) to RGBA(32 bit 8888).
TCVN_CST_BGR_555_TO_BGRA	Transform the color space from BGR(16 bit 555) to BGRA(32 bit 8888).
TCVN_CST_GRAY_TO_BGR_555	Transform the color space from Gray to BGR(16 bit 555).
TCVN_CST_BGR_555_TO_GRAY	Transform the color space from BGR(16 bit 555) to Gray.
TCVN_CST_BGR_TO_XYZ	Transform the color space from BGR to CIE XYZ (scaled to the full value range of the image).
TCVN_CST_RGB_TO_XYZ	Transform the color space from RGB to CIE XYZ (scaled to the full value range of the image).
TCVN_CST_XYZ_TO_BGR	Transform the color space from CIE XYZ (scaled to the full value range of the image) to BGR.
TCVN_CST_XYZ_TO_RGB	Transform the color space from CIE XYZ (scaled to the full value range of the image) to RGB.
TCVN_CST_BGR_TO_YCRCB	Transform the color space from BGR to YCrCb (scaled to the full value range of the image).
TCVN_CST_RGB_TO_YCRCB	Transform the color space from RGB to YCrCb (scaled to the full value range of the image).
TCVN_CST_YCRCB_TO_BGR	Transform the color space from YCrCb (scaled to the full value range of the image) to BGR.
TCVN_CST_YCRCB_TO_RGB	Transform the color space from YCrCb (scaled to the full value range of the image) to RGB.
TCVN_CST_BGR_TO_HSV	Transform the color space from BGR to HSV (for images of type USINT, H is scaled to a range from 0 to 180 and S,V from 0 to 255).
TCVN_CST_RGB_TO_HSV	Transform the color space from RGB to HSV (for images of type USINT, H is scaled to a range from 0 to 180 and S,V from 0 to 255).
TCVN_CST_BGR_TO_LAB	Transform the color space from BGR to CIE L*a*b* (for images of type USINT, all channels are scaled to a range from 0 to 255).
TCVN_CST_RGB_TO_LAB	Transform the color space from RGB to CIE L*a*b* (for images of type USINT, all channels are scaled to a range from 0 to 255).
TCVN_CST_BAYER_RG_TO_BGR	Transform the color space from BayerRG to BGR.
TCVN_CST_BAYER_BG_TO_RGB	Transform the color space from BayerBG to RGB.
TCVN_CST_BAYER_GR_TO_BGR	Transform the color space from BayerGR to BGR.
TCVN_CST_BAYER_GB_TO_RGB	Transform the color space from BayerGB to RGB.
TCVN_CST_BAYER_BG_TO_BGR	Transform the color space from BayerBG to BGR.
TCVN_CST_BAYER_RG_TO_RGB	Transform the color space from BayerRG to RGB.
TCVN_CST_BAYER_GB_TO_BGR	Transform the color space from BayerGB to BGR.
TCVN_CST_BAYER_GR_TO_RGB	Transform the color space from BayerGR to RGB.
TCVN_CST_BGR_TO_LUV	Transform the color space from BGR to CIE L*u*v* (for images of type USINT, all channels are scaled to a range from 0 to 255).
TCVN_CST_RGB_TO_LUV	Transform the color space from RGB to CIE L*u*v* (for images of type USINT, all channels are scaled to a range from 0 to 255).
TCVN_CST_BGR_TO_HLS	Transform the color space from BGR to HLS (for images of type USINT, H is scaled to a range from 0 to 180 and L,S from 0 to 255).

Name	Description
TCVN_CST_RGB_TO_HLS	Transform the color space from RGB to HLS (for images of type USINT, H is scaled to a range from 0 to 180 and L,S from 0 to 255).
TCVN_CST_HSV_TO_BGR	Transform the color space from HSV (for images of type USINT, H is scaled to a range from 0 to 180 and S,V from 0 to 255) to BGR.
TCVN_CST_HSV_TO_RGB	Transform the color space from HSV (for images of type USINT, H is scaled to a range from 0 to 180 and S,V from 0 to 255) to RGB.
TCVN_CST_LAB_TO_BGR	Transform the color space from CIE L*a*b* (for images of type USINT, all channels are scaled to a range from 0 to 255) to BGR.
TCVN_CST_LAB_TO_RGB	Transform the color space from CIE L*a*b* (for images of type USINT, all channels are scaled to a range from 0 to 255) to RGB.
TCVN_CST_LUV_TO_BGR	Transform the color space from CIE L*u*v* (for images of type USINT, all channels are scaled to a range from 0 to 255) to BGR.
TCVN_CST_LUV_TO_RGB	Transform the color space from CIE L*u*v* (for images of type USINT, all channels are scaled to a range from 0 to 255) to RGB.
TCVN_CST_HLS_TO_BGR	Transform the color space from HLS (for images of type USINT, H is scaled to a range from 0 to 180 and L,S from 0 to 255) to BGR.
TCVN_CST_HLS_TO_RGB	Transform the color space from HLS (for images of type USINT, H is scaled to a range from 0 to 180 and L,S from 0 to 255) to RGB.
TCVN_CST_BAYER_RG_TO_BGR_VNG	Transform the color space from BayerRG to BGR, using Variable Number of Gradients.
TCVN_CST_BAYER_GR_TO_BGR_VNG	Transform the color space from BayerGR to BGR, using Variable Number of Gradients.
TCVN_CST_BAYER_BG_TO_BGR_VNG	Transform the color space from BayerBG to BGR, using Variable Number of Gradients.
TCVN_CST_BAYER_GB_TO_BGR_VNG	Transform the color space from BayerGB to BGR, using Variable Number of Gradients.
TCVN_CST_BAYER_RG_TO_RGB_VNG	Transform the color space from BayerRG to RGB, using Variable Number of Gradients.
TCVN_CST_BAYER_GR_TO_RGB_VNG	Transform the color space from BayerGR to RGB, using Variable Number of Gradients.
TCVN_CST_BAYER_BG_TO_RGB_VNG	Transform the color space from BayerBG to RGB, using Variable Number of Gradients.
TCVN_CST_BAYER_GB_TO_RGB_VNG	Transform the color space from BayerGB to RGB, using Variable Number of Gradients.
TCVN_CST_BGR_TO_HSV_FULL	Transform the color space from BGR to HSV (for images of type USINT, all channels are scaled to a range from 0 to 255).
TCVN_CST_RGB_TO_HSV_FULL	Transform the color space from RGB to HSV (for images of type USINT, all channels are scaled to a range from 0 to 255).
TCVN_CST_BGR_TO_HLS_FULL	Transform the color space from BGR to HLS (for images of type USINT, all channels are scaled to a range from 0 to 255).
TCVN_CST_RGB_TO_HLS_FULL	Transform the color space from RGB to HLS (for images of type USINT, all channels are scaled to a range from 0 to 255).

Name	Description
TCVN_CST_HSV_TO_BGR_FULL	Transform the color space from HSV (for images of type USINT, all channels are scaled to a range from 0 to 255) to BGR.
TCVN_CST_HSV_TO_RGB_FULL	Transform the color space from HSV (for images of type USINT, all channels are scaled to a range from 0 to 255) to RGB.
TCVN_CST_HLS_TO_BGR_FULL	Transform the color space from HLS (for images of type USINT, all channels are scaled to a range from 0 to 255) to BGR.
TCVN_CST_HLS_TO_RGB_FULL	Transform the color space from HLS (for images of type USINT, all channels are scaled to a range from 0 to 255) to RGB.
TCVN_CST_LBGR_TO_LAB	Transform the color space from LBGR to CIE L*a*b* (for images of type USINT, all channels are scaled to a range from 0 to 255).
TCVN_CST_LRGB_TO_LAB	Transform the color space from LRGB to CIE L*a*b* (for images of type USINT, all channels are scaled to a range from 0 to 255).
TCVN_CST_LBGR_TO_LUV	Transform the color space from LBGR to CIE L*u*v* (for images of type USINT, all channels are scaled to a range from 0 to 255).
TCVN_CST_LRGB_TO_LUV	Transform the color space from LRGB to CIE L*u*v* (for images of type USINT, all channels are scaled to a range from 0 to 255).
TCVN_CST_LAB_TO_LBGR	Transform the color space from CIE L*a*b* (for images of type USINT, all channels are scaled to a range from 0 to 255) to LBGR.
TCVN_CST_LAB_TO_LRGB	Transform the color space from CIE L*a*b* (for images of type USINT, all channels are scaled to a range from 0 to 255) to LRGB.
TCVN_CST_LUV_TO_LBGR	Transform the color space from CIE L*u*v* (for images of type USINT, all channels are scaled to a range from 0 to 255) to LBGR.
TCVN_CST_LUV_TO_LRGB	Transform the color space from CIE L*u*v* (for images of type USINT, all channels are scaled to a range from 0 to 255) to LRGB.
TCVN_CST_BGR_TO_YUV	Transform the color space from BGR to YUV (scaled to the full value range of the image).
TCVN_CST_RGB_TO_YUV	Transform the color space from RGB to YUV (scaled to the full value range of the image).
TCVN_CST_YUV_TO_BGR	Transform the color space from YUV (scaled to the full value range of the image) to BGR.
TCVN_CST_YUV_TO_RGB	Transform the color space from YUV (scaled to the full value range of the image) to RGB.
TCVN_CST_BAYER_RG_TO_GRAY	Transform the color space from BayerRG to Gray.
TCVN_CST_BAYER_GR_TO_GRAY	Transform the color space from BayerGR to Gray.
TCVN_CST_BAYER_BG_TO_GRAY	Transform the color space from BayerBG to Gray.
TCVN_CST_BAYER_GB_TO_GRAY	Transform the color space from BayerGB to Gray.
TCVN_CST_YUV_420_NV12_TO_RGB	Transform the color space from YUV420 NV12 to RGB.
TCVN_CST_YUV_420_NV12_TO_BGR	Transform the color space from YUV420 NV12 to BGR.
TCVN_CST_YUV_420_NV21_TO_RGB	Transform the color space from YUV420 NV21 (SP) to RGB.
TCVN_CST_YUV_420_NV21_TO_BGR	Transform the color space from YUV420 NV21 (SP) to BGR.

Name	Description
TCVN_CST_YUV_420_SP_TO_RGB	Transform the color space from YUV420 NV21 (SP) to RGB.
TCVN_CST_YUV_420_SP_TO_BGR	Transform the color space from YUV420 NV21 (SP) to BGR.
TCVN_CST_YUV_420_NV12_TO_RGBA	Transform the color space from YUV420 NV12 to RGBA.
TCVN_CST_YUV_420_NV12_TO_BGRA	Transform the color space from YUV420 NV12 to BGRA.
TCVN_CST_YUV_420_NV21_TO_RGBA	Transform the color space from YUV420 NV21 (SP) to RGBA.
TCVN_CST_YUV_420_NV21_TO_BGRA	Transform the color space from YUV420 NV21 (SP) to BGRA.
TCVN_CST_YUV_420_SP_TO_RGBA	Transform the color space from YUV420 NV21 (SP) to RGBA.
TCVN_CST_YUV_420_SP_TO_BGRA	Transform the color space from YUV420 NV21 (SP) to BGRA.
TCVN_CST_YUV_420_YV12_TO_RGB	Transform the color space from YUV420 YV12 (P) to RGB.
TCVN_CST_YUV_420_YV12_TO_BGR	Transform the color space from YUV420 YV12 (P) to BGR.
TCVN_CST_YUV_420_IYUV_TO_RGB	Transform the color space from YUV420 IYUV (I420) to RGB.
TCVN_CST_YUV_420_IYUV_TO_BGR	Transform the color space from YUV420 IYUV (I420) to BGR.
TCVN_CST_YUV_420_I420_TO_RGB	Transform the color space from YUV420 IYUV (I420) to RGB.
TCVN_CST_YUV_420_I420_TO_BGR	Transform the color space from YUV420 IYUV (I420) to BGR.
TCVN_CST_YUV_420_P_TO_RGB	Transform the color space from YUV420 YV12 (P) to RGB.
TCVN_CST_YUV_420_P_TO_BGR	Transform the color space from YUV420 YV12 (P) to BGR.
TCVN_CST_YUV_420_YV12_TO_RGBA	Transform the color space from YUV420 YV12 (P) to RGBA.
TCVN_CST_YUV_420_YV12_TO_BGRA	Transform the color space from YUV420 YV12 (P) to BGRA.
TCVN_CST_YUV_420_IYUV_TO_RGBA	Transform the color space from YUV420 IYUV (I420) to RGBA.
TCVN_CST_YUV_420_IYUV_TO_BGRA	Transform the color space from YUV420 IYUV (I420) to BGRA.
TCVN_CST_YUV_420_I420_TO_RGBA	Transform the color space from YUV420 IYUV (I420) to RGBA.
TCVN_CST_YUV_420_I420_TO_BGRA	Transform the color space from YUV420 IYUV (I420) to BGRA.
TCVN_CST_YUV_420_P_TO_RGBA	Transform the color space from YUV420 YV12 (P) to RGBA.
TCVN_CST_YUV_420_P_TO_BGRA	Transform the color space from YUV420 YV12 (P) to BGRA.
TCVN_CST_YUV_420_TO_GRAY	Transform the color space from YUV420 to Gray.
TCVN_CST_YUV_420_NV21_TO_GRAY	Transform the color space from YUV420 to Gray.
TCVN_CST_YUV_420_NV12_TO_GRAY	Transform the color space from YUV420 to Gray.
TCVN_CST_YUV_420_YV12_TO_GRAY	Transform the color space from YUV420 to Gray.
TCVN_CST_YUV_420_IYUV_TO_GRAY	Transform the color space from YUV420 to Gray.
TCVN_CST_YUV_420_I420_TO_GRAY	Transform the color space from YUV420 to Gray.

Name	Description
TCVN_CST_YUV_420_SP_TO_GRAY	Transform the color space from YUV420 to Gray.
TCVN_CST_YUV_420_P_TO_GRAY	Transform the color space from YUV420 to Gray.
TCVN_CST_YUV_422_UYVY_TO_RGB	Transform the color space from YUV422 UYVY (Y422, UYNV) to RGB.
TCVN_CST_YUV_422_UYVY_TO_BGR	Transform the color space from YUV422 UYVY (Y422, UYNV) to BGR.
TCVN_CST_YUV_422_Y422_TO_RGB	Transform the color space from YUV422 UYVY (Y422, UYNV) to RGB.
TCVN_CST_YUV_422_Y422_TO_BGR	Transform the color space from YUV422 UYVY (Y422, UYNV) to BGR.
TCVN_CST_YUV_422_UYNV_TO_RGB	Transform the color space from YUV422 UYVY (Y422, UYNV) to RGB.
TCVN_CST_YUV_422_UYNV_TO_BGR	Transform the color space from YUV422 UYVY (Y422, UYNV) to BGR.
TCVN_CST_YUV_422_UYVY_TO_RGBA	Transform the color space from YUV422 UYVY (Y422, UYNV) to RGBA.
TCVN_CST_YUV_422_UYVY_TO_BGRA	Transform the color space from YUV422 UYVY (Y422, UYNV) to BGRA.
TCVN_CST_YUV_422_Y422_TO_RGBA	Transform the color space from YUV422 UYVY (Y422, UYNV) to RGB.
TCVN_CST_YUV_422_Y422_TO_BGRA	Transform the color space from YUV422 UYVY (Y422, UYNV) to BGRA.
TCVN_CST_YUV_422_UYNV_TO_RGBA	Transform the color space from YUV422 UYVY (Y422, UYNV) to RGB.
TCVN_CST_YUV_422_UYNV_TO_BGRA	Transform the color space from YUV422 UYVY (Y422, UYNV) to BGRA.
TCVN_CST_YUV_422_YUY2_TO_RGB	Transform the color space from YUV422 YUYV (YUY2, YUNV) to RGB.
TCVN_CST_YUV_422_YUY2_TO_BGR	Transform the color space from YUV422 YUYV (YUY2, YUNV) to BGR.
TCVN_CST_YUV_422_YVYU_TO_RGB	Transform the color space from YUV422 YVYU to RGB.
TCVN_CST_YUV_422_YVYU_TO_BGR	Transform the color space from YUV422 YVYU to BGR.
TCVN_CST_YUV_422_YUYV_TO_RGB	Transform the color space from YUV422 YUYV (YUY2, YUNV) to RGB.
TCVN_CST_YUV_422_YUYV_TO_BGR	Transform the color space from YUV422 YUYV (YUY2, YUNV) to BGR.
TCVN_CST_YUV_422_YUNV_TO_RGB	Transform the color space from YUV422 YUYV (YUY2, YUNV) to RGB.
TCVN_CST_YUV_422_YUNV_TO_BGR	Transform the color space from YUV422 YUYV (YUY2, YUNV) to BGR.
TCVN_CST_YUV_422_YUY2_TO_RGBA	Transform the color space from YUV422 YUYV (YUY2, YUNV) to RGBA.
TCVN_CST_YUV_422_YUY2_TO_BGRA	Transform the color space from YUV422 YUYV (YUY2, YUNV) to BGRA.
TCVN_CST_YUV_422_YVYU_TO_RGBA	Transform the color space from YUV422 YVYU to RGBA.
TCVN_CST_YUV_422_YVYU_TO_BGRA	Transform the color space from YUV422 YVYU to BGRA.
TCVN_CST_YUV_422_YUYV_TO_RGBA	Transform the color space from YUV422 YUYV (YUY2, YUNV) to RGBA.
TCVN_CST_YUV_422_YUYV_TO_BGRA	Transform the color space from YUV422 YUYV (YUY2, YUNV) to BGRA.

Name	Description
TCVN_CST_YUV_422_YUNV_TO_RGBA	Transform the color space from YUV422 YUYV (YUY2, YUNV) to RGBA.
TCVN_CST_YUV_422_YUNV_TO_BGRA	Transform the color space from YUV422 YUYV (YUY2, YUNV) to BGRA.
TCVN_CST_YUV_422_UYVY_TO_GRAY	Transform the color space from YUV422 UYVY (Y422, UYNV) to Gray.
TCVN_CST_YUV_422_YUY2_TO_GRAY	Transform the color space from YUV422 YUYV (YUY2, YUNV) to Gray.
TCVN_CST_YUV_422_Y422_TO_GRAY	Transform the color space from YUV422 UYVY (Y422, UYNV) to Gray.
TCVN_CST_YUV_422_UYNV_TO_GRAY	Transform the color space from YUV422 UYVY (Y422, UYNV) to Gray.
TCVN_CST_YUV_422_YVYU_TO_GRAY	Transform the color space from YUV422 YVYU to Gray.
TCVN_CST_YUV_422_YUYV_TO_GRAY	Transform the color space from YUV422 YUYV (YUY2, YUNV) to Gray.
TCVN_CST_YUV_422_YUNV_TO_GRAY	Transform the color space from YUV422 YUYV (YUY2, YUNV) to Gray.
TCVN_CST_RGBA_TO_PREMULTIPLICATED_RGBA	Transform the color space from RGBA to premultiplicated RGBA.
TCVN_CST_PREMULTIPLICATED_RGBA_TO_RGBA	Transform the color space from premultiplicated RGBA to RGBA.
TCVN_CST_RGB_TO_YUV_420_I420	Transform the color space from RGB to YUV420 IYUV (I420).
TCVN_CST_BGR_TO_YUV_420_I420	Transform the color space from BGR to YUV420 IYUV (I420).
TCVN_CST_RGB_TO_YUV_420_IYUV	Transform the color space from RGB to YUV420 IYUV (I420).
TCVN_CST_BGR_TO_YUV_420_IYUV	Transform the color space from BGR to YUV420 IYUV (I420).
TCVN_CST_RGBA_TO_YUV_420_I420	Transform the color space from RGBA to YUV420 IYUV (I420).
TCVN_CST_BGRA_TO_YUV_420_I420	Transform the color space from BGRA to YUV420 IYUV (I420).
TCVN_CST_RGBA_TO_YUV_420_IYUV	Transform the color space from RGBA to YUV420 IYUV (I420).
TCVN_CST_BGRA_TO_YUV_420_IYUV	Transform the color space from BGRA to YUV420 IYUV (I420).
TCVN_CST_RGB_TO_YUV_420_YV12	Transform the color space from RGB to YUV420 YV12 (P).
TCVN_CST_BGR_TO_YUV_420_YV12	Transform the color space from BGR to YUV420 YV12 (P).
TCVN_CST_RGBA_TO_YUV_420_YV12	Transform the color space from RGBA to YUV420 YV12 (P).
TCVN_CST_BGRA_TO_YUV_420_YV12	Transform the color space from BGRA to YUV420 YV12 (P).
TCVN_CST_BAYER_RG_TO_BGR_EA	Transform the color space from BayerRG to BGR, using an Edge Aware algorithm.
TCVN_CST_BAYER_GR_TO_BGR_EA	Transform the color space from BayerGR to BGR, using an Edge Aware algorithm.
TCVN_CST_BAYER_BG_TO_BGR_EA	Transform the color space from BayerBG to BGR, using an Edge Aware algorithm.
TCVN_CST_BAYER_GB_TO_BGR_EA	Transform the color space from BayerGB to BGR, using an Edge Aware algorithm.

Name	Description
TCVN_CST_BAYER_RG_TO_RGB_EA	Transform the color space from BayerRG to RGB, using an Edge Aware algorithm.
TCVN_CST_BAYER_GR_TO_RGB_EA	Transform the color space from BayerGR to RGB, using an Edge Aware algorithm.
TCVN_CST_BAYER_BG_TO_RGB_EA	Transform the color space from BayerBG to RGB, using an Edge Aware algorithm.
TCVN_CST_BAYER_GB_TO_RGB_EA	Transform the color space from BayerGB to RGB, using an Edge Aware algorithm.
TCVN_CST_BAYER_RG_TO_BGRA	Transform the color space from BayerRG to BGRA.
TCVN_CST_BAYER_GR_TO_BGRA	Transform the color space from BayerGR to BGRA.
TCVN_CST_BAYER_BG_TO_BGRA	Transform the color space from BayerBG to BGRA.
TCVN_CST_BAYER_GB_TO_BGRA	Transform the color space from BayerGB to BGRA.
TCVN_CST_BAYER_RG_TO_RGBA	Transform the color space from BayerRG to RGBA.
TCVN_CST_BAYER_GR_TO_RGBA	Transform the color space from BayerGR to RGBA.
TCVN_CST_BAYER_BG_TO_RGBA	Transform the color space from BayerBG to RGBA.
TCVN_CST_BAYER_GB_TO_RGBA	Transform the color space from BayerGB to RGBA.
TCVN_CST_MAX	For internal use only, adapted when adding new values.

相关函数

- F_VN_ConvertColorSpace [▶ 1197]

6.1.2.4.15 ETcVnColorTrainingMethod

Offers color training methods.

Syntax

Definition:

```
TYPE ETcVnColorTrainingMethod :
(
    TCVN_CTM_LAB := 0,
    TCVN_CTM_RGB := 1
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_CTM_LAB	Trains the color in CIE L*a*b* color space.
TCVN_CTM_RGB	Trains the color in RGB color space.

相关函数

- F_VN_TrainImageColorExp [▶ 1221]

6.1.2.4.16 ETcVnConnectedComponentsAlgorithm

Offers connected components algorithms.

Syntax

Definition:

```
TYPE ETcVnConnectedComponentsAlgorithm :
(
    TCVN_CCA_WU     := 0,
```

```
    TCVN_CCA_GRANA := 1
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_CCA_WU	SAUF algorithm
TCVN_CCA_GRANA	BBDT algorithm for 8-way connectivity, SAUF algorithm for 4-way connectivity

6.1.2.4.17 ETcVnContainerExportFormat

Offers container export formats.

Syntax

Definition:

```
TYPE ETcVnContainerExportFormat :
(
    TCVN_CEF_XML            := 0,
    TCVN_CEF_XML_SERIALIZED := 1,
    TCVN_CEF_CSV            := 2,
    TCVN_CEF_BINARY         := 3
)UINT;
END_TYPE
```

Values

Name	Description
TCVN_CEF_XML	XML format with human readable data
TCVN_CEF_XML_SERIALIZED	XML format with serialized data
TCVN_CEF_CSV	CSV format (limited to two dimensional representation, so not applicable for all container types)
TCVN_CEF_BINARY	Binary serialized data (recommended for large containers)

相关元素

- FB_VN_WriteContainer [▶ 1534]

6.1.2.4.18 ETcVnContourApproximationMethod

Offers methods for contour approximation.

Syntax

Definition:

```
TYPE ETcVnContourApproximationMethod :
(
    TCVN_CAM_NONE       := 1,
    TCVN_CAM_SIMPLE     := 2,
    TCVN_CAM_TC89_L1    := 3,
    TCVN_CAM_TC89_KCOS := 4
)DINT;
END_TYPE
```

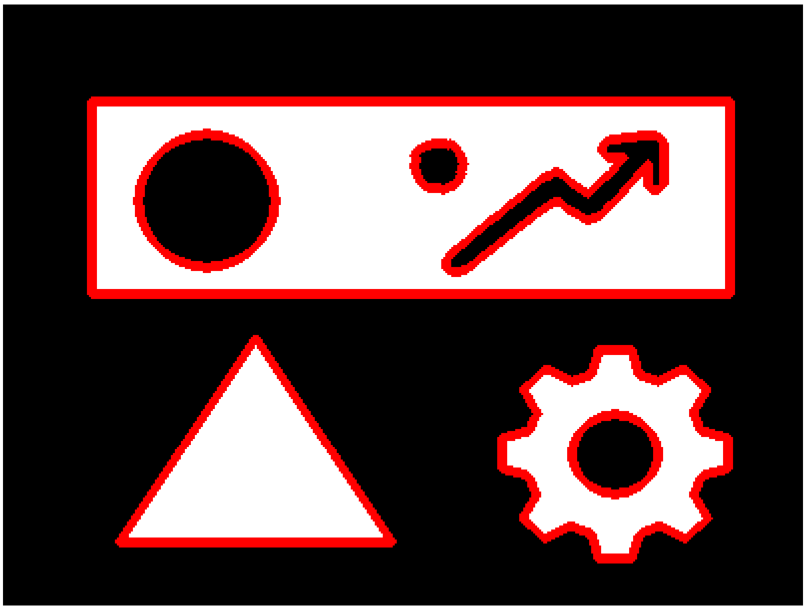
Values

Name	Description
TCVN_CAM_NONE	No approximation, every single point is stored, i.e. two subsequent points are always direct neighbors (horizontal, vertical or diagonal).
TCVN_CAM_SIMPLE	Lossless compression of segments that form straight lines in horizontal, vertical or diagonal direction leaving only their endpoints (For instance, an up-right rectangle is reduced to its four corner points.)
TCVN_CAM_TC89_L1	Compression using a Teh-Chin chain approximation algorithm (IEEE-paper C. Teh, R. Chin, On the Detection of Dominant Points on Digital Curves, 1989).
TCVN_CAM_TC89_KCOS	Compression using a Teh-Chin chain approximation algorithm (IEEE-paper C. Teh, R. Chin, On the Detection of Dominant Points on Digital Curves, 1989).

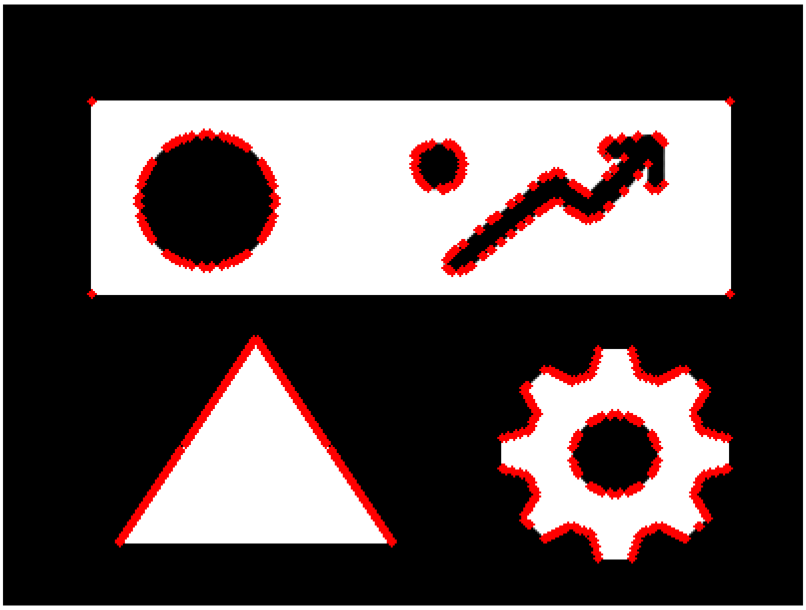
更多信息

这种类型表明是否要简化描述轮廓的点以及如果需要，如何简化。

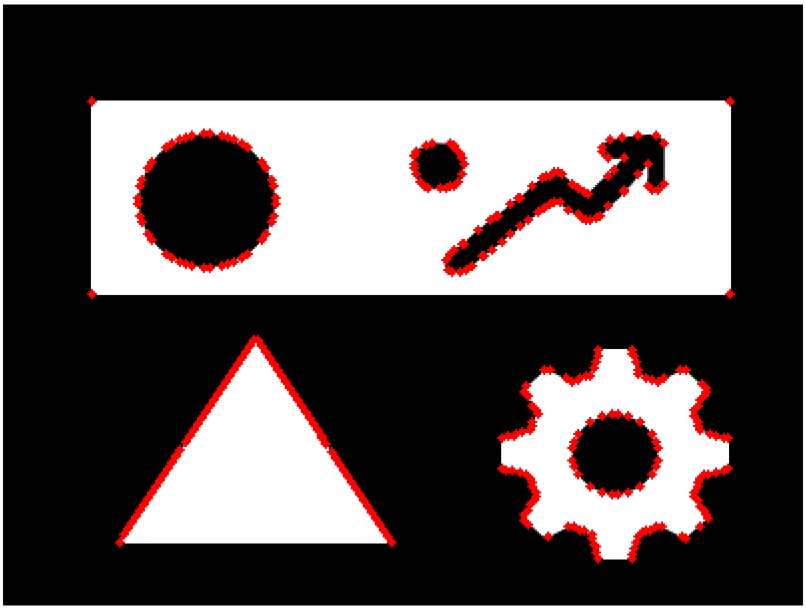
- TCVN_CAM_NONE
  未对轮廓描述进行简化，轮廓由所有包围点描述。



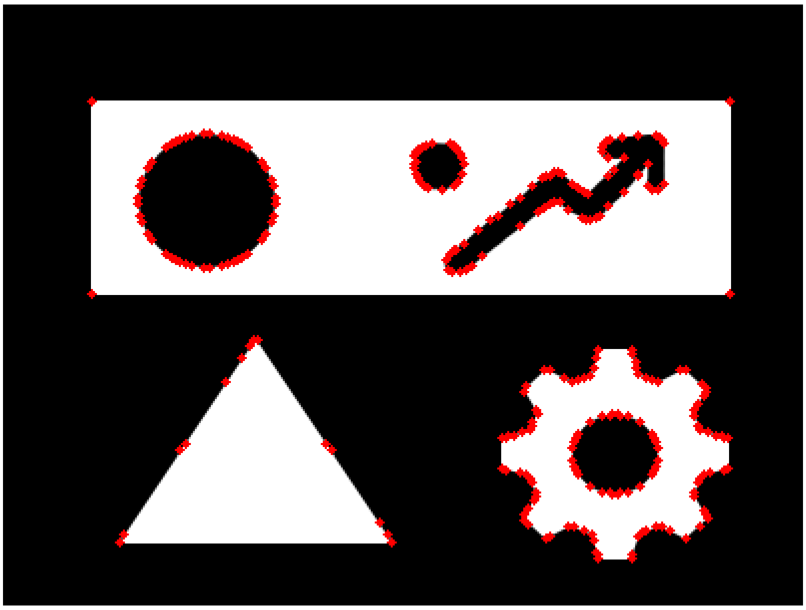
- TCVN_CAM_SIMPLE
  无损简化



- TCVN_CAM_TC89_L1
  使用 Teh-Chin 链近似算法简化轮廓点（IEEE-paper C. Teh, R. Chin, On the Detection of Dominant Points on Digital Curves, 1989）。



- TCVN_CAM_TC89_KCOS
  使用 Teh-Chin 链近似算法简化轮廓点（IEEE-paper C. Teh, R. Chin, On the Detection of Dominant Points on Digital Curves, 1989）。



相关函数

- F_VN_FindContoursExp [▶ 1149]
- F_VN_FindContourHierarchyExp [▶ 1143]

6.1.2.4.19 ETcVnContourRetrievalMode

Offers retrieval modes for a contour search.

Syntax

Definition:

```
TYPE ETcVnContourRetrievalMode :
(
    TCVN_CRM_EXTERNAL                := 0,
    TCVN_CRM_LIST                    := 1,
    TCVN_CRM_CONNECTED_COMPONENTS    := 2,
    TCVN_CRM_TREE                    := 3,
    TCVN_CRM_FLOODFILL               := 4
)DINT;
END_TYPE
```

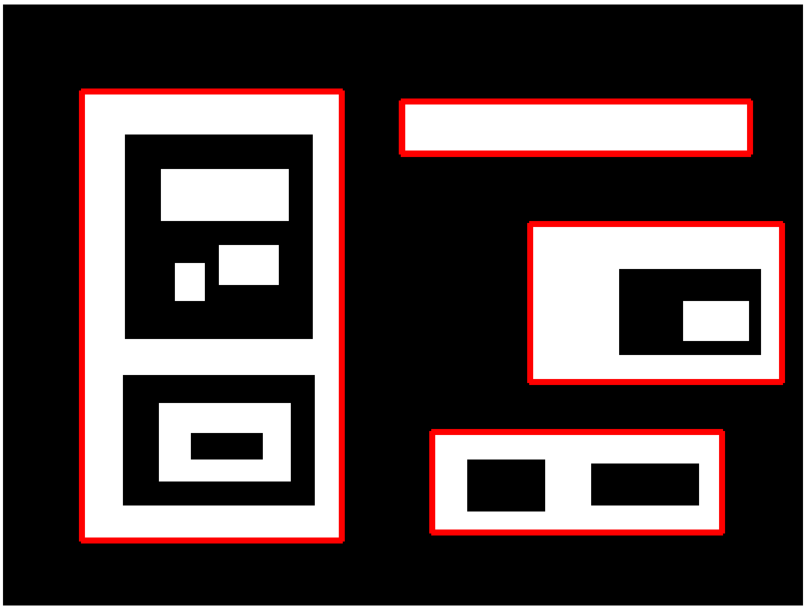
Values

Name	Description
TCVN_CRM_EXTERNAL	Returns only external contours.
TCVN_CRM_LIST	Returns all found contours, not considering their hierarchy.
TCVN_CRM_CONNECTED_COMPONENTS	Returns all contours with a 2-level hierarchy (External contours are assigned level 0, internal contours level 1).
TCVN_CRM_TREE	Returns all contours and their full hierarchy.
TCVN_CRM_FLOODFILL	Returns the found contours using a floodfill algorithm (only available for DINT images).

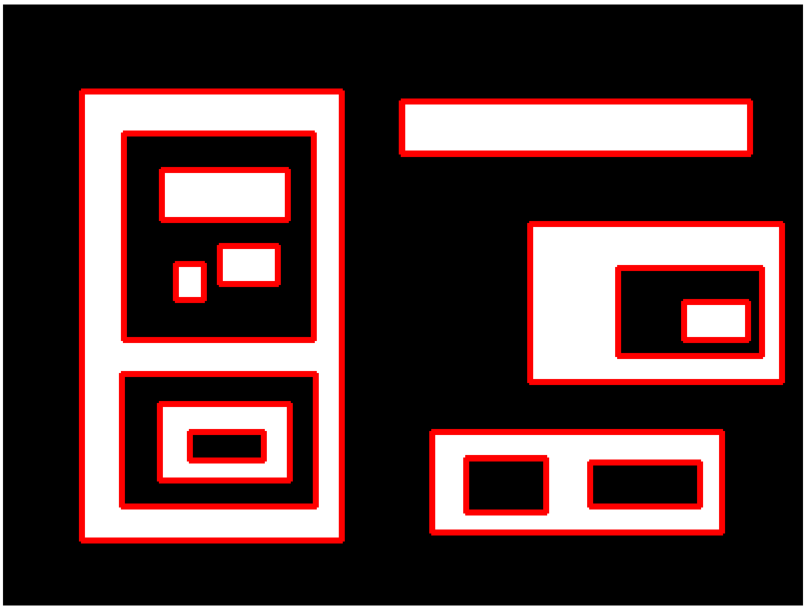
更多信息

该类型指定了在轮廓搜索中如何考虑轮廓层次的问题。

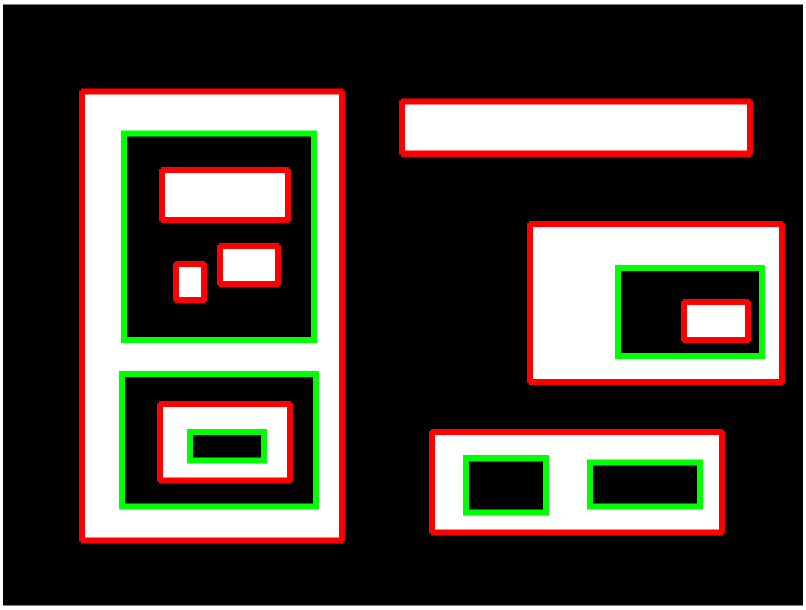
- TCVN_CRM_EXTERNAL
  只返回外部轮廓，不考虑内部轮廓。



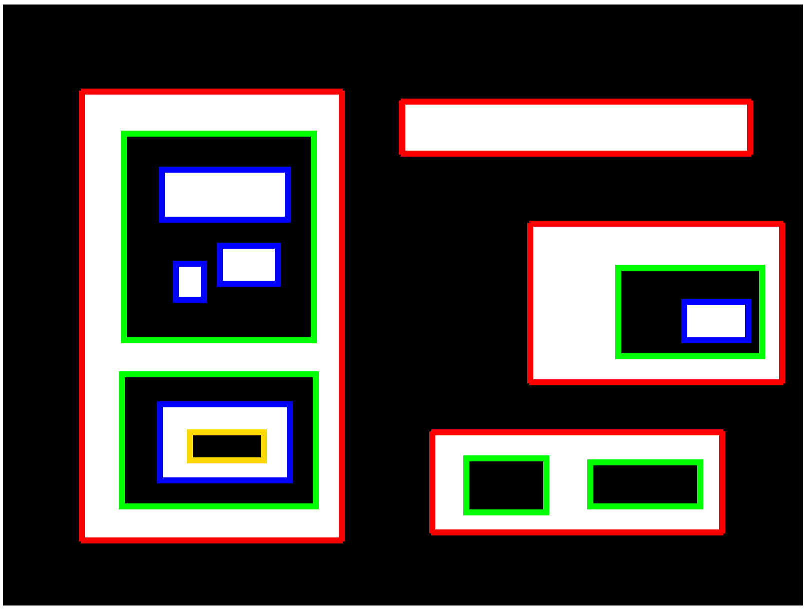
- TCVN_CRM_LIST
  所有轮廓都被返回。轮廓层次未考虑在内。



- TCVN_CRM_CONNECTED_COMPONENTS
  返回所有的轮廓，并创建一个 2 级层次结构。在图形中，1 级的所有轮廓线都显示为红色，2 级显示为绿色。很明显，在 2 级轮廓之后的下一个内部轮廓又是在 1 级。



- TCVN_CRM_TREE
  返回所有的轮廓，并根据树状结构创建一个层次结构。在图像中，1 级是红色，2 级是绿色，3 级是蓝色，4 级是黄色。



相关函数

- F_VN_FindContoursExp [▶ 1149]
- F_VN_FindContourHierarchyExp [▶ 1143]

6.1.2.4.20 ETcVnContoursMatchComparisonMethod

Offers comparison methods for contour matching.

Syntax

Definition:

```
TYPE ETcVnContoursMatchComparisonMethod :
(
    TCVN_CMCM_CONTOURS_MATCH_I1 := 1,
```

```
    TCVN_CMCM_CONTOURS_MATCH_I2 := 2,
    TCVN_CMCM_CONTOURS_MATCH_I3 := 3
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_CMCM_CONTOURS_MATCH_I1	Contours are compared using the sum over the differences between the reciprocal individual characteristics.
TCVN_CMCM_CONTOURS_MATCH_I2	Contours are compared using the sum over the differences between the individual characteristics.
TCVN_CMCM_CONTOURS_MATCH_I3	Contours are compared using only the maximum difference between the individual characteristics.

更多信息

可用方法基于 Hu 不变量。使用以下算法，其中 A 代表第一个轮廓，而 B 代表第二个轮廓。

$$I_1(A,B) = \sum_{i=1..7} \left| \frac{1}{m_i^A} - \frac{1}{m_i^B} \right|$$

$$I_2(A,B) = \sum_{i=1..7} \left| m_i^A - m_i^B \right|$$

$$I_3(A,B) = \max_{i=1..7} \frac{\left| m_i^A - m_i^B \right|}{\left| m_i^A \right|}$$

$$m_i^A = sign\left(h_i^A\right) \cdot \log\left(h_i^A\right)$$

$$m_i^B = sign\left(h_i^B\right) \cdot \log\left(h_i^B\right)$$

h_i^A 和 h_i^B （其中i在[1, 7]之间）是 A 和 B 的 Hu 矩。

相关函数

- F_VN_MatchContours(Exp) [▸ 941]
- F_VN_MatchContours1vsN(Exp) [▸ 943]
- F_VN_MatchImageHuMoments [▸ 1158]

6.1.2.4.21 ETcVnDiffusivityTypeKAZE

Offers diffusivity types for feature detection methods KAZE and AKAZE.

Syntax

Definition:

```
TYPE ETcVnDiffusivityTypeKAZE :
(
    TCVN_DT1_KAZE_PM_G1        := 0,
    TCVN_DT1_KAZE_PM_G2        := 1,
    TCVN_DT1_KAZE_WEICKERT     := 2,
    TCVN_DT1_KAZE_CHARBONNIER  := 3
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_DT1_KAZE_PM_G1	Uses the Perona-Malik diffusivity filter g1.
TCVN_DT1_KAZE_PM_G2	Uses the Perona-Malik diffusivity filter g2.
TCVN_DT1_KAZE_WEICKERT	Uses the Weickert diffusivity filter.
TCVN_DT1_KAZE_CHARBONNIER	Uses the Charbonnier diffusivity filter.

更多信息

枚举ETcVnDiffusivityTypeKAZE用于结构TcVnParamsKAZE [▶ 219]。

相关函数

- F_VN_KeyPointsAndDescriptorsKAZEExp [▶ 1314]

6.1.2.4.22 ETcVnDistanceTransformationLabel

Offers types of the label array to build.

Syntax

Definition:

```
TYPE ETcVnDistanceTransformationLabel :
(
    TCVN_DTL_CCOMP := 0,
    TCVN_DTL_PIXEL := 1
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_DTL_CCOMP	Labels connected components (Each connected component of zeros in the source image and the pixels closest to the connected component will be assigned the same label).
TCVN_DTL_PIXEL	Labels pixels (Each zero pixel and the non-zero pixels closest to it get their own label).

相关函数

- F_VN_DistanceTransformationExp [▶ 1175]

6.1.2.4.23 ETcVnDistanceTransformationMask

Offers different sizes of the distance transformation mask. Please note, that some sizes are not supported by certain distance types.

Syntax

Definition:

```
TYPE ETcVnDistanceTransformationMask :
(
    TCVN_DTM_PRECISE := 0,
    TCVN_DTM_3       := 3,
    TCVN_DTM_5       := 5
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_DTM_PRECISE	Uses a precise mask.
TCVN_DTM_3	Uses a 3x3 mask.
TCVN_DTM_5	Uses a 5x5 mask.

相关函数

- F_VN_DistanceTransformation(Exp) [▸ 1173]

6.1.2.4.24 ETcVnDistanceType

Offers distance types.

Syntax

Definition:

```
TYPE ETcVnDistanceType :
(
    TCVN_DT_USER   := -1,
    TCVN_DT_L1     := 1,
    TCVN_DT_L2     := 2,
    TCVN_DT_C      := 3,
    TCVN_DT_L12    := 4,
    TCVN_DT_FAIR   := 5,
    TCVN_DT_WELSCH := 6,
    TCVN_DT_HUBER  := 7
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_DT_USER	User defined distance
TCVN_DT_L1	|x1 - x2| + |y1 - y2|
TCVN_DT_L2	Euclidean distance
TCVN_DT_C	max(|x1 - x2|, |y1 - y2|)
TCVN_DT_L12	L1 - L2 : 2*(sqrt(1 + x*x/2) - 1))
TCVN_DT_FAIR	c^2(|x|/c-log(1+|x|/c)), c = 1.3998
TCVN_DT_WELSCH	c^2/2(1-exp(-(x/c)^2)), c = 2.9846
TCVN_DT_HUBER	|x|<c ? x^2/2 : c(|x|-c/2), c=1.345

相关函数

- F_VN_DistanceTransformation(Exp) [▸ 1173]
- F_VN_FitLineExp [▸ 939]

6.1.2.4.25 ETcVnDrawMatchesFlags

Offers a combination of flags to support overdrawing an existing destination image and/or skipping single (i.e. non-matched) keypoints and/or drawing additional (rich-)keypoint information (size and orientation). Used by F_VN_DrawKeypointsExp.

Syntax

Definition:

```
TYPE ETcVnDrawMatchesFlags :
(
    TCVN_DMF_DEFAULT                        := 0,
    TCVN_DMF_OVERDRAW                       := 1,
    TCVN_DMF_SKIPSINGLE                     := 2,
```

```
    TCVN_DMF_OVERDRAW_SKIPSINGLE                := 3,
    TCVN_DMF_RICHKEYPOINT                       := 4,
    TCVN_DMF_OVERDRAW_RICHKEYPOINT              := 5,
    TCVN_DMF_SKIPSINGLE_RICHKEYPOINT            := 6,
    TCVN_DMF_OVERDRAW_SKIPSINGLE_RICHKEYPOINT := 7
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_DMF_DEFAULT	Draw all keypoints into a new image.
TCVN_DMF_OVERDRAW	Draw all keypoints into the existing destination image.
TCVN_DMF_SKIPSINGLE	Draw the keypoints into a new image but skip single keypoints.
TCVN_DMF_OVERDRAW_SKIPSINGLE	Draw the keypoints into the existing destination image but skip single keypoints.
TCVN_DMF_RICHKEYPOINT	Draw all keypoints with rich information into a new image.
TCVN_DMF_OVERDRAW_RICHKEYPOINT	Draw all keypoints with rich information into the existing destination image.
TCVN_DMF_SKIPSINGLE_RICHKEYPOINT	Draw the keypoints with rich information into a new image but skip single keypoints.
TCVN_DMF_OVERDRAW_SKIPSINGLE_RICHKEYPOINT	Draw the keypoints with rich information into the existing destination image but skip single keypoints.

相关函数

- F_VN_DrawKeypointsExp [▶ 974]
- F_VN_DrawMatchesExp [▶ 986]

6.1.2.4.26 ETcVnDrawShape

Offers shapes to be drawn.

Syntax

Definition:

```
TYPE ETcVnDrawShape :
(
    TCVN_DS_RANDOM  := -1,
    TCVN_DS_CIRCLE  := 0,
    TCVN_DS_SQUARE  := 1,
    TCVN_DS_PLUS    := 2,
    TCVN_DS_X       := 3,
    TCVN_DS_DIAMOND := 4,
    TCVN_DS_MAX     := 5
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_DS_RANDOM	Randomly select a circle, square, plus, X or diamond.
TCVN_DS_CIRCLE	Circle
TCVN_DS_SQUARE	Square
TCVN_DS_PLUS	+
TCVN_DS_X	x
TCVN_DS_DIAMOND	Diamond
TCVN_DS_MAX	For internal use only, adapted when adding new values.

相关函数

- F_VN_DrawPoint(Exp) [▶ 990]
- F_VN_DrawPoints(Exp) [▶ 992]

6.1.2.4.27 ETcVnEdgeDetectionAlgorithm

Offers edge detection algorithms.

Syntax

Definition:

```
TYPE ETcVnEdgeDetectionAlgorithm :
(
    TCVN_EDA_INTERPOLATION   := 0,
    TCVN_EDA_APPROX_ERF      := 1,
    TCVN_EDA_APPROX_GAUSSIAN := 2
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_EDA_INTERPOLATION	Interpolates pixels (bilinear) and then finds the maximum gradient. This approach is fast and stable, but usually less precise than the function approximation methods.
TCVN_EDA_APPROX_ERF	Approximates the edge with an erf function. This approach is slower than the interpolation method, but usually more precise. However, it can be inaccurate if the edge does not suit the erf model.
TCVN_EDA_APPROX_GAUSSIAN	Approximates the edge with a gaussian function. This method is intended to find the center of relatively thin lines, so it is likely to be inaccurate for other edges.

相关函数

- 测量 [▶ 1428]
    - F_VN_LocateCircularArcExp [▶ 1438]
    - F_VN_LocateEdgeExp [▶ 1444]
    - F_VN_LocateEdgesExp [▶ 1450]
    - F_VN_LocateEllipseExp [▶ 1455]
    - F_VN_MeasureAngleBetweenEdgesExp [▶ 1460]
    - F_VN_MeasureEdgeDistanceExp [▶ 1465]

- ◦ F_VN_MeasureMinEdgeDistanceExp [▶ 1470]

6.1.2.4.28 ETcVnEdgeDirection

Offers edge directions relative to the search direction.

Syntax

Definition:

```
TYPE ETcVnEdgeDirection :
(
    TCVN_ED_DARK_TO_LIGHT := 0,
    TCVN_ED_LIGHT_TO_DARK := 1
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_ED_DARK_TO_LIGHT	Dark to light
TCVN_ED_LIGHT_TO_DARK	Light to dark

相关函数

- 测量 [▶ 1428]

6.1.2.4.29 ETcVnElementType

Offers element types.

Syntax

Definition:

```
TYPE ETcVnElementType :
(
    TCVN_ET_SAME_AS_SOURCE := -1,
    TCVN_ET_USINT          := 0,
    TCVN_ET_SINT           := 1,
    TCVN_ET_UINT           := 2,
    TCVN_ET_INT            := 3,
    TCVN_ET_DINT           := 4,
    TCVN_ET_REAL           := 5,
    TCVN_ET_LREAL          := 6
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_ET_SAME_AS_SOURCE	Sets the element type of the destination image to the source image element type.
TCVN_ET_USINT	USINT (depth: 8 bit, 0..255)
TCVN_ET_SINT	SINT (depth: 8 bit, -128..127)
TCVN_ET_UINT	UINT (depth: 16 bit, 0..65535)
TCVN_ET_INT	INT (depth: 16 bit, -32768..32767)
TCVN_ET_DINT	DINT (depth: 32 bit, -2147483648..2147483647)
TCVN_ET_REAL	REAL (depth: 32 bit, ˜ -3.402823E 10ˆ38 .. ˜ 3.402823E 38)
TCVN_ET_LREAL	LREAL (depth: 64 bit, ˜ -1.79769313486231E 308 .. ˜ 1.79769313486232E 308)

更多信息

枚举ETcVnElementType主要用于指定图像的像素类型或位深。此外，它还包含在结构TcVnMatrix [▶ 211]中。

相关函数

- F_VN_ConvertElementType(Exp) [▶ 747]
- F_VN_CreateImage [▶ 755]
- F_VN_CreateImageAndSetPixels [▶ 756]
- F_VN_CreateImageFromArray [▶ 758]
- F_VN_CustomFilter(Exp) [▶ 1242]
- F_VN_InitMatrixStruct [▶ 1484]
- F_VN_LaplacianFilter(Exp) [▶ 1254]
- F_VN_NormalizeImageExp [▶ 1208]
- F_VN_ScharrFilter(Exp) [▶ 1268]
- F_VN_SeparableCustomFilter(Exp) [▶ 1272]
- F_VN_SobelFilter(Exp) [▶ 1276]

6.1.2.4.30 ETcVnEstimationAlgorithm

Offers estimation algorithms for matching point sets.

Syntax

Definition:

```
TYPE ETcVnEstimationAlgorithm :
(
    TCVN_EA_DEFAULT := 0,
    TCVN_EA_LMEDS   := 4,
    TCVN_EA_RANSAC  := 8,
    TCVN_EA_RHO     := 16
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_EA_DEFAULT	Use all points.
TCVN_EA_LMEDS	Least Median of Squares
TCVN_EA_RANSAC	Random Sample Consensus
TCVN_EA_RHO	Progressive Sample Consensus

相关函数

- F_VN_FindReferenceKeyPointsInImageORBExp [▶ 1303]
- F_VN_HomographyExp [▶ 1108]

6.1.2.4.31 ETcVnExtremePointDirection

Offers search directions for the extreme point.

Syntax

Definition:

```
TYPE ETcVnExtremePointDirection :
(
    TCVN_EPD_TOP_LEFT       := 0,
    TCVN_EPD_TOP_MEDIAN     := 1,
    TCVN_EPD_TOP_RIGHT      := 2,
```

```
    TCVN_EPD_BOTTOM_LEFT   := 3,
    TCVN_EPD_BOTTOM_MEDIAN := 4,
    TCVN_EPD_BOTTOM_RIGHT  := 5,
    TCVN_EPD_LEFT_TOP      := 6,
    TCVN_EPD_LEFT_MEDIAN   := 7,
    TCVN_EPD_LEFT_BOTTOM   := 8,
    TCVN_EPD_RIGHT_TOP     := 9,
    TCVN_EPD_RIGHT_MEDIAN  := 10,
    TCVN_EPD_RIGHT_BOTTOM  := 11
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_EPD_TOP_LEFT	Find the topmost point (min y, take left one if more than 1).
TCVN_EPD_TOP_MEDIAN	Find the topmost point (min y, take median one if more than 1).
TCVN_EPD_TOP_RIGHT	Find the topmost point (min y, take right one if more than 1).
TCVN_EPD_BOTTOM_LEFT	Find the bottommost point (max y, take left one if more than 1).
TCVN_EPD_BOTTOM_MEDIAN	Find the bottommost point (max y, take median one if more than 1).
TCVN_EPD_BOTTOM_RIGHT	Find the bottommost point (max y, take right one if more than 1).
TCVN_EPD_LEFT_TOP	Find the leftmost point (min x, take top one if more than 1).
TCVN_EPD_LEFT_MEDIAN	Find the leftmost point (min x, take median one if more than 1).
TCVN_EPD_LEFT_BOTTOM	Find the leftmost point (min x, take bottom one if more than 1).
TCVN_EPD_RIGHT_TOP	Find the rightmost point (max x, take top one if more than 1).
TCVN_EPD_RIGHT_MEDIAN	Find the rightmost point (max x, take median one if more than 1).
TCVN_EPD_RIGHT_BOTTOM	Find the rightmost point (max x, take bottom one if more than 1).

相关函数

- F_VN_ContourExtremePoint [▶ 922]

6.1.2.4.32 ETcVnFeatureDescriptorTypeAKAZE

Offers descriptor types for AKAZE method.

Syntax

Definition:

```
TYPE ETcVnFeatureDescriptorTypeAKAZE :
(
    TCVN_FDT_AKAZE_KAZE_UPRIGHT := 2,
    TCVN_FDT_AKAZE_KAZE         := 3,
    TCVN_FDT_AKAZE_MLDB_UPRIGHT := 4,
    TCVN_FDT_AKAZE_MLDB         := 5
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_FDT_AKAZE_KAZE_UPRIGHT	KAZE descriptor (upright, i.e. rotation dependend).
TCVN_FDT_AKAZE_KAZE	KAZE descriptor (rotation invariant).
TCVN_FDT_AKAZE_MLDB_UPRIGHT	MLDB descriptor (upright, i.e. rotation dependend).
TCVN_FDT_AKAZE_MLDB	MLDB descriptor (rotation invariant).

更多信息

枚举ETcVnFeatureDescriptorTypAKAZE用于结构TcVnParamsAKAZE [▸ 213]。

6.1.2.4.33 ETcVnFeatureScalingType

Offers feature scaling types

Syntax

Definition:

```
TYPE ETcVnFeatureScalingType :
(
    TCVN_FST1_MAXABS          := 0,
    TCVN_FST1_MINMAX          := 1,
    TCVN_FST1_STANDARDIZATION := 2
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_FST1_MAXABS	Use the max of the absolute values of the feature vector. The normlized values are in the range [-1 to 1] or [0 to 1] if all the original values are positive.
TCVN_FST1_MINMAX	Use the min and max values of the feature vector. The normlized values are in the range [0 to 1]
TCVN_FST1_STANDARDIZATION	Use the mean and the standard deviation of the feature vector. The normalized vector has a mean value equal to zero and standard deviation equal to one

6.1.2.4.34 ETcVnFeatureScoreTypeORB

Offers algorithms used to rank features.

Syntax

Definition:

```
TYPE ETcVnFeatureScoreTypeORB :
(
    TCVN_FST_ORB_HARRIS := 0,
    TCVN_FST_ORB_FAST   := 1
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_FST_ORB_HARRIS	Harris algorithm (produces more stable keypoints, but computation takes longer).
TCVN_FST_ORB_FAST	FAST algorithm (produces slightly less stable keypoints, but computation is faster).

更多信息

枚举ETcVnFeatureScoreTypeORB用于结构TcVnParamsORB [▶ 220]。

6.1.2.4.35 ETcVnFilterDirection

Offers directions, in which to apply filter.

Syntax

Definition:

```
TYPE ETcVnFilterDirection :
(
    TCVN_FD_X := 0,
    TCVN_FD_Y := 1
)INT;
END_TYPE
```

Values

Name	Description
TCVN_FD_X	X-direction
TCVN_FD_Y	Y-direction

相关函数

- F_VN_ScharrFilter(Exp) [▶ 1268]

6.1.2.4.36 ETcVnFlipAxis

Defines the axis around which to flip (mirror) the image.

Syntax

Definition:

```
TYPE ETcVnFlipAxis :
(
    TCVN_FA_XY := -1,
    TCVN_FA_X  := 0,
    TCVN_FA_Y  := 1
)INT;
END_TYPE
```

Values

Name	Description
TCVN_FA_XY	Flip the image around both axes
TCVN_FA_X	Flip the image around the X axis
TCVN_FA_Y	Flip the image around the Y axis

相关函数

- F_VN_FlipImage [▶ 1101]

6.1.2.4.37 ETcVnFontType

Offers font types.

Syntax

Definition:

```
TYPE ETcVnFontType :
(
    TCVN_FT_HERSHEY_SIMPLEX              := 0,
    TCVN_FT_HERSHEY_PLAIN                := 1,
    TCVN_FT_HERSHEY_DUPLEX               := 2,
    TCVN_FT_HERSHEY_COMPLEX              := 3,
    TCVN_FT_HERSHEY_TRIPLEX              := 4,
    TCVN_FT_HERSHEY_COMPLEX_SMALL        := 5,
    TCVN_FT_HERSHEY_SCRIPT_SIMPLEX       := 6,
    TCVN_FT_HERSHEY_SCRIPT_COMPLEX       := 7,
    TCVN_FT_HERSHEY_PLAIN_ITALIC         := 17,
    TCVN_FT_HERSHEY_COMPLEX_ITALIC       := 19,
    TCVN_FT_HERSHEY_TRIPLEX_ITALIC       := 20,
    TCVN_FT_HERSHEY_COMPLEX_SMALL_ITALIC := 21
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_FT_HERSHEY_SIMPLEX	Normal size sans-serif font.
TCVN_FT_HERSHEY_PLAIN	Small size sans-serif font.
TCVN_FT_HERSHEY_DUPLEX	More complex normal size sans-serif font.
TCVN_FT_HERSHEY_COMPLEX	Normal size serif font.
TCVN_FT_HERSHEY_TRIPLEX	More complex normal size serif font.
TCVN_FT_HERSHEY_COMPLEX_SMALL	Small size serif font.
TCVN_FT_HERSHEY_SCRIPT_SIMPLEX	Hand-writing style font.
TCVN_FT_HERSHEY_SCRIPT_COMPLEX	More complex hand-writing style font.
TCVN_FT_HERSHEY_PLAIN_ITALIC	Small size sans-serif font (italic).
TCVN_FT_HERSHEY_COMPLEX_ITALIC	Normal size serif font (italic).
TCVN_FT_HERSHEY_TRIPLEX_ITALIC	More complex normal size serif font (italic).
TCVN_FT_HERSHEY_COMPLEX_SMALL_ITALIC	Small size serif font (italic).

更多信息

以下字体可用：

相关函数

- F_VN_PutLabelExp [▶ 1011]
- F_VN_PutText(Exp) [▶ 1012]

6.1.2.4.38 ETcVnHoughMethod

Offers Hough methods.

Syntax

Definition:

```
TYPE ETcVnHoughMethod :
(
    TCVN_HM_STANDARD        := 0,
    TCVN_HM_PROBABILISTIC := 1,
    TCVN_HM_MULTI_SCALE     := 2,
    TCVN_HM_GRADIENT        := 3,
    TCVN_HM_GRADIENT_ALT    := 4
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_HM_STANDARD	Classical or standard Hough transform. Every line is represented by two floating-point numbers (f1, f2), where f1 is a distance between (0,0) origin and the line, and f2 is the angle between x-axis and the normal to the line.
TCVN_HM_PROBABILISTIC	Probabilistic Hough transform. More efficient for pictures containing long linear segments. Returns line segments rather than the whole line, while each segment is represented by start and end point.
TCVN_HM_MULTI_SCALE	Multi scale variant of classical Hough transform.
TCVN_HM_GRADIENT	Method designed to find circles. See paper: HK Yuen, John Princen, John Illingworth, and Josef Kittler. Comparative study of hough transform methods for circle finding. Image and Vision Computing, 8(1):71 - 77, 1990
TCVN_HM_GRADIENT_ALT	Variation of the GRADIENT method.

相关函数

- F_VN_HoughCirclesExp [▶ 1152]

6.1.2.4.39 ETcVnInitializableFunction

Offers initializable functions.

Syntax

Definition:

```
TYPE ETcVnInitializableFunction :
(
    TCVN_IF_OCR := 0,
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_IF_OCR	F_VN_OCR

6.1.2.4.40 ETcVnInterpolationType

Offers interpolation types.

Syntax

Definition:

```
TYPE ETcVnInterpolationType :
(
    TCVN_IT_NEAREST_NEIGHBOR := 0,
    TCVN_IT_BILINEAR         := 1,
    TCVN_IT_BICUBIC          := 2,
    TCVN_IT_AREA_BASED       := 3,
    TCVN_IT_LANCZOS4         := 4
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_IT_NEAREST_NEIGHBOR	Nearest neighbor interpolation
TCVN_IT_BILINEAR	Bilinear interpolation
TCVN_IT_BICUBIC	Bicubic interpolation
TCVN_IT_AREA_BASED	Area based interpolation
TCVN_IT_LANCZOS4	Lanczos4 interpolation

相关函数

- F_VN_MatchTemplateAndEvaluateExp [▶ 1161]
- F_VN_RemapImageToLogPolarSpaceExp [▶ 1114]
- F_VN_RemapImageToPolarSpaceExp [▶ 1117]
- F_VN_ResizeImage [▶ 1120]
- F_VN_WarpAffineExp [▶ 1129]
- F_VN_WarpPerspectiveExp [▶ 1135]

6.1.2.4.41 ETcVnKeypointDetectionTypeAGAST

Offers different neighborhood types for AGAST method (For details see paper: E. Mair et al.: Adaptive and Generic Corner Detection Based on the Accelerated Segment Test, 2010).

Syntax

Definition:

```
TYPE ETcVnKeypointDetectionTypeAGAST :
(
    TCVN_KDT_AGAST_5_8   := 0,
    TCVN_KDT_AGAST_7_12d := 1,
    TCVN_KDT_AGAST_7_12s := 2,
    TCVN_KDT_AGAST_9_16  := 3
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_KDT_AGAST_5_8	5 of 8
TCVN_KDT_AGAST_7_12d	7 of 12 diamond
TCVN_KDT_AGAST_7_12s	7 of 12 square
TCVN_KDT_AGAST_9_16	9 of 16

更多信息

枚举ETcVnKeypointDetectionTypeAGAST用于结构TcVnParamsAGAST [▶ 212]。

6.1.2.4.42 ETcVnKeypointDetectionTypeFAST

Offers different neighborhood types for FAST method (For details see paper: E. Rosten: Machine Learning for High-speed Corner Detection, 2006).

Syntax

Definition:

```
TYPE ETcVnKeypointDetectionTypeFAST :
(
    TCVN_KDT_FAST_5_8  := 0,
    TCVN_KDT_FAST_7_12 := 1,
```

```
    TCVN_KDT_FAST_9_16 := 2
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_KDT_FAST_5_8	5 of 8
TCVN_KDT_FAST_7_12	7 of 12
TCVN_KDT_FAST_9_16	9 of 16

更多信息

枚举ETcVnKeypointDetectionTypeFAST用于结构TcVnParamsFAST [▶ 218]。

6.1.2.4.43 ETcVnKnn

Offers kNN model types.

Syntax

Definition:

```
TYPE ETcVnKnn :
(
    TCVN_KNN_CLASSIFIER       := 1,
    TCVN_KNN_NOVELTY_DETECTOR := 2,
    TCVN_KNN_REGRESSOR        := 4
)UDINT;
END_TYPE
```

Values

Name	Description
TCVN_KNN_CLASSIFIER	Classifier
TCVN_KNN_NOVELTY_DETECTOR	Novelty detector
TCVN_KNN_REGRESSOR	Regressor

6.1.2.4.44 ETcVnLineType

Offers line types.

Syntax

Definition:

```
TYPE ETcVnLineType :
(
    TCVN_LT_4_CONNECTED := 4,
    TCVN_LT_8_CONNECTED := 8,
    TCVN_LT_ANTIALIASED := 16
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_LT_4_CONNECTED	4-connected (pixels are connected horizontally and vertically).
TCVN_LT_8_CONNECTED	8-connected (pixels are connected horizontally, vertically and diagonally).
TCVN_LT_ANTIALIASED	Antialiased (drawn using Gaussian filtering, only implemented for 8-bit images).

相关函数

- F_VN_DrawCircleExp [▸ 960]
- F_VN_DrawCirclesExp [▸ 962]
- F_VN_DrawCircularArcExp [▸ 964]
- F_VN_DrawContoursExp [▸ 969]
- F_VN_DrawEllipseExp [▸ 972]
- F_VN_DrawLineExp [▸ 978]
- F_VN_DrawLineExp_TcVnVector4_DINT [▸ 980]
- F_VN_DrawLineExp_TcVnVector4_LREAL [▸ 981]
- F_VN_DrawLinesExp [▸ 983]
- F_VN_DrawPointExp [▸ 991]
- F_VN_DrawPointsExp [▸ 993]
- F_VN_DrawRotatedRectangleExp [▸ 1000]
- F_VN_PutLabelExp [▸ 1011]
- F_VN_PutTextExp [▸ 1014]

6.1.2.4.45 ETcVnMorphologicalOperator

Offers morphological operators.

Syntax

Definition:

```
TYPE ETcVnMorphologicalOperator :
(
    TCVN_MO_EROSION                         := 0x0,
    TCVN_MO_DILATION                        := 0x1,
    TCVN_MO_OPENING                         := 0x2,
    TCVN_MO_CLOSING                         := 0x3,
    TCVN_MO_GRADIENT                        := 0x4,
    TCVN_MO_WHITE_TOPHAT                    := 0x5,
    TCVN_MO_BLACK_TOPHAT                    := 0x6,
    TCVN_MO_OPENING_BY_RECONSTRUCTION       := 0x40000002,
    TCVN_MO_CLOSING_BY_RECONSTRUCTION       := 0x40000003,
    TCVN_MO_WHITE_TOPHAT_BY_RECONSTRUCTION := 0x40000005,
    TCVN_MO_BLACK_TOPHAT_BY_RECONSTRUCTION := 0x40000006
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_MO_EROSION	Shrinks objects (regions of foreground (i.e. white) pixels), removing regions smaller than the structuring element completely.
TCVN_MO_DILATION	Expands objects and thereby closes small holes inside objects.
TCVN_MO_OPENING	Applies an erosion first, a dilation second. Objects smaller than the structuring element are removed while outer shapes remain largely the same.
TCVN_MO_CLOSING	Applies a dilation first, an erosion second. Holes inside objects that fit into the structuring element are closed completely while outer shapes remain largely the same.
TCVN_MO_GRADIENT	Difference between the dilation and the erosion of an image.
TCVN_MO_WHITE_TOPHAT	Difference between an input image and its opening.
TCVN_MO_BLACK_TOPHAT	Difference between an input image and its closing.
TCVN_MO_OPENING_BY_RECONSTRUCTION	Opening with subsequent reconstruction of objects that were not removed by the opening.
TCVN_MO_CLOSING_BY_RECONSTRUCTION	Closing with subsequent reconstruction of objects that were not removed by the closing.
TCVN_MO_WHITE_TOPHAT_BY_RECONSTRUCTION	White tophat with subsequent reconstruction of objects that were not removed by the white tophat.
TCVN_MO_BLACK_TOPHAT_BY_RECONSTRUCTION	Black tophat with subsequent reconstruction of objects that were not removed by the black tophat.

相关函数

- F_VN_MorphologicalOperator [▶ 1264]

6.1.2.4.46 ETcVnNbc

Offers normal Bayes classifier types.

Syntax

Definition:

```
TYPE ETcVnNbc :
(
    TCVN_NBC_CLASSIFIER        := 1,
    TCVN_NBC_NOVELTY_DETECTOR := 2
)UDINT;
END_TYPE
```

Values

Name	Description
TCVN_NBC_CLASSIFIER	Classifier
TCVN_NBC_NOVELTY_DETECTOR	Novelty detector

6.1.2.4.47 ETcVnNormalizationType

Offers normalization types.

Syntax

Definition:

```
TYPE ETcVnNormalizationType :
(
    TCVN_NT_INF                := 1,
    TCVN_NT_L1                 := 2,
    TCVN_NT_L2                 := 4,
    TCVN_NT_L2SQR              := 5,
    TCVN_NT_HAMMING            := 6,
    TCVN_NT_HAMMING2           := 7,
    TCVN_NT_RELATIVE_INF       := 9,
    TCVN_NT_RELATIVE_L1        := 10,
    TCVN_NT_RELATIVE_L2        := 12,
    TCVN_NT_RELATIVE_L2SQR     := 13,
    TCVN_NT_RELATIVE_HAMMING   := 14,
    TCVN_NT_RELATIVE_HAMMING2 := 15,
    TCVN_NT_MINMAX             := 32
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_NT_INF	Infinity norm
TCVN_NT_L1	L1 norm
TCVN_NT_L2	L2 (euclidean) norm
TCVN_NT_L2SQR	Squared L2 norm
TCVN_NT_HAMMING	Hamming distance (bitwise)
TCVN_NT_HAMMING2	Hamming distance (each 2 bits combined to one)
TCVN_NT_RELATIVE_INF	Infinity norm (when comparing 2 arrays, the norm of their difference is divided by the norm of the second array)
TCVN_NT_RELATIVE_L1	L1 norm (when comparing 2 arrays, the norm of their difference is divided by the norm of the second array)
TCVN_NT_RELATIVE_L2	L2 (euclidean) norm (when comparing 2 arrays, the norm of their difference is divided by the norm of the second array)
TCVN_NT_RELATIVE_L2SQR	Squared L2 norm (when comparing 2 arrays, the norm of their difference is divided by the norm of the second array)
TCVN_NT_RELATIVE_HAMMING	Hamming distance (bitwise; when comparing 2 arrays, the norm of their difference is divided by the norm of the second array)
TCVN_NT_RELATIVE_HAMMING2	Hamming distance (each 2 bits combined to one; when comparing 2 arrays, the norm of their difference is divided by the norm of the second array)
TCVN_NT_MINMAX	Normalize the values to a range given by a minimum and a maximum value.

相关函数

- F_VN_FindReferenceKeyPointsInImageORBExp [▶ 1305]
- F_VN_MatchDescriptorsBFExp [▶ 1326]
- F_VN_MatchDescriptorsKnnBFExp [▶ 1330]
- F_VN_NormalizeImageExp [▶ 1208]

6.1.2.4.48 ETcVnOcrModelType

Offers different OCR model types

Syntax

Definition:

```
TYPE ETcVnOcrModelType :
(
    TCVN_OMT_NUMBERS                := 580542139465730,
    TCVN_OMT_NUMBERS_SC             := 580542139465732,
    TCVN_OMT_UCLETTERS              := 580542139465736,
    TCVN_OMT_NUMBERS_SC_UCLETTERS := 580542139465744
)ULINT;
END_TYPE
```

Values

Name	Description
TCVN_OMT_NUMBERS	Classify numbers
TCVN_OMT_NUMBERS_SC	Classify numbers and special characters
TCVN_OMT_UCLETTERS	Classify uppercase letters
TCVN_OMT_NUMBERS_SC_UCLETTERS	Classify numbers, special characters and uppercase letters

6.1.2.4.49 ETcVnOcrOptions

Provides different options to indicate whether certain actions must be performed to achieve the intended result.

Syntax

Definition:

```
TYPE ETcVnOcrOptions :
(
    TCVN_OO_NONE        := 0,
    TCVN_OO_WITHBLANKS := 1
)UDINT;
END_TYPE
```

Values

Name	Description
TCVN_OO_NONE	No options
TCVN_OO_WITHBLANKS	Indicates that if blanks were found, they should be included in the result. By default, all blanks are omitted from the result.

6.1.2.4.50 ETcVnOrientationMethod

Offers methods to calculate the orientation of a set of points.

Syntax

Definition:

```
TYPE ETcVnOrientationMethod :
(
    TCVN_OM_PCA             := 0,
    TCVN_OM_FITELLIPSE      := 1,
    TCVN_OM_MOMENTS         := 2,
    TCVN_OM_ENCLOSINGRECT := 3
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_OM_PCA	Apply the PCA on the points and calculate the mean point as the center point and the rotation angle of the main principal axis
TCVN_OM_FITELLIPSE	Fit an ellipse around the points and calculate the center point and the rotation angle of the main axis
TCVN_OM_MOMENTS	Calculate the moments of the points and estimate the center and the rotation angle
TCVN_OM_ENCLOSINGRECT	Calculate a minimum area rectangle enclosing the points and calculate the center point and the rotation angle of the rectangle

6.1.2.4.51 ETcVnPaddingMode

Offers padding modes.

Syntax

Definition:

```
TYPE ETcVnPaddingMode :
(
    TCVN_PM_NONE        := 0,
    TCVN_PM_CROP_CENTER := 1,
    TCVN_PM_LETTERBOX   := 2
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_PM_NONE	Resize the image to the desired size without extra processing
TCVN_PM_CROP_CENTER	Resize and crop the image
TCVN_PM_LETTERBOX	Resize the image to the desired size while preserving the aspect ratio of the original image

6.1.2.4.52 ETcVnPixelConnectivity

Offers pixel connectivities.

Syntax

Definition:

```
TYPE ETcVnPixelConnectivity :
(
    TCVN_PC_4 := 4,
    TCVN_PC_8 := 8
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_PC_4	4-way
TCVN_PC_8	8-way

6.1.2.4.53 ETcVnPixelEncoding

Offers pixel encodings.

Syntax

Definition:

```
TYPE ETcVnPixelEncoding :
(
    TCVN_PE_NONE                := 0,
    TCVN_PE_BAYER_GR            := 1,
    TCVN_PE_BAYER_RG            := 2,
    TCVN_PE_BAYER_GB            := 3,
    TCVN_PE_BAYER_BG            := 4,
    TCVN_PE_YUV_411_UYYVYY      := 5,
    TCVN_PE_YUV_422_UYVY        := 6,
    TCVN_PE_YUV_422_YUYV        := 7,
    TCVN_PE_YCBCR_411_CBYYCRYY := 8,
    TCVN_PE_YCBCR_422_CBYCRY    := 9,
    TCVN_PE_YCBCR_422_YCBYCR    := 10
)BYTE;
END_TYPE
```

Values

Name	Description
TCVN_PE_NONE	No encoding available, i.e. every pixel value is independent of other pixels.
TCVN_PE_BAYER_GR	Pixels are encoded as a BayerGR pattern.
TCVN_PE_BAYER_RG	Pixels are encoded as a BayerRG pattern.
TCVN_PE_BAYER_GB	Pixels are encoded as a BayerGB pattern.
TCVN_PE_BAYER_BG	Pixels are encoded as a BayerBG pattern.
TCVN_PE_YUV_411_UYYVYY	Pixels are encoded as YUV411 (UYYVYY).
TCVN_PE_YUV_422_UYVY	Pixels are encoded as YUV422 (UYVY).
TCVN_PE_YUV_422_YUYV	Pixels are encoded as YUV422 (YUYV).
TCVN_PE_YCBCR_411_CBYYCRYY	Pixels are encoded as YCbCr411 (CbYYCrYY).
TCVN_PE_YCBCR_422_CBYCRY	Pixels are encoded as YCbCr422 (CbYCrY).
TCVN_PE_YCBCR_422_YCBYCR	Pixels are encoded as YCbCr422 (YCbYCr).

更多信息

枚举ETcVnPixelEncoding用于结构TcVnPixelFormat [▸ 224]。关于像素编码的信息，可以通过查询图像的像素格式信息获得。常见使用案例是查询 Bayer 模式，以了解一张图片是否还需要转换为 RGB 格式等。

6.1.2.4.54 ETcVnPixelPackMode

Offers pixel packing modes.

Syntax

Definition:

```
TYPE ETcVnPixelPackMode :
(
    TCVN_PPM_NONE            := 0,
    TCVN_PPM_MONO1P          := 1,
    TCVN_PPM_MONO2P          := 2,
    TCVN_PPM_MONO4P          := 3,
    TCVN_PPM_MONO10PACKED    := 4,
    TCVN_PPM_MONO12PACKED    := 5,
    TCVN_PPM_RGB10V1PACKED := 6,
    TCVN_PPM_RGB10P32        := 7,
    TCVN_PPM_RGB12V1PACKED := 8,
    TCVN_PPM_RGB565P         := 9,
    TCVN_PPM_BGR565P         := 10,
    TCVN_PPM_MONO10P         := 11,
```

```
    TCVN_PPM_MONO12P        := 12,
    TCVN_PPM_MONO14P        := 13
)BYTE;
END_TYPE
```

Values

Name	Description
TCVN_PPM_NONE	No packing
TCVN_PPM_MONO1P	Mono1p
TCVN_PPM_MONO2P	Mono2p
TCVN_PPM_MONO4P	Mono4p
TCVN_PPM_MONO10PACKED	Mono10Packed or BayerXX10Packed
TCVN_PPM_MONO12PACKED	Mono12Packed or BayerXX12Packed
TCVN_PPM_RGB10V1PACKED	RGB10V1Packed
TCVN_PPM_RGB10P32	RGB10V2Packed (RGB10p32)
TCVN_PPM_RGB12V1PACKED	RGB12V1Packed
TCVN_PPM_RGB565P	RGB565p
TCVN_PPM_BGR565P	BGR565p
TCVN_PPM_MONO10P	Mono10p or BayerXX10p
TCVN_PPM_MONO12P	Mono12p or BayerXX12p
TCVN_PPM_MONO14P	Mono14p or BayerXX14p

更多信息

枚举ETcVnPixelPackMode用于结构TcVnPixelFormat [▸ 224]。

6.1.2.4.55 ETcVnPrototypeClusterer

Offers prototype clusterer model types.

Syntax

Definition:

```
TYPE ETcVnPrototypeClusterer :
(
    TCVN_PC_NOVELTY_DETECTOR := 2,
    TCVN_PC_CLUSTERER        := 8
)UDINT;
END_TYPE
```

Values

Name	Description
TCVN_PC_NOVELTY_DETECTOR	Novelty detector
TCVN_PC_CLUSTERER	Clusterer

6.1.2.4.56 ETcVnRectangleIntersection

Offers rectangle intersection types.

Syntax

Definition:

```
TYPE ETcVnRectangleIntersection :
(
    TCVN_RI_NONE    := 0,
    TCVN_RI_PARTIAL := 1,
```

```
    TCVN_RI_FULL    := 2
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_RI_NONE	No intersection
TCVN_RI_PARTIAL	Partial intersection
TCVN_RI_FULL	Full intersection

6.1.2.4.57 ETcVnRotationAngle

Offers rotation angles.

Syntax

Definition:

```
TYPE ETcVnRotationAngle :
(
    TCVN_RA_90_DEG  := 0,
    TCVN_RA_180_DEG := 1,
    TCVN_RA_270_DEG := 2
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_RA_90_DEG	90 degrees
TCVN_RA_180_DEG	180 degrees
TCVN_RA_270_DEG	270 degrees

相关函数

- F_VN_RotateImage [▶ 1122]

6.1.2.4.58 ETcVnRTrees

Offers RTrees model types.

Syntax

Definition:

```
TYPE ETcVnRTrees :
(
    TCVN_RT_CLASSIFIER := 1,
    TCVN_RT_REGRESSOR  := 4
)UDINT;
END_TYPE
```

Values

Name	Description
TCVN_RT_CLASSIFIER	Classifier
TCVN_RT_REGRESSOR	Regressor

6.1.2.4.59 ETcVnSignedNormalization

Offers options for normalizing images with signed data types.

Syntax

Definition:

```
TYPE ETcVnSignedNormalization :
(
    TCVN_SN_FIX_ZERO   := 0,
    TCVN_SN_FULL_SCALE := 1
)INT;
END_TYPE
```

Values

Name	Description
TCVN_SN_FIX_ZERO	The value 0 is fixed, i.e. the normalized image might only reach either the minimum or maximum value.
TCVN_SN_FULL_SCALE	The image is normalized to the full value range, i.e. a prior value of 0 might be != 0 after normalization.

相关函数

- F_VN_NormalizeImageForDisplay [▶ 1209]

6.1.2.4.60 ETcVnSolvePnPMethod

Offers SolvePnP methods.

Syntax

Definition:

```
TYPE ETcVnSolvePnPMethod :
(
    TCVN_SPM_ITERATIVE   := 0,
    TCVN_SPM_EPNP        := 1,
    TCVN_SPM_P3P         := 2,
    TCVN_SPM_AP3P        := 5,
    TCVN_SPM_IPPE        := 6,
    TCVN_SPM_IPPE_SQUARE := 7,
    TCVN_SPM_SQPNP       := 8
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_SPM_ITERATIVE	Levenberg-Marquardt optimization. Requires at least 4 planar or 6 non-planar reference points.
TCVN_SPM_EPNP	Efficient perspective-n-point camera pose estimation. Requires at least 4 reference points.
TCVN_SPM_P3P	Complete solution classification for the perspective-three-point problem. Requires exactly 4 reference points to find a unique solution.
TCVN_SPM_AP3P	An efficient algebraic solution to the perspective-three-point problem. Requires exactly 4 reference points to find a unique solution.
TCVN_SPM_IPPE	Infinitesimal plane-based pose estimation. Requires at least 4 coplanar reference points.
TCVN_SPM_IPPE_SQUARE	IPPE specialization for squares. Requires exactly 4 reference points in the following order (hsl: half square length): (-hsl, hsl, 0), (hsl, hsl, 0), (hsl, -hsl, 0), (-hsl, -hsl, 0)
TCVN_SPM_SQPNP	A consistently fast and globally optimal solution to the perspective-n-point problem. Requires at least 4 reference points.

6.1.2.4.61 ETcVnSta

Offers Simplified TopoART model types.

Syntax

Definition:

```
TYPE ETcVnSta :
(
    TCVN_STA_CLASSIFIER        := 1,
    TCVN_STA_NOVELTY_DETECTOR := 2,
    TCVN_STA_REGRESSOR         := 4,
    TCVN_STA_CLUSTERER         := 8
)UDINT;
END_TYPE
```

Values

Name	Description
TCVN_STA_CLASSIFIER	Classifier
TCVN_STA_NOVELTY_DETECTOR	Novelty detector
TCVN_STA_REGRESSOR	Regressor
TCVN_STA_CLUSTERER	Clusterer

6.1.2.4.62 ETcVnStructuringElementShape

Offers shapes for a structuring element.

Syntax

Definition:

```
TYPE ETcVnStructuringElementShape :
(
    TCVN_SES_RECTANGLE := 0,
    TCVN_SES_CROSS     := 1,
    TCVN_SES_ELLIPSE   := 2
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_SES_RECTANGLE	Rectangle
TCVN_SES_CROSS	Cross
TCVN_SES_ELLIPSE	Ellipse

相关函数

- F_VN_CreateStructuringElement [▶ 1240]

6.1.2.4.63 ETcVnSvm

Offers SVM model types.

Syntax

Definition:

```
TYPE ETcVnSvm :
(
    TCVN_SVM_C_CLASSIFIER      := 1,
    TCVN_SVM_NU_CLASSIFIER     := 65537,
    TCVN_SVM_NOVELTY_DETECTOR := 2,
    TCVN_SVM_EPS_REGRESSOR     := 4,
    TCVN_SVM_NU_REGRESSOR      := 65540
)UDINT;
END_TYPE
```

Values

Name	Description
TCVN_SVM_C_CLASSIFIER	Classifier
TCVN_SVM_NU_CLASSIFIER	Classifier
TCVN_SVM_NOVELTY_DETECTOR	Novelty detector
TCVN_SVM_EPS_REGRESSOR	Regressor
TCVN_SVM_NU_REGRESSOR	Regressor

6.1.2.4.64 ETcVnSvmKernelType

Offers SVM kernel types.

Syntax

Definition:

```
TYPE ETcVnSvmKernelType :
(
    TCVN_SKT_LINEAR  := 0,
    TCVN_SKT_POLY    := 1,
    TCVN_SKT_RBF     := 2,
    TCVN_SKT_SIGMOID := 3,
    TCVN_SKT_CHI2    := 4,
    TCVN_SKT_INTER   := 5
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_SKT_LINEAR	Linear kernel (fast)
TCVN_SKT_POLY	Polynomial kernel
TCVN_SKT_RBF	Radial basis function (RBF) kernel (good default)
TCVN_SKT_SIGMOID	Sigmoid kernel
TCVN_SKT_CHI2	Chi-squared kernel
TCVN_SKT_INTER	Histogram intersection kernel

6.1.2.4.65 ETcVnSvmSgdClassifierMarginType

Offers different margin types for SVMSGD classifiers.

Syntax

Definition:

```
TYPE ETcVnSvmSgdClassifierMarginType :
(
    TCVN_SSCMT_SOFT_MARGIN := 0,
    TCVN_SSCMT_HARD_MARGIN := 1
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_SSCMT_SOFT_MARGIN	Soft margin allowing outliers. (beneficial for classes that cannot be separated linearly)
TCVN_SSCMT_HARD_MARGIN	Hard margin. (best-suited for linearly separable classes)

6.1.2.4.66 ETcVnSvmSgdClassifierType

Offers different types of SVMSGD classifiers.

Syntax

Definition:

```
TYPE ETcVnSvmSgdClassifierType :
(
    TCVN_SSCT_SGD  := 0,
    TCVN_SSCT_ASGD := 1
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_SSCT_SGD	Stochastic Gradient Descent
TCVN_SSCT_ASGD	Average Stochastic Gradient Descent

6.1.2.4.67 ETcVnTemplateMatchMethod

Offers methods for template matching.

Syntax

Definition:

```
TYPE ETcVnTemplateMatchMethod :
(
    TCVN_TMM_SQDIFF         := 0,
    TCVN_TMM_SQDIFF_NORMED  := 1,
    TCVN_TMM_CCORR          := 2,
    TCVN_TMM_CCORR_NORMED   := 3,
    TCVN_TMM_CCOEFF         := 4,
    TCVN_TMM_CCOEFF_NORMED  := 5
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_TMM_SQDIFF	Squared difference (supports template mask)
TCVN_TMM_SQDIFF_NORMED	Normalized squared difference
TCVN_TMM_CCORR	Cross-correlation
TCVN_TMM_CCORR_NORMED	Normalized cross-correlation (supports template mask)
TCVN_TMM_CCOEFF	Correlation coefficient
TCVN_TMM_CCOEFF_NORMED	Normalized correlation coefficient

相关函数

- F_VN_MatchTemplateAndEvaluateExp [▶ 1161]
- F_VN_MatchTemplateExp [▶ 1163]

6.1.2.4.68 ETcVnThresholdType

Offers threshold types.

Syntax

Definition:

```
TYPE ETcVnThresholdType :
(
    TCVN_TT_BINARY              := 0,
    TCVN_TT_BINARY_INV          := 1,
    TCVN_TT_TRUNC               := 2,
    TCVN_TT_TOZERO              := 3,
    TCVN_TT_TOZERO_INV          := 4,
    TCVN_TT_OTSU_BINARY         := 8,
    TCVN_TT_OTSU_BINARY_INV     := 9,
    TCVN_TT_OTSU_TRUNC          := 10,
    TCVN_TT_OTSU_TOZERO         := 11,
    TCVN_TT_OTSU_TOZERO_INV     := 12,
    TCVN_TT_TRIANGLE_BINARY     := 16,
    TCVN_TT_TRIANGLE_BINARY_INV := 17,
    TCVN_TT_TRIANGLE_TRUNC      := 18,
    TCVN_TT_TRIANGLE_TOZERO     := 19,
    TCVN_TT_TRIANGLE_TOZERO_INV := 20
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_TT_BINARY	Binary threshold
TCVN_TT_BINARY_INV	Inverted binary threshold
TCVN_TT_TRUNC	Truncated threshold (pixels > thresh are set to thresh, others keep their value)
TCVN_TT_TOZERO	To zero threshold (pixels < thresh are set to zero, others keep their value)
TCVN_TT_TOZERO_INV	Inverted to zero threshold (pixels > thresh are set to zero, others keep their value)
TCVN_TT_OTSU_BINARY	Binary threshold with the threshold value selected according to the Otsu algorithm
TCVN_TT_OTSU_BINARY_INV	Inverted binary threshold with the threshold value selected according to the Otsu algorithm
TCVN_TT_OTSU_TRUNC	Truncated threshold with the threshold value selected according to the Otsu algorithm
TCVN_TT_OTSU_TOZERO	To zero threshold with the threshold value selected according to the Otsu algorithm
TCVN_TT_OTSU_TOZERO_INV	Inverted to zero threshold with the threshold value selected according to the Otsu algorithm
TCVN_TT_TRIANGLE_BINARY	Binary threshold with the threshold value selected according to the Triangle algorithm
TCVN_TT_TRIANGLE_BINARY_INV	Inverted binary threshold with the threshold value selected according to the Triangle algorithm
TCVN_TT_TRIANGLE_TRUNC	Truncated threshold with the threshold value selected according to the Triangle algorithm
TCVN_TT_TRIANGLE_TOZERO	To zero threshold with the threshold value selected according to the Triangle algorithm
TCVN_TT_TRIANGLE_TOZERO_INV	Inverted to zero threshold with the threshold value selected according to the Triangle algorithm

更多信息

枚举ETcVnThresholdType提供了计算阈值所实施的方法的选择。它也被用于结构TcVnParamsBlobDetection [▸ 213]。

相关函数

- F_VN_AdaptiveThresholdExp [▸ 1283]
- F_VN_Threshold [▸ 1287]

6.1.2.4.69 ETcVnTimestamp

Offers image acquisition timestamps.

Syntax

Definition:

```
TYPE ETcVnTimestamp :
(
    TCVN_TS_IMAGE_COMPLETED := 1
)ULINT;
END_TYPE
```

Values

Name	Description
TCVN_TS_IMAGE_COMPLETED	Timestamp when the image was completed, i.e. all packets were received by the GigEVision driver.

6.1.2.4.70 ETcVnVectorCompareMethod

Offers methods for vector comparison.

Syntax

Definition:

```
TYPE ETcVnVectorCompareMethod :
(
    TCVN_VCM_EUCLIDEAN   := 0,
    TCVN_VCM_ELEMENTWISE := 1
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_VCM_EUCLIDEAN	Euclidean
TCVN_VCM_ELEMENTWISE	Elementwise

相关函数

- F_VN_MaxPixelValueExp [▶ 1187]
- F_VN_MinPixelValueExp [▶ 1189]

6.1.2.4.71 ETcVnWorldCoordinateSystem

Offers world coordinate system orientations.

Syntax

Definition:

```
TYPE ETcVnWorldCoordinateSystem :
(
    TCVN_WCS_X_RIGHT_Y_DOWN := 0,
    TCVN_WCS_X_DOWN_Y_RIGHT := 1,
    TCVN_WCS_X_LEFT_Y_DOWN  := 2,
    TCVN_WCS_X_UP_Y_RIGHT   := 3,
    TCVN_WCS_X_RIGHT_Y_UP   := 4,
    TCVN_WCS_X_DOWN_Y_LEFT  := 5,
    TCVN_WCS_X_LEFT_Y_UP    := 6,
    TCVN_WCS_X_UP_Y_LEFT    := 7
)DINT;
END_TYPE
```

Values

Name	Description
TCVN_WCS_X_RIGHT_Y_DOWN	X axis points right, Y axis down (same orientation as image coordinate system)
TCVN_WCS_X_DOWN_Y_RIGHT	X axis points down, Y axis right
TCVN_WCS_X_LEFT_Y_DOWN	X axis points left, Y axis down
TCVN_WCS_X_UP_Y_RIGHT	X axis points up, Y axis right
TCVN_WCS_X_RIGHT_Y_UP	X axis points right, Y axis up
TCVN_WCS_X_DOWN_Y_LEFT	X axis points down, Y axis left
TCVN_WCS_X_LEFT_Y_UP	X axis points left, Y axis up
TCVN_WCS_X_UP_Y_LEFT	X axis points up, Y axis left

6.1.2.4.72 ETcWatchdogAccumulationType

Offers watchdog accumulation types to compute the fraction processed.

Syntax

Definition:

```
TYPE ETcWatchdogAccumulationType :
(
    WD_ACC_TYPE_MEAN    := 1,
    WD_ACC_TYPE_PRODUCT := 2
)UINT;
END_TYPE
```

Values

Name	Description
WD_ACC_TYPE_MEAN	Calculates the mean of the individual fractions processed over all monitored functions (recommended for independent functions).
WD_ACC_TYPE_PRODUCT	Calculates the product of the individual fractions processed over all monitored functions (recommended for dependent functions).

更多信息

通过这个枚举，在设置当几个函数串联时，可以设置看门狗 [▸ 127]如何计算已执行函数的分数（nFractionProcessed）。

作为说明操作模式的一个示例，假设有三个函数受到监测。第一次是 100% 执行，第二次是 50%，第三次是 0%。

WD_ACC_TYPE_MEAN	所有函数的执行分数的平均值将作为结果返回。在例子中，结果是：（100% + 50% + 0%） / 3 = 50%。
WD_ACC_TYPE_PRODUCT	所有函数的执行分数的乘积将作为结果返回。在这个例子中，结果将是：100% * 50% * 0% = 0%。

根据程序结构和机器的安全状态，使用情况有所不同。

相关函数

- F_VN_StartAbsWatchdogExp [▸ 949]
- F_VN_StartRelWatchdogExp [▸ 950]

6.1.2.4.73 GEV_CAMERA_STATE

Describes the GigEVision camera connection state.

Syntax

Definition:

```
TYPE GEV_CAMERA_STATE :
(
    GEV_CAMERA_IDLE                := 0,
    GEV_CONTROL_CHANNEL_OPEN_MASK := 1,
    GEV_STREAM_CHANNELS_OPEN_MASK := 2,
    GEV_CONTROL_CHANNEL_OPEN      := 1,
    GEV_STREAM_CHANNELS_OPEN      := 3
)BYTE;
END_TYPE
```

Values

Name	Description
GEV_CAMERA_IDLE	Camera is idle (no control or stream channel open).
GEV_CONTROL_CHANNEL_OPEN_MASK	The control channel is open.
GEV_STREAM_CHANNELS_OPEN_MASK	At least 1 stream channel is open.
GEV_CONTROL_CHANNEL_OPEN	The control channel is open, but no stream channel.
GEV_STREAM_CHANNELS_OPEN	At least 1 stream channel and the control channel are open.

6.1.2.5 Structs

6.1.2.5.1 GVSP_IMAGE_INFO

Shows GVSP (GigE Vision Streaming Protocol) meta information.

Syntax

Definition:

```
TYPE GVSP_IMAGE_INFO :
STRUCT
    CameraIpAddress : UDINT;
    LocalIpAddress  : UDINT;
    CameraUdpPort   : UINT;
    LocalUdpPort    : UINT;
    GvspChannelId   : UINT;
    GevStatus       : UINT;
    BlockId         : ULINT;
    LeaderInfo      : GVSP_LEADER_PAYLOAD_IMAGE;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
CameraIpAddress	UDINT	Camera IP address
LocalIpAddress	UDINT	Local IP address
CameraUdpPort	UINT	Camera UDP port
LocalUdpPort	UINT	Local UDP port
GvspChannelId	UINT	ID of GVSP channel
GevStatus	UINT	GigE Vision status code
BlockId	ULINT	Block Id (incremented for each acquired image, but reset to 0 on overflow)
LeaderInfo	GVSP_LEADER_PAYLOAD_IMAGE [▶ 205]	Contains information about timestamp, pixel format and size.

6.1.2.5.2 GVSP_LEADER_PAYLOAD_IMAGE

Shows information over the GVSP (GigE Vision Streaming Protocol) leader payload image.

Syntax

Definition:

```
TYPE GVSP_LEADER_PAYLOAD_IMAGE :
STRUCT
    Timestamp   : ULINT;
    PixelFormat : GVSP_PIXEL_FORMAT;
    SizeX       : UDINT;
    SizeY       : UDINT;
    OffsetX     : UDINT;
    OffsetY     : UDINT;
    PaddingX    : UINT;
    PaddingY    : UINT;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
Timestamp	ULINT	Image timestamp
PixelFormat	GVSP_PIXEL_FORMAT [▶ 205]	Image pixel format
SizeX	UDINT	Image size in x direction
SizeY	UDINT	Image size in y direction
OffsetX	UDINT	Image x-offset from (0,0) origin
OffsetY	UDINT	Image y-offset from (0,0) origin
PaddingX	UINT	Image padding in x direction
PaddingY	UINT	Image padding in y direction

6.1.2.5.3 GVSP_PIXEL_FORMAT

Shows the GVSP (GigE Vision Streaming Protocol) pixel format.

Syntax

Definition:

```
TYPE GVSP_PIXEL_FORMAT :
STRUCT
    Color               : BYTE;
```

```
    EffectivePixelSize : BYTE;
    Id                 : UINT;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
Color	BYTE	Indicates if the pixel format is mono or color.
EffectivePixelSize	BYTE	Effective pixel size in bit
Id	UINT	GVSP pixel format ID

6.1.2.5.4 TcVnCameraCalibrationOptions

Offers camera calibration options.

Syntax

Definition:

```
TYPE TcVnCameraCalibrationOptions :
STRUCT
    bUseIntrinsicGuess : BIT;
    bFixAspectRatio    : BIT;
    bFixPrincipalPoint : BIT;
    bZeroTangentDist   : BIT;
    bFixFocalLength    : BIT;
    bFixK1             : BIT;
    bFixK2             : BIT;
    bFixK3             : BIT;
    bFixK4             : BIT;
    bFixK5             : BIT;
    bFixK6             : BIT;
    bRationalModel     : BIT;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
bUseIntrinsicGuess	BIT	The camera matrix and distortion coefficients already contain a valid initial guess, which is optimized further.
bFixAspectRatio	BIT	The ratio fx/fy stays the same as in the input camera matrix.
bFixPrincipalPoint	BIT	The principal point is fixed to the image center (or provided cx,cy if bUseIntrinsicGuess).
bZeroTangentDist	BIT	The tangential distortion coefficients (p1, p2) are forced to 0.
bFixFocalLength	BIT	The parameters fx and fy stay the same as in the input camera matrix.
bFixK1	BIT	The radial distortion coefficient k1 is fixed to 0 (or provided input if bUseIntrinsicGuess)
bFixK2	BIT	The radial distortion coefficient k2 is fixed to 0 (or provided input if bUseIntrinsicGuess)
bFixK3	BIT	The radial distortion coefficient k3 is fixed to 0 (or provided input if bUseIntrinsicGuess)
bFixK4	BIT	The radial distortion coefficient k4 is fixed to 0 (or provided input if bUseIntrinsicGuess)
bFixK5	BIT	The radial distortion coefficient k5 is fixed to 0 (or provided input if bUseIntrinsicGuess)
bFixK6	BIT	The radial distortion coefficient k6 is fixed to 0 (or provided input if bUseIntrinsicGuess)
bRationalModel	BIT	The radial distortion coefficients k4, k5, k6 are enabled.

6.1.2.5.5 TcVnCircularArc

Describes a circular arc.

Syntax

Definition:

```
TYPE TcVnCircularArc :
STRUCT
    aCenter      : TcVnPoint2_REAL;
    fRadius      : REAL;
    fStartAngle : REAL;
    fEndAngle    : REAL;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
aCenter	TcVnPoint2_REAL [▶ 139]	Center of the circular arc
fRadius	REAL	Radius of the circular arc
fStartAngle	REAL	Start angle of the circular arc
fEndAngle	REAL	End angle of the circular arc

6.1.2.5.6 TcVnCodeGrades1D

Describes code quality grades for 1D barcodes according to ISO / IEC 15416:2016. The grades range from 0 (very bad) to 4 (very good) and are averaged over several scan lines.

Syntax

Definition:

```
TYPE TcVnCodeGrades1D :
STRUCT
    fDecode           : REAL;
    fSymbolContrast   : REAL;
    fMinReflectance   : REAL;
    fMinEdgeContrast  : REAL;
    fModulation       : REAL;
    fDefects          : REAL;
    fDecodability     : REAL;
    fOverall          : REAL;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
fDecode	REAL	Decode grade
fSymbolContrast	REAL	Symbol contrast grade
fMinReflectance	REAL	Minimum reflectance grade
fMinEdgeContrast	REAL	Minimum edge contrast grade
fModulation	REAL	Modulation grade
fDefects	REAL	Defects grade
fDecodability	REAL	Decodability grade
fOverall	REAL	Overall code grade, i.e. the minimum achieved individual grade averaged over all scan lines. If different scan lines lead to different decoded data, the overall grade is 0.

6.1.2.5.7 TcVnCodeGradesDM

Describes code quality grades for Data Matrix codes according to ISO / IEC 15415:2011. The grades range from 0 (very bad) to 4 (very good).

Syntax

Definition:

```
TYPE TcVnCodeGradesDM :
STRUCT
    nDecode                 : USINT;
    nSymbolContrast         : USINT;
    nModulation             : USINT;
    nReflectanceMargin      : USINT;
    nFixedPatternDamage     : USINT;
    nAxialNonuniformity     : USINT;
```

```
    nGridNonuniformity     : USINT;
    nUnusedErrorCorrection : USINT;
    nOverall               : USINT;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
nDecode	USINT	Decode grade
nSymbolContrast	USINT	Symbol contrast grade
nModulation	USINT	Modulation grade
nReflectanceMargin	USINT	Reflectance margin grade
nFixedPatternDamage	USINT	Fixed pattern damage grade
nAxialNonuniformity	USINT	Axial nonuniformity grade
nGridNonuniformity	USINT	Grid nonuniformity grade
nUnusedErrorCorrection	USINT	Unused error correction grade
nOverall	USINT	Overall code grade, i.e. the minimum achieved individual grade

6.1.2.5.8 TcVnCodeGradesQR

Describes code quality grades for QR codes according to ISO / IEC 15415:2011. The grades range from 0 (very bad) to 4 (very good).

Syntax

Definition:

```
TYPE TcVnCodeGradesQR :
STRUCT
    nDecode                : USINT;
    nSymbolContrast        : USINT;
    nModulation            : USINT;
    nReflectanceMargin     : USINT;
    nFixedPatternDamage    : USINT;
    nAxialNonuniformity    : USINT;
    nGridNonuniformity     : USINT;
    nUnusedErrorCorrection : USINT;
    nFormatInfo            : USINT;
    nVersionInfo           : USINT;
    nOverall               : USINT;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
nDecode	USINT	Decode grade
nSymbolContrast	USINT	Symbol contrast grade
nModulation	USINT	Modulation grade
nReflectanceMargin	USINT	Reflectance margin grade
nFixedPatternDamage	USINT	Fixed pattern damage grade
nAxialNonuniformity	USINT	Axial nonuniformity grade
nGridNonuniformity	USINT	Grid nonuniformity grade
nUnusedErrorCorrection	USINT	Unused error correction grade
nFormatInfo	USINT	Format information grade
nVersionInfo	USINT	Version information grade
nOverall	USINT	Overall code grade, i.e. the minimum achieved individual grade

6.1.2.5.9 TcVnDMatch

Describes a descriptor match

Syntax

Definition:

```
TYPE TcVnDMatch :
STRUCT
    nQueryIdx : DINT;
    nTrainIdx : DINT;
    nImageIdx : DINT;
    fDistance : REAL;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
nQueryIdx	DINT	Query descriptor index
nTrainIdx	DINT	Train descriptor index
nImageIdx	DINT	Train image index
fDistance	REAL	Distance between the descriptors (smaller distance means better match)

6.1.2.5.10 TcVnImageInfo

Shows image information.

Syntax

Definition:

```
TYPE TcVnImageInfo :
STRUCT
    nImageSize    : ULINT;
    nWidth        : UDINT;
    nHeight       : UDINT;
    nXPadding     : UINT;
    nYPadding     : UINT;
    stPixelFormat : TcVnPixelFormat;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
nImageSize	ULINT	Image size (number of pixels)
nWidth	UDINT	Image width
nHeight	UDINT	Image height
nXPadding	UINT	Image x-padding
nYPadding	UINT	Image y-padding
stPixelFormat	TcVnPixelFormat [▶ 224]	Pixel format

6.1.2.5.11 TcVnKeyPoint

Describes a key point.

Syntax

Definition:

```
TYPE TcVnKeyPoint :
STRUCT
    aPoint    : TcVnPoint2_REAL;
    fDiameter : REAL;
    fAngle    : REAL;
    fResponse : REAL;
    nOctave   : DINT;
    nClassId  : DINT;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
aPoint	TcVnPoint2_REAL [▶ 139]	Position
fDiameter	REAL	Diameter
fAngle	REAL	Angle
fResponse	REAL	Response
nOctave	DINT	Octave
nClassId	DINT	Class ID

6.1.2.5.12 TcVnMatrix

Offers a user-defined matrix with variable rows, columns and element-type.

Syntax

Definition:

```
TYPE TcVnMatrix :
STRUCT
    nRows : UDINT;
    nCols : UDINT;
    eType : ETcVnElementType;
    pData : PVOID;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
nRows	UDINT	Number of rows
nCols	UDINT	Number of columns
eType	ETcVnElementType [▶ 178]	Element type
pData	PVOID	Pointer to the data

6.1.2.5.13 TcVnMoments

Offers image or contour moments.

Syntax

Definition:

```
TYPE TcVnMoments :
STRUCT
    fM00  : LREAL;
    fM10  : LREAL;
    fM01  : LREAL;
    fM20  : LREAL;
    fM11  : LREAL;
    fM02  : LREAL;
    fM30  : LREAL;
    fM21  : LREAL;
    fM12  : LREAL;
    fM03  : LREAL;
```

```
    fMu20 : LREAL;
    fMu11 : LREAL;
    fMu02 : LREAL;
    fMu30 : LREAL;
    fMu21 : LREAL;
    fMu12 : LREAL;
    fMu03 : LREAL;
    fNu20 : LREAL;
    fNu11 : LREAL;
    fNu02 : LREAL;
    fNu30 : LREAL;
    fNu21 : LREAL;
    fNu12 : LREAL;
    fNu03 : LREAL;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
fM00	LREAL	Spatial moment 00
fM10	LREAL	Spatial moment 10
fM01	LREAL	Spatial moment 01
fM20	LREAL	Spatial moment 20
fM11	LREAL	Spatial moment 11
fM02	LREAL	Spatial moment 02
fM30	LREAL	Spatial moment 30
fM21	LREAL	Spatial moment 21
fM12	LREAL	Spatial moment 12
fM03	LREAL	Spatial moment 03
fMu20	LREAL	Central moment 20
fMu11	LREAL	Central moment 11
fMu02	LREAL	Central moment 02
fMu30	LREAL	Central moment 30
fMu21	LREAL	Central moment 21
fMu12	LREAL	Central moment 12
fMu03	LREAL	Central moment 03
fNu20	LREAL	Normalized central moment 20
fNu11	LREAL	Normalized central moment 11
fNu02	LREAL	Normalized central moment 02
fNu30	LREAL	Normalized central moment 30
fNu21	LREAL	Normalized central moment 21
fNu12	LREAL	Normalized central moment 12
fNu03	LREAL	Normalized central moment 03

6.1.2.5.14 TcVnParamsAGAST

Offers parameters for AGAST method.

Syntax

Definition:

```
TYPE TcVnParamsAGAST :
STRUCT
    nThreshold          : DINT;
    bNonMaxSuppression : BOOL;
    eType               : ETcVnKeypointDetectionTypeAGAST;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
nThreshold	DINT	10	Threshold for the intensity difference between the center pixel and its surrounding circle
bNonMaxSuppression	BOOL	TRUE	If true, non-maximum suppression is applied.
eType	ETcVnKeypointDetectionTypeAGAST [▸ 186]	TCVN_KDT_AGAST_9_16	Neighborhood type

6.1.2.5.15 TcVnParamsAKAZE

Offers parameters for AKAZE method.

Syntax

Definition:

```
TYPE TcVnParamsAKAZE :
STRUCT
    eDescrType      : ETcVnFeatureDescriptorTypeAKAZE;
    nDescrSize      : UDINT;
    nDescrChannels  : UDINT;
    fThreshold      : REAL;
    nOctaves        : UDINT;
    nOctaveLayers   : UDINT;
    eDiffusivity    : ETcVnDiffusivityTypeKAZE;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
eDescrType	ETcVnFeatureDescriptorTypeAKAZE [▸ 180]	TCVN_FDT_AKAZE_MLDB	Type of the descriptor
nDescrSize	UDINT	0	Size of the descriptor in bits (only for MLDB; 0 = full size)
nDescrChannels	UDINT	3	Number of descriptor channels (currently only 3 supported for MLDB)
fThreshold	REAL	0.001	Detector response threshold
nOctaves	UDINT	2	Maximum octave evolution
nOctaveLayers	UDINT	1	Number of sublevels per scale level
eDiffusivity	ETcVnDiffusivityTypeKAZE [▸ 173]	TCVN_DT1_KAZE_PM_G2	Diffusivity type

6.1.2.5.16 TcVnParamsBlobDetection

Offers parameters for blob detection

Syntax

Definition:

```
TYPE TcVnParamsBlobDetection :
STRUCT
    bFilterByArea          : BOOL;
    bFilterByCircularity   : BOOL;
    bFilterByConvexity     : BOOL;
    bFilterByEccentricity  : BOOL;
    bFilterByInertiaRatio  : BOOL;
    fMinArea               : REAL;
    fMaxArea               : REAL;
    fMinCircularity        : REAL;
    fMaxCircularity        : REAL;
    fMinConvexity          : REAL;
    fMaxConvexity          : REAL;
    fMinEccentricity       : REAL;
    fMaxEccentricity       : REAL;
    fMinInertiaRatio       : REAL;
    fMaxInertiaRatio       : REAL;
    eThresholdType         : ETcVnThresholdType;
    fMinThreshold          : REAL;
    fMaxThreshold          : REAL;
    fThresholdStep         : REAL;
    fMinBlobDistance       : REAL;
    nMinRepeatability      : UDINT;
    eBlobCombination       : ETcVnBlobCombination;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
bFilterByArea	BOOL	TRUE	Enable filtering by area (fMinArea <= area <= fMaxArea); strongly recommended to activate for filtering noise with fMinArea.
bFilterByCircularity	BOOL	FALSE	Enable filtering by circularity (fMinCircularity <= circularity(4pi * area / perimeter^2) <= fMaxCircularity).
bFilterByConvexity	BOOL	FALSE	Enable filtering by convexity (fMinConvexity <= convexity(area / hullArea) <= fMaxConvexity).
bFilterByEccentricity	BOOL	FALSE	Enable filtering by eccentricity (fMinEccentricity <= eccentricity <= fMaxEccentricity).
bFilterByInertiaRatio	BOOL	FALSE	Enable filtering by inertia ratio (fMinInertiaRatio <= inertia ratio <= fMaxInertiaRatio).
fMinArea	REAL	10	Min estimated blob area in pixel
fMaxArea	REAL	100000000	Max estimated blob area in pixel
fMinCircularity	REAL	0	Min circularity (1.0: ideal circle, < 1: less circular, 0: not circular at all)
fMaxCircularity	REAL	1	Max circularity (1.0: ideal circle, < 1: less circular, 0: not circular at all)
fMinConvexity	REAL	0	Min convexity (1.0: blob fully convex, < 1: less convex)
fMaxConvexity	REAL	1	Max convexity (1.0: blob fully convex, < 1: less convex)
fMinEccentricity	REAL	0	Min eccentricity (0.0: circular, 1.0: linear)
fMaxEccentricity	REAL	1	Max eccentricity (0.0: circular, 1.0: linear)
fMinInertiaRatio	REAL	0	Min inertia ratio (1.0: equal width and height, 0.0: linear)
fMaxInertiaRatio	REAL	1	Max inertia ratio (1.0: equal width and height, 0.0: linear)

Name	Type	Default	Description
eThresholdType	ETcVnThresholdType [▶ 200]	TCVN_TT_BINARY	Threshold type for internally applied threshold method (OTSU_XXX only supported for USINT images).
fMinThreshold	REAL	30	Threshold to start with (if fThresholdStep > 0, otherwise this is the only threshold used).
fMaxThreshold	REAL	225	Threshold to end with (if fThresholdStep > 0, otherwise this value is not used).
fThresholdStep	REAL	0	Sets to 0 if only 1 threshold should be used (much faster than multiple thresholds and combining the results).
fMinBlobDistance	REAL	5	Minimum distance between the center points of two different blobs (only used if fThresholdStep > 0; if distance < fMinBlobDistance, the blobs are treated as the same).
nMinRepeatability	UDINT	2	Minimum number of threshold steps, for which the same contour has to be detected (only used if fThresholdStep > 0; same means center point distance < fMinBlobDistance).
eBlobCombination	ETcVnBlobCombination [▶ 144]	TCVN_BC_MEDIAN_THRESHOLD	Selects, which of the multi-threshold blob contours should be returned.

6.1.2.5.17 TcVnParamsBRISK

Offers parameters for BRISK method.

Syntax

Definition:

```
TYPE TcVnParamsBRISK :
STRUCT
    nThreshold    : DINT;
    nOctaves      : UDINT;
    fPatternScale : REAL;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
nThreshold	DINT	30	Detection threshold
nOctaves	UDINT	3	Detection octaves (0 for single scale)
fPatternScale	REAL	1	Scale factor for the neighborhood pattern

6.1.2.5.18 TcVnParamsFAST

Offers parameters for FAST method.

Syntax

Definition:

```
TYPE TcVnParamsFAST :
STRUCT
    nThreshold          : DINT;
    bNonMaxSuppression : BOOL;
    eType               : ETcVnKeypointDetectionTypeFAST;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
nThreshold	DINT	10	Threshold for the intensity difference between the center pixel and its surrounding circle
bNonMaxSuppression	BOOL	TRUE	If true, non-maximum suppression is applied.
eType	ETcVnKeypointDetectionTypeFAST [▶ 186]	TCVN_KDT_FAST_9_16	Neighborhood type

6.1.2.5.19 TcVnParamsGFTT

Offers parameters for GFTT method.

Syntax

Definition:

```
TYPE TcVnParamsGFTT :
STRUCT
    nMaxCorners         : UDINT;
    fQualityLevel       : LREAL;
    fMinDistance        : LREAL;
    nBlockSize          : UDINT;
    bUseHarrisDetector : BOOL;
    fHarrisK            : LREAL;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
nMaxCorners	UDINT	1000	Maximum number of corners to return (strongest ones)
fQualityLevel	LREAL	0.01	Minimum accepted corner quality, relative to the strongest one
fMinDistance	LREAL	1	Minimum euclidean distance between returned keypoints
nBlockSize	UDINT	3	Neighborhood block size
bUseHarrisDetector	BOOL	FALSE	If true, a Harris detector is used instead of the default method.
fHarrisK	LREAL	0.04	Free parameter of the Harris detector

6.1.2.5.20 TcVnParamsKAZE

Offers parameters for KAZE method

Syntax

Definition:

```
TYPE TcVnParamsKAZE :
STRUCT
    bExtended      : BOOL;
    bUpright       : BOOL;
    fThreshold     : REAL;
    nOctaves       : UDINT;
    nOctaveLayers  : UDINT;
    eDiffusivity   : ETcVnDiffusivityTypeKAZE;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
bExtended	BOOL	FALSE	If true, the descriptor size is extended from the default 64 byte to 128 byte.
bUpright	BOOL	FALSE	If true, upright (rotation dependent) descriptors are computed.
fThreshold	REAL	0.001	Detector response threshold
nOctaves	UDINT	4	Maximum octave evolution of the image
nOctaveLayers	UDINT	2	Number of sublevels per scale level
eDiffusivity	ETcVnDiffusivityTypeKAZE [▶ 173]	TCVN_DT1_KAZE_PM_G2	Diffusivity type

6.1.2.5.21 TcVnParamsMSER

Offers parameters for MSER method

Syntax

Definition:

```
TYPE TcVnParamsMSER :
STRUCT
    nDelta          : DINT;
    nMinArea        : UDINT;
    nMaxArea        : UDINT;
    fMaxVariation   : LREAL;
    fMinDiversity   : LREAL;
    nMaxEvolution   : UDINT;
    fAreaThreshold  : LREAL;
    fMinMargin      : LREAL;
    nEdgeBlurSize   : UDINT;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
nDelta	DINT	5	Delta for size comparison
nMinArea	UDINT	60	Minimum blob size
nMaxArea	UDINT	14400	Maximum blob size
fMaxVariation	LREAL	0.25	Maximum blob size variation
fMinDiversity	LREAL	0.2	Minimum MSER diversity
nMaxEvolution	UDINT	200	Maximum evolution steps (only used for color images)
fAreaThreshold	LREAL	1.01	Area threshold to cause re-initialization (only used for color images)
fMinMargin	LREAL	0.003	Minimum margin (only used for color images)
nEdgeBlurSize	UDINT	5	Aperture size for edge bluring (only used for color images)

6.1.2.5.22 TcVnParamsORB

Offers parameters for ORB method.

Syntax

Definition:

```
TYPE TcVnParamsORB :
STRUCT
    nMaxPoints      : UDINT;
    fPyramidScale   : REAL;
    nPyramidLevels  : UDINT;
    nEdgeThreshold  : UDINT;
    nFirstLevel     : UDINT;
    nBriefPoints    : UDINT;
    eScoreType      : ETcVnFeatureScoreTypeORB;
    nPatchSize      : UDINT;
    nFastThreshold  : DINT;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
nMaxPoints	UDINT	500	Maximum number of returned keypoints
fPyramidScale	REAL	1.2	Pyramid decimation ratio (must be greater than 1, should be smaller than 2)
nPyramidLevels	UDINT	8	Number of pyramid levels
nEdgeThreshold	UDINT	31	Size of the border, where no features are detected (should match nPatchSize)
nFirstLevel	UDINT	0	First pyramid level (currently, only 0 is supported)
nBriefPoints	UDINT	2	Number of points to produce each BRIEF descriptor element (2, 3, 4)
eScoreType	ETcVnFeatureScoreTypeORB [▶ 181]	TCVN_FST_ORB_HARRIS	Score type (HARRIS is more stable but slightly slower than FAST)
nPatchSize	UDINT	31	Patch size of the BRIEF descriptor
nFastThreshold	DINT	20	Threshold for the FAST keypoint detection

6.1.2.5.23 TcVnParamsSB

Offers parameters for SB method (a simple blob detector with multiple thresholds).

Syntax

Definition:

```
TYPE TcVnParamsSB :
STRUCT
    bFilterByArea         : BOOL;
    bFilterByCircularity  : BOOL;
    bFilterByColor        : BOOL;
    bFilterByConvexity    : BOOL;
    bFilterByInertia      : BOOL;
    fMinArea              : REAL;
    fMaxArea              : REAL;
    fMinCircularity       : REAL;
    fMaxCircularity       : REAL;
    nBlobColor            : USINT;
    fMinConvexity         : REAL;
    fMaxConvexity         : REAL;
    fMinInertiaRatio      : REAL;
    fMaxInertiaRatio      : REAL;
    fMinBlobDist          : REAL;
    nMinRepeatability     : UDINT;
    fMinThreshold         : REAL;
    fMaxThreshold         : REAL;
    fThresholdStep        : REAL;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Default	Description
bFilterByArea	BOOL	TRUE	Enable filtering by area (fMinArea <= area < fMaxArea).
bFilterByCircularity	BOOL	FALSE	Enable filtering by circularity (fMinCircularity <= circularity(4pi * area / perimeter^2) < fMaxCircularity).
bFilterByColor	BOOL	FALSE	Enable filtering by color (thresholdedColor(0 or 255) = nBlobColor).
bFilterByConvexity	BOOL	FALSE	Enable filtering by convexity (fMinConvexity <= convexity(area / hullArea) < fMaxConvexity).
bFilterByInertia	BOOL	FALSE	Enable filtering by inertia ratio (fMinInertiaRatio <= inertia ratio < fMaxInertiaRatio).
fMinArea	REAL	25	Min estimated blob area in pixel
fMaxArea	REAL	15000	Max estimated blob area in pixel
fMinCircularity	REAL	0	Min circularity (1.0: ideal circle, < 1: less circular, 0: not circular at all)
fMaxCircularity	REAL	1	Max circularity (1.0: ideal circle, < 1: less circular, 0: not circular at all)
nBlobColor	USINT	255	0 or 255
fMinConvexity	REAL	0	Min convexity (1.0: blob fully convex, < 1: less convex)
fMaxConvexity	REAL	1	Max convexity (1.0: blob fully convex, < 1: less convex)
fMinInertiaRatio	REAL	0	Min inertia ratio (0.0 .. 1.0)
fMaxInertiaRatio	REAL	1	Max inertia ratio (0.0 .. 1.0)
fMinBlobDist	REAL	5	Min distance between different blobs
nMinRepeatability	UDINT	2	Min number of same detected blobs (dist < fMinBlobDist) in different thresholds
fMinThreshold	REAL	30	Min threshold (start)
fMaxThreshold	REAL	225	Max threshold (stop)
fThresholdStep	REAL	10	Threshold step between min and max threshold

6.1.2.5.24 TcVnPixelFormat

Contains detailed information about the pixel format.

Syntax

Definition:

```
TYPE TcVnPixelFormat :
STRUCT
    bSupported     : BIT;
    bSigned        : BIT;
    bPlanar        : BIT;
    bFloat         : BIT;
    nChannels      : BYTE;
    ePixelEncoding : ETcVnPixelEncoding;
    ePixelPackMode : ETcVnPixelPackMode;
    nElementSize   : UINT;
    nTotalSize     : UINT;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
bSupported	BIT	If false, the pixel format is not supported.
bSigned	BIT	If true, pixel intensities are signed values.
bPlanar	BIT	If true, the image channels are stored planar instead of interleaved (e.g. RRRRRGGGGGBBBBB instead of RGBRGBRGBRGBRGB).
bFloat	BIT	If true, the pixel format is floating point.
nChannels	BYTE	Number of channels
ePixelEncoding	ETcVnPixelEncoding [▶ 193]	Pixel encoding
ePixelPackMode	ETcVnPixelPackMode [▶ 193]	Pixel pack mode
nElementSize	UINT	Size (bit) of a single pixel channel
nTotalSize	UINT	Size (bit) of all pixel channels

6.1.2.5.25 TcVnRectangle_DINT

Contains origin and size of a rectangle.

Syntax

Definition:

```
TYPE TcVnRectangle_DINT :
STRUCT
    nX      : DINT;
    nY      : DINT;
    nWidth  : DINT;
    nHeight : DINT;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
nX	DINT	X coordinate of the top-left corner
nY	DINT	Y coordinate of the top-left corner
nWidth	DINT	Width
nHeight	DINT	Height

6.1.2.5.26 TcVnRectangle_UDINT

Contains origin and size of a rectangle.

Syntax

Definition:

```
TYPE TcVnRectangle_UDINT :
STRUCT
    nX      : UDINT;
    nY      : UDINT;
    nWidth  : UDINT;
    nHeight : UDINT;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
nX	UDINT	X coordinate of the top-left corner
nY	UDINT	Y coordinate of the top-left corner
nWidth	UDINT	Width
nHeight	UDINT	Height

6.1.2.5.27 TcVnRotatedRectangle

Contains center, size and angle of a rotated rectangle.

Syntax

Definition:

```
TYPE TcVnRotatedRectangle :
STRUCT
    aCenter : TcVnPoint2_REAL;
    stSize  : TcVnSize2_REAL;
    fAngle  : REAL;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
aCenter	TcVnPoint2_REAL [▸ 139]	Center point
stSize	TcVnSize2_REAL [▸ 226]	Size composed of fWidth and fHeight
fAngle	REAL	Angle in degree

6.1.2.5.28 TcVnSize2_REAL

Contains width and height.

Syntax

Definition:

```
TYPE TcVnSize2_REAL :
STRUCT
    fWidth  : REAL;
    fHeight : REAL;
END_STRUCT
END_TYPE
```

Parameters

Name	Type	Description
fWidth	REAL	Width
fHeight	REAL	Height

6.1.3 Interfaces

接口不需要许可证。

接口分组

TwinCAT Vision 库的所有接口按主题分类为以下几组：

- 图像 [▸ 354] ()
- 容器 [▸ 226] ()
- 机器学习 [▸ 397] ()
- 其他 [▸ 401] ()

6.1.3.1 Containers

该组包含用于处理容器 [▸ 132]的接口。

常规信息

容器基本上是几个元素组成的矢量。元素的数量可以动态地改变。路由器内存 [▸ 49]中的相应内存为此自动分配和释放。

容器类型

容器的所有元素都为相同的类型。根据容器包含的元素类型，容器的特点为独特的 GUID。为了便于使用，这些 GUID 由常量 [▸ 139]定义。容器的所有类型名称以“ContainerType_Vector_”开始（“ContainerType_String_SINT”除外）。对于所有使用容器的 API 元素，类型要求都有记录。

容器类型可按以下方式确定：

```
ipContainer.GetElementTypeGuid(nTypeGuid);
```

6.1.3.1.1 Function Blocks

该组包含用于实现某些功能块的接口。

6.1.3.1.1.1 ITcVnCustomContainerOperation_ITcVnContainer

Offers an interface for custom, elementwise container operations (for containers of containers) with up to 3 different containers.

Inheritance Hierarchy

ITcVnCustomContainerOperation_ITcVnContainer

Methods

Name	Origin	Description
Execute [▶ 227]	ITcVnCustomContainerOperation_ITcVnContainer	Executes the custom operation on the container elements.

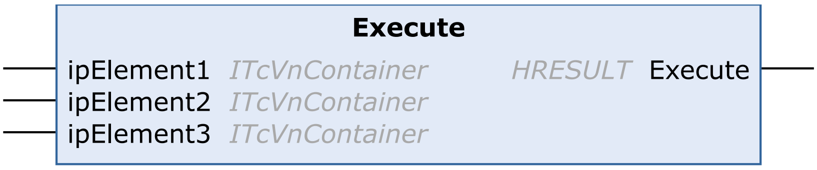
相关函数

- F_VN_CustomElementWiseContainerOperation_ITcVnContainer [▶ 932]

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.1.1.1 Execute



Executes the custom operation on the container elements.

Syntax

Definition:

```
METHOD Execute : HRESULT
VAR_INPUT
    ipElement1 : ITcVnContainer;
    ipElement2 : ITcVnContainer;
    ipElement3 : ITcVnContainer;
END_VAR
```

Inputs

Name	Type	Description
ipElement1	ITcVnContainer [▶ 349]	Current element of the 1st container.
ipElement2	ITcVnContainer [▶ 349]	Current element of the 2nd container.
ipElement3	ITcVnContainer [▶ 349]	Current element of the 3rd container.

Return value

HRESULT [▶ 122]

6.1.3.1.1.2 ITcVnCustomContainerOperation_ITcVnForwardIterator

Offers an interface for custom, elementwise container operations with up to 3 different containers.

Inheritance Hierarchy

ITcVnCustomContainerOperation_ITcVnForwardIterator

Methods

Name	Origin	Description
Execute [▶ 228]	ITcVnCustomContainerOperation_ITcVnForwardIterator	Executes the custom operation on the container elements.

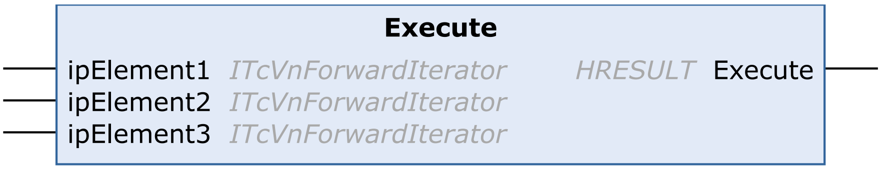
相关函数

- F_VN_CustomElementWiseContainerOperation_ITcVnForwardIterator [▶ 933]

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.1.2.1 Execute



Executes the custom operation on the container elements.

Syntax

Definition:

```
METHOD Execute : HRESULT
VAR_INPUT
    ipElement1 : ITcVnForwardIterator;
    ipElement2 : ITcVnForwardIterator;
    ipElement3 : ITcVnForwardIterator;
END_VAR
```

Inputs

Name	Type	Description
ipElement1	ITcVnForwardIterator [▶ 339]	Current element of the 1st container.
ipElement2	ITcVnForwardIterator [▶ 339]	Current element of the 2nd container.
ipElement3	ITcVnForwardIterator [▶ 339]	Current element of the 3rd container.

Return value

HRESULT [▶ 122]

6.1.3.1.1.3 ITcVnCustomElementCondition_ITcVnContainer

Offers an interface for a custom condition computed for a container.

Inheritance Hierarchy

ITcVnCustomElementCondition_ITcVnContainer

Methods

Name	Origin	Description
Condition [▶ 229]	ITcVnCustomElementCondition_ITcVnContainer	Evaluates the condition for the container.

相关函数

- F_VN_CopyContainerElementsConditional_ITcVnContainer [▶ 722]

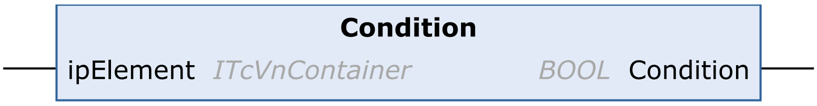
样本

- 容器元素选择 [▶ 2640]

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.1.3.1 Condition



Evaluates the condition for the container.

Syntax

Definition:

```
METHOD Condition : BOOL
VAR_INPUT
    ipElement : ITcVnContainer;
END_VAR
```

Inputs

Name	Type	Description
ipElement	ITcVnContainer [▶ 349]	The container for which the condition is evaluated.

Return value

BOOL

6.1.3.1.1.4 ITcVnCustomElementCondition_ITcVnForwardIterator

Offers an interface for a custom condition computed for an element represented by an ITcVnForwardIterator.

Inheritance Hierarchy

ITcVnCustomElementCondition_ITcVnForwardIterator

Methods

Name	Origin	Description
Condition [▶ 230]	ITcVnCustomElementCondition_ITcVnForwardIterator	Evaluates the condition for the provided element.

相关函数

- F_VN_CopyContainerElementsConditional_ITcVnForwardIterator [▶ 724]

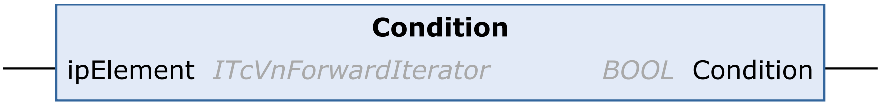
样本

- 容器元素选择 [▶ 2640]

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.1.4.1 Condition



Evaluates the condition for the provided element.

Syntax

Definition:

```
METHOD Condition : BOOL
VAR_INPUT
    ipElement : ITcVnForwardIterator;
END_VAR
```

Inputs

Name	Type	Description
ipElement	ITcVnForwardIterator [▶ 339]	Element for which the condition is evaluated.

Return value

BOOL

6.1.3.1.2 ITcVnAccess

接口ITcVnAccess用于访问容器元素 [▶ 133]，并由迭代器 [▶ 337]提供。与ITcVnRandomAccess [▶ 281]不同，只有相关迭代器在那一刻指向的元素可以被寻址。

6.1.3.1.2.1 ITcVnAccess_DINT

Offers an access interface for DINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
 ITcVnAccess_DINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 231]	ITcVnAccess_DINT	Gets the value
Set [▶ 231]	ITcVnAccess_DINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.1.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    nValue : Reference To DINT;
END_VAR
```

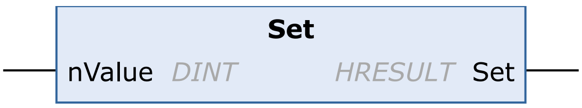
Inputs

Name	Type	Description
nValue	Reference To DINT	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.1.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    nValue : DINT;
END_VAR
```

Inputs

Name	Type	Description
nValue	DINT	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.2 ITcVnAccess_INT

Offers an access interface for INT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnAccess_INT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 232]	ITcVnAccess_INT	Gets the value
Set [▶ 233]	ITcVnAccess_INT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.2.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    nValue : Reference To INT;
END_VAR
```

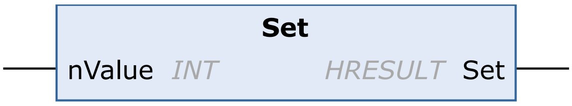
Inputs

Name	Type	Description
nValue	Reference To INT	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.2.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    nValue : INT;
END_VAR
```

Inputs

Name	Type	Description
nValue	INT	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.3 ITcVnAccess_ITcVnImage

Offers an access interface for images.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_ITcVnImage

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 234]	ITcVnAccess_ITcVnImage	Gets the image.
Set [▶ 234]	ITcVnAccess_ITcVnImage	Sets the image.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.3.1 Get

Gets the image.

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    ipImage : Reference To ITcVnImage;
END_VAR
```

Inputs

Name	Type	Description
ipImage	Reference To ITcVnImage [▶ 390]	Returns the image.

Return value

HRESULT [▶ 122]

6.1.3.1.2.3.2 Set

Sets the image.

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    ipImage : ITcVnImage;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	The image to set.

Return value

HRESULT [▶ 122]

6.1.3.1.2.4 ITcVnAccess_LREAL

Offers an access interface for LREAL values.

Inheritance Hierarchy

ITcUnknown [▶ 407]

ITcVnAccess_LREAL

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 235]	ITcVnAccess_LREAL	Gets the value
Set [▶ 235]	ITcVnAccess_LREAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.4.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    fValue : Reference To LREAL;
END_VAR
```

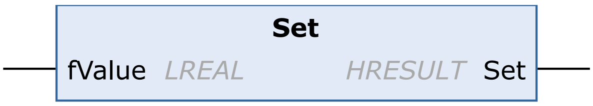
Inputs

Name	Type	Description
fValue	Reference To LREAL	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.4.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    fValue : LREAL;
END_VAR
```

Inputs

Name	Type	Description
fValue	LREAL	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.5 ITcVnAccess_REAL

Offers an access interface for REAL values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnAccess_REAL

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 236]	ITcVnAccess_REAL	Gets the value
Set [▶ 237]	ITcVnAccess_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.5.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    fValue : Reference To REAL;
END_VAR
```

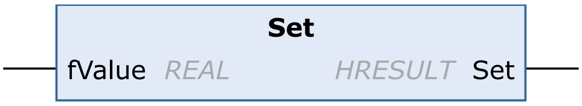
Inputs

Name	Type	Description
fValue	Reference To REAL	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.5.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    fValue : REAL;
END_VAR
```

Inputs

Name	Type	Description
fValue	REAL	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.6 ITcVnAccess_SINT

Offers an access interface for SINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnAccess_SINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 238]	ITcVnAccess_SINT	Gets the value
Set [▶ 238]	ITcVnAccess_SINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.6.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    nValue : Reference To SINT;
END_VAR
```

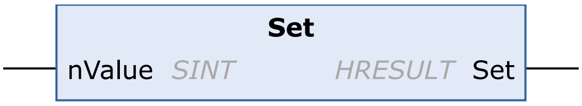
Inputs

Name	Type	Description
nValue	Reference To SINT	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.6.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    nValue : SINT;
END_VAR
```

Inputs

Name	Type	Description
nValue	SINT	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.7 ITcVnAccess_TcVnDMatch

Offers an access interface for TcVnDMatch values.

Inheritance Hierarchy

ITcUnknown [▶ 407]

ITcVnAccess_TcVnDMatch

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 239]	ITcVnAccess_TcVnDMatch	Gets the value
Set [▶ 239]	ITcVnAccess_TcVnDMatch	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.7.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    stDMatch : Reference To TcVnDMatch;
END_VAR
```

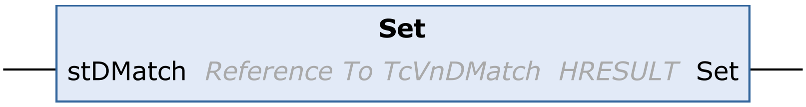
Inputs

Name	Type	Description
stDMatch	Reference To TcVnDMatch [▶ 210]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.7.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    stDMatch : Reference To TcVnDMatch;
END_VAR
```

Inputs

Name	Type	Description
stDMatch	Reference To TcVnDMatch [▶ 210]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.8 ITcVnAccess_TcVnKeyPoint

Offers an access interface for TcVnKeyPoint values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnAccess_TcVnKeyPoint

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 240]	ITcVnAccess_TcVnKeyPoint	Gets the value.
Set [▶ 241]	ITcVnAccess_TcVnKeyPoint	Sets the value.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.8.1 Get

Gets the value.

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    stKeyPoint : Reference To TcVnKeyPoint;
END_VAR
```

Inputs

Name	Type	Description
stKeyPoint	Reference To TcVnKeyPoint [▶ 210]	Returns the value.

Return value

HRESULT [▶ 122]

6.1.3.1.2.8.2 Set

Sets the value.

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    stKeyPoint : Reference To TcVnKeyPoint;
END_VAR
```

Inputs

Name	Type	Description
stKeyPoint	Reference To TcVnKeyPoint [▶ 210]	The value to set.

Return value

HRESULT [▶ 122]

6.1.3.1.2.9 ITcVnAccess_TcVnPoint2_DINT

Offers an access interface for TcVnPoint2_DINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]

ITcVnAccess_TcVnPoint2_DINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 242]	ITcVnAccess_TcVnPoint2_DINT	Gets the value.
Set [▶ 242]	ITcVnAccess_TcVnPoint2_DINT	Sets the value.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.9.1 Get

Gets the value.

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aPoint : Reference To TcVnPoint2_DINT;
END_VAR
```

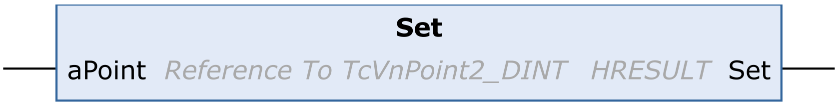
Inputs

Name	Type	Description
aPoint	Reference To TcVnPoint2_DINT [▸ 139]	Returns the value.

Return value

HRESULT [▸ 122]

6.1.3.1.2.9.2 Set



Sets the value.

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aPoint : Reference To TcVnPoint2_DINT;
END_VAR
```

Inputs

Name	Type	Description
aPoint	Reference To TcVnPoint2_DINT [▸ 139]	The value to set.

Return value

HRESULT [▸ 122]

6.1.3.1.2.10 ITcVnAccess_TcVnPoint2_LREAL

Offers an access interface for TcVnPoint2_LREAL values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
        ITcVnAccess_TcVnPoint2_LREAL

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 243]	ITcVnAccess_TcVnPoint2_LREAL	Gets the value
Set [▶ 244]	ITcVnAccess_TcVnPoint2_LREAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.10.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aPoint : Reference To TcVnPoint2_LREAL;
END_VAR
```

Inputs

Name	Type	Description
aPoint	Reference To TcVnPoint2_LREAL [▶ 139]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.10.2 Set

Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aPoint : Reference To TcVnPoint2_LREAL;
END_VAR
```

Inputs

Name	Type	Description
aPoint	Reference To TcVnPoint2_LREAL [▶ 139]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.11 ITcVnAccess_TcVnPoint2_REAL

Offers an access interface for TcVnPoint2_REAL values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnAccess_TcVnPoint2_REAL

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 245]	ITcVnAccess_TcVnPoint2_REAL	Gets the value
Set [▶ 245]	ITcVnAccess_TcVnPoint2_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.11.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aPoint : Reference To TcVnPoint2_REAL;
END_VAR
```

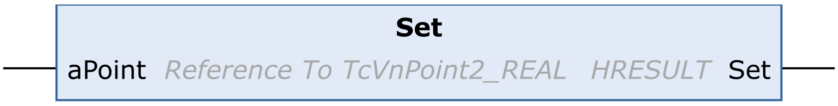
Inputs

Name	Type	Description
aPoint	Reference To TcVnPoint2_REAL [▶ 139]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.11.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aPoint : Reference To TcVnPoint2_REAL;
END_VAR
```

Inputs

Name	Type	Description
aPoint	Reference To TcVnPoint2_REAL [▶ 139]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.12 ITcVnAccess_TcVnPoint3_LREAL

Offers an access interface for TcVnPoint3_LREAL values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
        ITcVnAccess_TcVnPoint3_LREAL

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 246]	ITcVnAccess_TcVnPoint3_LREAL	Gets the value
Set [▶ 246]	ITcVnAccess_TcVnPoint3_LREAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.12.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aPoint : Reference To TcVnPoint3_LREAL;
END_VAR
```

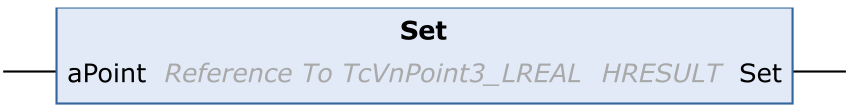
Inputs

Name	Type	Description
aPoint	Reference To TcVnPoint3_LREAL [▶ 139]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.12.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aPoint : Reference To TcVnPoint3_LREAL;
END_VAR
```

Inputs

Name	Type	Description
aPoint	Reference To TcVnPoint3_LREAL [▶ 139]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.13 ITcVnAccess_TcVnPoint3_REAL

Offers an access interface for TcVnPoint3_REAL values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnAccess_TcVnPoint3_REAL

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 247]	ITcVnAccess_TcVnPoint3_REAL	Gets the value
Set [▶ 248]	ITcVnAccess_TcVnPoint3_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.13.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aPoint : Reference To TcVnPoint3_REAL;
END_VAR
```

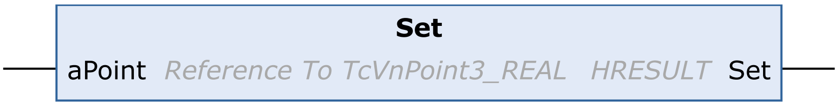
Inputs

Name	Type	Description
aPoint	Reference To TcVnPoint3_REAL [▶ 139]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.13.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aPoint : Reference To TcVnPoint3_REAL;
END_VAR
```

Inputs

Name	Type	Description
aPoint	Reference To TcVnPoint3_REAL [▶ 139]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.14 ITcVnAccess_TcVnRectangle_DINT

Offers an access interface for TcVnRectangle_DINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
        ITcVnAccess_TcVnRectangle_DINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 249]	ITcVnAccess_TcVnRectangle_DINT	Gets the value
Set [▶ 249]	ITcVnAccess_TcVnRectangle_DINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.14.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    stRectangle : Reference To TcVnRectangle_DINT;
END_VAR
```

Inputs

Name	Type	Description
stRectangle	Reference To TcVnRectangle_DINT [▶ 224]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.14.2 Set

Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    stRectangle : Reference To TcVnRectangle_DINT;
END_VAR
```

Inputs

Name	Type	Description
stRectangle	Reference To TcVnRectangle_DINT [▸ 224]	The value to set

Return value

HRESULT [▸ 122]

6.1.3.1.2.15 ITcVnAccess_TcVnVector2_DINT

Offers an access interface for TcVnVector2_DINT values.

Inheritance Hierarchy

ITcUnknown [▸ 407]
    ITcVnAccess_TcVnVector2_DINT

Methods

Name	Origin	Description
TcAddRef [▸ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 408]	ITcUnknown	Decrements the reference counter.
Get [▸ 250]	ITcVnAccess_TcVnVector2_DINT	Gets the value
Set [▸ 251]	ITcVnAccess_TcVnVector2_DINT	Sets the value.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.15.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector2_DINT;
END_VAR
```

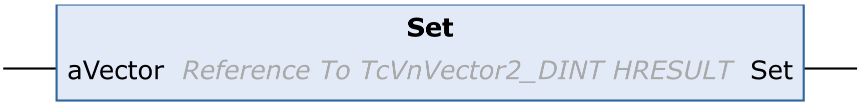
Inputs

Name	Type	Description
aVector	Reference To TcVnVector2_DINT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.15.2 Set



Sets the value.

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector2_DINT;
END_VAR
```

Inputs

Name	Type	Description
aVector	Reference To TcVnVector2_DINT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.16 ITcVnAccess_TcVnVector2_INT

Offers an access interface for TcVnVector2_INT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnAccess_TcVnVector2_INT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 252]	ITcVnAccess_TcVnVector2_INT	Gets the value
Set [▶ 252]	ITcVnAccess_TcVnVector2_INT	Sets the value.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.16.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector2_INT;
END_VAR
```

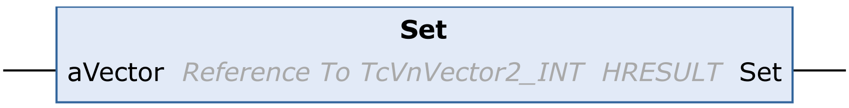
Inputs

Name	Type	Description
aVector	Reference To TcVnVector2_INT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.16.2 Set



Sets the value.

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector2_INT;
END_VAR
```

Inputs

Name	Type	Description
aVector	Reference To TcVnVector2_INT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.17 ITcVnAccess_TcVnVector2_REAL

Offers an access interface for TcVnVector2_REAL values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_TcVnVector2_REAL

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 253]	ITcVnAccess_TcVnVector2_REAL	Gets the value
Set [▶ 254]	ITcVnAccess_TcVnVector2_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.17.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector2_REAL;
END_VAR
```

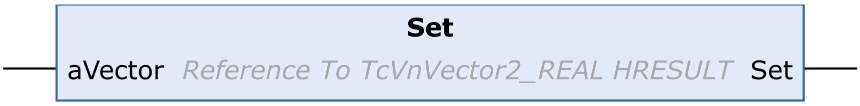
Inputs

Name	Type	Description
aVector	Reference To TcVnVector2_REAL [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.17.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector2_REAL;
END_VAR
```

Inputs

Name	Type	Description
aVector	Reference To TcVnVector2_REAL [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.18 ITcVnAccess_TcVnVector2_SINT

Offers an access interface for TcVnVector2_SINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnAccess_TcVnVector2_SINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 255]	ITcVnAccess_TcVnVector2_SINT	Gets the value
Set [▶ 255]	ITcVnAccess_TcVnVector2_SINT	Sets the value.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.18.1 Get

Get
aVector *Reference To TcVnVector2_SINT HRESULT* Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector2_SINT;
END_VAR
```

Inputs

Name	Type	Description
aVector	Reference To TcVnVector2_SINT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.18.2 Set

Set
aVector *Reference To TcVnVector2_SINT HRESULT* Set

Sets the value.

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector2_SINT;
END_VAR
```

Inputs

Name	Type	Description
aVector	Reference To TcVnVector2_SINT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.19 ITcVnAccess_TcVnVector2_UINT

Offers an access interface for TcVnVector2_UINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
        ITcVnAccess_TcVnVector2_UINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 256]	ITcVnAccess_TcVnVector2_UINT	Gets the value
Set [▶ 257]	ITcVnAccess_TcVnVector2_UINT	Sets the value.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.19.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector2_UINT;
END_VAR
```

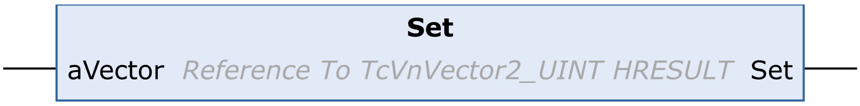
Inputs

Name	Type	Description
aVector	Reference To TcVnVector2_UINT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.19.2 Set



Sets the value.

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector2_UINT;
END_VAR
```

Inputs

Name	Type	Description
aVector	Reference To TcVnVector2_UINT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.20 ITcVnAccess_TcVnVector2_USINT

Offers an access interface for TcVnVector2_USINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnAccess_TcVnVector2_USINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 258]	ITcVnAccess_TcVnVector2_USINT	Gets the value
Set [▶ 258]	ITcVnAccess_TcVnVector2_USINT	Sets the value.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.20.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector2_USINT;
END_VAR
```

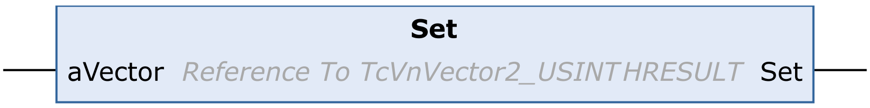
Inputs

Name	Type	Description
aVector	Reference To TcVnVector2_USINT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.20.2 Set



Sets the value.

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector2_USINT;
END_VAR
```

Inputs

Name	Type	Description
aVector	Reference To TcVnVector2_USINT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.21 ITcVnAccess_TcVnVector3_INT

Offers an access interface for TcVnVector3_INT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
        ITcVnAccess_TcVnVector3_INT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 259]	ITcVnAccess_TcVnVector3_INT	Gets the value
Set [▶ 260]	ITcVnAccess_TcVnVector3_INT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.21.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector3_INT;
END_VAR
```

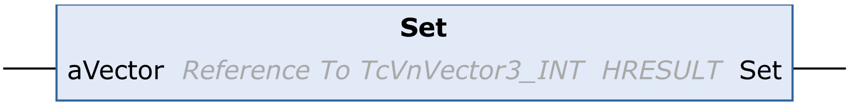
Inputs

Name	Type	Description
aVector	Reference To TcVnVector3_INT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.21.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector3_INT;
END_VAR
```

Inputs

Name	Type	Description
aVector	Reference To TcVnVector3_INT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.22 ITcVnAccess_TcVnVector3_REAL

Offers an access interface for TcVnVector3_REAL values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
        ITcVnAccess_TcVnVector3_REAL

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 261]	ITcVnAccess_TcVnVector3_REAL	Gets the value
Set [▶ 261]	ITcVnAccess_TcVnVector3_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.22.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector3_REAL;
END_VAR
```

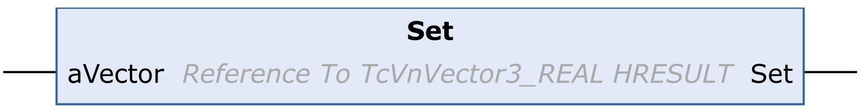
Inputs

Name	Type	Description
aVector	Reference To TcVnVector3_REAL [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.22.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector3_REAL;
END_VAR
```

Inputs

Name	Type	Description
aVector	Reference To TcVnVector3_REAL [▸ 141]	The value to set

Return value

HRESULT [▸ 122]

6.1.3.1.2.23 ITcVnAccess_TcVnVector3_SINT

Offers an access interface for TcVnVector3_SINT values.

Inheritance Hierarchy

ITcUnknown [▸ 407]
ITcVnAccess_TcVnVector3_SINT

Methods

Name	Origin	Description
TcAddRef [▸ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 408]	ITcUnknown	Decrements the reference counter.
Get [▸ 262]	ITcVnAccess_TcVnVector3_SINT	Gets the value
Set [▸ 263]	ITcVnAccess_TcVnVector3_SINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.23.1 Get

Get
aVector *Reference To TcVnVector3_SINT HRESULT* Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector3_SINT;
END_VAR
```

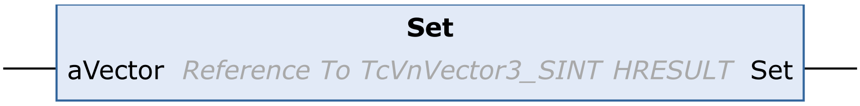
Inputs

Name	Type	Description
aVector	Reference To TcVnVector3_SINT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.23.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector3_SINT;
END_VAR
```

Inputs

Name	Type	Description
aVector	Reference To TcVnVector3_SINT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.24 ITcVnAccess_TcVnVector3_UINT

Offers an access interface for TcVnVector3_UINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_TcVnVector3_UINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 264]	ITcVnAccess_TcVnVector3_UINT	Gets the value
Set [▶ 264]	ITcVnAccess_TcVnVector3_UINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.24.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector3_UINT;
END_VAR
```

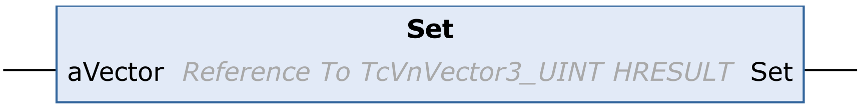
Inputs

Name	Type	Description
aVector	Reference To TcVnVector3_UINT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.24.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector3_UINT;
END_VAR
```

Inputs

Name	Type	Description
aVector	Reference To TcVnVector3_UINT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.25 ITcVnAccess_TcVnVector3_USINT

Offers an access interface for TcVnVector3_USINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnAccess_TcVnVector3_USINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 265]	ITcVnAccess_TcVnVector3_USINT	Gets the value
Set [▶ 266]	ITcVnAccess_TcVnVector3_USINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.25.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector3_USINT;
END_VAR
```

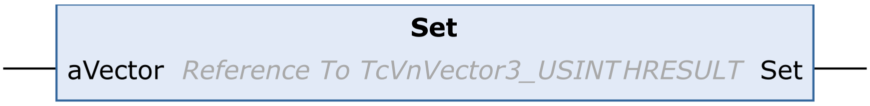
Inputs

Name	Type	Description
aVector	Reference To TcVnVector3_USINT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.25.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector3_USINT;
END_VAR
```

Inputs

Name	Type	Description
aVector	Reference To TcVnVector3_USINT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.26 ITcVnAccess_TcVnVector4_DINT

Offers an access interface for TcVnVector4_DINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_TcVnVector4_DINT

Methods

Name	Origin	Description
TcAddRef [▸ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 408]	ITcUnknown	Decrements the reference counter.
Get [▸ 267]	ITcVnAccess_TcVnVector4_DINT	Gets the value
Set [▸ 267]	ITcVnAccess_TcVnVector4_DINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.26.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector4_DINT;
END_VAR
```

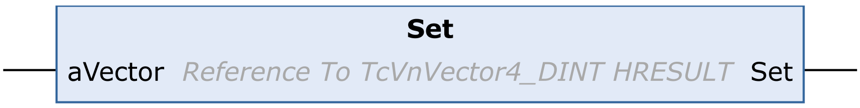
Inputs

Name	Type	Description
aVector	Reference To TcVnVector4_DINT [▸ 141]	Returns the value

Return value

HRESULT [▸ 122]

6.1.3.1.2.26.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector4_DINT;
END_VAR
```

Inputs

Name	Type	Description
aVector	Reference To TcVnVector4_DINT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.27 ITcVnAccess_TcVnVector4_INT

Offers an access interface for TcVnVector4_INT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnAccess_TcVnVector4_INT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 268]	ITcVnAccess_TcVnVector4_INT	Gets the value
Set [▶ 269]	ITcVnAccess_TcVnVector4_INT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.27.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector4_INT;
END_VAR
```

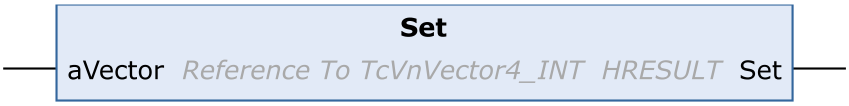
Inputs

Name	Type	Description
aVector	Reference To TcVnVector4_INT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.27.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector4_INT;
END_VAR
```

Inputs

Name	Type	Description
aVector	Reference To TcVnVector4_INT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.28 ITcVnAccess_TcVnVector4_LREAL

Offers an access interface for TcVnVector4_LREAL values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_TcVnVector4_LREAL

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 270]	ITcVnAccess_TcVnVector4_LREAL	Gets the value
Set [▶ 270]	ITcVnAccess_TcVnVector4_LREAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.28.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector4_LREAL;
END_VAR
```

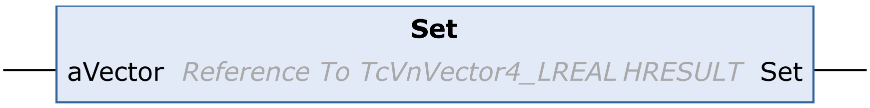
Inputs

Name	Type	Description
aVector	Reference To TcVnVector4_LREAL [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.28.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector4_LREAL;
END_VAR
```

Inputs

Name	Type	Description
aVector	Reference To TcVnVector4_LREAL [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.29 ITcVnAccess_TcVnVector4_SINT

Offers an access interface for TcVnVector4_SINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_TcVnVector4_SINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 271]	ITcVnAccess_TcVnVector4_SINT	Gets the value
Set [▶ 272]	ITcVnAccess_TcVnVector4_SINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.29.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector4_SINT;
END_VAR
```

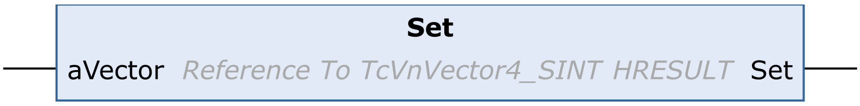
Inputs

Name	Type	Description
aVector	Reference To TcVnVector4_SINT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.29.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector4_SINT;
END_VAR
```

Inputs

Name	Type	Description
aVector	Reference To TcVnVector4_SINT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.30 ITcVnAccess_TcVnVector4_UINT

Offers an access interface for TcVnVector4_UINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_TcVnVector4_UINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 273]	ITcVnAccess_TcVnVector4_UINT	Gets the value
Set [▶ 273]	ITcVnAccess_TcVnVector4_UINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.30.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector4_UINT;
END_VAR
```

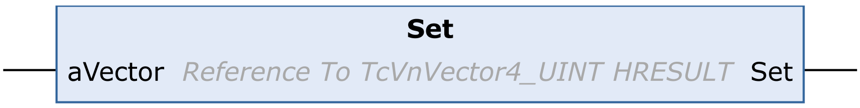
Inputs

Name	Type	Description
aVector	Reference To TcVnVector4_UINT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.30.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector4_UINT;
END_VAR
```

Inputs

Name	Type	Description
aVector	Reference To TcVnVector4_UINT [▸ 141]	The value to set

Return value

HRESULT [▸ 122]

6.1.3.1.2.31 ITcVnAccess_TcVnVector4_USINT

Offers an access interface for TcVnVector4_USINT values.

Inheritance Hierarchy

ITcUnknown [▸ 407]
        ITcVnAccess_TcVnVector4_USINT

Methods

Name	Origin	Description
TcAddRef [▸ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 408]	ITcUnknown	Decrements the reference counter.
Get [▸ 274]	ITcVnAccess_TcVnVector4_USINT	Gets the value
Set [▸ 275]	ITcVnAccess_TcVnVector4_USINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.31.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector4_USINT;
END_VAR
```

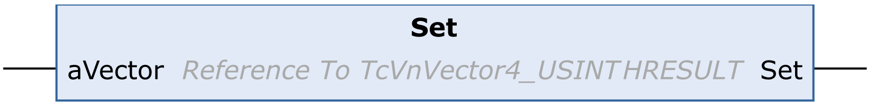
Inputs

Name	Type	Description
aVector	Reference To TcVnVector4_USINT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.31.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    aVector : Reference To TcVnVector4_USINT;
END_VAR
```

Inputs

Name	Type	Description
aVector	Reference To TcVnVector4_USINT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.32 ITcVnAccess_UDINT

Offers an access interface for UDINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_UDINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 276]	ITcVnAccess_UDINT	Gets the value
Set [▶ 276]	ITcVnAccess_UDINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.32.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    nValue : Reference To UDINT;
END_VAR
```

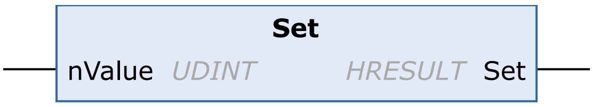
Inputs

Name	Type	Description
nValue	Reference To UDINT	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.32.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    nValue : UDINT;
END_VAR
```

Inputs

Name	Type	Description
nValue	UDINT	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.33 ITcVnAccess_UINT

Offers an access interface for UINT values.

Inheritance Hierarchy

ITcUnknown [▸ 407]
        ITcVnAccess_UINT

Methods

Name	Origin	Description
TcAddRef [▸ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 408]	ITcUnknown	Decrements the reference counter.
Get [▸ 277]	ITcVnAccess_UINT	Gets the value
Set [▸ 277]	ITcVnAccess_UINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.33.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    nValue : Reference To UINT;
END_VAR
```

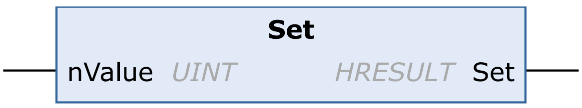
Inputs

Name	Type	Description
nValue	Reference To UINT	Returns the value

Return value

HRESULT [▸ 122]

6.1.3.1.2.33.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    nValue : UINT;
END_VAR
```

Inputs

Name	Type	Description
nValue	UINT	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.34 ITcVnAccess_ULINT

Offers an access interface for ULINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnAccess_ULINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 278]	ITcVnAccess_ULINT	Gets the value
Set [▶ 279]	ITcVnAccess_ULINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.34.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    nValue : Reference To ULINT;
END_VAR
```

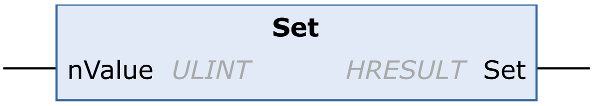
Inputs

Name	Type	Description
nValue	Reference To ULINT	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.34.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    nValue : ULINT;
END_VAR
```

Inputs

Name	Type	Description
nValue	ULINT	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.2.35 ITcVnAccess_USINT

Offers an access interface for USINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_USINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 280]	ITcVnAccess_USINT	Gets the value
Set [▶ 280]	ITcVnAccess_USINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.2.35.1 Get

Gets the value

Syntax

Definition:

```
METHOD Get : HRESULT
VAR_INPUT
    nValue : Reference To USINT;
END_VAR
```

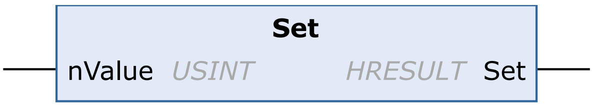
Inputs

Name	Type	Description
nValue	Reference To USINT	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.2.35.2 Set



Sets the value

Syntax

Definition:

```
METHOD Set : HRESULT
VAR_INPUT
    nValue : USINT;
END_VAR
```

Inputs

Name	Type	Description
nValue	USINT	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3 ITcVnRandomAccess

接口ITcVnRandomAccess用于访问容器的任何元素 [▶ 133]，并由ITcVnRandomAccessIterator [▶ 345]提供。与ITcVnAccess [▶ 230]不同，容器中的任何元素都可以被寻址。

6.1.3.1.3.1 ITcVnRandomAccess_DINT

Offers a random access interface for DINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_DINT [▶ 230]
        ITcVnRandomAccess_DINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 231]	ITcVnAccess_DINT	Gets the value
Set [▶ 231]	ITcVnAccess_DINT	Sets the value
GetAt [▶ 281]	ITcVnRandomAccess_DINT	Gets the value
SetAt [▶ 282]	ITcVnRandomAccess_DINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.1.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    nValue  : Reference To DINT;
END_VAR
```

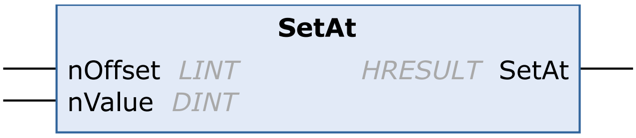
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
nValue	Reference To DINT	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.1.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    nValue  : DINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
nValue	DINT	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.2 ITcVnRandomAccess_INT

Offers a random access interface for INT values

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_INT [▶ 232]
        ITcVnRandomAccess_INT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 232]	ITcVnAccess_INT	Gets the value
Set [▶ 233]	ITcVnAccess_INT	Sets the value
GetAt [▶ 283]	ITcVnRandomAccess_INT	Gets the value
SetAt [▶ 283]	ITcVnRandomAccess_INT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.2.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    nValue  : Reference To INT;
END_VAR
```

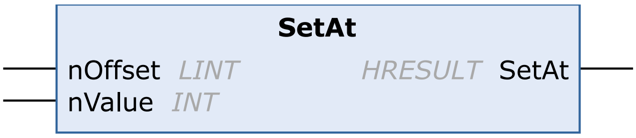
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
nValue	Reference To INT	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.2.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    nValue  : INT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
nValue	INT	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.3 ITcVnRandomAccess_ITcVnImage

Offers an random access interface for images.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnAccess_ITcVnImage [▶ 233]
ITcVnRandomAccess_ITcVnImage

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 234]	ITcVnAccess_ITcVnImage	Gets the image.
Set [▶ 234]	ITcVnAccess_ITcVnImage	Sets the image.
GetAt [▶ 284]	ITcVnRandomAccess_ITcVnImage	Gets the image.
SetAt [▶ 285]	ITcVnRandomAccess_ITcVnImage	Sets the image.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.3.1 GetAt

Gets the image.

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    ipImage : Reference To ITcVnImage;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position.
ipImage	Reference To ITcVnImage [▶ 390]	Returns the image.

Return value

HRESULT [▶ 122]

6.1.3.1.3.3.2 SetAt

Sets the image.

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    ipImage : ITcVnImage;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
ipImage	ITcVnImage [▶ 390]	The image to set.

Return value

HRESULT [▶ 122]

6.1.3.1.3.4 ITcVnRandomAccess_LREAL

Offers a random access interface for LREAL values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_LREAL [▶ 234]
        ITcVnRandomAccess_LREAL

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 235]	ITcVnAccess_LREAL	Gets the value
Set [▶ 235]	ITcVnAccess_LREAL	Sets the value
GetAt [▶ 286]	ITcVnRandomAccess_LREAL	Gets the value
SetAt [▶ 286]	ITcVnRandomAccess_LREAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.4.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    fValue  : Reference To LREAL;
END_VAR
```

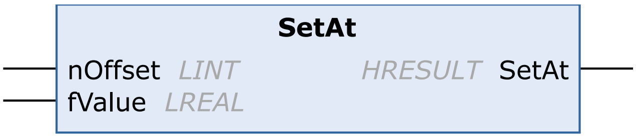
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
fValue	Reference To LREAL	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.4.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    fValue  : LREAL;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position.
fValue	LREAL	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.5 ITcVnRandomAccess_REAL

Offers a random access interface for REAL values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_REAL [▶ 236]
        ITcVnRandomAccess_REAL

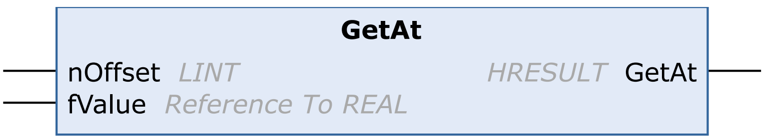
Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 236]	ITcVnAccess_REAL	Gets the value
Set [▶ 237]	ITcVnAccess_REAL	Sets the value
GetAt [▶ 287]	ITcVnRandomAccess_REAL	Gets the value
SetAt [▶ 288]	ITcVnRandomAccess_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.5.1 GetAt



Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    fValue  : Reference To REAL;
END_VAR
```

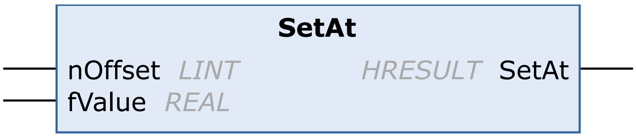
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
fValue	Reference To REAL	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.5.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    fValue  : REAL;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
fValue	REAL	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.6 ITcVnRandomAccess_SINT

Offers a random access interface for SINT values

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnAccess_SINT [▶ 237]
ITcVnRandomAccess_SINT

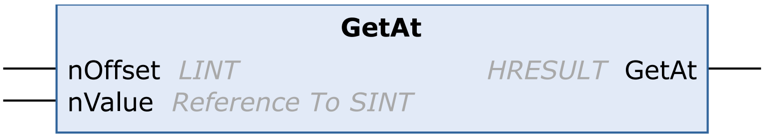
Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 238]	ITcVnAccess_SINT	Gets the value
Set [▶ 238]	ITcVnAccess_SINT	Sets the value
GetAt [▶ 289]	ITcVnRandomAccess_SINT	Gets the value
SetAt [▶ 289]	ITcVnRandomAccess_SINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.6.1 GetAt



Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    nValue  : Reference To SINT;
END_VAR
```

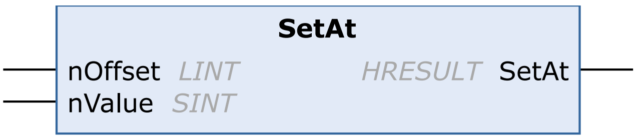
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
nValue	Reference To SINT	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.6.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    nValue  : SINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
nValue	SINT	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.7 ITcVnRandomAccess_TcVnDMatch

Offers an random access interface for TcVnDMatch values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
        ITcVnAccess_TcVnDMatch [▶ 238]
                ITcVnRandomAccess_TcVnDMatch

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 239]	ITcVnAccess_TcVnDMatch	Gets the value
Set [▶ 239]	ITcVnAccess_TcVnDMatch	Sets the value
GetAt [▶ 290]	ITcVnRandomAccess_TcVnDMatch	Gets the value
SetAt [▶ 291]	ITcVnRandomAccess_TcVnDMatch	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.7.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset  : LINT;
    stDMatch : Reference To TcVnDMatch;
END_VAR
```

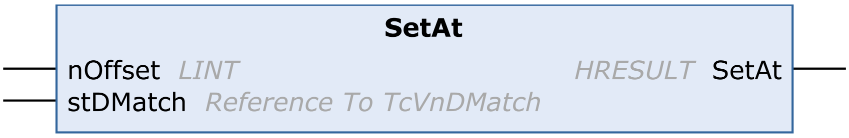
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position.
stDMatch	Reference To TcVnDMatch [▶ 210]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.7.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset  : LINT;
    stDMatch : Reference To TcVnDMatch;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
stDMatch	Reference To TcVnDMatch [▶ 210]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.8 ITcVnRandomAccess_TcVnKeyPoint

Offers a random access interface for TcVnKeyPoint values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
        ITcVnAccess_TcVnKeyPoint [▶ 240]
                ITcVnRandomAccess_TcVnKeyPoint

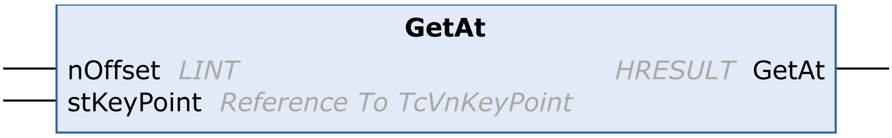
Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 240]	ITcVnAccess_TcVnKeyPoint	Gets the value.
Set [▶ 241]	ITcVnAccess_TcVnKeyPoint	Sets the value.
GetAt [▶ 292]	ITcVnRandomAccess_TcVnKeyPoint	Gets the value
SetAt [▶ 292]	ITcVnRandomAccess_TcVnKeyPoint	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.8.1 GetAt



Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset    : LINT;
    stKeyPoint : Reference To TcVnKeyPoint;
END_VAR
```

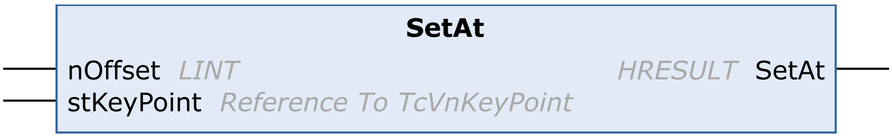
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
stKeyPoint	Reference To TcVnKeyPoint [▸ 210]	Returns the value

Return value

HRESULT [▸ 122]

6.1.3.1.3.8.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset    : LINT;
    stKeyPoint : Reference To TcVnKeyPoint;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
stKeyPoint	Reference To TcVnKeyPoint [▸ 210]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.9 ITcVnRandomAccess_TcVnPoint2_DINT

Offers a random access interface for TcVnPoint2_DINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_TcVnPoint2_DINT [▶ 241]
        ITcVnRandomAccess_TcVnPoint2_DINT

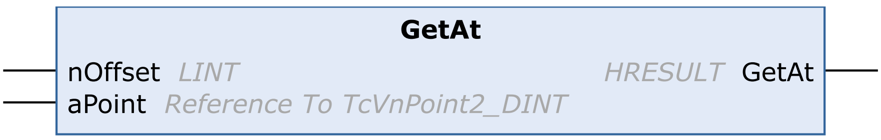
Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 242]	ITcVnAccess_TcVnPoint2_DINT	Gets the value.
Set [▶ 242]	ITcVnAccess_TcVnPoint2_DINT	Sets the value.
GetAt [▶ 293]	ITcVnRandomAccess_TcVnPoint2_DINT	Gets the value
SetAt [▶ 294]	ITcVnRandomAccess_TcVnPoint2_DINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.9.1 GetAt



Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aPoint  : Reference To TcVnPoint2_DINT;
END_VAR
```

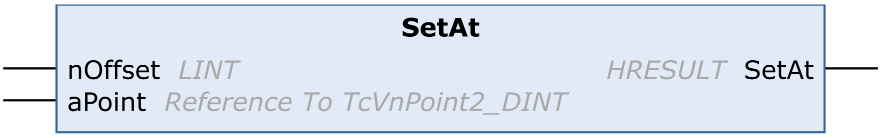
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aPoint	Reference To TcVnPoint2_DINT [▶ 139]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.9.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aPoint  : Reference To TcVnPoint2_DINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aPoint	Reference To TcVnPoint2_DINT [▶ 139]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.10 ITcVnRandomAccess_TcVnPoint2_LREAL

Offers a random access interface for TcVnPoint2_LREAL values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_TcVnPoint2_LREAL [▶ 243]
        ITcVnRandomAccess_TcVnPoint2_LREAL

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 243]	ITcVnAccess_TcVnPoint2_LREAL	Gets the value
Set [▶ 244]	ITcVnAccess_TcVnPoint2_LREAL	Sets the value
GetAt [▶ 295]	ITcVnRandomAccess_TcVnPoint2_LREAL	Gets the value
SetAt [▶ 295]	ITcVnRandomAccess_TcVnPoint2_LREAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.10.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aPoint  : Reference To TcVnPoint2_LREAL;
END_VAR
```

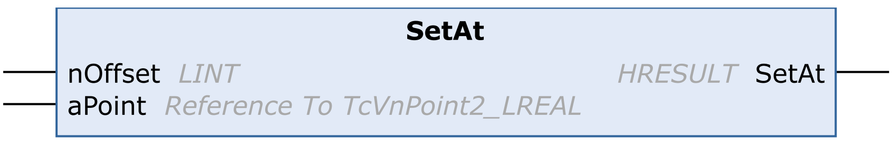
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aPoint	Reference To TcVnPoint2_LREAL [▶ 139]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.10.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aPoint  : Reference To TcVnPoint2_LREAL;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aPoint	Reference To TcVnPoint2_LREAL [▶ 139]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.11 ITcVnRandomAccess_TcVnPoint2_REAL

Offers a random access interface for TcVnPoint2_REAL values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_TcVnPoint2_REAL [▶ 244]
        ITcVnRandomAccess_TcVnPoint2_REAL

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 245]	ITcVnAccess_TcVnPoint2_REAL	Gets the value
Set [▶ 245]	ITcVnAccess_TcVnPoint2_REAL	Sets the value
GetAt [▶ 297]	ITcVnRandomAccess_TcVnPoint2_REAL	Gets the value
SetAt [▶ 297]	ITcVnRandomAccess_TcVnPoint2_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.11.1 GetAt

GetAt
nOffset *LINT* — *HRESULT* GetAt
aPoint *Reference To TcVnPoint2_REAL*

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aPoint  : Reference To TcVnPoint2_REAL;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aPoint	Reference To TcVnPoint2_REAL [▶ 139]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.11.2 SetAt

SetAt
nOffset *LINT* — *HRESULT* SetAt
aPoint *Reference To TcVnPoint2_REAL*

Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aPoint  : Reference To TcVnPoint2_REAL;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aPoint	Reference To TcVnPoint2_REAL [▶ 139]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.12 ITcVnRandomAccess_TcVnPoint3_LREAL

Offers a random access interface for TcVnPoint3_LREAL values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
        ITcVnAccess_TcVnPoint3_LREAL [▶ 245]
                ITcVnRandomAccess_TcVnPoint3_LREAL

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 246]	ITcVnAccess_TcVnPoint3_LREAL	Gets the value
Set [▶ 246]	ITcVnAccess_TcVnPoint3_LREAL	Sets the value
GetAt [▶ 298]	ITcVnRandomAccess_TcVnPoint3_LREAL	Gets the value
SetAt [▶ 299]	ITcVnRandomAccess_TcVnPoint3_LREAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.12.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aPoint  : Reference To TcVnPoint3_LREAL;
END_VAR
```

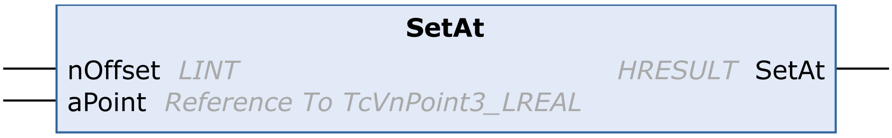
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aPoint	Reference To TcVnPoint3_LREAL [▶ 139]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.12.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aPoint  : Reference To TcVnPoint3_LREAL;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aPoint	Reference To TcVnPoint3_LREAL [▶ 139]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.13 ITcVnRandomAccess_TcVnPoint3_REAL

Offers a random access interface for TcVnPoint3_REAL values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_TcVnPoint3_REAL [▶ 247]
        ITcVnRandomAccess_TcVnPoint3_REAL

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 247]	ITcVnAccess_TcVnPoint3_REAL	Gets the value
Set [▶ 248]	ITcVnAccess_TcVnPoint3_REAL	Sets the value
GetAt [▶ 300]	ITcVnRandomAccess_TcVnPoint3_REAL	Gets the value
SetAt [▶ 300]	ITcVnRandomAccess_TcVnPoint3_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.13.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aPoint  : Reference To TcVnPoint3_REAL;
END_VAR
```

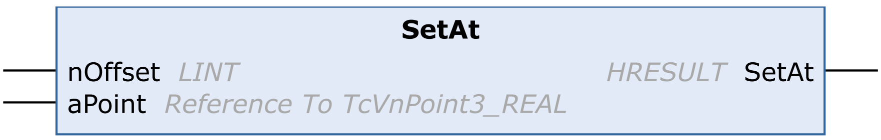
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aPoint	Reference To TcVnPoint3_REAL [▶ 139]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.13.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aPoint  : Reference To TcVnPoint3_REAL;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aPoint	Reference To TcVnPoint3_REAL [▸ 139]	The value to set

Return value

HRESULT [▸ 122]

6.1.3.1.3.14 ITcVnRandomAccess_TcVnRectangle_DINT

Offers a random access interface for TcVnRectangle_DINT values.

Inheritance Hierarchy

ITcUnknown [▸ 407]
ITcVnAccess_TcVnRectangle_DINT [▸ 248]
ITcVnRandomAccess_TcVnRectangle_DINT

Methods

Name	Origin	Description
TcAddRef [▸ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 408]	ITcUnknown	Decrements the reference counter.
Get [▸ 249]	ITcVnAccess_TcVnRectangle_DINT	Gets the value
Set [▸ 249]	ITcVnAccess_TcVnRectangle_DINT	Sets the value
GetAt [▸ 302]	ITcVnRandomAccess_TcVnRectangle_DINT	Gets the value
SetAt [▸ 302]	ITcVnRandomAccess_TcVnRectangle_DINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.14.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset     : LINT;
    stRectangle : Reference To TcVnRectangle_DINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
stRectangle	Reference To TcVnRectangle_DINT [▸ 224]	Returns the value

Return value

HRESULT [▸ 122]

6.1.3.1.3.14.2 SetAt

Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset     : LINT;
    stRectangle : Reference To TcVnRectangle_DINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
stRectangle	Reference To TcVnRectangle_DINT [▸ 224]	The value to set

Return value

HRESULT [▸ 122]

6.1.3.1.3.15 ITcVnRandomAccess_TcVnVector2_DINT

Offers a random access interface for TcVnVector2_DINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
        ITcVnAccess_TcVnVector2_DINT [▶ 250]
                ITcVnRandomAccess_TcVnVector2_DINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 250]	ITcVnAccess_TcVnVector2_DINT	Gets the value
Set [▶ 251]	ITcVnAccess_TcVnVector2_DINT	Sets the value.
GetAt [▶ 303]	ITcVnRandomAccess_TcVnVector2_DINT	Gets the value
SetAt [▶ 304]	ITcVnRandomAccess_TcVnVector2_DINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.15.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector2_DINT;
END_VAR
```

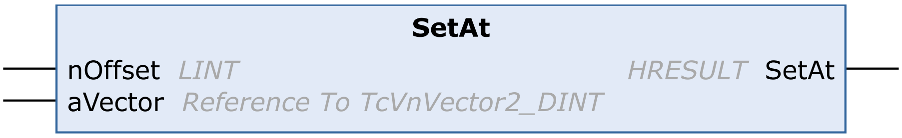
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector2_DINT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.15.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector2_DINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector2_DINT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.16 ITcVnRandomAccess_TcVnVector2_INT

Offers a random access interface for TcVnVector2_INT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_TcVnVector2_INT [▶ 251]
        ITcVnRandomAccess_TcVnVector2_INT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 252]	ITcVnAccess_TcVnVector2_INT	Gets the value
Set [▶ 252]	ITcVnAccess_TcVnVector2_INT	Sets the value.
GetAt [▶ 305]	ITcVnRandomAccess_TcVnVector2_INT	Gets the value
SetAt [▶ 305]	ITcVnRandomAccess_TcVnVector2_INT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.16.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector2_INT;
END_VAR
```

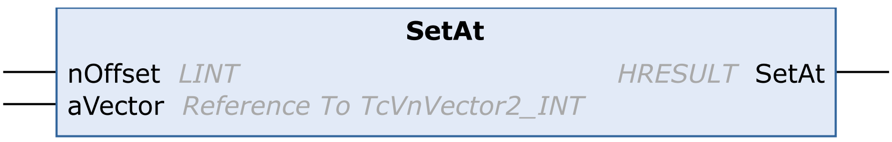
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector2_INT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.16.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector2_INT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector2_INT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.17 ITcVnRandomAccess_TcVnVector2_REAL

Offers a random access interface for TcVnVector2_REAL values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_TcVnVector2_REAL [▶ 253]
        ITcVnRandomAccess_TcVnVector2_REAL

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 253]	ITcVnAccess_TcVnVector2_REAL	Gets the value
Set [▶ 254]	ITcVnAccess_TcVnVector2_REAL	Sets the value
GetAt [▶ 307]	ITcVnRandomAccess_TcVnVector2_REAL	Gets the value
SetAt [▶ 307]	ITcVnRandomAccess_TcVnVector2_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.17.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector2_REAL;
END_VAR
```

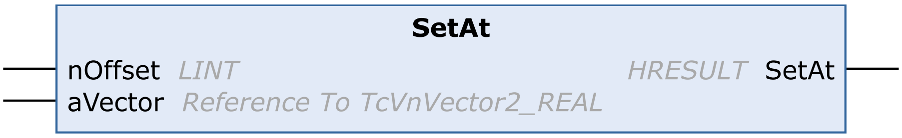
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector2_REAL [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.17.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector2_REAL;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector2_REAL [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.18 ITcVnRandomAccess_TcVnVector2_SINT

Offers a random access interface for TcVnVector2_SINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_TcVnVector2_SINT [▶ 254]
        ITcVnRandomAccess_TcVnVector2_SINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 255]	ITcVnAccess_TcVnVector2_SINT	Gets the value
Set [▶ 255]	ITcVnAccess_TcVnVector2_SINT	Sets the value.
GetAt [▶ 308]	ITcVnRandomAccess_TcVnVector2_SINT	Gets the value
SetAt [▶ 309]	ITcVnRandomAccess_TcVnVector2_SINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.18.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector2_SINT;
END_VAR
```

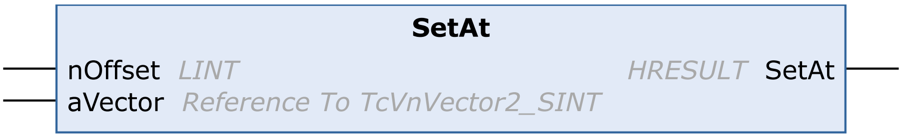
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector2_SINT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.18.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector2_SINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector2_SINT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.19 ITcVnRandomAccess_TcVnVector2_UINT

Offers a random access interface for TcVnVector2_UINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_TcVnVector2_UINT [▶ 256]
        ITcVnRandomAccess_TcVnVector2_UINT

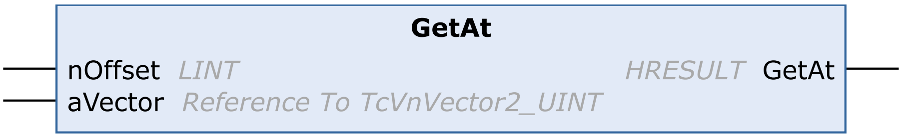
Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 256]	ITcVnAccess_TcVnVector2_UINT	Gets the value
Set [▶ 257]	ITcVnAccess_TcVnVector2_UINT	Sets the value.
GetAt [▶ 310]	ITcVnRandomAccess_TcVnVector2_UINT	Gets the value
SetAt [▶ 310]	ITcVnRandomAccess_TcVnVector2_UINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.19.1 GetAt



Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector2_UINT;
END_VAR
```

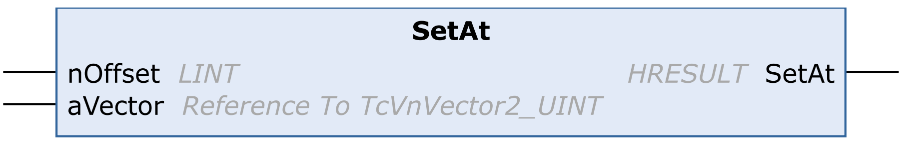
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector2_UINT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.19.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector2_UINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector2_UINT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.20 ITcVnRandomAccess_TcVnVector2_USINT

Offers a random access interface for TcVnVector2_USINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_TcVnVector2_USINT [▶ 257]
        ITcVnRandomAccess_TcVnVector2_USINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 258]	ITcVnAccess_TcVnVector2_USINT	Gets the value
Set [▶ 258]	ITcVnAccess_TcVnVector2_USINT	Sets the value.
GetAt [▶ 312]	ITcVnRandomAccess_TcVnVector2_USINT	Gets the value
SetAt [▶ 312]	ITcVnRandomAccess_TcVnVector2_USINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.20.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector2_USINT;
END_VAR
```

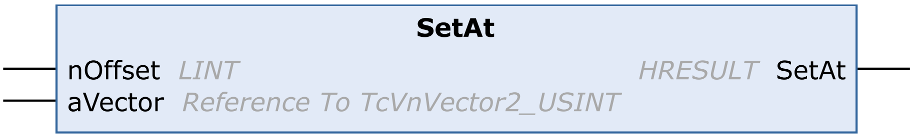
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector2_USINT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.20.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector2_USINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector2_USINT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.21 ITcVnRandomAccess_TcVnVector3_INT

Offers a random access interface for TcVnVector3_INT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_TcVnVector3_INT [▶ 259]
        ITcVnRandomAccess_TcVnVector3_INT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 259]	ITcVnAccess_TcVnVector3_INT	Gets the value
Set [▶ 260]	ITcVnAccess_TcVnVector3_INT	Sets the value
GetAt [▶ 313]	ITcVnRandomAccess_TcVnVector3_INT	Gets the value
SetAt [▶ 314]	ITcVnRandomAccess_TcVnVector3_INT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.21.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector3_INT;
END_VAR
```

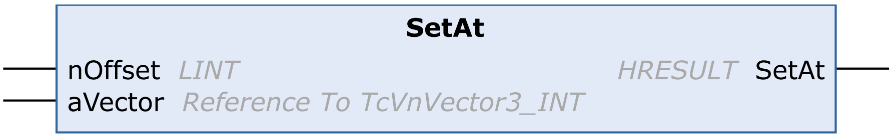
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector3_INT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.21.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector3_INT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector3_INT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.22 ITcVnRandomAccess_TcVnVector3_REAL

Offers a random access interface for TcVnVector3_REAL values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
        ITcVnAccess_TcVnVector3_REAL [▶ 260]
                ITcVnRandomAccess_TcVnVector3_REAL

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 261]	ITcVnAccess_TcVnVector3_REAL	Gets the value
Set [▶ 261]	ITcVnAccess_TcVnVector3_REAL	Sets the value
GetAt [▶ 315]	ITcVnRandomAccess_TcVnVector3_REAL	Gets the value
SetAt [▶ 315]	ITcVnRandomAccess_TcVnVector3_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.22.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector3_REAL;
END_VAR
```

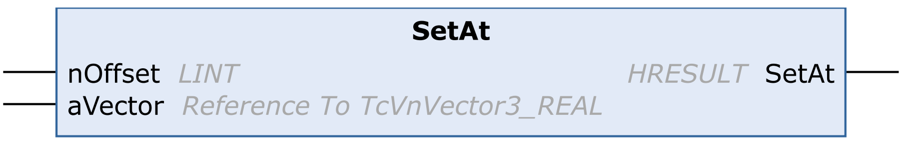
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector3_REAL [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.22.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector3_REAL;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector3_REAL [▸ 141]	The value to set

Return value

HRESULT [▸ 122]

6.1.3.1.3.23 ITcVnRandomAccess_TcVnVector3_SINT

Offers a random access interface for TcVnVector3_SINT values.

Inheritance Hierarchy

ITcUnknown [▸ 407]
    ITcVnAccess_TcVnVector3_SINT [▸ 262]
        ITcVnRandomAccess_TcVnVector3_SINT

Methods

Name	Origin	Description
TcAddRef [▸ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 408]	ITcUnknown	Decrements the reference counter.
Get [▸ 262]	ITcVnAccess_TcVnVector3_SINT	Gets the value
Set [▸ 263]	ITcVnAccess_TcVnVector3_SINT	Sets the value
GetAt [▸ 317]	ITcVnRandomAccess_TcVnVector3_SINT	Gets the value
SetAt [▸ 317]	ITcVnRandomAccess_TcVnVector3_SINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.23.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector3_SINT;
END_VAR
```

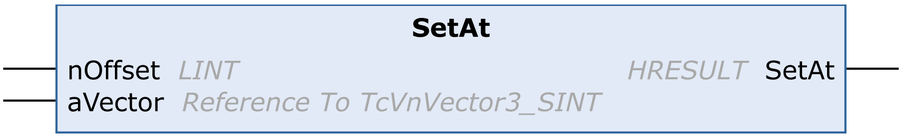
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector3_SINT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.23.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector3_SINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector3_SINT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.24 ITcVnRandomAccess_TcVnVector3_UINT

Offers a random access interface for TcVnVector3_UINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
  ITcVnAccess_TcVnVector3_UINT [▶ 263]
    ITcVnRandomAccess_TcVnVector3_UINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 264]	ITcVnAccess_TcVnVector3_UINT	Gets the value
Set [▶ 264]	ITcVnAccess_TcVnVector3_UINT	Sets the value
GetAt [▶ 318]	ITcVnRandomAccess_TcVnVector3_UINT	Gets the value
SetAt [▶ 319]	ITcVnRandomAccess_TcVnVector3_UINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.24.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector3_UINT;
END_VAR
```

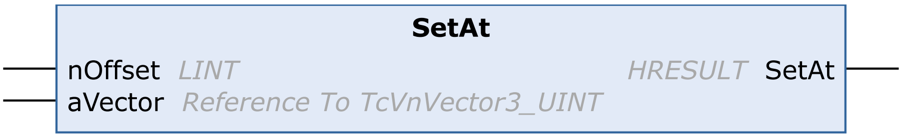
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector3_UINT [▶ 141]	Returns the value

Return value

HRESULT [▸ 122]

6.1.3.1.3.24.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector3_UINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector3_UINT [▸ 141]	The value to set

Return value

HRESULT [▸ 122]

6.1.3.1.3.25 ITcVnRandomAccess_TcVnVector3_USINT

Offers a random access interface for TcVnVector3_USINT values.

Inheritance Hierarchy

ITcUnknown [▸ 407]
  ITcVnAccess_TcVnVector3_USINT [▸ 265]
    ITcVnRandomAccess_TcVnVector3_USINT

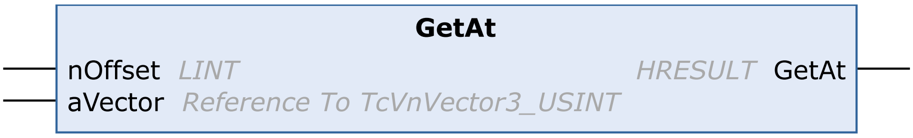
Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 265]	ITcVnAccess_TcVnVector3_USINT	Gets the value
Set [▶ 266]	ITcVnAccess_TcVnVector3_USINT	Sets the value
GetAt [▶ 320]	ITcVnRandomAccess_TcVnVector3_USINT	Gets the value
SetAt [▶ 320]	ITcVnRandomAccess_TcVnVector3_USINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.25.1 GetAt



Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector3_USINT;
END_VAR
```

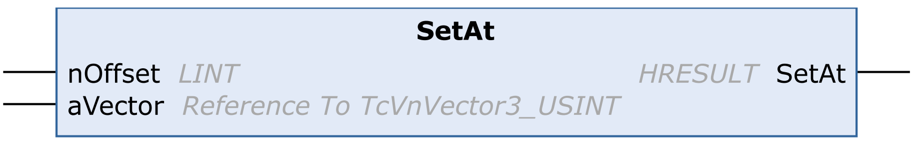
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector3_USINT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.25.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector3_USINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector3_USINT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.26 ITcVnRandomAccess_TcVnVector4_DINT

Offers a random access interface for TcVnVector4_DINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_TcVnVector4_DINT [▶ 266]
        ITcVnRandomAccess_TcVnVector4_DINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 267]	ITcVnAccess_TcVnVector4_DINT	Gets the value
Set [▶ 267]	ITcVnAccess_TcVnVector4_DINT	Sets the value
GetAt [▶ 322]	ITcVnRandomAccess_TcVnVector4_DINT	Gets the value
SetAt [▶ 322]	ITcVnRandomAccess_TcVnVector4_DINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.26.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector4_DINT;
END_VAR
```

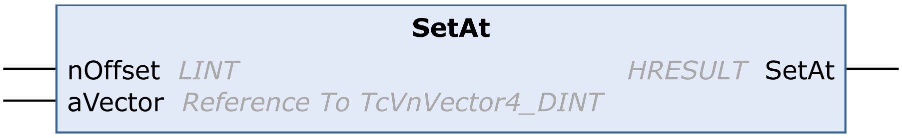
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector4_DINT [▸ 141]	Returns the value

Return value

HRESULT [▸ 122]

6.1.3.1.3.26.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector4_DINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector4_DINT [▸ 141]	The value to set

Return value

HRESULT [▸ 122]

6.1.3.1.3.27 ITcVnRandomAccess_TcVnVector4_INT

Offers a random access interface for TcVnVector4_INT values.

Inheritance Hierarchy

ITcUnknown [▸ 407]
ITcVnAccess_TcVnVector4_INT [▸ 268]
ITcVnRandomAccess_TcVnVector4_INT

Methods

Name	Origin	Description
TcAddRef [▸ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 408]	ITcUnknown	Decrements the reference counter.
Get [▸ 268]	ITcVnAccess_TcVnVector4_INT	Gets the value
Set [▸ 269]	ITcVnAccess_TcVnVector4_INT	Sets the value
GetAt [▸ 323]	ITcVnRandomAccess_TcVnVector4_INT	Gets the value
SetAt [▸ 324]	ITcVnRandomAccess_TcVnVector4_INT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.27.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector4_INT;
END_VAR
```

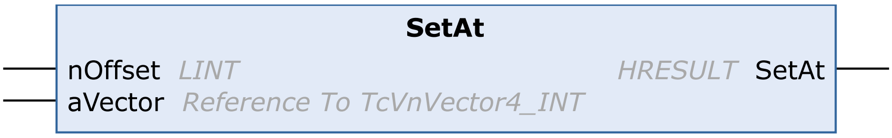
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector4_INT [▸ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.27.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector4_INT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector4_INT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.28 ITcVnRandomAccess_TcVnVector4_LREAL

Offers a random access interface for TcVnVector4_LREAL values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
        ITcVnAccess_TcVnVector4_LREAL [▶ 269]
                ITcVnRandomAccess_TcVnVector4_LREAL

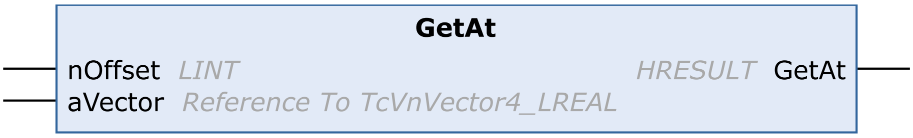
Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 270]	ITcVnAccess_TcVnVector4_LREAL	Gets the value
Set [▶ 270]	ITcVnAccess_TcVnVector4_LREAL	Sets the value
GetAt [▶ 325]	ITcVnRandomAccess_TcVnVector4_LREAL	Gets the value
SetAt [▶ 325]	ITcVnRandomAccess_TcVnVector4_LREAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.28.1 GetAt



Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector4_LREAL;
END_VAR
```

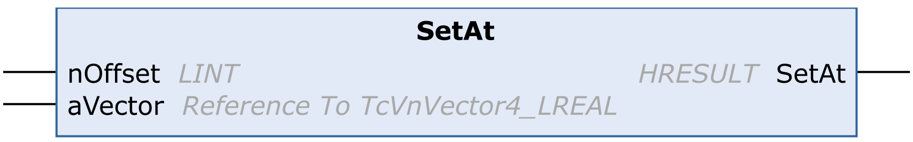
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector4_LREAL [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.28.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector4_LREAL;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector4_LREAL [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.29 ITcVnRandomAccess_TcVnVector4_SINT

Offers a random access interface for TcVnVector4_SINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_TcVnVector4_SINT [▶ 271]
        ITcVnRandomAccess_TcVnVector4_SINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 271]	ITcVnAccess_TcVnVector4_SINT	Gets the value
Set [▶ 272]	ITcVnAccess_TcVnVector4_SINT	Sets the value
GetAt [▶ 327]	ITcVnRandomAccess_TcVnVector4_SINT	Gets the value
SetAt [▶ 327]	ITcVnRandomAccess_TcVnVector4_SINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.29.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector4_SINT;
END_VAR
```

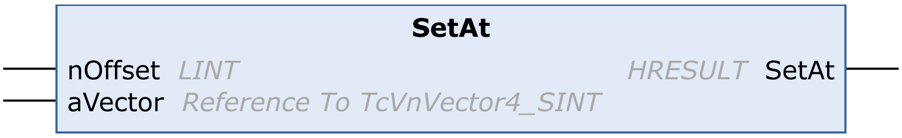
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector4_SINT [▸ 141]	Returns the value

Return value

HRESULT [▸ 122]

6.1.3.1.3.29.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector4_SINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector4_SINT [▸ 141]	The value to set

Return value

HRESULT [▸ 122]

6.1.3.1.3.30 ITcVnRandomAccess_TcVnVector4_UINT

Offers a random access interface for TcVnVector4_UINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
        ITcVnAccess_TcVnVector4_UINT [▶ 272]
                ITcVnRandomAccess_TcVnVector4_UINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 273]	ITcVnAccess_TcVnVector4_UINT	Gets the value
Set [▶ 273]	ITcVnAccess_TcVnVector4_UINT	Sets the value
GetAt [▶ 328]	ITcVnRandomAccess_TcVnVector4_UINT	Gets the value
SetAt [▶ 329]	ITcVnRandomAccess_TcVnVector4_UINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.30.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector4_UINT;
END_VAR
```

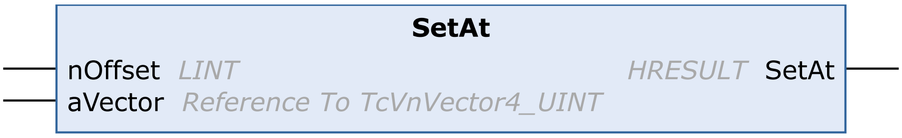
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector4_UINT [▶ 141]	Returns the value

Return value

HRESULT [▸ 122]

6.1.3.1.3.30.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector4_UINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector4_UINT [▸ 141]	The value to set

Return value

HRESULT [▸ 122]

6.1.3.1.3.31 ITcVnRandomAccess_TcVnVector4_USINT

Offers a random access interface for TcVnVector4_USINT values.

Inheritance Hierarchy

ITcUnknown [▸ 407]
    ITcVnAccess_TcVnVector4_USINT [▸ 274]
        ITcVnRandomAccess_TcVnVector4_USINT

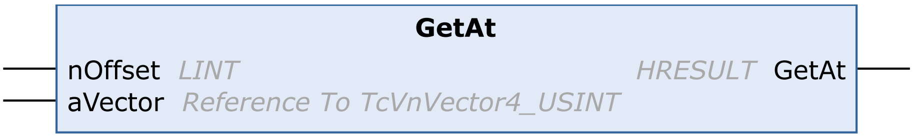
Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 274]	ITcVnAccess_TcVnVector4_USINT	Gets the value
Set [▶ 275]	ITcVnAccess_TcVnVector4_USINT	Sets the value
GetAt [▶ 330]	ITcVnRandomAccess_TcVnVector4_USINT	Gets the value
SetAt [▶ 330]	ITcVnRandomAccess_TcVnVector4_USINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.31.1 GetAt



Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector4_USINT;
END_VAR
```

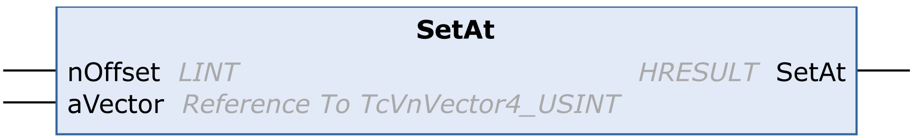
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector4_USINT [▶ 141]	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.31.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    aVector : Reference To TcVnVector4_USINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
aVector	Reference To TcVnVector4_USINT [▶ 141]	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.32 ITcVnRandomAccess_UDINT

Offers a random access interface for UDINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_UDINT [▶ 275]
        ITcVnRandomAccess_UDINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 276]	ITcVnAccess_UDINT	Gets the value
Set [▶ 276]	ITcVnAccess_UDINT	Sets the value
GetAt [▶ 331]	ITcVnRandomAccess_UDINT	Gets the value
SetAt [▶ 332]	ITcVnRandomAccess_UDINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.32.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    nValue  : Reference To UDINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
nValue	Reference To UDINT	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.32.2 SetAt

Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    nValue  : UDINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
nValue	UDINT	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.33 ITcVnRandomAccess_UINT

Offers a random access interface for UINT values

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_UINT [▶ 276]
        ITcVnRandomAccess_UINT

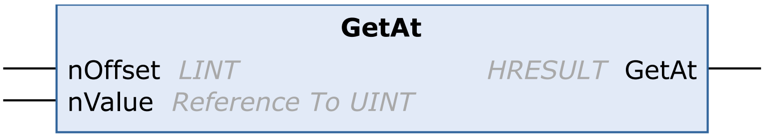
Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 277]	ITcVnAccess_UINT	Gets the value
Set [▶ 277]	ITcVnAccess_UINT	Sets the value
GetAt [▶ 333]	ITcVnRandomAccess_UINT	Gets the value
SetAt [▶ 333]	ITcVnRandomAccess_UINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.33.1 GetAt



Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    nValue  : Reference To UINT;
END_VAR
```

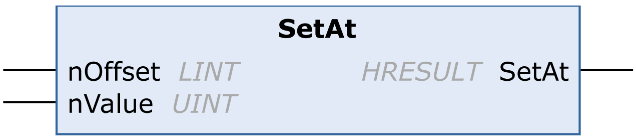
Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
nValue	Reference To UINT	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.33.2 SetAt



Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    nValue  : UINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
nValue	UINT	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.34 ITcVnRandomAccess_ULINT

Offers a randowm access interface for ULINT values.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_ULINT [▶ 278]
        ITcVnRandomAccess_ULINT

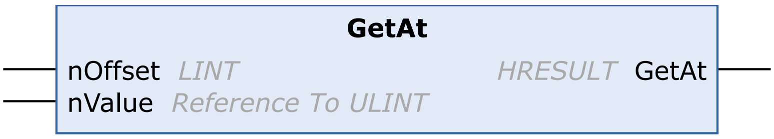
Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 278]	ITcVnAccess_ULINT	Gets the value
Set [▶ 279]	ITcVnAccess_ULINT	Sets the value
GetAt [▶ 334]	ITcVnRandomAccess_ULINT	Gets the value
SetAt [▶ 335]	ITcVnRandomAccess_ULINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.34.1 GetAt



Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    nValue  : Reference To ULINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
nValue	Reference To ULINT	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.34.2 SetAt

Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    nValue  : ULINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
nValue	ULINT	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.3.35 ITcVnRandomAccess_USINT

Offers a random access interface for USINT values

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnAccess_USINT [▶ 279]
        ITcVnRandomAccess_USINT

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
Get [▶ 280]	ITcVnAccess_USINT	Gets the value
Set [▶ 280]	ITcVnAccess_USINT	Sets the value
GetAt [▶ 336]	ITcVnRandomAccess_USINT	Gets the value
SetAt [▶ 336]	ITcVnRandomAccess_USINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.3.35.1 GetAt

Gets the value

Syntax

Definition:

```
METHOD GetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    nValue  : Reference To USINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
nValue	Reference To USINT	Returns the value

Return value

HRESULT [▶ 122]

6.1.3.1.3.35.2 SetAt

Sets the value

Syntax

Definition:

```
METHOD SetAt : HRESULT
VAR_INPUT
    nOffset : LINT;
    nValue  : USINT;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset to the current position
nValue	USINT	The value to set

Return value

HRESULT [▶ 122]

6.1.3.1.4 Iterators

该组包含迭代器处理的接口，以便访问容器 [▶ 132]的元素。

6.1.3.1.4.1 ITcVnBidirectionalIterator

Offers an interface for a bidirectional iterator.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnIteratorBase [▶ 341]
        ITcVnForwardIterator [▶ 339]
            ITcVnBidirectionalIterator

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
CheckIfBegin [▶ 342]	ITcVnIteratorBase	Checks if the iterator points to the first element.
CheckIfEnd [▶ 342]	ITcVnIteratorBase	Checks if the iterator points to the past-the-end element (alternatively use F_VN_CheckIfIteratorIsAtEnd).
GetValueSize [▶ 343]	ITcVnIteratorBase	Gets the memory size (in byte) required by the value the iterator points to.
GetValueTypeGuid [▶ 343]	ITcVnIteratorBase	Gets the type GUID of the value the iterator points to.
SetToBegin [▶ 344]	ITcVnIteratorBase	Sets the iterator to the first element (alternatively use F_VN_SetIteratorToBegin).
SetToEnd [▶ 344]	ITcVnIteratorBase	Sets the iterator to the past-the-end element.
CheckIfEqualTo [▶ 339]	ITcVnForwardIterator	Checks if iterator is equal to another iterator.
GetContainer [▶ 340]	ITcVnForwardIterator	Gets a pointer to the current element converted into an ITcVnContainer interface and increment its reference counter (only possible for container types). (Alternatively use F_VN_GetContainer.)
Increment [▶ 340]	ITcVnForwardIterator	Increments the iterator. (Alternatively use F_VN_IncrementIterator.)
SetContainer [▶ 341]	ITcVnForwardIterator	Sets the current element using an ITcVnContainer interface (only possible for container types). (Alternatively use F_VN_SetContainer.)
Decrement [▶ 338]	ITcVnBidirectionalIterator	Decrements the iterator.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.4.1.1 Decrement

Decrements the iterator.

Syntax

Definition:

```
METHOD Decrement : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.1.4.2 ITcVnForwardIterator

Offers an interface for a forward iterator.

Inheritance Hierarchy

ITcUnknown [▸ 407]
    ITcVnIteratorBase [▸ 341]
        ITcVnForwardIterator

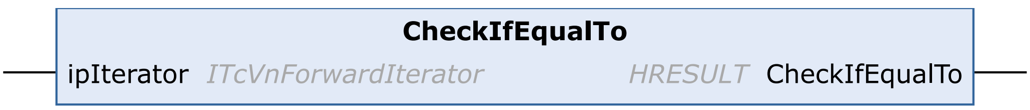
Methods

Name	Origin	Description
TcAddRef [▸ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 408]	ITcUnknown	Decrements the reference counter.
CheckIfBegin [▸ 342]	ITcVnIteratorBase	Checks if the iterator points to the first element.
CheckIfEnd [▸ 342]	ITcVnIteratorBase	Checks if the iterator points to the past-the-end element (alternatively use F_VN_CheckIfIteratorIsAtEnd).
GetValueSize [▸ 343]	ITcVnIteratorBase	Gets the memory size (in byte) required by the value the iterator points to.
GetValueTypeGuid [▸ 343]	ITcVnIteratorBase	Gets the type GUID of the value the iterator points to.
SetToBegin [▸ 344]	ITcVnIteratorBase	Sets the iterator to the first element (alternatively use F_VN_SetIteratorToBegin).
SetToEnd [▸ 344]	ITcVnIteratorBase	Sets the iterator to the past-the-end element.
CheckIfEqualTo [▸ 339]	ITcVnForwardIterator	Checks if iterator is equal to another iterator.
GetContainer [▸ 340]	ITcVnForwardIterator	Gets a pointer to the current element converted into an ITcVnContainer interface and increment its reference counter (only possible for container types). (Alternatively use F_VN_GetContainer.)
Increment [▸ 340]	ITcVnForwardIterator	Increments the iterator. (Alternatively use F_VN_IncrementIterator.)
SetContainer [▸ 341]	ITcVnForwardIterator	Sets the current element using an ITcVnContainer interface (only possible for container types). (Alternatively use F_VN_SetContainer.)

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.4.2.1 CheckIfEqualTo



Checks if iterator is equal to another iterator.

Syntax

Definition:

```
METHOD CheckIfEqualTo : HRESULT
VAR_INPUT
    ipIterator : ITcVnForwardIterator;
END_VAR
```

Inputs

Name	Type	Description
ipIterator	ITcVnForwardIterator [▸ 339]	Iterator interface to compare with

Return value

HRESULT [▸ 122]

6.1.3.1.4.2.2 GetContainer

Gets a pointer to the current element converted into an ITcVnContainer interface and increment its reference counter (only possible for container types). (Alternatively use F_VN_GetContainer.)

Syntax

Definition:

```
METHOD GetContainer : HRESULT
VAR_INPUT
    pipContainer : Pointer To ITcVnContainer;
END_VAR
```

Inputs

Name	Type	Description
pipContainer	Pointer To ITcVnContainer [▸ 349]	Returns the container interface.

Return value

HRESULT [▸ 122]

6.1.3.1.4.2.3 Increment

Increments the iterator. (Alternatively use F_VN_IncrementIterator.)

Syntax

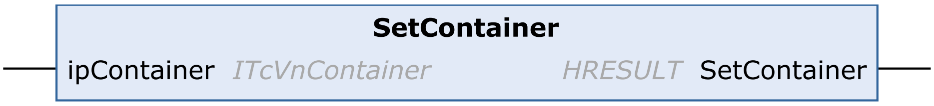
Definition:

```
METHOD Increment : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.1.4.2.4 SetContainer



Sets the current element using an ITcVnContainer interface (only possible for container types). (Alternatively use F_VN_SetContainer.)

Syntax

Definition:

```
METHOD SetContainer : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container interface of which the content is to be assigned to the current element.

Return value

HRESULT [▶ 122]

6.1.3.1.4.3 ITcVnIteratorBase

Offers a base interface for iterators

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnIteratorBase

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
CheckIfBegin [▶ 342]	ITcVnIteratorBase	Checks if the iterator points to the first element.
CheckIfEnd [▶ 342]	ITcVnIteratorBase	Checks if the iterator points to the past-the-end element (alternatively use F_VN_CheckIfIteratorIsAtEnd).
GetValueSize [▶ 343]	ITcVnIteratorBase	Gets the memory size (in byte) required by the value the iterator points to.
GetValueTypeGuid [▶ 343]	ITcVnIteratorBase	Gets the type GUID of the value the iterator points to.
SetToBegin [▶ 344]	ITcVnIteratorBase	Sets the iterator to the first element (alternatively use F_VN_SetIteratorToBegin).
SetToEnd [▶ 344]	ITcVnIteratorBase	Sets the iterator to the past-the-end element.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.4.3.1 CheckIfBegin

Checks if the iterator points to the first element.

Syntax

Definition:

```
METHOD CheckIfBegin : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.1.4.3.2 CheckIfEnd

Checks if the iterator points to the past-the-end element (alternatively use F_VN_CheckIfIteratorIsAtEnd).

Syntax

Definition:

```
METHOD CheckIfEnd : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.1.4.3.3 GetValueSize

Gets the memory size (in byte) required by the value the iterator points to.

Syntax

Definition:

```
METHOD GetValueSize : HRESULT
VAR_INPUT
    nSize : Reference To ULINT;
END_VAR
```

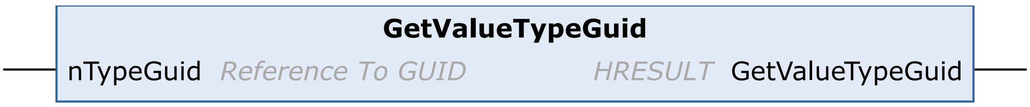
Inputs

Name	Type	Description
nSize	Reference To ULINT	Returns the element size in bytes.

Return value

HRESULT [▶ 122]

6.1.3.1.4.3.4 GetValueTypeGuid



Gets the type GUID of the value the iterator points to.

Syntax

Definition:

```
METHOD GetValueTypeGuid : HRESULT
VAR_INPUT
    nTypeGuid : Reference To GUID;
END_VAR
```

Inputs

Name	Type	Description
nTypeGuid	Reference To GUID	Returns the type GUID.

Return value

HRESULT [▶ 122]

6.1.3.1.4.3.5 SetToBegin

Sets the iterator to the first element (alternatively use F_VN_SetIteratorToBegin).

Syntax

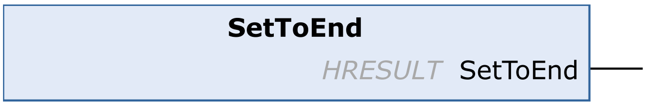
Definition:

```
METHOD SetToBegin : HRESULT
```

Return value

HRESULT [▸ 122]

6.1.3.1.4.3.6 SetToEnd



Sets the iterator to the past-the-end element.

Syntax

Definition:

```
METHOD SetToEnd : HRESULT
```

Return value

HRESULT [▸ 122]

6.1.3.1.4.4 ITcVnIteratorCopyCreator

Offers an interface providing a method for creating a new iterator to the same position.

Inheritance Hierarchy

ITcUnknown [▸ 407]
 ITcVnIteratorCopyCreator

Methods

Name	Origin	Description
TcAddRef [▸ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 408]	ITcUnknown	Decrements the reference counter.
Create [▸ 345]	ITcVnIteratorCopyCreator	Create a new iterator pointing to the position of the calling iterator.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.4.4.1 Create

Create a new iterator pointing to the position of the calling iterator.

Syntax

Definition:

```
METHOD Create : HRESULT
VAR_INPUT
    pipIterator : Pointer To ITcVnForwardIterator;
END_VAR
```

Inputs

Name	Type	Description
pipIterator	Pointer To ITcVnForwardIterator [▶ 339]	Returns the created iterator.

Return value

HRESULT [▶ 122]

6.1.3.1.4.5 ITcVnRandomAccessIterator

Offers an interface for a random access iterator.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnIteratorBase [▶ 341]
        ITcVnForwardIterator [▶ 339]
            ITcVnBidirectionalIterator [▶ 337]
                ITcVnRandomAccessIterator

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
CheckIfBegin [▶ 342]	ITcVnIteratorBase	Checks if the iterator points to the first element.
CheckIfEnd [▶ 342]	ITcVnIteratorBase	Checks if the iterator points to the past-the-end element (alternatively use F_VN_CheckIfIteratorIsAtEnd).
GetValueSize [▶ 343]	ITcVnIteratorBase	Gets the memory size (in byte) required by the value the iterator points to.
GetValueTypeGuid [▶ 343]	ITcVnIteratorBase	Gets the type GUID of the value the iterator points to.
SetToBegin [▶ 344]	ITcVnIteratorBase	Sets the iterator to the first element (alternatively use F_VN_SetIteratorToBegin).
SetToEnd [▶ 344]	ITcVnIteratorBase	Sets the iterator to the past-the-end element.
CheckIfEqualTo [▶ 339]	ITcVnForwardIterator	Checks if iterator is equal to another iterator.
GetContainer [▶ 340]	ITcVnForwardIterator	Gets a pointer to the current element converted into an ITcVnContainer interface and increment its reference counter (only possible for container types). (Alternatively use F_VN_GetContainer.)
Increment [▶ 340]	ITcVnForwardIterator	Increments the iterator. (Alternatively use F_VN_IncrementIterator.)
SetContainer [▶ 341]	ITcVnForwardIterator	Sets the current element using an ITcVnContainer interface (only possible for container types). (Alternatively use F_VN_SetContainer.)
Decrement [▶ 338]	ITcVnBidirectionalIterator	Decrements the iterator.
CheckIfGreaterThan [▶ 347]	ITcVnRandomAccessIterator	Checks if the iterator is greater than another iterator.
CheckIfLessThan [▶ 347]	ITcVnRandomAccessIterator	Checks if the iterator is less than another iterator.
GetContainerAt [▶ 348]	ITcVnRandomAccessIterator	Gets a pointer to the element at a specific offset from the current element converted into an ITcVnContainer interface and increment its reference counter (only possible for container types). (Alternatively use F_VN_GetAt_ITcVnContainer.)
SetContainerAt [▶ 348]	ITcVnRandomAccessIterator	Sets the element at a specific offset from the current element using an ITcVnContainer interface (only possible for container types). (Alternatively use F_VN_SetAt_ITcVnContainer.)

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.4.5.1 CheckIfGreaterThan

Checks if the iterator is greater than another iterator.

Syntax

Definition:

```
METHOD CheckIfGreaterThan : HRESULT
VAR_INPUT
    ipIterator : ITcVnRandomAccessIterator;
END_VAR
```

Inputs

Name	Type	Description
ipIterator	ITcVnRandomAccessIterator [▶ 345]	Iterator interface to compare with.

Return value

HRESULT [▶ 122]

6.1.3.1.4.5.2 CheckIfLessThan

Checks if the iterator is less than another iterator.

Syntax

Definition:

```
METHOD CheckIfLessThan : HRESULT
VAR_INPUT
    ipIterator : ITcVnRandomAccessIterator;
END_VAR
```

Inputs

Name	Type	Description
ipIterator	ITcVnRandomAccessIterator [▶ 345]	Iterator interface to compare with.

Return value

HRESULT [▶ 122]

6.1.3.1.4.5.3 GetContainerAt

Gets a pointer to the element at a specific offset from the current element converted into an ITcVnContainer interface and increment its reference counter (only possible for container types). (Alternatively use F_VN_GetAt_ITcVnContainer.)

Syntax

Definition:

```
METHOD GetContainerAt : HRESULT
VAR_INPUT
    nOffset      : LINT;
    pipContainer : Pointer To ITcVnContainer;
END_VAR
```

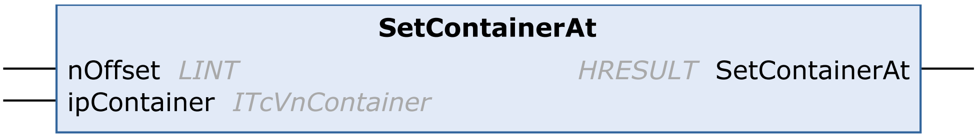
Inputs

Name	Type	Description
nOffset	LINT	Offset from the current element.
pipContainer	Pointer To ITcVnContainer [▶ 349]	Returns the pointer to the container interface.

Return value

HRESULT [▶ 122]

6.1.3.1.4.5.4 SetContainerAt



Sets the element at a specific offset from the current element using an ITcVnContainer interface (only possible for container types). (Alternatively use F_VN_SetAt_ITcVnContainer.)

Syntax

Definition:

```
METHOD SetContainerAt : HRESULT
VAR_INPUT
    nOffset     : LINT;
    ipContainer : ITcVnContainer;
END_VAR
```

Inputs

Name	Type	Description
nOffset	LINT	Offset from the current element.
ipContainer	ITcVnContainer [▶ 349]	Container interface of which the content is to be assigned to the current element.

Return value

HRESULT [▶ 122]

6.1.3.1.5 ITcVnContainer

Offers an interface for an object container.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnContainer

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
CheckIfBasicContainer [▶ 350]	ITcVnContainer	Checks if the container contains only basic (non-container) elements.
CheckIfEmpty [▶ 350]	ITcVnContainer	Checks if the container is empty. (Alternatively use F_VN_CheckIfEmpty.)
GetBidirectionalIterator [▶ 350]	ITcVnContainer	Gets an interface pointer to a bidirectional iterator (if this iterator type is supported by the container) and increment its reference counter.
GetElementNum [▶ 351]	ITcVnContainer	Gets the size (number of elements) of the container. (Alternatively use F_VN_GetNumberOfElements.)
GetElementSize [▶ 351]	ITcVnContainer	Gets the size (in byte) of each element in the container.
GetElementTypeGuid [▶ 352]	ITcVnContainer	Gets the GUID of the container elements.
GetExportSize [▶ 352]	ITcVnContainer	Gets combined size (in byte) of all elements in the container.
GetForwardIterator [▶ 353]	ITcVnContainer	Gets an interface pointer to a forward iterator (if this iterator type is supported by the container) and increment its reference counter. (Alternatively use F_VN_GetForwardIterator.)
GetRandomAccessIterator [▶ 353]	ITcVnContainer	Gets an interface pointer to a random access iterator (if this iterator type is supported by the container) and increment its reference counter. (Alternatively use F_VN_GetRandomAccessIterator.)
GetTypeGuid [▶ 354]	ITcVnContainer	Gets GUID of the container.
GetTypeName [▶ 354]	ITcVnContainer	Gets the container type name as a string.

更多信息

接口ITcVnContainer用于处理容器 [▶ 132]。请注意接口指针 [▶ 119]一章中的说明。

覆写容器

一般来说，具有 API 功能的容器可以简单地覆盖。然而，目前，如果现有容器和待写入容器类型不同，则无法覆盖。这是由错误代码 70E（INCOMPATIBLE）表示。

相关函数

- 基本容器操作 [▶ 520]
- 代数容器操作 [▶ 436]
- 容器统计 [▶ 818]

- 轮廓分析 [▸ 913]

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.1.5.1 CheckIfBasicContainer

Checks if the container contains only basic (non-container) elements.

Syntax

Definition:

```
METHOD CheckIfBasicContainer : HRESULT
```

Return value

HRESULT [▸ 122]

6.1.3.1.5.2 CheckIfEmpty

Checks if the container is empty. (Alternatively use F_VN_CheckIfEmpty.)

Syntax

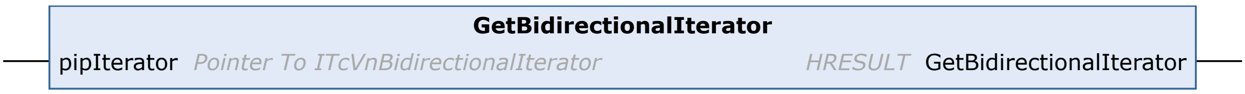
Definition:

```
METHOD CheckIfEmpty : HRESULT
```

Return value

HRESULT [▸ 122]

6.1.3.1.5.3 GetBidirectionalIterator



Gets an interface pointer to a bidirectional iterator (if this iterator type is supported by the container) and increment its reference counter.

Syntax

Definition:

```
METHOD GetBidirectionalIterator : HRESULT
VAR_INPUT
    pipIterator : Pointer To ITcVnBidirectionalIterator;
END_VAR
```

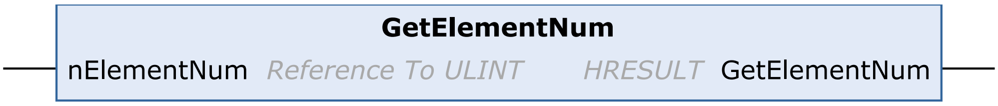
Inputs

Name	Type	Description
pipIterator	Pointer To ITcVnBidirectionalIterator [▶ 337]	Returns the iterator interface.

Return value

HRESULT [▶ 122]

6.1.3.1.5.4 GetElementNum



Gets the size (number of elements) of the container. (Alternatively use F_VN_GetNumberOfElements.)

Syntax

Definition:

```
METHOD GetElementNum : HRESULT
VAR_INPUT
    nElementNum : Reference To ULINT;
END_VAR
```

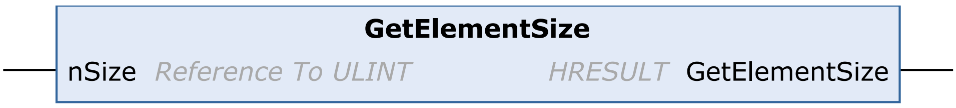
Inputs

Name	Type	Description
nElementNum	Reference To ULINT	Returns the number of elements in the container.

Return value

HRESULT [▶ 122]

6.1.3.1.5.5 GetElementSize



Gets the size (in byte) of each element in the container.

Syntax

Definition:

```
METHOD GetElementSize : HRESULT
VAR_INPUT
    nSize : Reference To ULINT;
END_VAR
```

Inputs

Name	Type	Description
nSize	Reference To ULINT	Returns the size in byte of a single element in the container.

Return value

HRESULT [▶ 122]

6.1.3.1.5.6 GetElementTypeGuid

Gets the GUID of the container elements.

Syntax

Definition:

```
METHOD GetElementTypeGuid : HRESULT
VAR_INPUT
    nTypeGuid : Reference To GUID;
END_VAR
```

Inputs

Name	Type	Description
nTypeGuid	Reference To GUID	Returns the GUID of the container elements.

Return value

HRESULT [▶ 122]

6.1.3.1.5.7 GetExportSize

Gets combined size (in byte) of all elements in the container.

Syntax

Definition:

```
METHOD GetExportSize : HRESULT
VAR_INPUT
    nExportSize : Reference To ULINT;
END_VAR
```

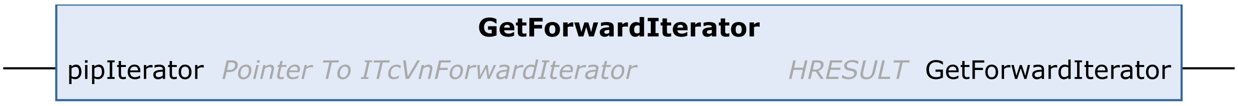
Inputs

Name	Type	Description
nExportSize	Reference To ULINT	Returns the combined size (in byte) of all elements in the container.

Return value

HRESULT [▶ 122]

6.1.3.1.5.8 GetForwardIterator



Gets an interface pointer to a forward iterator (if this iterator type is supported by the container) and increment its reference counter. (Alternatively use F_VN_GetForwardIterator.)

Syntax

Definition:

```
METHOD GetForwardIterator : HRESULT
VAR_INPUT
    pipIterator : Pointer To ITcVnForwardIterator;
END_VAR
```

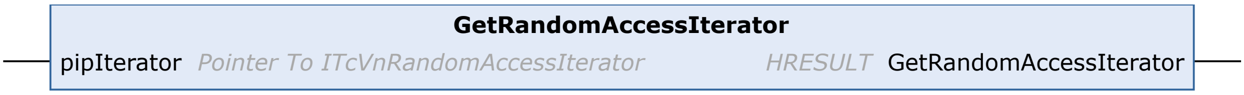
Inputs

Name	Type	Description
pipIterator	Pointer To ITcVnForwardIterator [▸ 339]	Returns the iterator interface.

Return value

HRESULT [▸ 122]

6.1.3.1.5.9 GetRandomAccessIterator



Gets an interface pointer to a random access iterator (if this iterator type is supported by the container) and increment its reference counter. (Alternatively use F_VN_GetRandomAccessIterator.)

Syntax

Definition:

```
METHOD GetRandomAccessIterator : HRESULT
VAR_INPUT
    pipIterator : Pointer To ITcVnRandomAccessIterator;
END_VAR
```

Inputs

Name	Type	Description
pipIterator	Pointer To ITcVnRandomAccessIterator [▸ 345]	Returns the iterator interface.

Return value

HRESULT [▸ 122]

6.1.3.1.5.10 GetTypeGuid

Gets GUID of the container.

Syntax

Definition:

```
METHOD GetTypeGuid : HRESULT
VAR_INPUT
    nTypeGuid : Reference To GUID;
END_VAR
```

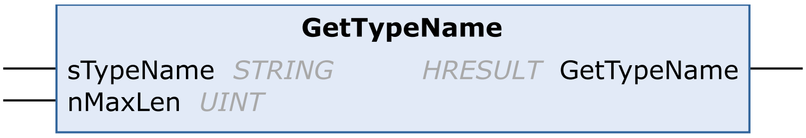
Inputs

Name	Type	Description
nTypeGuid	Reference To GUID	Returns the GUID of the container.

Return value

HRESULT [▶ 122]

6.1.3.1.5.11 GetTypeName



Gets the container type name as a string.

Syntax

Definition:

```
METHOD GetTypeName : HRESULT
VAR_INPUT
    sTypeName : STRING;
    nMaxLen   : UINT;
END_VAR
```

Inputs

Name	Type	Description
sTypeName	STRING	Returns the container type name as a string.
nMaxLen	UINT	Maximum string length allowed to be written in sTypeName

Return value

HRESULT [▶ 122]

6.1.3.2 Images

该组包含用于记录和处理图像 [▶ 130]的接口。

6.1.3.2.1 Acquisition

该组包含用于处理图像采集的接口。

6.1.3.2.1.1 ITcIoFileImageAcquisition

Interface for file image acquisition.

Inheritance Hierarchy

ITcUnknown [▶ 407]
  ITcIoImageAcquisition [▶ 359]
    ITcIoFileImageAcquisition

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
RegisterReceiver [▶ 360]	ITcIoImageAcquisition	Register callback interface.
UnregisterReceiver [▶ 361]	ITcIoImageAcquisition	Unregister callback interface.
OpenCamera [▶ 361]	ITcIoImageAcquisition	Open camera (open the control channel, open a stream channel, and activate the GVSP receiver).
CloseCamera [▶ 361]	ITcIoImageAcquisition	Close camera (deactivate the GVSP receiver, close the stream channel, and close the control channel).
StartAcquisition [▶ 361]	ITcIoImageAcquisition	Start the image acquisition.
StopAcquisition [▶ 362]	ITcIoImageAcquisition	Stop the image acquisition.
SoftwareTrigger [▶ 362]	ITcIoImageAcquisition	Initialize the software trigger and trigger it. The initialization step is skipped if it was executed before.
InitializeCamera [▶ 363]	ITcIoImageAcquisition	Send initialization commands to the camera.
CheckConnection [▶ 356]	ITcIoFileImageAcquisition	Checks the connection
TriggerImage [▶ 356]	ITcIoFileImageAcquisition	Initialize the software trigger and trigger a single image. Skips nSkipImages and triggers the capturing of image nSkipImages+1.
TriggerImageByName [▶ 356]	ITcIoFileImageAcquisition	Initialize the software trigger and trigger a single image specified by its name in the client assistant.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.1.1.1 CheckConnection

Checks the connection

Syntax

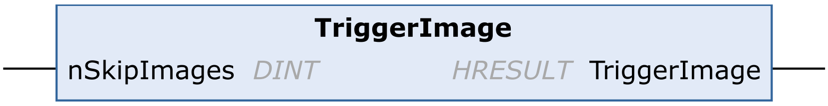
Definition:

```
METHOD CheckConnection : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.2.1.1.2 TriggerImage



Initialize the software trigger and trigger a single image. Skips nSkipImages and triggers the capturing of image nSkipImages+1.

Syntax

Definition:

```
METHOD TriggerImage : HRESULT
VAR_INPUT
    nSkipImages : DINT;
END_VAR
```

Inputs

Name	Type	Description
nSkipImages	DINT	Number of images to skip

Return value

HRESULT [▶ 122]

6.1.3.2.1.1.3 TriggerImageByName

Initialize the software trigger and trigger a single image specified by its name in the client assistant.

Syntax

Definition:

```
METHOD TriggerImageByName : HRESULT
VAR_INPUT
    sImageName : STRING;
END_VAR
```

Inputs

Name	Type	Description
sImageName	STRING	Image name to trigger

Return value

HRESULT [▶ 122]

6.1.3.2.1.2 ITcIoFileImageRecv

Interface for a image receiver.

Inheritance Hierarchy

ITcUnknown [▶ 407]
  ITcIoImageRecv [▶ 363]
    ITcIoFileImageRecv

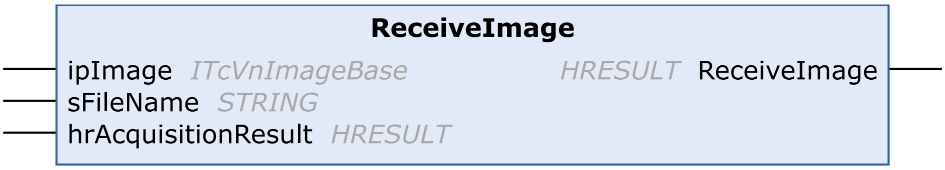
Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
ReceiveImage [▶ 363]	ITcIoImageRecv	Receive an image from an instance of ITcIoGevImageAcquisition.
ReceiveOpResult [▶ 364]	ITcIoImageRecv	Receive an operation result from an instance of ITcIoGevImageAcquisition.
ReceiveImage [▶ 357]	ITcIoFileImageRecv	Receive an image from an instance of ITcIoFileImageAcquisition.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.1.2.1 ReceiveImage



Receive an image from an instance of ITcIoFileImageAcquisition.

Syntax

Definition:

```
METHOD ReceiveImage : HRESULT
VAR_INPUT
    ipImage             : ITcVnImageBase;
    sFileName           : STRING;
    hrAcquisitionResult : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImageBase [▸ 393]	Interface pointer to the image.
sFileName	STRING	Returns the filename to the image.
hrAcquisition Result	HRESULT [▸ 122]	HRESULT indicating the status of the image acquisition.

Return value

HRESULT [▸ 122]

6.1.3.2.1.3 ITcIoGevImageAcquisition

Interface for GigE Vision image acquisition.

Inheritance Hierarchy

ITcUnknown [▸ 407]
    ITcIoImageAcquisition [▸ 359]
        ITcIoGevImageAcquisition

Methods

Name	Origin	Description
TcAddRef [▸ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 408]	ITcUnknown	Decrements the reference counter.
RegisterReceiver [▸ 360]	ITcIoImageAcquisition	Register callback interface.
UnregisterReceiver [▸ 361]	ITcIoImageAcquisition	Unregister callback interface.
OpenCamera [▸ 361]	ITcIoImageAcquisition	Open camera (open the control channel, open a stream channel, and activate the GVSP receiver).
CloseCamera [▸ 361]	ITcIoImageAcquisition	Close camera (deactivate the GVSP receiver, close the stream channel, and close the control channel).
StartAcquisition [▸ 361]	ITcIoImageAcquisition	Start the image acquisition.
StopAcquisition [▸ 362]	ITcIoImageAcquisition	Stop the image acquisition.
SoftwareTrigger [▸ 362]	ITcIoImageAcquisition	Initialize the software trigger and trigger it. The initialization step is skipped if it was executed before.
InitializeCamera [▸ 363]	ITcIoImageAcquisition	Send initialization commands to the camera.
CheckConnection [▸ 359]	ITcIoGevImageAcquisition	Checks the camera connection

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.1.3.1 CheckConnection

Checks the camera connection

Syntax

Definition:

```
METHOD CheckConnection : HRESULT
VAR_INPUT
    eAssumedState : Reference To GEV_CAMERA_STATE;
    eActualState  : Reference To GEV_CAMERA_STATE;
END_VAR
```

Inputs

Name	Type	Description
eAssumedState	Reference To GEV_CAMERA_STATE [▸ 204]	The internally assumed state of the camera.
eActualState	Reference To GEV_CAMERA_STATE [▸ 204]	The actually observable state of the camera.

Return value

HRESULT [▸ 122]

6.1.3.2.1.4 ITcIoImageAcquisition

Interface for image acquisition.

Inheritance Hierarchy

ITcUnknown [▸ 407]
ITcIoImageAcquisition

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
RegisterReceiver [▶ 360]	ITcIoImageAcquisition	Register callback interface.
UnregisterReceiver [▶ 361]	ITcIoImageAcquisition	Unregister callback interface.
OpenCamera [▶ 361]	ITcIoImageAcquisition	Open camera (open the control channel, open a stream channel, and activate the GVSP receiver).
CloseCamera [▶ 361]	ITcIoImageAcquisition	Close camera (deactivate the GVSP receiver, close the stream channel, and close the control channel).
StartAcquisition [▶ 361]	ITcIoImageAcquisition	Start the image acquisition.
StopAcquisition [▶ 362]	ITcIoImageAcquisition	Stop the image acquisition.
SoftwareTrigger [▶ 362]	ITcIoImageAcquisition	Initialize the software trigger and trigger it. The initialization step is skipped if it was executed before.
InitializeCamera [▶ 363]	ITcIoImageAcquisition	Send initialization commands to the camera.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.1.4.1 RegisterReceiver

Register callback interface.

Syntax

Definition:

```
METHOD RegisterReceiver : HRESULT
VAR_INPUT
    ipRecv : ITcIoImageRecv;
END_VAR
```

Inputs

Name	Type	Description
ipRecv	ITcIoImageRecv [▶ 363]	Pointer to an interface containing the callback function.

Return value

HRESULT [▶ 122]

6.1.3.2.1.4.2 UnregisterReceiver

UnregisterReceiver

HRESULT UnregisterReceiver

Unregister callback interface.

Syntax

Definition:

```
METHOD UnregisterReceiver : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.2.1.4.3 OpenCamera

OpenCamera

HRESULT OpenCamera

Open camera (open the control channel, open a stream channel, and activate the GVSP receiver).

Syntax

Definition:

```
METHOD OpenCamera : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.2.1.4.4 CloseCamera

CloseCamera

HRESULT CloseCamera

Close camera (deactivate the GVSP receiver, close the stream channel, and close the control channel).

Syntax

Definition:

```
METHOD CloseCamera : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.2.1.4.5 StartAcquisition

StartAcquisition

HRESULT StartAcquisition

Start the image acquisition.

Syntax

Definition:

```
METHOD StartAcquisition : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.2.1.4.6 StopAcquisition

Stop the image acquisition.

Syntax

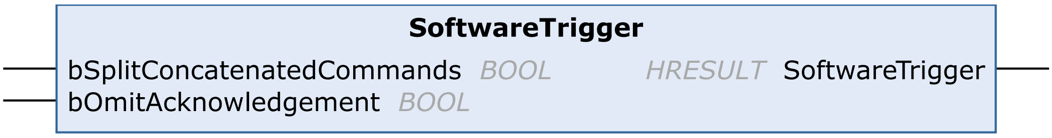
Definition:

```
METHOD StopAcquisition : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.2.1.4.7 SoftwareTrigger



Initialize the software trigger and trigger it. The initialization step is skipped if it was executed before.

Syntax

Definition:

```
METHOD SoftwareTrigger : HRESULT
VAR_INPUT
    bSplitConcatenatedCommands : BOOL;
    bOmitAcknowledgement       : BOOL;
END_VAR
```

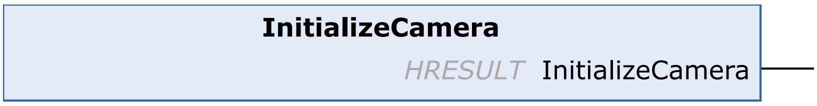
Inputs

Name	Type	Description
bSplitConcatenatedCommands	BOOL	If true, multi-read and multi-write commands are split into sequences of single read and single write commands, respectively.
bOmitAcknowledgement	BOOL	Indicates that no acknowledge packet should be requested.

Return value

HRESULT [▶ 122]

6.1.3.2.1.4.8 InitializeCamera



Send initialization commands to the camera.

Syntax

Definition:

```
METHOD InitializeCamera : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.2.1.5 ITcIoImageRecv

Interface for a image receiver.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcIoImageRecv

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
ReceiveImage [▶ 363]	ITcIoImageRecv	Receive an image from an instance of ITcIoGevImageAcquisition.
ReceiveOpResult [▶ 364]	ITcIoImageRecv	Receive an operation result from an instance of ITcIoGevImageAcquisition.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.1.5.1 ReceiveImage

Receive an image from an instance of ITcIoGevImageAcquisition.

Syntax

Definition:

```
METHOD ReceiveImage : HRESULT
VAR_INPUT
    ipImage              : ITcVnImageBase;
    hrAcquisitionResult : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImageBase [▶ 393]	Interface pointer to the image.
hrAcquisition Result	HRESULT [▶ 122]	HRESULT indicating the status of the image acquisition.

Return value

HRESULT [▶ 122]

6.1.3.2.1.5.2 ReceiveOpResult

Receive an operation result from an instance of ITcIoGevImageAcquisition.

Syntax

Definition:

```
METHOD ReceiveOpResult : HRESULT
VAR_INPUT
    hrOperationResult : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
hrOperationRe sult	HRESULT [▶ 122]	HRESULT indicating the status of the performed operation.

Return value

HRESULT [▶ 122]

6.1.3.2.1.6 ITcVnFileImageProvider

Interface for an image provider for images from outside realtime environment.

Inheritance Hierarchy

ITcUnknown [▶ 407]
        ITcVnImageProvider [▶ 369]
                ITcVnFileImageProvider

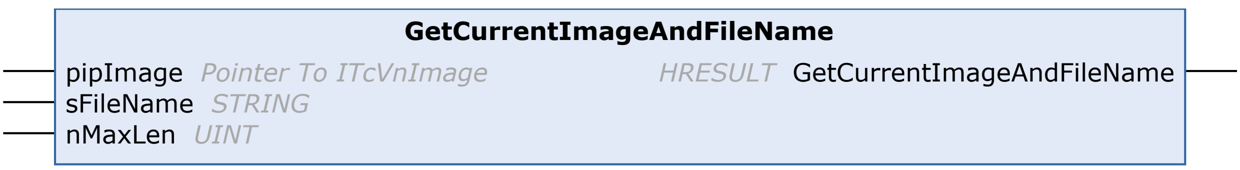
Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
CheckIfBusy [▶ 370]	ITcVnImageProvider	Check whether the module is busy; i.e, an operation is in progress.
CheckIfCameraConnected [▶ 371]	ITcVnImageProvider	Check if the camera is properly connected (Returns S_OK if connected and S_FALSE if not. Can only check software connection, i.e. the check is unable to determine if the cable is connected or not in case that S_FALSE is returned).
GetCurrentImage [▶ 371]	ITcVnImageProvider	Gets the current image by detaching its internal reference.
GetError [▶ 371]	ITcVnImageProvider	Gets result of the last operation performed.
OpenCamera [▶ 372]	ITcVnImageProvider	Open camera (open the control channel, open a stream channel, and activate the GVSP receiver).
CloseCamera [▶ 372]	ITcVnImageProvider	Close camera (deactivate the GVSP receiver, close the stream channel, and close the control channel).
StartAcquisition [▶ 372]	ITcVnImageProvider	Start image acquisition.
StopAcquisition [▶ 373]	ITcVnImageProvider	Stop image acquisition.
SoftwareTrigger [▶ 373]	ITcVnImageProvider	Initialize the software trigger and trigger it. The initialization step is skipped if it was executed before.
InitializeCamera [▶ 373]	ITcVnImageProvider	Set camera to initial state.
GetCurrentImageAndFileName [▶ 365]	ITcVnFileImageProvider	Get the current image by detaching its internal reference.
TriggerImage [▶ 366]	ITcVnFileImageProvider	Initialize the software trigger and trigger a single image. Skips nSkipImages and triggers the capturing of image nSkipImages+1.
TriggerImageByName [▶ 366]	ITcVnFileImageProvider	Initialize the software trigger and trigger a single image specified by its name in the client assistant.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.1.6.1 GetCurrentImageAndFileName



Get the current image by detaching its internal reference.

Syntax

Definition:

```
METHOD GetCurrentImageAndFileName : HRESULT
VAR_INPUT
    pipImage  : Pointer To ITcVnImage;
    sFileName : STRING;
    nMaxLen   : UINT;
END_VAR
```

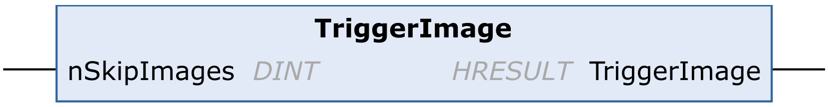
Inputs

Name	Type	Description
pipImage	Pointer To ITcVnImage [▶ 390]	pointer to the interface pointer to be returned, might be 0 if the image acquisition failed
sFileName	STRING	returns the filename as a string
nMaxLen	UINT	maximum string length allowed to be written in sFileName

Return value

HRESULT [▶ 122]

6.1.3.2.1.6.2 TriggerImage



Initialize the software trigger and trigger a single image. Skips nSkipImages and triggers the capturing of image nSkipImages+1.

Syntax

Definition:

```
METHOD TriggerImage : HRESULT
VAR_INPUT
    nSkipImages : DINT;
END_VAR
```

Inputs

Name	Type	Description
nSkipImages	DINT	Number of images to skip

Return value

HRESULT [▶ 122]

6.1.3.2.1.6.3 TriggerImageByName

Initialize the software trigger and trigger a single image specified by its name in the client assistant.

Syntax

Definition:

```
METHOD TriggerImageByName : HRESULT
VAR_INPUT
    sImageName : STRING;
END_VAR
```

Inputs

Name	Type	Description
sImageName	STRING	Image name to trigger

Return value

HRESULT [▶ 122]

6.1.3.2.1.7 ITcVnGevImageProvider

Interface for a GigE Vision image provider.

Inheritance Hierarchy

ITcUnknown [▶ 407]
        ITcVnImageProvider [▶ 369]
                ITcVnGevImageProvider

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
CheckIfBusy [▶ 370]	ITcVnImageProvider	Check whether the module is busy; i.e, an operation is in progress.
CheckIfCameraConnected [▶ 371]	ITcVnImageProvider	Check if the camera is properly connected (Returns S_OK if connected and S_FALSE if not. Can only check software connection, i.e. the check is unable to determine if the cable is connected or not in case that S_FALSE is returned).
GetCurrentImage [▶ 371]	ITcVnImageProvider	Gets the current image by detaching its internal reference.
GetError [▶ 371]	ITcVnImageProvider	Gets result of the last operation performed.
OpenCamera [▶ 372]	ITcVnImageProvider	Open camera (open the control channel, open a stream channel, and activate the GVSP receiver).
CloseCamera [▶ 372]	ITcVnImageProvider	Close camera (deactivate the GVSP receiver, close the stream channel, and close the control channel).
StartAcquisition [▶ 372]	ITcVnImageProvider	Start image acquisition.
StopAcquisition [▶ 373]	ITcVnImageProvider	Stop image acquisition.
SoftwareTrigger [▶ 373]	ITcVnImageProvider	Initialize the software trigger and trigger it. The initialization step is skipped if it was executed before.
InitializeCamera [▶ 373]	ITcVnImageProvider	Set camera to initial state.
GetCurrentGevImage [▶ 368]	ITcVnGevImageProvider	Gets the current GigE Vision image by detaching its internal reference.
GetCurrentImageWithGvspInfo [▶ 369]	ITcVnGevImageProvider	Gets the current image by detaching its internal reference and additionally provide the GVSP info.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.1.7.1 GetCurrentGevImage

Gets the current GigE Vision image by detaching its internal reference.

Syntax

Definition:

```
METHOD GetCurrentGevImage : HRESULT
VAR_INPUT
    pipGevImage : Pointer To ITcIoGevImage;
END_VAR
```

Inputs

Name	Type	Description
pipGevImage	Pointer To ITcIoGevImage [▶ 386]	Pointer to the interface pointer to be returned, might be 0 if the image acquisition failed.

Return value

HRESULT [▶ 122]

6.1.3.2.1.7.2 GetCurrentImageWithGvspInfo

Gets the current image by detaching its internal reference and additionally provide the GVSP info.

Syntax

Definition:

```
METHOD GetCurrentImageWithGvspInfo : HRESULT
VAR_INPUT
    pipImage   : Pointer To ITcVnImage;
    stGvspInfo : Reference To GVSP_IMAGE_INFO;
END_VAR
```

Inputs

Name	Type	Description
pipImage	Pointer To ITcVnImage [▶ 390]	Pointer to the interface pointer to be returned, might be 0 if the image acquisition failed.
stGvspInfo	Reference To GVSP_IMAGE_INFO [▶ 204]	Contains useful meta information like image id, time stamp etc.

Return value

HRESULT [▶ 122]

6.1.3.2.1.8 ITcVnImageProvider

Interface for an image provider.

Inheritance Hierarchy

ITcUnknown [▶ 407]
        ITcVnImageProvider

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
CheckIfBusy [▶ 370]	ITcVnImageProvider	Check whether the module is busy; i.e, an operation is in progress.
CheckIfCameraConnected [▶ 371]	ITcVnImageProvider	Check if the camera is properly connected (Returns S_OK if connected and S_FALSE if not. Can only check software connection, i.e. the check is unable to determine if the cable is connected or not in case that S_FALSE is returned).
GetCurrentImage [▶ 371]	ITcVnImageProvider	Gets the current image by detaching its internal reference.
GetError [▶ 371]	ITcVnImageProvider	Gets result of the last operation performed.
OpenCamera [▶ 372]	ITcVnImageProvider	Open camera (open the control channel, open a stream channel, and activate the GVSP receiver).
CloseCamera [▶ 372]	ITcVnImageProvider	Close camera (deactivate the GVSP receiver, close the stream channel, and close the control channel).
StartAcquisition [▶ 372]	ITcVnImageProvider	Start image acquisition.
StopAcquisition [▶ 373]	ITcVnImageProvider	Stop image acquisition.
SoftwareTrigger [▶ 373]	ITcVnImageProvider	Initialize the software trigger and trigger it. The initialization step is skipped if it was executed before.
InitializeCamera [▶ 373]	ITcVnImageProvider	Set camera to initial state.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.1.8.1 CheckIfBusy

Check whether the module is busy; i.e, an operation is in progress.

Syntax

Definition:

```
METHOD CheckIfBusy : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.2.1.8.2 CheckIfCameraConnected

CheckIfCameraConnected

HRESULT CheckIfCameraConnected

Check if the camera is properly connected (Returns S_OK if connected and S_FALSE if not. Can only check software connection, i.e. the check is unable to determine if the cable is connected or not in case that S_FALSE is returned).

Syntax

Definition:

```
METHOD CheckIfCameraConnected : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.2.1.8.3 GetCurrentImage

GetCurrentImage

pipImage *Pointer To ITcVnImage* *HRESULT* GetCurrentImage

Gets the current image by detaching its internal reference.

Syntax

Definition:

```
METHOD GetCurrentImage : HRESULT
VAR_INPUT
    pipImage : Pointer To ITcVnImage;
END_VAR
```

Inputs

Name	Type	Description
pipImage	Pointer To ITcVnImage [▶ 390]	Pointer to the interface pointer to be returned, might be 0 if the image acquisition failed.

Return value

HRESULT [▶ 122]

6.1.3.2.1.8.4 GetError

GetError

HRESULT GetError

Gets result of the last operation performed.

Syntax

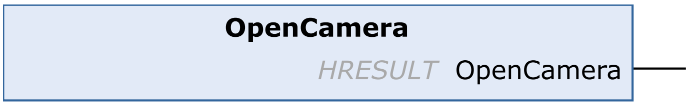
Definition:

```
METHOD GetError : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.2.1.8.5 OpenCamera



Open camera (open the control channel, open a stream channel, and activate the GVSP receiver).

Syntax

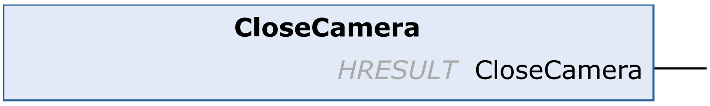
Definition:

```
METHOD OpenCamera : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.2.1.8.6 CloseCamera



Close camera (deactivate the GVSP receiver, close the stream channel, and close the control channel).

Syntax

Definition:

```
METHOD CloseCamera : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.2.1.8.7 StartAcquisition

Start image acquisition.

Syntax

Definition:

```
METHOD StartAcquisition : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.2.1.8.8 StopAcquisition

StopAcquisition
HRESULT StopAcquisition

Stop image acquisition.

Syntax

Definition:

```
METHOD StopAcquisition : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.2.1.8.9 SoftwareTrigger

SoftwareTrigger
HRESULT SoftwareTrigger

Initialize the software trigger and trigger it. The initialization step is skipped if it was executed before.

Syntax

Definition:

```
METHOD SoftwareTrigger : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.2.1.8.10 InitializeCamera

InitializeCamera
HRESULT InitializeCamera

Set camera to initial state.

Syntax

Definition:

```
METHOD InitializeCamera : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.2.2 Export

该组包含处理与图像有关的数据输出的接口。

6.1.3.2.2.1 ITcVnBitmapExport

Interface for exporting an image as a Windows Bitmap (BMP).

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnBitmapExport

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
GetBitmapSize [▶ 374]	ITcVnBitmapExport	Gets size of the image if it is converted to a Windows Bitmap (BMP).
GetBitmapImage [▶ 375]	ITcVnBitmapExport	Export the image as a Windows Bitmap (BMP) into a given buffer.

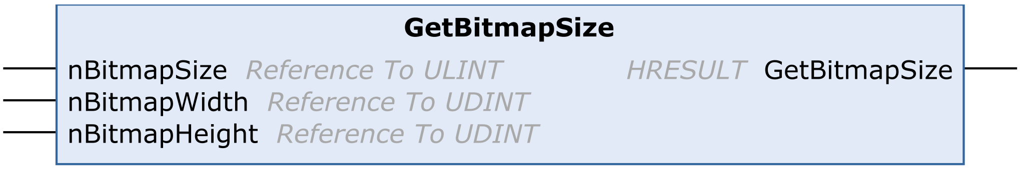
更多信息

根据定义，一个位图只能有 8 位（对于单通道图像）或 24 位（对于 3 通道和 4 通道图像）。因此，16 位图像以及元素类型为 REAL 或 LREAL 的图像被转换为位图输出。在 16 位图像的情况下，数值范围被简单地缩减为 8 位（0 对应于 0，而 255 对应于 65,280）。在 (L)REAL 图像的情况下，值范围 [-1, +1] 被缩放为 [-127, +127] 并作为有符号整数传输。在这些限制之外的值将分别被解释为 -1 或 +1。

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.2.1.1 *GetBitmapSize*



Gets size of the image if it is converted to a Windows Bitmap (BMP).

Syntax

Definition:

```
METHOD GetBitmapSize : HRESULT
VAR_INPUT
    nBitmapSize   : Reference To ULINT;
    nBitmapWidth  : Reference To UDINT;
    nBitmapHeight : Reference To UDINT;
END_VAR
```

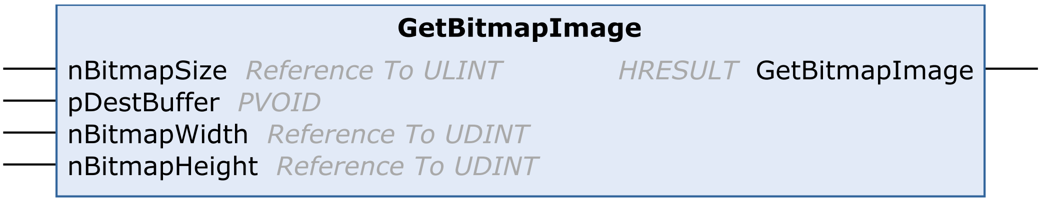
Inputs

Name	Type	Description
nBitmapSize	Reference To ULINT	Output parameter containing the required buffer size.
nBitmapWidth	Reference To UDINT	Desired width or 0 to keep the original width (in) and actual width (out).
nBitmapHeight	Reference To UDINT	Desired height or 0 to keep the original height (in) and actual height (out).

Return value

HRESULT [▶ 122]

6.1.3.2.2.1.2 GetBitmapImage



Export the image as a Windows Bitmap (BMP) into a given buffer.

Syntax

Definition:

```
METHOD GetBitmapImage : HRESULT
VAR_INPUT
    nBitmapSize   : Reference To ULINT;
    pDestBuffer   : PVOID;
    nBitmapWidth  : Reference To UDINT;
    nBitmapHeight : Reference To UDINT;
END_VAR
```

Inputs

Name	Type	Description
nBitmapSize	Reference To ULINT	Maximum buffer size (in) and actual buffer size (out).
pDestBuffer	PVOID	Pointer to the destination buffer.
nBitmapWidth	Reference To UDINT	Desired width or 0 to keep the original width (in) and actual width (out).
nBitmapHeight	Reference To UDINT	Desired height or 0 to keep the original height (in) and actual height (out).

Return value

HRESULT [▶ 122]

6.1.3.2.2.2 ITcVnBitmapExportNotification

Interface required for sending displayable images as ADS notifications

Inheritance Hierarchy

ITcUnknown [▶ 407]
        ITcVnBitmapExportNotification

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
GetTimestamp [▶ 376]	ITcVnBitmapExportNotification	Gets the timestamp of the latest image change.
GetBitmapImageRpcUnlocked [▶ 376]	ITcVnBitmapExportNotification	Export the image as a Windows Bitmap (BMP) into a given buffer by means of an unlocked remote procedure call. It lies within the responsibility of the user to ensure that no conflicting accesses can occur.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.2.2.1 GetTimestamp

Gets the timestamp of the latest image change.

Syntax

Definition:

```
METHOD GetTimestamp : HRESULT
VAR_INPUT
    nTimestamp : Reference To LINT;
END_VAR
```

Inputs

Name	Type	Description
nTimestamp	Reference To LINT	Returns the timestamp.

Return value

HRESULT [▶ 122]

6.1.3.2.2.2.2 GetBitmapImageRpcUnlocked

Export the image as a Windows Bitmap (BMP) into a given buffer by means of an unlocked remote procedure call. It lies within the responsibility of the user to ensure that no conflicting accesses can occur.

Syntax

Definition:

```
METHOD GetBitmapImageRpcUnlocked : HRESULT
VAR_INPUT
    nBitmapSize   : Reference To ULINT;
    pDestBuffer   : PVOID;
    nBitmapWidth  : Reference To UDINT;
    nBitmapHeight : Reference To UDINT;
END_VAR
```

Inputs

Name	Type	Description
nBitmapSize	Reference To ULINT	Maximum buffer size (in) and actual buffer size (out).
pDestBuffer	PVOID	Pointer to the destination buffer.
nBitmapWidth	Reference To UDINT	Desired width or 0 to keep the original width (in) and actual width (out).
nBitmapHeight	Reference To UDINT	Desired height or 0 to keep the original height (in) and actual height (out).

Return value

HRESULT [▸ 122]

6.1.3.2.2.3 ITcVnBitmapExportRpcUnlocked

Interface to export an image as a Windows Bitmap (BMP) into a given buffer by means of an unlocked remote procedure call.

Inheritance Hierarchy

ITcUnknown [▸ 407]
    ITcVnBitmapExport [▸ 373]
        ITcVnBitmapExportRpcUnlocked

Methods

Name	Origin	Description
TcAddRef [▸ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 408]	ITcUnknown	Decrements the reference counter.
GetBitmapSize [▸ 374]	ITcVnBitmapExport	Gets size of the image if it is converted to a Windows Bitmap (BMP).
GetBitmapImage [▸ 375]	ITcVnBitmapExport	Export the image as a Windows Bitmap (BMP) into a given buffer.
GetBitmapImageRpcUnlocked [▸ 378]	ITcVnBitmapExportRpcUnlocked	Export the image as a Windows Bitmap (BMP) into a given buffer by means of an unlocked remote procedure call. It lies within the responsibility of the user to ensure that no conflicting accesses can occur.

更多信息

根据定义，一个位图只能有 8 位（对于单通道图像）或 24 位（对于 3 通道和 4 通道图像）。因此，16 位图像以及元素类型为 REAL 或 LREAL 的图像被转换为位图输出。在 16 位图像的情况下，数值范围被简单地缩减为 8 位（0 对应于 0，而 255 对应于 65,280）。在 (L)REAL 图像的情况下，值范围 [-1, +1] 被缩放为 [-127, +127] 并作为有符号整数传输。在这些限制之外的值将分别被解释为 -1 或 +1。

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.2.3.1 GetBitmapImageRpcUnlocked

Export the image as a Windows Bitmap (BMP) into a given buffer by means of an unlocked remote procedure call. It lies within the responsibility of the user to ensure that no conflicting accesses can occur.

Syntax

Definition:

```
METHOD GetBitmapImageRpcUnlocked : HRESULT
VAR_INPUT
    nBitmapSize   : Reference To ULINT;
    pDestBuffer   : PVOID;
    nBitmapWidth  : Reference To UDINT;
    nBitmapHeight : Reference To UDINT;
END_VAR
```

Inputs

Name	Type	Description
nBitmapSize	Reference To ULINT	Maximum buffer size (in) and actual buffer size (out).
pDestBuffer	PVOID	Pointer to the destination buffer.
nBitmapWidth	Reference To UDINT	Desired width or 0 to keep the original width (in) and actual width (out).
nBitmapHeight	Reference To UDINT	Desired height or 0 to keep the original height (in) and actual height (out).

Return value

HRESULT [▶ 122]

6.1.3.2.2.4 ITcVnHistogramExport

Interface for exporting a histogram for an image as an array.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnHistogramExport

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
GetHistogramSize [▶ 379]	ITcVnHistogramExport	Gets the size of the histogram array.
GetHistogramArray [▶ 379]	ITcVnHistogramExport	Gets the histogram array.
GetCustomHistogramSize [▶ 380]	ITcVnHistogramExport	Gets the size of a custom histogram array.
GetCustomHistogramArray [▶ 381]	ITcVnHistogramExport	Gets a custom histogram array.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.2.4.1 GetHistogramSize

Gets the size of the histogram array.

Syntax

Definition:

```
METHOD GetHistogramSize : HRESULT
VAR_INPUT
    nHistogramSize : Reference To ULINT;
END_VAR
```

Inputs

Name	Type	Description
nHistogramSize	Reference To ULINT	Output parameter containing the required buffer size.

Return value

HRESULT [▶ 122]

6.1.3.2.2.4.2 GetHistogramArray

Gets the histogram array.

Syntax

Definition:

```
METHOD GetHistogramArray : HRESULT
VAR_INPUT
    nHistogramSize : Reference To ULINT;
    pDestBuffer    : PVOID;
END_VAR
```

Inputs

Name	Type	Description
nHistogramSize	Reference To ULINT	Maximum buffer size (in) and actual buffer size (out).
pDestBuffer	PVOID	Pointer to the destination buffer.

Return value

HRESULT [▶ 122]

6.1.3.2.2.4.3 GetCustomHistogramSize

Gets the size of a custom histogram array.

Syntax

Definition:

```
METHOD GetCustomHistogramSize : HRESULT
VAR_INPUT
    nHistogramSize : Reference To ULINT;
    nBins          : Reference To UDINT;
END_VAR
```

Inputs

Name	Type	Description
nHistogramSize	Reference To ULINT	Output parameter containing the required buffer size.
nBins	Reference To UDINT	Desired number of bins or 0 to keep the default for the corresponding image format (in) and default number of bins (out).

Return value

HRESULT [▶ 122]

6.1.3.2.2.4.4 GetCustomHistogramArray

GetCustomHistogramArray	
nHistogramSize *Reference To ULINT*	*HRESULT* GetCustomHistogramArray
pDestBuffer *PVOID*	
nBins *Reference To UDINT*	
fLowerBound *Reference To LREAL*	
fUpperBound *Reference To LREAL*	

Gets a custom histogram array.

Syntax

Definition:

```
METHOD GetCustomHistogramArray : HRESULT
VAR_INPUT
    nHistogramSize : Reference To ULINT;
    pDestBuffer    : PVOID;
    nBins          : Reference To UDINT;
    fLowerBound    : Reference To LREAL;
    fUpperBound    : Reference To LREAL;
END_VAR
```

Inputs

Name	Type	Description
nHistogramSize	Reference To ULINT	Maximum buffer size (in) and actual buffer size (out).
pDestBuffer	PVOID	Pointer to the destination buffer.
nBins	Reference To UDINT	Desired number of bins or 0 to keep the default for the corresponding image format (in) and default number of bins (out).
fLowerBound	Reference To LREAL	Lower (inclusive) boundary of the 0-th histogram bin (in), or receive the default if fLowerBound AND fUpperBound are set to 0 (out).
fUpperBound	Reference To LREAL	Upper (exclusive) boundary of the last histogram bin nBins-1 (in), or receive the default if fLowerBound AND fUpperBound are set to 0 (out).

Return value

HRESULT [▶ 122]

6.1.3.2.2.5 ITcVnTiffExport

Interface for exporting an image as tiff.

Inheritance Hierarchy

ITcUnknown [▶ 407]
        ITcVnTiffExport

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
GetTiffSize [▶ 382]	ITcVnTiffExport	Gets size of the image if it is converted to tiff.
GetTiffImage [▶ 382]	ITcVnTiffExport	Export the image as tiff into a given buffer.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.2.5.1 GetTiffSize

Gets size of the image if it is converted to tiff.

Syntax

Definition:

```
METHOD GetTiffSize : HRESULT
VAR_INPUT
    nTiffSize : Reference To ULINT;
END_VAR
```

Inputs

Name	Type	Description
nTiffSize	Reference To ULINT	Output parameter containing the required buffer size.

Return value

HRESULT [▶ 122]

6.1.3.2.2.5.2 GetTiffImage

Export the image as tiff into a given buffer.

Syntax

Definition:

```
METHOD GetTiffImage : HRESULT
VAR_INPUT
    nTiffSize   : Reference To ULINT;
    pDestBuffer : PVOID;
END_VAR
```

Inputs

Name	Type	Description
nTiffSize	Reference To ULINT	Maximum buffer size (in) and actual buffer size (out).
pDestBuffer	PVOID	Pointer to the destination buffer.

Return value

HRESULT [▸ 122]

6.1.3.2.2.6 ITcVnTiffExportNotification

Interface for sending displayable images as tiff via ADS notifications.

Inheritance Hierarchy

ITcUnknown [▸ 407]
    ITcVnTiffExportNotification

Methods

Name	Origin	Description
TcAddRef [▸ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 408]	ITcUnknown	Decrements the reference counter.
GetTimestamp [▸ 383]	ITcVnTiffExportNotification	Gets the timestamp of the latest image change.
GetTiffImageRpcUnlocked [▸ 384]	ITcVnTiffExportNotification	Export the image as tiff into a given buffer by means of an unlocked remote procedure call. It lies within the responsibility of the user to ensure that no conflicting accesses can occur.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.2.6.1 GetTimestamp

Gets the timestamp of the latest image change.

Syntax

Definition:

```
METHOD GetTimestamp : HRESULT
VAR_INPUT
    nTimestamp : Reference To LINT;
END_VAR
```

Inputs

Name	Type	Description
nTimestamp	Reference To LINT	Returns the timestamp.

Return value

HRESULT [▶ 122]

6.1.3.2.2.6.2 GetTiffImageRpcUnlocked

Export the image as tiff into a given buffer by means of an unlocked remote procedure call. It lies within the responsibility of the user to ensure that no conflicting accesses can occur.

Syntax

Definition:

```
METHOD GetTiffImageRpcUnlocked : HRESULT
VAR_INPUT
    nTiffSize   : Reference To ULINT;
    pDestBuffer : PVOID;
END_VAR
```

Inputs

Name	Type	Description
nTiffSize	Reference To ULINT	Maximum buffer size (in) and actual buffer size (out).
pDestBuffer	PVOID	Pointer to the destination buffer.

Return value

HRESULT [▶ 122]

6.1.3.2.2.7 ITcVnTiffExportRpcUnlocked

Interface for exporting an image as tiff by means of an unlocked remote procedure call.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnTiffExport [▶ 381]
        ITcVnTiffExportRpcUnlocked

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
GetTiffSize [▶ 382]	ITcVnTiffExport	Gets size of the image if it is converted to tiff.
GetTiffImage [▶ 382]	ITcVnTiffExport	Export the image as tiff into a given buffer.
GetTiffImageRpcUnlocked [▶ 385]	ITcVnTiffExportRpcUnlocked	Export the image as tiff into a given buffer by means of an unlocked remote procedure call. It lies within the responsibility of the user to ensure that no conflicting accesses can occur.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.2.7.1 GetTiffImageRpcUnlocked

Export the image as tiff into a given buffer by means of an unlocked remote procedure call. It lies within the responsibility of the user to ensure that no conflicting accesses can occur.

Syntax

Definition:

```
METHOD GetTiffImageRpcUnlocked : HRESULT
VAR_INPUT
    nTiffSize   : Reference To ULINT;
    pDestBuffer : PVOID;
END_VAR
```

Inputs

Name	Type	Description
nTiffSize	Reference To ULINT	Maximum buffer size (in) and actual buffer size (out).
pDestBuffer	PVOID	Pointer to the destination buffer.

Return value

HRESULT [▶ 122]

6.1.3.2.3 Import

该组包含用于处理与图像相关的数据导入的接口。

6.1.3.2.3.1 ITcVnTiffImport

Interface for importing a tiff image.

Inheritance Hierarchy

ITcUnknown [▸ 407]
        ITcVnTiffImport

Methods

Name	Origin	Description
TcAddRef [▸ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 408]	ITcUnknown	Decrements the reference counter.
ImportTiffImage [▸ 386]	ITcVnTiffImport	Import a tiff image.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.3.1.1 ImportTiffImage

Import a tiff image.

Syntax

Definition:

```
METHOD ImportTiffImage : HRESULT
VAR_INPUT
    pTiffImage : PVOID;
END_VAR
```

Inputs

Name	Type	Description
pTiffImage	PVOID	Pointer to the tiff file.

Return value

HRESULT [▸ 122]

6.1.3.2.4 ITcIoGevImage

Interface for GigE Vision images.

Inheritance Hierarchy

ITcUnknown [▸ 407]
        ITcVnImageBase [▸ 393]
                ITcIoGevImage

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
GetImageData [▶ 394]	ITcVnImageBase	Gets the data interface (Do not explicitly free the pointer in there!).
GetImageSize [▶ 394]	ITcVnImageBase	Gets the image size.
GetWidth [▶ 395]	ITcVnImageBase	Gets the image width (alternatively use F_VN_GetImageWidth).
GetHeight [▶ 395]	ITcVnImageBase	Gets the image height (alternatively use F_VN_GetImageHeight).
GetXPadding [▶ 395]	ITcVnImageBase	Gets the horizontal padding.
GetYPadding [▶ 396]	ITcVnImageBase	Gets the vertical padding.
GetPixelFormat [▶ 396]	ITcVnImageBase	Gets the pixel format (alternatively use F_VN_GetPixelFormat).
GetImageInfo [▶ 397]	ITcVnImageBase	Gets a struct containing all common meta infos of the image. This basically encompasses all meta information accessible via this interface. (Alternatively use F_VN_GetImageInfo.)
CheckIfCopy [▶ 387]	ITcIoGevImage	Check if image data is a copy.
DecreaseHeight [▶ 388]	ITcIoGevImage	Sets image height to newHeight if it is smaller than the current value.
GetBlockId [▶ 388]	ITcIoGevImage	Gets the GVSP block ID.
GetGevStatus [▶ 388]	ITcIoGevImage	Gets the block status.
GetGvspImageInfo [▶ 389]	ITcIoGevImage	Gets a pointer to the GVSP leader payload.
SetGevStatus [▶ 389]	ITcIoGevImage	Sets the block status.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.4.1 CheckIfCopy

Check if image data is a copy.

Syntax

Definition:

```
METHOD CheckIfCopy : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.2.4.2 DecreaseHeight

Sets image height to newHeight if it is smaller than the current value.

Syntax

Definition:

```
METHOD DecreaseHeight : HRESULT
VAR_INPUT
    nNewHeight : UDINT;
END_VAR
```

Inputs

Name	Type	Description
nNewHeight	UDINT	Height value to set.

Return value

HRESULT [▸ 122]

6.1.3.2.4.3 GetBlockId

Gets the GVSP block ID.

Syntax

Definition:

```
METHOD GetBlockId : HRESULT
VAR_INPUT
    nBlockId : Reference To ULINT;
END_VAR
```

Inputs

Name	Type	Description
nBlockId	Reference To ULINT	Returns the block ID.

Return value

HRESULT [▸ 122]

6.1.3.2.4.4 GetGevStatus

Gets the block status.

Syntax

Definition:

```
METHOD GetGevStatus : HRESULT
VAR_INPUT
    nGevStatus : Reference To UINT;
END_VAR
```

Inputs

Name	Type	Description
nGevStatus	Reference To UINT	Returns the GigE Vision status code.

Return value

HRESULT [▶ 122]

6.1.3.2.4.5 GetGvspImageInfo

GetGvspImageInfo

ppGvspImageInfo *Pointer To Pointer To GVSP_IMAGE_INFO* *HRESULT* GetGvspImageInfo

Gets a pointer to the GVSP leader payload.

Syntax

Definition:

```
METHOD GetGvspImageInfo : HRESULT
VAR_INPUT
    ppGvspImageInfo : Pointer To Pointer To GVSP_IMAGE_INFO;
END_VAR
```

Inputs

Name	Type	Description
ppGvspImageInfo	Pointer To Pointer To GVSP_IMAGE_INFO [▶ 204]	Pointer to the GVSP meta information.

Return value

HRESULT [▶ 122]

6.1.3.2.4.6 SetGevStatus

SetGevStatus

nGevStatus *UINT* *HRESULT* SetGevStatus

Sets the block status.

Syntax

Definition:

```
METHOD SetGevStatus : HRESULT
VAR_INPUT
    nGevStatus : UINT;
END_VAR
```

Inputs

Name	Type	Description
nGevStatus	UINT	GigE Vision status code.

Return value

HRESULT [▶ 122]

6.1.3.2.5 ITcVnDisplayableImage

Interface for displayable images.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnDisplayableImage

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.

更多信息

该接口用于通过 ADS 显示图像（例如使用 ADS 图像查看）。它不适合用于处理图像。处理图像需要 ITcVnImage [▶ 390]接口。关于更多信息，请参见可处理和可显示图像之间的区别 [▶ 130]。

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.6 ITcVnImage

Basic interface for images.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnImageBase [▶ 393]
ITcVnImage

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
GetImageData [▶ 394]	ITcVnImageBase	Gets the data interface (Do not explicitly free the pointer in there!).
GetImageSize [▶ 394]	ITcVnImageBase	Gets the image size.
GetWidth [▶ 395]	ITcVnImageBase	Gets the image width (alternatively use F_VN_GetImageWidth).
GetHeight [▶ 395]	ITcVnImageBase	Gets the image height (alternatively use F_VN_GetImageHeight).
GetXPadding [▶ 395]	ITcVnImageBase	Gets the horizontal padding.
GetYPadding [▶ 396]	ITcVnImageBase	Gets the vertical padding.
GetPixelFormat [▶ 396]	ITcVnImageBase	Gets the pixel format (alternatively use F_VN_GetPixelFormat).
GetImageInfo [▶ 397]	ITcVnImageBase	Gets a struct containing all common meta infos of the image. This basically encompasses all meta information accessible via this interface. (Alternatively use F_VN_GetImageInfo.)
Init [▶ 391]	ITcVnImage	Initialize an image with an ITcVnImageBase interface.
GetRowPointer [▶ 392]	ITcVnImage	Gets a pointer to a specific row of an image.
ReleaseRowPointer [▶ 392]	ITcVnImage	Release the pointer to a specific row of an image.

更多信息

接口ITcVnImage用于处理图像 [▶ 130]。它提供了获取元数据（如图像尺寸、通道数、像素类型等）的方法以及访问原始图像数据的方法。

这个接口不适合通过 ADS 显示图像（例如通过ADS 图像查看 [▶ 115]）。通过 ADS 显示图像需要ITcVnDisplayableImage [▶ 390]接口。可以通过以下功能进行转换：

- F_VN_CopyIntoDisplayableImage [▶ 753]
- F_VN_TransformIntoDisplaybleImage [▶ 784]

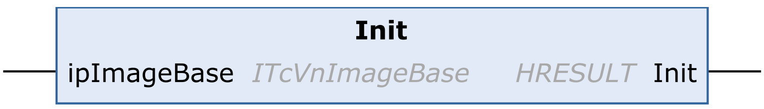
相关函数

- 基本图像操作 [▶ 745]
- 代数图像操作 [▶ 477]

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.6.1 Init



Initialize an image with an ITcVnImageBase interface.

Syntax

Definition:

```
METHOD Init : HRESULT
VAR_INPUT
    ipImageBase : ITcVnImageBase;
END_VAR
```

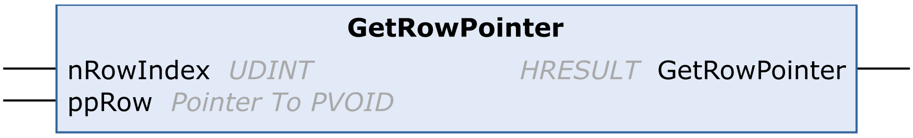
Inputs

Name	Type	Description
ipImageBase	ITcVnImageBase [▶ 393]	Image from which to obtain the data.

Return value

HRESULT [▶ 122]

6.1.3.2.6.2 GetRowPointer



Gets a pointer to a specific row of an image.

Syntax

Definition:

```
METHOD GetRowPointer : HRESULT
VAR_INPUT
    nRowIndex : UDINT;
    ppRow     : Pointer To PVOID;
END_VAR
```

Inputs

Name	Type	Description
nRowIndex	UDINT	Row index.
ppRow	Pointer To PVOID	Returns a pointer to the requested image row.

Return value

HRESULT [▶ 122]

更多信息

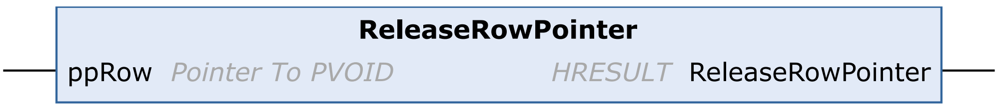
确保描述指针的基本类型与图像的像素类型相同。

注意

启用行指针

如果使用 GetRowPointer 创建了一个指针，你也必须用 ReleaseRowPointer 方法再次释放它！

6.1.3.2.6.3 ReleaseRowPointer



Release the pointer to a specific row of an image.

Syntax

Definition:

```
METHOD ReleaseRowPointer : HRESULT
VAR_INPUT
    ppRow : Pointer To PVOID;
END_VAR
```

Inputs

Name	Type	Description
ppRow	Pointer To PVOID	Pointer to the row pointer to be released.

Return value

HRESULT [▶ 122]

6.1.3.2.7 ITcVnImageBase

Base interface for all image types.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnImageBase

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
GetImageData [▶ 394]	ITcVnImageBase	Gets the data interface (Do not explicitly free the pointer in there!).
GetImageSize [▶ 394]	ITcVnImageBase	Gets the image size.
GetWidth [▶ 395]	ITcVnImageBase	Gets the image width (alternatively use F_VN_GetImageWidth).
GetHeight [▶ 395]	ITcVnImageBase	Gets the image height (alternatively use F_VN_GetImageHeight).
GetXPadding [▶ 395]	ITcVnImageBase	Gets the horizontal padding.
GetYPadding [▶ 396]	ITcVnImageBase	Gets the vertical padding.
GetPixelFormat [▶ 396]	ITcVnImageBase	Gets the pixel format (alternatively use F_VN_GetPixelFormat).
GetImageInfo [▶ 397]	ITcVnImageBase	Gets a struct containing all common meta infos of the image. This basically encompasses all meta information accessible via this interface. (Alternatively use F_VN_GetImageInfo.)

更多信息

接口ITcVnImageBase用于内部用途，而用户通常不需要。

对于处理图像 [▶ 130]，请使用ITcVnImage [▶ 390]接口代替。

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.2.7.1 GetImageData

Gets the data interface (Do not explicitly free the pointer in there!).

Syntax

Definition:

```
METHOD GetImageData : HRESULT
VAR_INPUT
    pipImageData : Pointer To ITcVnData;
END_VAR
```

Inputs

Name	Type	Description
pipImageData	Pointer To ITcVnData [▶ 402]	Returns a pointer to the image data interface.

Return value

HRESULT [▶ 122]

6.1.3.2.7.2 GetImageSize

Gets the image size.

Syntax

Definition:

```
METHOD GetImageSize : HRESULT
VAR_INPUT
    nSize : Reference To ULINT;
END_VAR
```

Inputs

Name	Type	Description
nSize	Reference To ULINT	Returns the image size in bytes.

Return value

HRESULT [▶ 122]

6.1.3.2.7.3 GetWidth

GetWidth

nWidth *Reference To UDINT* *HRESULT* GetWidth

Gets the image width (alternatively use F_VN_GetImageWidth).

Syntax

Definition:

```
METHOD GetWidth : HRESULT
VAR_INPUT
    nWidth : Reference To UDINT;
END_VAR
```

Inputs

Name	Type	Description
nWidth	Reference To UDINT	Returns the image width in pixels.

Return value

HRESULT [▶ 122]

6.1.3.2.7.4 GetHeight

GetHeight

nHeight *Reference To UDINT* *HRESULT* GetHeight

Gets the image height (alternatively use F_VN_GetImageHeight).

Syntax

Definition:

```
METHOD GetHeight : HRESULT
VAR_INPUT
    nHeight : Reference To UDINT;
END_VAR
```

Inputs

Name	Type	Description
nHeight	Reference To UDINT	Returns the image height in pixels.

Return value

HRESULT [▶ 122]

6.1.3.2.7.5 GetXPadding

GetXPadding

nXPadding *Reference To UINT* *HRESULT* GetXPadding

Gets the horizontal padding.

Syntax

Definition:

```
METHOD GetXPadding : HRESULT
VAR_INPUT
    nXPadding : Reference To UINT;
END_VAR
```

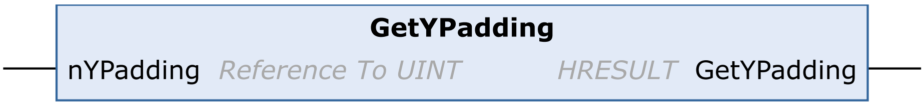
Inputs

Name	Type	Description
nXPadding	Reference To UINT	Returns the horizontal padding in bytes.

Return value

HRESULT [▶ 122]

6.1.3.2.7.6 GetYPadding



Gets the vertical padding.

Syntax

Definition:

```
METHOD GetYPadding : HRESULT
VAR_INPUT
    nYPadding : Reference To UINT;
END_VAR
```

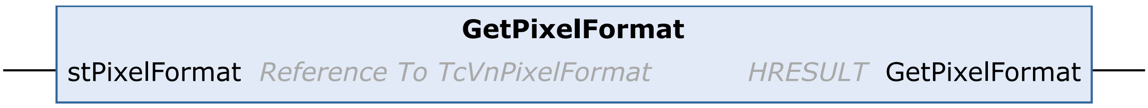
Inputs

Name	Type	Description
nYPadding	Reference To UINT	Returns the vertical padding in bytes.

Return value

HRESULT [▶ 122]

6.1.3.2.7.7 GetPixelFormat



Gets the pixel format (alternatively use F_VN_GetPixelFormat).

Syntax

Definition:

```
METHOD GetPixelFormat : HRESULT
VAR_INPUT
    stPixelFormat : Reference To TcVnPixelFormat;
END_VAR
```

Inputs

Name	Type	Description
stPixelFormat	Reference To TcVnPixelFormat [▸ 224]	Returns a struct describing the pixel format.

Return value

HRESULT [▸ 122]

6.1.3.2.7.8 GetImageInfo

GetImageInfo

stImageInfo *Reference To TcVnImageInfo* *HRESULT* GetImageInfo

Gets a struct containing all common meta infos of the image. This basically encompasses all meta information accessible via this interface. (Alternatively use F_VN_GetImageInfo.)

Syntax

Definition:

```
METHOD GetImageInfo : HRESULT
VAR_INPUT
    stImageInfo : Reference To TcVnImageInfo;
END_VAR
```

Inputs

Name	Type	Description
stImageInfo	Reference To TcVnImageInfo [▸ 210]	Returns a struct describing the image.

Return value

HRESULT [▸ 122]

6.1.3.3 Machine Learning

该组包含机器学习的接口。

6.1.3.3.1 ITcVnColorModel

Interface to train and classify an image color.

Inheritance Hierarchy

ITcUnknown [▸ 407]
        ITcVnMlModel [▸ 399]
                ITcVnColorModel

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
GetTypeGuid [▶ 400]	ITcVnMlModel	Gets the GUID of the model.
GetTypeName [▶ 400]	ITcVnMlModel	Gets the model type name as a string.
TrainImageColor [▶ 398]	ITcVnColorModel	Train the color of the provided image.
ClassifyImageColor [▶ 399]	ITcVnColorModel	Compute the similarity of each image pixel to the trained reference color.

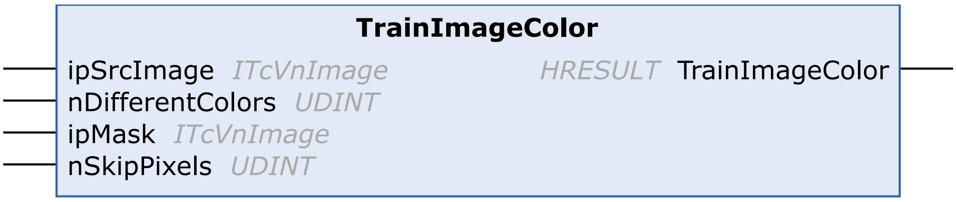
更多信息

接口ITcVnColorModel用于描述一个颜色区域。模型可以通过函数F_VN_TrainImageColor [▶ 1218]进行创建或调整，并使用函数F_VN_ReferenceColorSimilarity_ITcVnColorModel [▶ 1210]与图像的颜色进行比较。

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.3.1.1 TrainImageColor



Train the color of the provided image.

Syntax

Definition:

```
METHOD TrainImageColor : HRESULT
VAR_INPUT
    ipSrcImage       : ITcVnImage;
    nDifferentColors : UDINT;
    ipMask           : ITcVnImage;
    nSkipPixels      : UDINT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image.
nDifferentColors	UDINT	Number of different colors to differentiate.
ipMask	ITcVnImage [▶ 390]	Optional image mask.
nSkipPixels	UDINT	Number of pixels to skip between each evaluated color sample (to achieve a better performance). 0 takes every pixel into account and tends to be more accurate.

Return value

HRESULT [▶ 122]

6.1.3.3.1.2 ClassifyImageColor

Compute the similarity of each image pixel to the trained reference color.

Syntax

Definition:

```
METHOD ClassifyImageColor : HRESULT
VAR_INPUT
    ipSrcImage       : ITcVnImage;
    pipDestImage     : Pointer To ITcVnImage;
    fVariance        : REAL;
    fLuminanceWeight : REAL;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image.
pipDestImage	Pointer To ITcVnImage [▶ 390]	Returns the color similarity.
fVariance	REAL	Allowed color variance.
fLuminanceWeight	REAL	Weight the impact of the luminance.

Return value

HRESULT [▶ 122]

6.1.3.3.2 ITcVnMlModel

Interface for a machine learning model.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcVnMlModel

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
GetTypeGuid [▶ 400]	ITcVnMlModel	Gets the GUID of the model.
GetTypeName [▶ 400]	ITcVnMlModel	Gets the model type name as a string.

更多信息

接口ITcVnMlModel用于处理机器学习模型。目前，这只包括使用接口ITcVnColorModel [▸ 397]。

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.3.2.1 GetTypeGuid

Gets the GUID of the model.

Syntax

Definition:

```
METHOD GetTypeGuid : HRESULT
VAR_INPUT
    nTypeGuid : Reference To GUID;
END_VAR
```

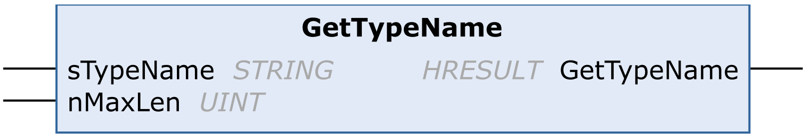
Inputs

Name	Type	Description
nTypeGuid	Reference To GUID	Returns the GUID of the model.

Return value

HRESULT [▸ 122]

6.1.3.3.2.2 GetTypeName



Gets the model type name as a string.

Syntax

Definition:

```
METHOD GetTypeName : HRESULT
VAR_INPUT
    sTypeName : STRING;
    nMaxLen   : UINT;
END_VAR
```

Inputs

Name	Type	Description
sTypeName	STRING	Returns the model type name as a string.
nMaxLen	UINT	Maximum string length allowed to be written in sTypeName.

Return value

HRESULT [▶ 122]

6.1.3.4 Miscellaneous

该组包含其他接口。

6.1.3.4.1 ITcSerializableNotification

Interface required for sending serialized objects as ADS notifications.

Inheritance Hierarchy

ITcUnknown [▶ 407]
    ITcSerializableNotification

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
GetComparisonValue [▶ 401]	ITcSerializableNotification	Gets a value that reflects changes of the underlying object (e.g., the timestamp of the latest object change).
SerializeRpcUnlocked [▶ 402]	ITcSerializableNotification	Serialize an object into a given buffer.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.4.1.1 GetComparisonValue

Gets a value that reflects changes of the underlying object (e.g., the timestamp of the latest object change).

Syntax

Definition:

```
METHOD GetComparisonValue : HRESULT
VAR_INPUT
    nComparisonValue : Reference To LINT;
END_VAR
```

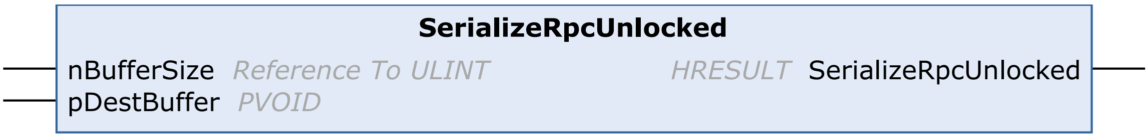
Inputs

Name	Type	Description
nComparisonValue	Reference To LINT	Returns the comparison value.

Return value

HRESULT [▶ 122]

6.1.3.4.1.2 SerializeRpcUnlocked



Serialize an object into a given buffer.

Syntax

Definition:

```
METHOD SerializeRpcUnlocked : HRESULT
VAR_INPUT
    nBufferSize : Reference To ULINT;
    pDestBuffer : PVOID;
END_VAR
```

Inputs

Name	Type	Description
nBufferSize	Reference To ULINT	Maximum buffer size (in) and actually used buffer size (out).
pDestBuffer	PVOID	Pointer to the destination buffer.

Return value

HRESULT [▶ 122]

6.1.3.4.2 ITcVnData

Interface for accessing data.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnData

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
CheckIfCopy [▶ 403]	ITcVnData	Check if the data are a copy (returns S_OK for copies and S_FALSE otherwise).
CheckIfImage [▶ 403]	ITcVnData	Check if the data belongs to an image (returns S_OK for images and S_FALSE otherwise).
GetSize [▶ 404]	ITcVnData	Gets the size of the data.
CheckDataPointer [▶ 404]	ITcVnData	Check if the data pointer is different from 0 (returns S_OK for non-zero data pointers and S_FALSE otherwise).
GetDataPointer [▶ 404]	ITcVnData	Gets the data pointer (the obtained data pointer must be released by calling ReleaseDataPointer).
ReleaseDataPointer [▶ 405]	ITcVnData	Release the data pointer.

更多信息

接口ITcVnData用于内部用途，而用户通常不需要。对于访问原始图像数据，请使用接口ITcVnImage [▶ 390]的函数GetRowPointer代替。

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.4.2.1 CheckIfCopy

Check if the data are a copy (returns S_OK for copies and S_FALSE otherwise).

Syntax

Definition:

```
METHOD CheckIfCopy : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.4.2.2 CheckIfImage

Check if the data belongs to an image (returns S_OK for images and S_FALSE otherwise).

Syntax

Definition:

```
METHOD CheckIfImage : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.4.2.3 GetSize

Gets the size of the data.

Syntax

Definition:

```
METHOD GetSize : HRESULT
VAR_INPUT
    nSize : Reference To ULINT;
END_VAR
```

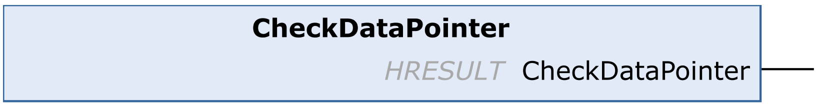
Inputs

Name	Type	Description
nSize	Reference To ULINT	Size of the stored data in bytes.

Return value

HRESULT [▶ 122]

6.1.3.4.2.4 CheckDataPointer



Check if the data pointer is different from 0 (returns S_OK for non-zero data pointers and S_FALSE otherwise).

Syntax

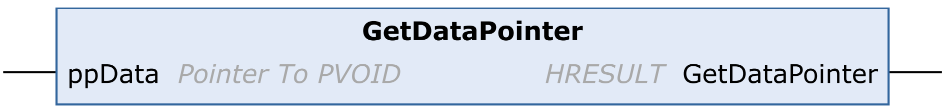
Definition:

```
METHOD CheckDataPointer : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.4.2.5 GetDataPointer



Gets the data pointer (the obtained data pointer must be released by calling ReleaseDataPointer).

Syntax

Definition:

```
METHOD GetDataPointer : HRESULT
VAR_INPUT
    ppData : Pointer To PVOID;
END_VAR
```

Inputs

Name	Type	Description
ppData	Pointer To PVOID	Returns the data pointer.

Return value

HRESULT [▶ 122]

6.1.3.4.2.6 ReleaseDataPointer

Release the data pointer.

Syntax

Definition:

```
METHOD ReleaseDataPointer : HRESULT
VAR_INPUT
    ppData : Pointer To PVOID;
END_VAR
```

Inputs

Name	Type	Description
ppData	Pointer To PVOID	Pointer to the data pointer to be released.

Return value

HRESULT [▶ 122]

6.1.3.4.3 ITcVnTimestamp

Offers an interface for timestamps.

Inheritance Hierarchy

ITcUnknown [▶ 407]
ITcVnTimestamp

Methods

Name	Origin	Description
TcAddRef [▶ 407]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 407]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 408]	ITcUnknown	Decrements the reference counter.
UpdateTimestamp [▶ 406]	ITcVnTimestamp	Updates the timestamp to the current time
GetTimestamp [▶ 406]	ITcVnTimestamp	Gets the timestamp

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.3.4.3.1 UpdateTimestamp

Updates the timestamp to the current time

Syntax

Definition:

```
METHOD UpdateTimestamp : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.3.4.3.2 GetTimestamp

Gets the timestamp

Syntax

Definition:

```
METHOD GetTimestamp : HRESULT
VAR_INPUT
    nTimestamp : Reference To LINT;
END_VAR
```

Inputs

Name	Type	Description
nTimestamp	Reference To LINT	Returns the timestamp

Return value

HRESULT [▶ 122]

6.1.3.4.4 接口 ITcUnknown

ITcUnknown 定义引用计数以及对更具体接口的引用查询。

语法

```
TCOM_DECL_INTERFACE("00000001-0000-0000-e000-000000000064", ITcUnknown)
```

声明形式：TcInterfaces.h

要求包括：

方法

名称	描述
TcAddRef [▸ 407]	增加引用计数器。
TcQueryInterface [▸ 407]	通过 IID 查询已实施接口的引用。
TcRelease [▸ 408]	减少引用计数器。

每个 TcCOM 界面都直接或间接从 ITcUnknown 派生 因此，每个 TcCOM 模块类均执行 ITcUnknown，因为它是从 ITComObject 派生。

ITcUnknown 的标准执行可确保在释放最后一个引用后删除对象。因此，在调用 TcRelease() 后，不得取消引用接口指针。

6.1.3.4.4.1 方法 ITcUnknown:TcAddRef

增加引用计数器并返回新值。

语法

```
ULONG TcAddRef( )
```

返回值

产生引用计数值。

6.1.3.4.4.2 方法 ITcUnknown:TcQueryInterface

根据接口 ID (IID) 给出的接口查询接口指针。

语法

```
HRESULT TcQueryInterface(RITCID iid, PPVOID pipItf )
```

返回值

如果成功，将返回 S_OK（"0"）或其它正值，请参见返回值。关于扩展信息，请特别参见 ADS 返回代码 [▸ 2753]中的 HRESULT 栏。

如果所要求的接口不可用，该方法将返回 ADSERR_DEVICE_NOINTERFACE。

参数

名称	类型	描述
iid	RITCID	接口 IID。
pipItf	PPVOID	指向接口指针。当所请求的接口类型在相应实例中可用时设置。

描述

通过 IID 查询已实施接口的引用。建议使用智能指针来初始化和保存接口指针。

变量 1：

```
HRESULT GetTraceLevel(ITcUnkown* ip, TcTraceLevel& tl)
{
HRESULT hr = S_OK;
if (ip != NULL)
{
ITComObjectPtr spObj;
hr = ip->TcQueryInterface(spObj.GetIID(), &spObj);
if (SUCCEEDED(hr))
{
hr = spObj->TcGetObjPara(PID_TcTraceLevel, &tl, sizeof(tl));
}
return hr;
}
}
```

与智能指针相关的接口 ID 可用作 TcQueryInterface 中的参数。“&”操作符将返回指向智能指针内部接口指针成员变量的指针。变量 1 假定当 TcQueryInterface 指示成功时，接口指针被初始化。如果范围仍然存在，则智能指针 spObj 的析构函数会释放引用。

变量 2：

```
HRESULT GetTraceLevel(ITcUnkown* ip, TcTraceLevel& tl)
{
HRESULT hr = S_OK;
ITComObjectPtr spObj = ip;
if (spObj != NULL)
{
spObj->TcGetObjParam(PID_TcTraceLevel, &tl);
}
else
{
hr = ADS_E_NOINTERFACE;
}
return hr;
}
```

如果接口指针 ip 被分配给智能指针 spObj，则将在 ip 引用的实例上使用 IID_ITComObject 隐式调用 TcQueryInterface 方法。这样会缩短代码，但会丢失 TcQueryInterface 的原始返回代码。

6.1.3.4.4.3 方法 ITcUnknown:TcRelease

该方法减少引用计数器。

语法

```
ULONG TcRelease( )
```

返回值

产生引用计数值。

描述

减少引用计数器并返回新值。

如果引用计数器为 0，则对象会自行删除。

6.1.4 Functions

图像处理算法被封装在函数中。更复杂的算法已作为功能块实现，可在 ImageProcessing [▶ 1581] 中找到。

函数调用的结构

函数结构如下：

```
hr := F_VN_ProcessImage(<...>, hr);
```

Status variable of type **HRESULT**
Naming convention of vision API functions
Parameters and return values as references

hr	类型 HRESULT 的状态变量 [▶ 122]。借此可跟踪每个函数的执行状态。
F_VN_	函数的标识前缀为“F_VN”。
参数 <...>	函数参数主要实现以下目的： • 待处理的输入参数（如图像或容器）。 • 用于设置算法的参数 • 作为返回结果引用的参数

函数分组

TwinCAT Vision 库的所有函数按主题分为以下几组：

图像

- 基本图像运算 [▶ 745]
- 代数图像运算 [▶ 477]
- 图像分割 [▶ 1281]
- 图像颜色和对比处理 [▶ 1192]
- 图像分析 [▶ 1136]
- 图像滤波 [▶ 1231]

容器

- 基本容器运算 [▶ 520]
- 代数容器运算 [▶ 436]
- 容器统计 [▶ 818]

轮廓

- 轮廓分析 [▶ 913]

测量

- 测量 [▶ 1428]
- 几何和坐标转换 [▶ 1030]

代码读取

- 代码读取 [▶ 796]

光学字符识别 (OCR)

- 光学字符识别 [▶ 1493]

绘图和文本

- 绘图 [▶ 952]

机器学习

- 机器学习 [▶ 1335]

高级函数

- 傅立叶分析 [▶ 1015]
- 关键点特征 [▶ 1290]

其它

- 控制函数 [▶ 948]
- 其它 [▶ 1473]

专家函数

许多函数都有专家版本。它们包含额外的参数，通常以名称后缀“Exp”标记。如果是专家版函数，则在“专家函数”标题下会另有说明。

例如，F_VN_MaxImage [▶ 502] 的专家版本称为 F_VN_MaxImagesExp [▶ 503]，并包含附加参数 ipMask。

运行时行为

某些图像处理算法的执行时间取决于其参数设置和图像内容。因此，它并不确定。尽管如此，为了保证实时行为，可对这些函数或函数组应用 Watchdog（看门狗） [▶ 127]，其可在指定时间后终止函数。

> **⚠ 警告**
>
> **函数使用不当会导致 TwinCAT 崩溃**
>
> 如果某些函数使用不当，可能会导致错误（并导致 TwinCAT 崩溃）。这些错误不会在函数中被拦截，因为相应的检查会大大延长执行时间。因此，您必须注意正确使用这些函数！例如，我们建议对输入参数进行验证。

还请参阅有关此

🗎 Code Quality [▶ 786]

6.1.4.1 License Overview

以下列表显示了 TwinCAT Vision 功能与可用许可证的分配情况（见授权模式 [▶ 11]）。它显示了应用某些功能需要哪些许可证。

> **TF7100**
>
> 始终需要基本许可证 TC3 Vision Base（TF7100）。

TC3 Vision Base

F_VN_AdaptiveThreshold [▶ 1282]

F_VN_AdaptiveThresholdExp [▶ 1283]

F_VN_AddContainers [▶ 463]

F_VN_AddImages [▶ 478]

F_VN_AddImagesWeighted [▶ 479]

F_VN_AddImagesWeightedExp [▶ 480]

F_VN_AddScalarToImage [▶ 481]

F_VN_AddToContainerElements_DINT [▶ 436]

F_VN_AddToContainerElements_INT [▶ 437]

F_VN_AddToContainerElements_LREAL [▶ 438]

F_VN_AddToContainerElements_REAL [▶ 438]

F_VN_AddToContainerElements_SINT [▶ 439]

TC3 Vision Code Reading

TC3 Vision Code Quality

TC3 Vision Metrology 2D

TC3 Vision Matching

TC3 Vision Machine Learning

F_VN_TrainSampleClass [▶ 1423]

F_VN_TrainSampleCluster [▶ 1424]

F_VN_TrainSampleScalar [▶ 1426]

F_VN_TrainSampleVector [▶ 1427]

TC3 Vision OCR

F_VN_OCR [▶ 1495]

F_VN_OCRExp [▶ 1496]

6.1.4.2 Algebraic Container Operations

该组包含代数算子以元素方式应用于容器 [▶ 132]的函数。

函数

代数

- F_VN_AddContainers [▶ 463]
- F_VN_AddToContainerElements [▶ 436]
- F_VN_DivideContainers(Exp) [▶ 467]
- F_VN_MultiplyContainers [▶ 471]
- F_VN_MultiplyWithContainerElements [▶ 472]
- F_VN_NegateContainer [▶ 475]
- F_VN_SubtractContainers [▶ 476]

逐位算子

- F_VN_BitwiseAndContainers [▶ 464]
- F_VN_BitwiseNotContainer [▶ 465]
- F_VN_BitwiseOrContainers [▶ 465]
- F_VN_BitwiseXorContainers [▶ 466]

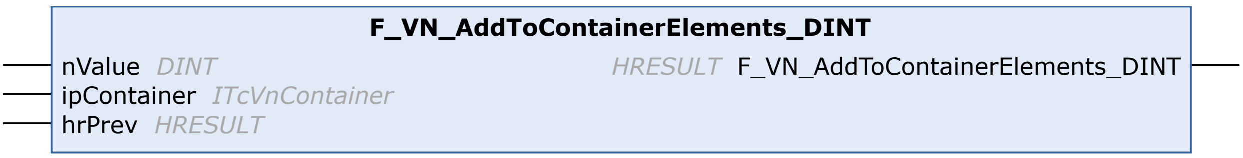
统计学特征

- F_VN_MaxContainer [▶ 469]
- F_VN_MinContainer [▶ 470]

6.1.4.2.1 F_VN_AddToContainerElements

本章包含根据数据类型向每个容器元素添加数值的函数。

6.1.4.2.1.1 F_VN_AddToContainerElements_DINT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_DINT : HRESULT
VAR_INPUT
    nValue      : DINT;
```

```
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nValue	DINT	Value
ipContainer	ITcVnContainer [▶ 349]	Container with DINT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

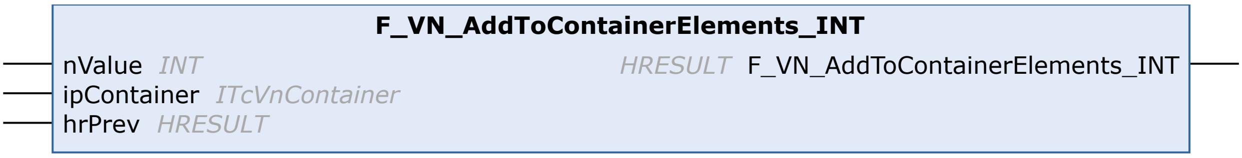
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.2 F_VN_AddToContainerElements_INT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_INT : HRESULT
VAR_INPUT
    nValue      : INT;
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nValue	INT	Value
ipContainer	ITcVnContainer [▶ 349]	Container with INT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

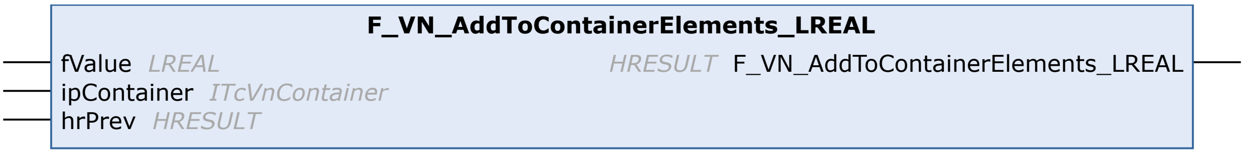
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.3 F_VN_AddToContainerElements_LREAL



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_LREAL : HRESULT
VAR_INPUT
    fValue      : LREAL;
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fValue	LREAL	Value
ipContainer	ITcVnContainer [▸ 349]	Container with LREAL elements
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

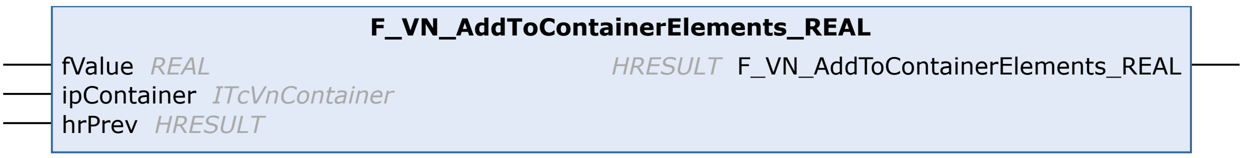
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.4 F_VN_AddToContainerElements_REAL



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_REAL : HRESULT
VAR_INPUT
    fValue       : REAL;
    ipContainer  : ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fValue	REAL	Value
ipContainer	ITcVnContainer [▶ 349]	Container with REAL elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

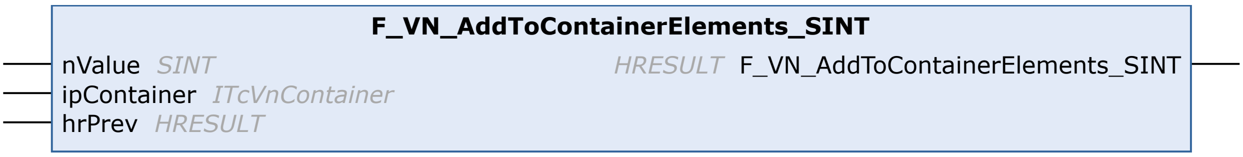
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.5 F_VN_AddToContainerElements_SINT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_SINT : HRESULT
VAR_INPUT
    nValue       : SINT;
    ipContainer  : ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nValue	SINT	Value
ipContainer	ITcVnContainer [▶ 349]	Container with SINT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

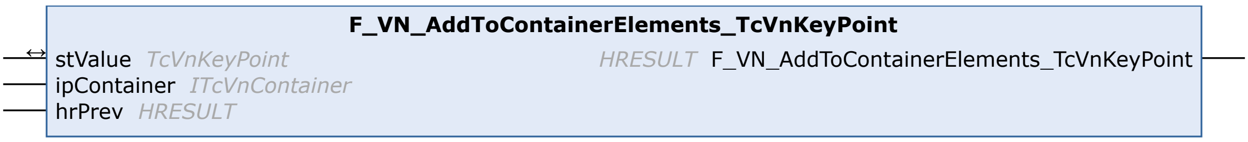
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.6 F_VN_AddToContainerElements_TcVnKeyPoint



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnKeyPoint : HRESULT
VAR_IN_OUT
    stValue     : TcVnKeyPoint;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnKeyPoint elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stValue	TcVnKeyPoint [▶ 210]	Value

Return value

HRESULT [▶ 122]

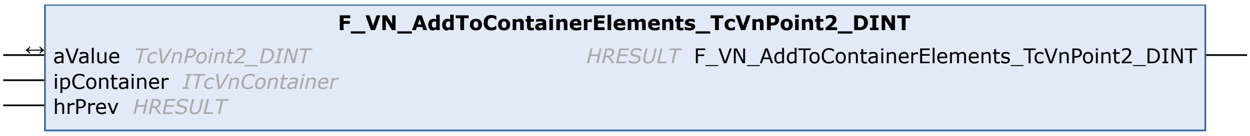
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.7 F_VN_AddToContainerElements_TcVnPoint2_DINT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnPoint2_DINT : HRESULT
VAR_IN_OUT
    aValue      : TcVnPoint2_DINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnPoint2_DINT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnPoint2_DINT [▶ 139]	Value

Return value

HRESULT [▶ 122]

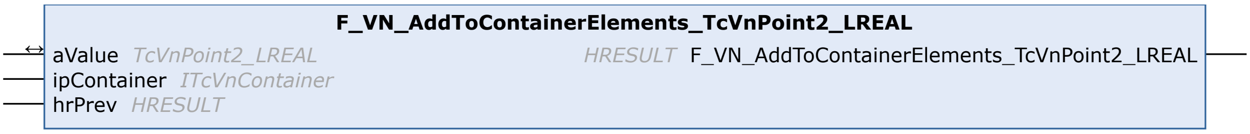
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.8 F_VN_AddToContainerElements_TcVnPoint2_LREAL



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnPoint2_LREAL : HRESULT
VAR_IN_OUT
    aValue      : TcVnPoint2_LREAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnPoint2_LREAL elements
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnPoint2_LREAL [▸ 139]	Value

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.9 F_VN_AddToContainerElements_TcVnPoint2_REAL

F_VN_AddToContainerElements_TcVnPoint2_REAL

aValue *TcVnPoint2_REAL* *HRESULT* F_VN_AddToContainerElements_TcVnPoint2_REAL
ipContainer *ITcVnContainer*
hrPrev *HRESULT*

Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnPoint2_REAL : HRESULT
VAR_IN_OUT
    aValue      : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnPoint2_REAL elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnPoint2_REAL [▶ 139]	Value

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.10 F_VN_AddToContainerElements_TcVnPoint3_LREAL

F_VN_AddToContainerElements_TcVnPoint3_LREAL

aValue *TcVnPoint3_LREAL* *HRESULT* F_VN_AddToContainerElements_TcVnPoint3_LREAL
ipContainer *ITcVnContainer*
hrPrev *HRESULT*

Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnPoint3_LREAL : HRESULT
VAR_IN_OUT
    aValue       : TcVnPoint3_LREAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnPoint3_LREAL elements
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnPoint3_LREAL [▸ 139]	Value

Return value

HRESULT [▸ 122]

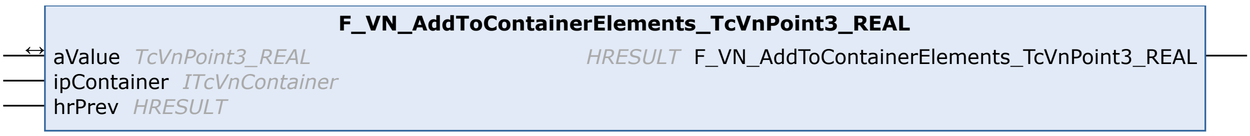
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.11 F_VN_AddToContainerElements_TcVnPoint3_REAL



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnPoint3_REAL : HRESULT
VAR_IN_OUT
    aValue       : TcVnPoint3_REAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnPoint3_REAL elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnPoint3_REAL [▶ 139]	Value

Return value

HRESULT [▶ 122]

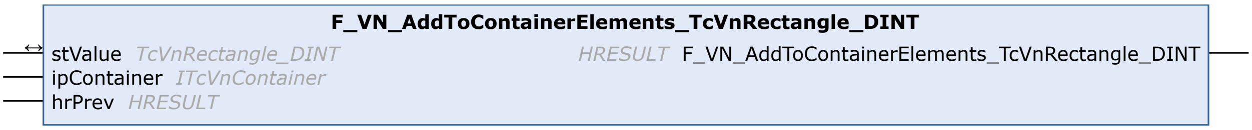
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.12 F_VN_AddToContainerElements_TcVnRectangle_DINT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnRectangle_DINT : HRESULT
VAR_IN_OUT
    stValue      : TcVnRectangle_DINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnRectangle_DINT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stValue	TcVnRectangle_DINT [▶ 224]	Value

Return value

HRESULT [▶ 122]

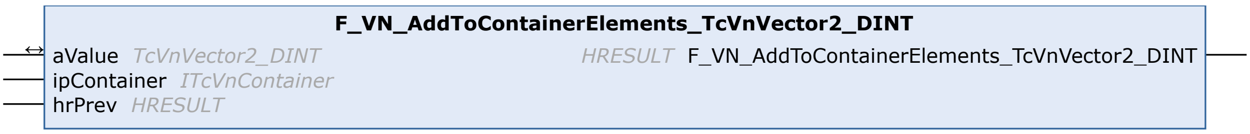
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.13 F_VN_AddToContainerElements_TcVnVector2_DINT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnVector2_DINT : HRESULT
VAR_IN_OUT
    aValue      : TcVnVector2_DINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector2_DINT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_DINT [▶ 141]	Value

Return value

HRESULT [▶ 122]

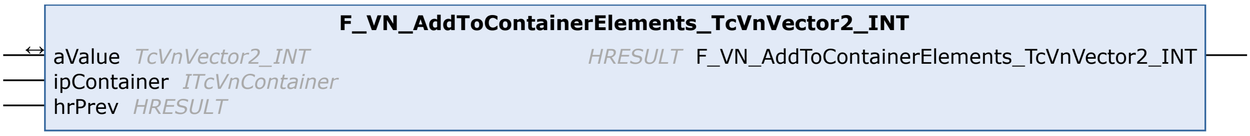
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.14 F_VN_AddToContainerElements_TcVnVector2_INT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnVector2_INT : HRESULT
VAR_IN_OUT
    aValue      : TcVnVector2_INT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector2_INT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_INT [▶ 141]	Value

Return value

HRESULT [▶ 122]

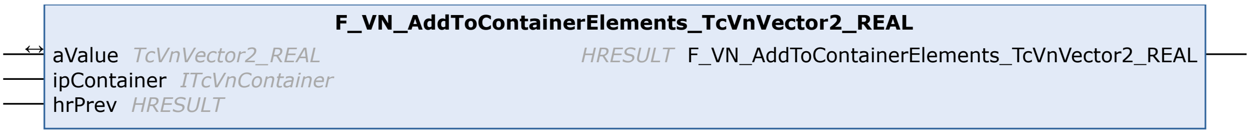
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.15 F_VN_AddToContainerElements_TcVnVector2_REAL



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnVector2_REAL : HRESULT
VAR_IN_OUT
    aValue      : TcVnVector2_REAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector2_REAL elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_REAL [▶ 141]	Value

Return value

HRESULT [▶ 122]

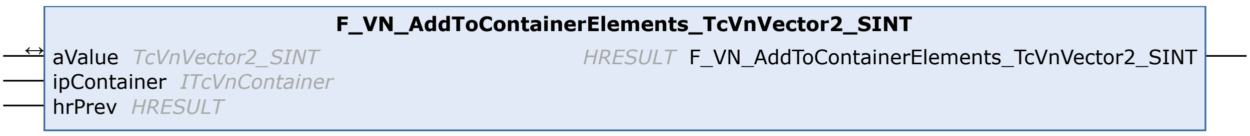
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.16 F_VN_AddToContainerElements_TcVnVector2_SINT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnVector2_SINT : HRESULT
VAR_IN_OUT
    aValue      : TcVnVector2_SINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnVector2_SINT elements
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_SINT [▸ 141]	Value

Return value

HRESULT [▸ 122]

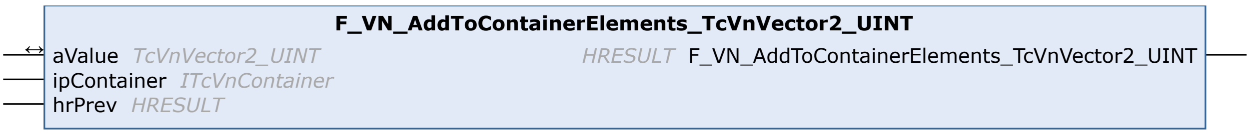
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.17 F_VN_AddToContainerElements_TcVnVector2_UINT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnVector2_UINT : HRESULT
VAR_IN_OUT
    aValue      : TcVnVector2_UINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector2_UINT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_UINT [▶ 141]	Value

Return value

HRESULT [▶ 122]

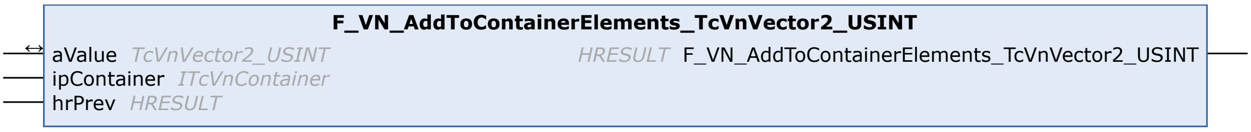
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.18 F_VN_AddToContainerElements_TcVnVector2_USINT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnVector2_USINT : HRESULT
VAR_IN_OUT
    aValue      : TcVnVector2_USINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector2_USINT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_USINT [▶ 141]	Value

Return value

HRESULT [▶ 122]

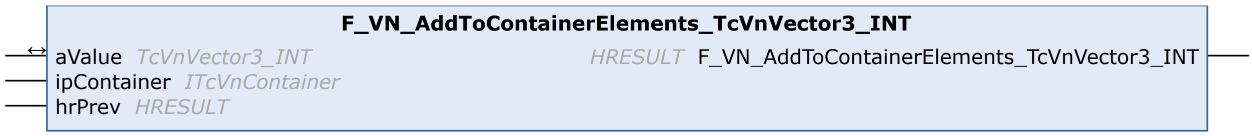
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.19 F_VN_AddToContainerElements_TcVnVector3_INT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnVector3_INT : HRESULT
VAR_IN_OUT
    aValue      : TcVnVector3_INT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector3_INT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector3_INT [▶ 141]	Value

Return value

HRESULT [▶ 122]

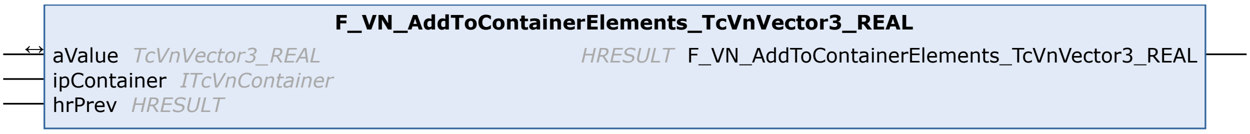
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.20 F_VN_AddToContainerElements_TcVnVector3_REAL



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnVector3_REAL : HRESULT
VAR_IN_OUT
    aValue      : TcVnVector3_REAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector3_REAL elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector3_REAL [▶ 141]	Value

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.21 F_VN_AddToContainerElements_TcVnVector3_SINT

F_VN_AddToContainerElements_TcVnVector3_SINT

aValue *TcVnVector3_SINT* — *HRESULT* F_VN_AddToContainerElements_TcVnVector3_SINT
ipContainer *ITcVnContainer*
hrPrev *HRESULT*

Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnVector3_SINT : HRESULT
VAR_IN_OUT
    aValue      : TcVnVector3_SINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector3_SINT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector3_SINT [▶ 141]	Value

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.22 F_VN_AddToContainerElements_TcVnVector3_UINT

F_VN_AddToContainerElements_TcVnVector3_UINT

aValue *TcVnVector3_UINT* — *HRESULT* F_VN_AddToContainerElements_TcVnVector3_UINT
ipContainer *ITcVnContainer*
hrPrev *HRESULT*

Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnVector3_UINT : HRESULT
VAR_IN_OUT
    aValue      : TcVnVector3_UINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector3_UINT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector3_UINT [▶ 141]	Value

Return value

HRESULT [▶ 122]

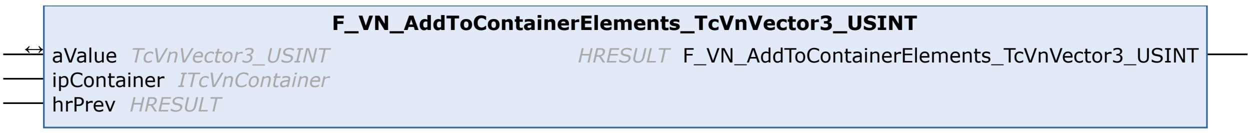
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.23 F_VN_AddToContainerElements_TcVnVector3_USINT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnVector3_USINT : HRESULT
VAR_IN_OUT
    aValue      : TcVnVector3_USINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector3_USINT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector3_USINT [▶ 141]	Value

Return value

HRESULT [▶ 122]

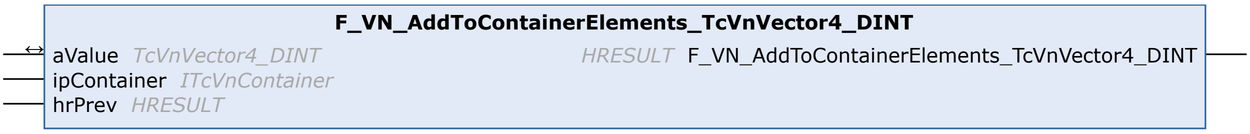
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.24 F_VN_AddToContainerElements_TcVnVector4_DINT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnVector4_DINT : HRESULT
VAR_IN_OUT
    aValue       : TcVnVector4_DINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector4_DINT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_DINT [▶ 141]	Value

Return value

HRESULT [▶ 122]

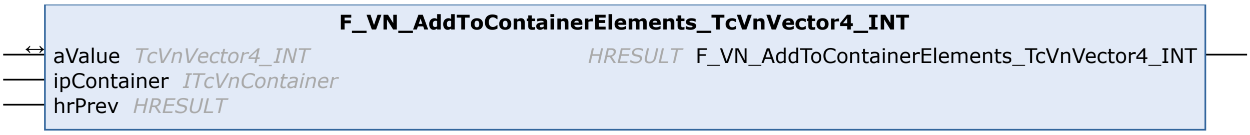
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.25 F_VN_AddToContainerElements_TcVnVector4_INT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnVector4_INT : HRESULT
VAR_IN_OUT
    aValue      : TcVnVector4_INT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector4_INT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_INT [▶ 141]	Value

Return value

HRESULT [▶ 122]

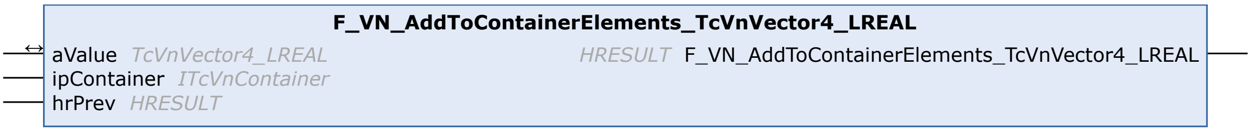
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.26 F_VN_AddToContainerElements_TcVnVector4_LREAL



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnVector4_LREAL : HRESULT
VAR_IN_OUT
    aValue      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector4_LREAL elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_LREAL [▶ 141]	Value

Return value

HRESULT [▶ 122]

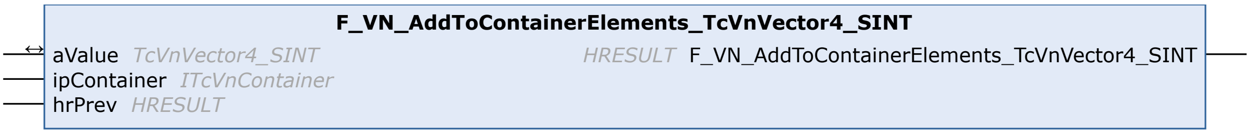
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.27 F_VN_AddToContainerElements_TcVnVector4_SINT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnVector4_SINT : HRESULT
VAR_IN_OUT
    aValue      : TcVnVector4_SINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnVector4_SINT elements
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_SINT [▸ 141]	Value

Return value

HRESULT [▸ 122]

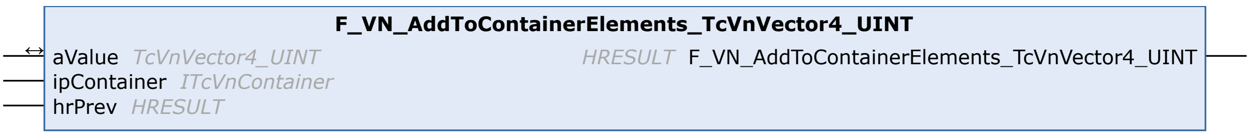
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.28 F_VN_AddToContainerElements_TcVnVector4_UINT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnVector4_UINT : HRESULT
VAR_IN_OUT
    aValue      : TcVnVector4_UINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector4_UINT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_UINT [▶ 141]	Value

Return value

HRESULT [▶ 122]

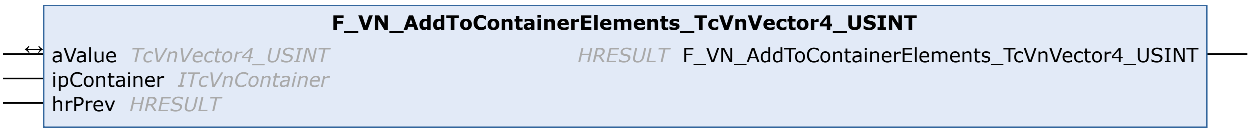
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.29 F_VN_AddToContainerElements_TcVnVector4_USINT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_TcVnVector4_USINT : HRESULT
VAR_IN_OUT
    aValue      : TcVnVector4_USINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector4_USINT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_USINT [▶ 141]	Value

Return value

HRESULT [▶ 122]

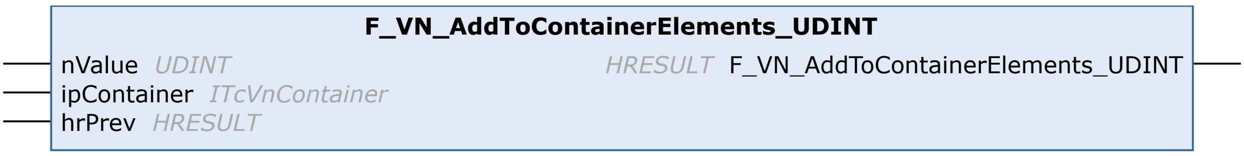
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.30 F_VN_AddToContainerElements_UDINT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_UDINT : HRESULT
VAR_INPUT
    nValue      : UDINT;
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nValue	UDINT	Value
ipContainer	ITcVnContainer [▶ 349]	Container with UDINT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

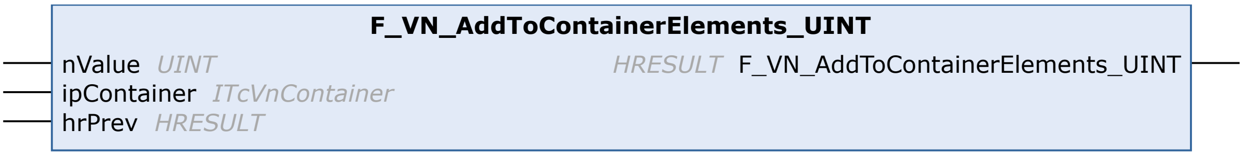
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.31 F_VN_AddToContainerElements_UINT



Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_UINT : HRESULT
VAR_INPUT
    nValue      : UINT;
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nValue	UINT	Value
ipContainer	ITcVnContainer [▶ 349]	Container with UINT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.32 F_VN_AddToContainerElements_ULINT

F_VN_AddToContainerElements_ULINT
nValue *ULINT* — *HRESULT* F_VN_AddToContainerElements_ULINT
ipContainer *ITcVnContainer*
hrPrev *HRESULT*

Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_ULINT : HRESULT
VAR_INPUT
    nValue      : ULINT;
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nValue	ULINT	Value
ipContainer	ITcVnContainer [▶ 349]	Container with ULINT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.1.33 F_VN_AddToContainerElements_USINT

F_VN_AddToContainerElements_USINT
nValue *USINT* — *HRESULT* F_VN_AddToContainerElements_USINT
ipContainer *ITcVnContainer*
hrPrev *HRESULT*

Add a value to each container element.

Syntax

Definition:

```
FUNCTION F_VN_AddToContainerElements_USINT : HRESULT
VAR_INPUT
    nValue      : USINT;
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nValue	USINT	Value
ipContainer	ITcVnContainer [▶ 349]	Container with USINT elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

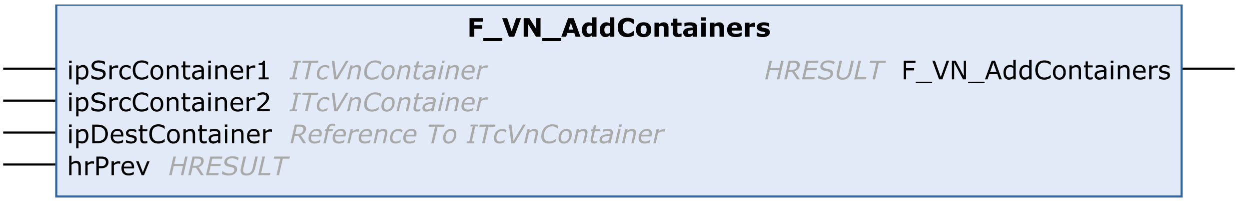
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.2 F_VN_AddContainers



Element-wise addition of two containers (same length and type).

Syntax

Definition:

```
FUNCTION F_VN_AddContainers : HRESULT
VAR_INPUT
    ipSrcContainer1 : ITcVnContainer;
    ipSrcContainer2 : ITcVnContainer;
    ipDestContainer : Reference To ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer1	ITcVnContainer [▶ 349]	Source container 1
ipSrcContainer2	ITcVnContainer [▶ 349]	Source container 2
ipDestContainer	Reference To ITcVnContainer [▶ 349]	Returns the resulting container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.3 F_VN_BitwiseAndContainers

Element-wise application of a bit-wise AND operator to two containers (same length and type, integer only).

Syntax

Definition:

```
FUNCTION F_VN_BitwiseAndContainers : HRESULT
VAR_INPUT
    ipSrcContainer1 : ITcVnContainer;
    ipSrcContainer2 : ITcVnContainer;
    ipDestContainer : Reference To ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer1	ITcVnContainer [▶ 349]	Source container 1
ipSrcContainer2	ITcVnContainer [▶ 349]	Source container 2
ipDestContainer	Reference To ITcVnContainer [▶ 349]	Returns the resulting container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

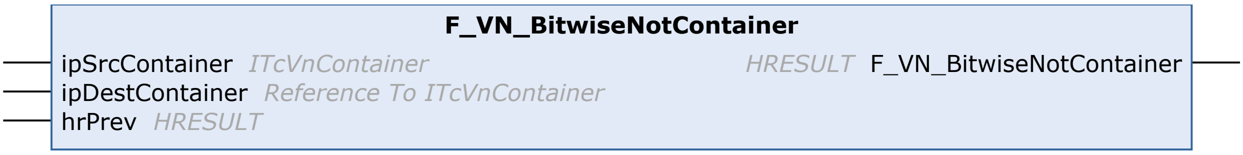
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.4 F_VN_BitwiseNotContainer



Element-wise application of a bit-wise NOT operator to a container (integer only).

Syntax

Definition:

```
FUNCTION F_VN_BitwiseNotContainer : HRESULT
VAR_INPUT
    ipSrcContainer  : ITcVnContainer;
    ipDestContainer : Reference To ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer	ITcVnContainer [▸ 349]	Source container
ipDestContainer	Reference To ITcVnContainer [▸ 349]	Returns the resulting container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.5 F_VN_BitwiseOrContainers

Element-wise application of a bit-wise OR operator to two containers (same length and type, integer only).

Syntax

Definition:

```
FUNCTION F_VN_BitwiseOrContainers : HRESULT
VAR_INPUT
    ipSrcContainer1 : ITcVnContainer;
    ipSrcContainer2 : ITcVnContainer;
    ipDestContainer : Reference To ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer1	ITcVnContainer [▶ 349]	Source container 1
ipSrcContainer2	ITcVnContainer [▶ 349]	Source container 2
ipDestContainer	Reference To ITcVnContainer [▶ 349]	Returns the resulting container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.6 F_VN_BitwiseXorContainers

Element-wise application of a bit-wise XOR operator to two containers (same length and type, integer only).

Syntax

Definition:

```
FUNCTION F_VN_BitwiseXorContainers : HRESULT
VAR_INPUT
    ipSrcContainer1 : ITcVnContainer;
    ipSrcContainer2 : ITcVnContainer;
```

```
    ipDestContainer : Reference To ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer1	ITcVnContainer [▶ 349]	Source container 1
ipSrcContainer2	ITcVnContainer [▶ 349]	Source container 2
ipDestContainer	Reference To ITcVnContainer [▶ 349]	Returns the resulting container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

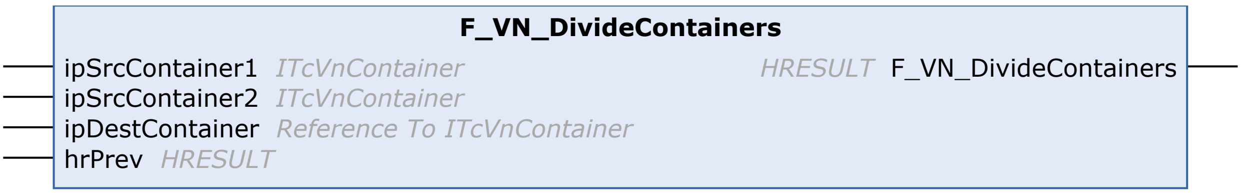
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.7 F_VN_DivideContainers



Element-wise division of two containers (same length and type). In case of integer division by zero, the resulting element is set to 0 and S_DIVISION_BY_ZERO is returned instead of S_OK (value can be changed in F_VN_DivideContainersExp).

Syntax

Definition:

```
FUNCTION F_VN_DivideContainers : HRESULT
VAR_INPUT
    ipSrcContainer1 : ITcVnContainer;
    ipSrcContainer2 : ITcVnContainer;
    ipDestContainer : Reference To ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer1	ITcVnContainer [▸ 349]	Source container 1
ipSrcContainer2	ITcVnContainer [▸ 349]	Source container 2
ipDestContainer	Reference To ITcVnContainer [▸ 349]	Returns the resulting container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

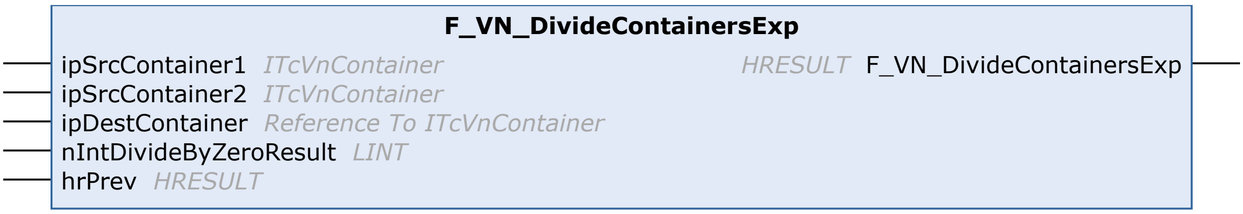
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.8 F_VN_DivideContainersExp



Element-wise division of two containers (same length and type). In case of integer division by zero, the resulting element is set to nIntDivideByZeroResult and S_DIVISION_BY_ZERO is returned instead of S_OK. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_DivideContainersExp : HRESULT
VAR_INPUT
    ipSrcContainer1        : ITcVnContainer;
    ipSrcContainer2        : ITcVnContainer;
    ipDestContainer        : Reference To ITcVnContainer;
    nIntDivideByZeroResult : LINT;
    hrPrev                 : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer1	ITcVnContainer [▶ 349]	Source container 1
ipSrcContainer2	ITcVnContainer [▶ 349]	Source container 2
ipDestContainer	Reference To ITcVnContainer [▶ 349]	Returns the resulting container
nIntDivideByZeroResult	LINT	Value set as result in case of integer division by zero
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.9 F_VN_MaxContainer

Element-wise maximum of two containers (same length and type).

Syntax

Definition:

```
FUNCTION F_VN_MaxContainer : HRESULT
VAR_INPUT
    ipSrcContainer1 : ITcVnContainer;
    ipSrcContainer2 : ITcVnContainer;
    ipDestContainer : Reference To ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer1	ITcVnContainer [▶ 349]	Source container 1
ipSrcContainer2	ITcVnContainer [▶ 349]	Source container 2
ipDestContainer	Reference To ITcVnContainer [▶ 349]	Returns the resulting container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.10 F_VN_MinContainer

Element-wise minimum of two containers (same length and type).

Syntax

Definition:

```
FUNCTION F_VN_MinContainer : HRESULT
VAR_INPUT
    ipSrcContainer1 : ITcVnContainer;
    ipSrcContainer2 : ITcVnContainer;
    ipDestContainer : Reference To ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer1	ITcVnContainer [▸ 349]	Source container 1
ipSrcContainer2	ITcVnContainer [▸ 349]	Source container 2
ipDestContainer	Reference To ITcVnContainer [▸ 349]	Returns the resulting container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.11 F_VN_MultiplyContainers

Element-wise multiplication of two containers (same length and type).

Syntax

Definition:

```
FUNCTION F_VN_MultiplyContainers : HRESULT
VAR_INPUT
    ipSrcContainer1 : ITcVnContainer;
    ipSrcContainer2 : ITcVnContainer;
    ipDestContainer : Reference To ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer1	ITcVnContainer [▶ 349]	Source container 1
ipSrcContainer2	ITcVnContainer [▶ 349]	Source container 2
ipDestContainer	Reference To ITcVnContainer [▶ 349]	Returns the resulting container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

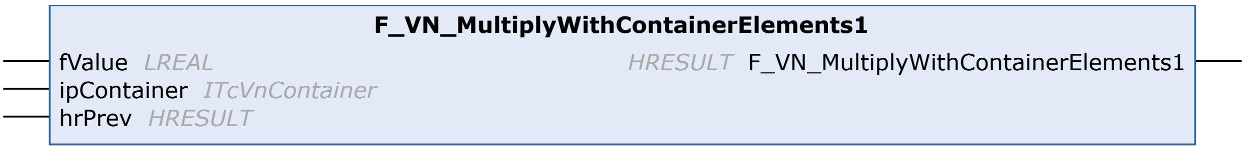
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.12 F_VN_MultiplyWithContainerElements1



Multiply each container element with a value.

Syntax

Definition:

```
FUNCTION F_VN_MultiplyWithContainerElements1 : HRESULT
VAR_INPUT
    fValue      : LREAL;
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fValue	LREAL	Value
ipContainer	ITcVnContainer [▶ 349]	Container with 1-dimensional elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.13 F_VN_MultiplyWithContainerElements2

F_VN_MultiplyWithContainerElements2	
↔ aValue *TcVnVector2_LREAL*	*HRESULT* F_VN_MultiplyWithContainerElements2
ipContainer *ITcVnContainer*	
hrPrev *HRESULT*	

Multiply each container element with a value.

Syntax

Definition:

```
FUNCTION F_VN_MultiplyWithContainerElements2 : HRESULT
VAR_IN_OUT
    aValue       : TcVnVector2_LREAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with 2-dimensional elements
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_LREAL [▸ 141]	Value

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.14 F_VN_MultiplyWithContainerElements3

F_VN_MultiplyWithContainerElements3
aValue *TcVnVector3_LREAL* — *HRESULT* F_VN_MultiplyWithContainerElements3
ipContainer *ITcVnContainer*
hrPrev *HRESULT*

Multiply each container element with a value.

Syntax

Definition:

```
FUNCTION F_VN_MultiplyWithContainerElements3 : HRESULT
VAR_IN_OUT
    aValue      : TcVnVector3_LREAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with 3-dimensional elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector3_LREAL [▶ 141]	Value

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.15 F_VN_MultiplyWithContainerElements4

F_VN_MultiplyWithContainerElements4
aValue *TcVnVector4_LREAL* — *HRESULT* F_VN_MultiplyWithContainerElements4
ipContainer *ITcVnContainer*
hrPrev *HRESULT*

Multiply each container element with a value.

Syntax

Definition:

```
FUNCTION F_VN_MultiplyWithContainerElements4 : HRESULT
VAR_IN_OUT
    aValue       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with 4-dimensional elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_LREAL [▶ 141]	Value

Return value

HRESULT [▶ 122]

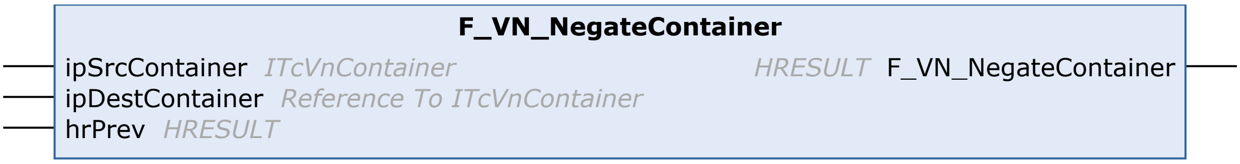
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.16 F_VN_NegateContainer



Element-wise negation of a container (two's complement).

Syntax

Definition:

```
FUNCTION F_VN_NegateContainer : HRESULT
VAR_INPUT
    ipSrcContainer   : ITcVnContainer;
    ipDestContainer : Reference To ITcVnContainer;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer	ITcVnContainer [▶ 349]	Source container
ipDestContainer	Reference To ITcVnContainer [▶ 349]	Returns the resulting container (same type as ipSrcContainer)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

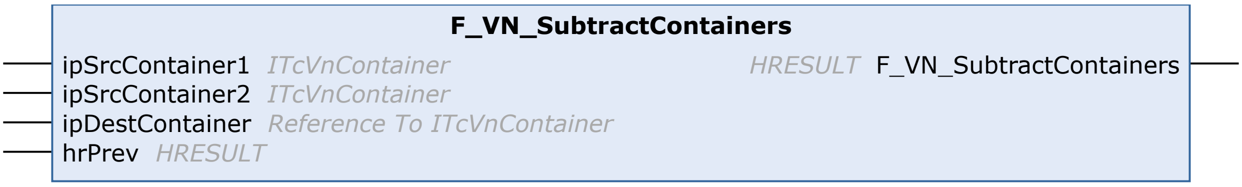
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.2.17 F_VN_SubtractContainers



Element-wise subtraction of two containers (same length and type).

Syntax

Definition:

```
FUNCTION F_VN_SubtractContainers : HRESULT
VAR_INPUT
    ipSrcContainer1 : ITcVnContainer;
    ipSrcContainer2 : ITcVnContainer;
    ipDestContainer : Reference To ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer1	ITcVnContainer [▶ 349]	Source container 1
ipSrcContainer2	ITcVnContainer [▶ 349]	Source container 2
ipDestContainer	Reference To ITcVnContainer [▶ 349]	Returns the resulting container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3 Algebraic Image Operations

这一组包含的函数用于将代数运算应用于图像 [▶ 130]每个像素。

函数

加法

- F_VN_AddImages [▶ 478]
- F_VN_AddImagesWeighted(Exp) [▶ 479]
- F_VN_AddScalarToImage [▶ 481]
- F_VN_AddVectorToImage [▶ 482]

减法

- F_VN_SubtractImages [▶ 517]
- F_VN_SubtractImageFromScalar [▶ 516]
- F_VN_SubtractImageFromVector [▶ 516]
- F_VN_SubtractScalarFromImage [▶ 518]
- F_VN_SubtractVectorFromImage [▶ 519]

乘法

- F_VN_MultiplyImages [▶ 513]
- F_VN_MultiplyImageWithScalar [▶ 514]
- F_VN_MultiplyImageWithVector [▶ 515]

除法

- F_VN_DivideImages [▶ 498]
- F_VN_DivideImageByScalar [▶ 496]
- F_VN_DivideImageByVector [▶ 497]
- F_VN_DivideScalarByImage [▶ 499]
- F_VN_DivideVectorByImage [▶ 500]

乘方和对数化

- F_VN_ElementwiseExp [▶ 501]
- F_VN_ElementwiseLog [▶ 502]

位数 AND

- F_VN_BitwiseAndImages(Exp) [▶ 483]
- F_VN_BitwiseAndScalarWithImage [▶ 485]
- F_VN_BitwiseAndVectorWithImage [▶ 485]

位数 NOT

- F_VN_BitwiseNotImage(Exp) [▶ 486]

位数 OR

- F_VN_BitwiseOrImages(Exp) [▶ 488]
- F_VN_BitwiseOrScalarWithImage [▶ 490]
- F_VN_BitwiseOrVectorWithImage [▶ 490]

位数 XOR

- F_VN_BitwiseXorImages(Exp) [▶ 491]
- F_VN_BitwiseXorScalarWithImage [▶ 493]
- F_VN_BitwiseXorVectorWithImage [▶ 494]

交叉淡化

- F_VN_BlendImages [▶ 495]

最大

- F_VN_MaxImage(Exp) [▶ 502]
- F_VN_MaxImageWithScalar(Exp) [▶ 504]
- F_VN_MaxImageWithVector(Exp) [▶ 506]

最小

- F_VN_MinImage(Exp) [▶ 508]
- F_VN_MinImageWithScalar(Exp) [▶ 509]
- F_VN_MinImageWithVector(Exp) [▶ 511]

6.1.4.3.1 F_VN_AddImages

Element-wise addition of two images using saturation arithmetics.

Syntax

Definition:

```
FUNCTION F_VN_AddImages : HRESULT
VAR_INPUT
    ipSrcImage1 : ITcVnImage;
    ipSrcImage2 : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▶ 390]	First source image
ipSrcImage2	ITcVnImage [▶ 390]	Second source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

样本

- 复制图像区域 [▶ 2643]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.2 F_VN_AddImagesWeighted

Weighted, element-wise addition of two images using saturation arithmetics.

Syntax

Definition:

```
FUNCTION F_VN_AddImagesWeighted : HRESULT
VAR_INPUT
    ipSrcImage1 : ITcVnImage;
    ipSrcImage2 : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    fWeight1    : LREAL;
    fWeight2    : LREAL;
    fDelta      : LREAL;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▶ 390]	First source image
ipSrcImage2	ITcVnImage [▶ 390]	Second source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
fWeight1	LREAL	Weight factor applied to ipSrcImage1
fWeight2	LREAL	Weight factor applied to ipSrcImage2
fDelta	LREAL	Value added to the weighted sum of both images
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

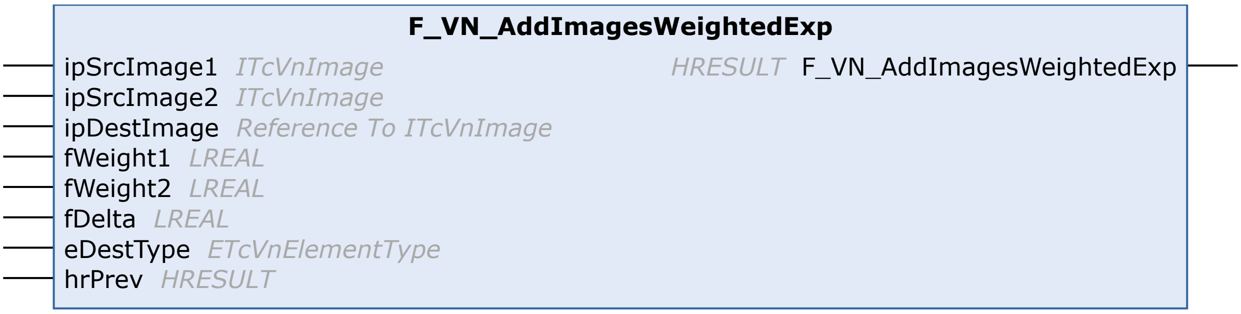
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.3 F_VN_AddImagesWeightedExp



Weighted, element-wise addition of two images using saturation arithmetics.

Syntax

Definition:

```
FUNCTION F_VN_AddImagesWeightedExp : HRESULT
VAR_INPUT
    ipSrcImage1 : ITcVnImage;
    ipSrcImage2 : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    fWeight1    : LREAL;
    fWeight2    : LREAL;
    fDelta      : LREAL;
    eDestType   : ETcVnElementType;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▶ 390]	First source image
ipSrcImage2	ITcVnImage [▶ 390]	Second source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
fWeight1	LREAL	Weight factor applied to ipSrcImage1
fWeight2	LREAL	Weight factor applied to ipSrcImage2
fDelta	LREAL	Value added to the weighted sum of both images
eDestType	ETcVnElementType [▶ 178]	Destination image depth
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

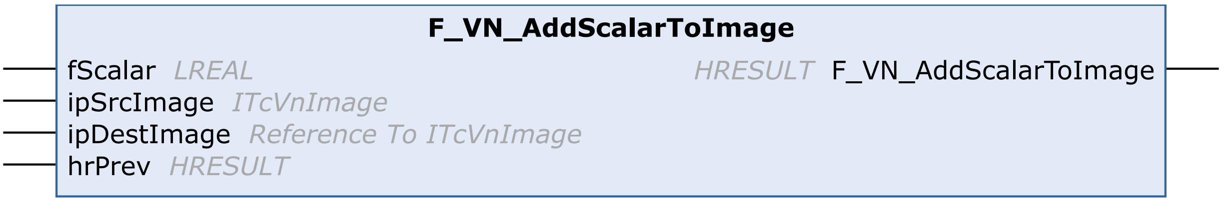
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.4 F_VN_AddScalarToImage



Add a scalar value to each image pixel using saturation arithmetics.

Syntax

Definition:

```
FUNCTION F_VN_AddScalarToImage : HRESULT
VAR_INPUT
    fScalar     : LREAL;
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fScalar	LREAL	Scalar value
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

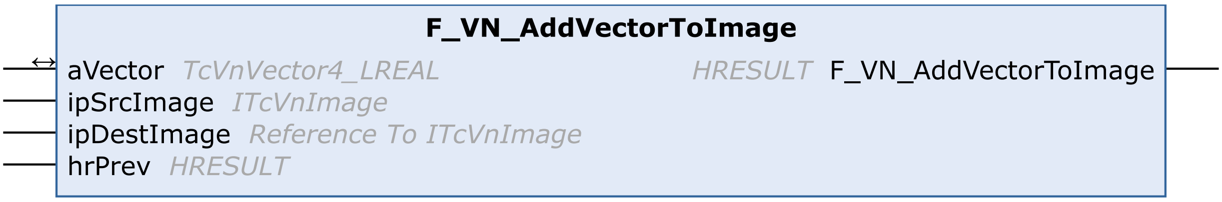
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.5 F_VN_AddVectorToImage



Add a vector (1 element for each image channel) to each image pixel using saturation arithmetics.

Syntax

Definition:

```
FUNCTION F_VN_AddVectorToImage : HRESULT
VAR_IN_OUT
    aVector     : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aVector	TcVnVector4_LREAL [▶ 141]	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)

Return value

HRESULT [▶ 122]

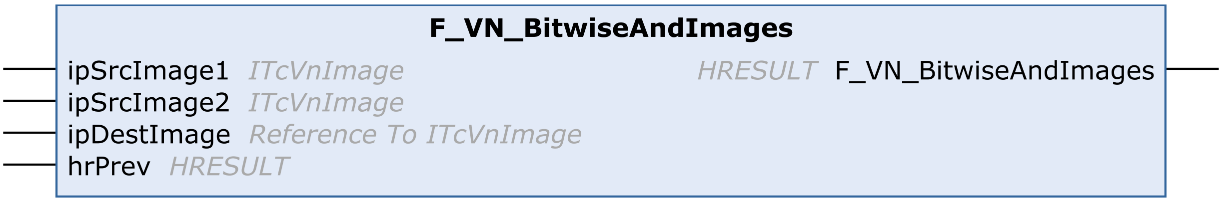
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.6 F_VN_BitwiseAndImages



Element-wise application of a bit-wise AND operator to two images.

Syntax

Definition:

```
FUNCTION F_VN_BitwiseAndImages : HRESULT
VAR_INPUT
    ipSrcImage1 : ITcVnImage;
    ipSrcImage2 : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▶ 390]	First source image
ipSrcImage2	ITcVnImage [▶ 390]	Second source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

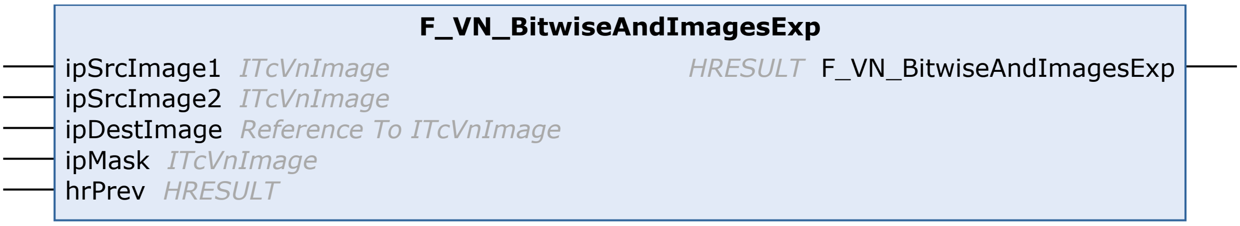
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.7 F_VN_BitwiseAndImagesExp



Element-wise application of a bit-wise AND operator to two images. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_BitwiseAndImagesExp : HRESULT
VAR_INPUT
    ipSrcImage1 : ITcVnImage;
    ipSrcImage2 : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    ipMask      : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▸ 390]	First source image
ipSrcImage2	ITcVnImage [▸ 390]	Second source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage [▸ 390]	Mask to limit the operation to specific pixel positions (mask value 0: skip pixel, mask value > 0: apply operation to pixel)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

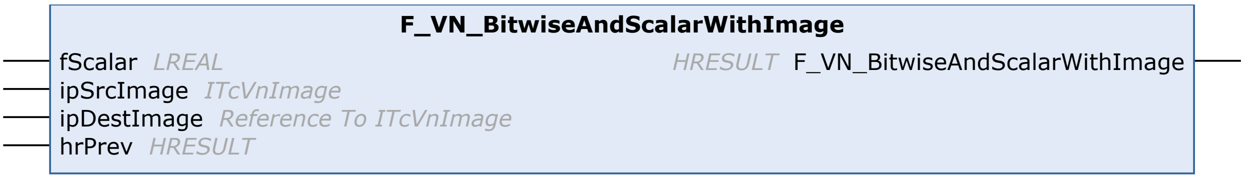
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.8 F_VN_BitwiseAndScalarWithImage



Bitwise and a scalar value with each image pixel.

Syntax

Definition:

```
FUNCTION F_VN_BitwiseAndScalarWithImage : HRESULT
VAR_INPUT
    fScalar      : LREAL;
    ipSrcImage   : ITcVnImage;
    ipDestImage  : Reference To ITcVnImage;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fScalar	LREAL	Scalar value
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

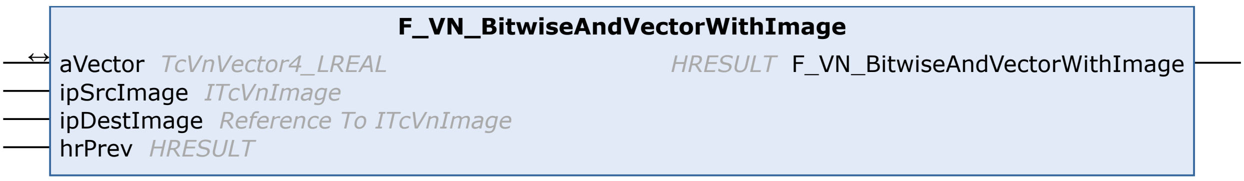
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.9 F_VN_BitwiseAndVectorWithImage



Bitwise and a vector (1 element for each image channel) with each image pixel.

Syntax

Definition:

```
FUNCTION F_VN_BitwiseAndVectorWithImage : HRESULT
VAR_IN_OUT
    aVector      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aVector	TcVnVector4_LREAL [▸ 141]	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)

Return value

HRESULT [▸ 122]

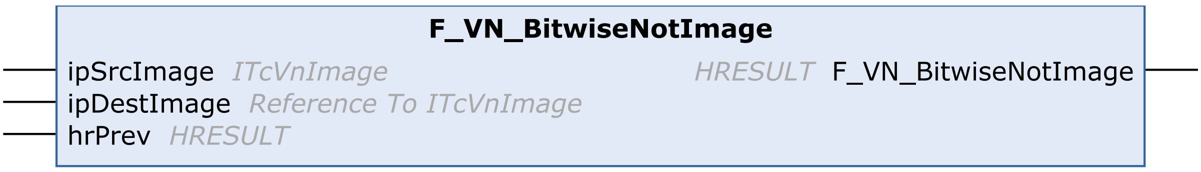
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.10 F_VN_BitwiseNotImage



Element-wise application of a bit-wise NOT operator to an image.

Syntax

Definition:

```
FUNCTION F_VN_BitwiseNotImage : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

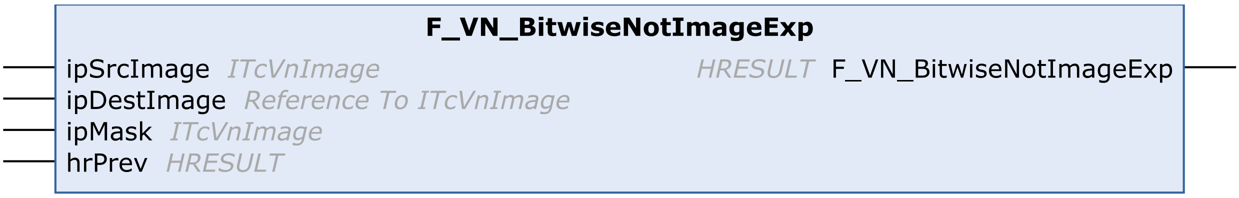
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.11 F_VN_BitwiseNotImageExp



Element-wise application of a bit-wise NOT operator to an image. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_BitwiseNotImageExp : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    ipMask      : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage [▶ 390]	Mask to limit the operation to specific pixel positions (mask value 0: skip pixel, mask value > 0: apply operation to pixel)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

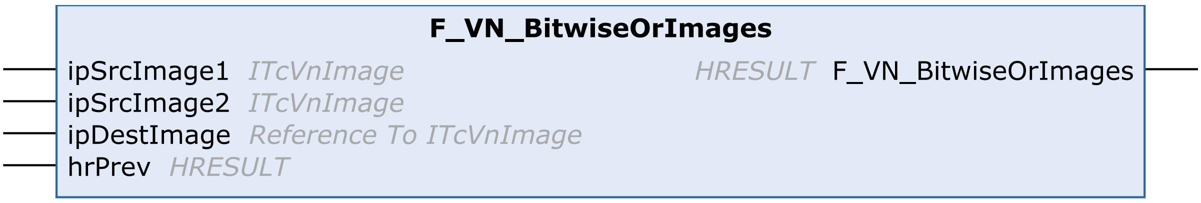
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.12 F_VN_BitwiseOrImages



Element-wise application of a bit-wise OR operator to two images.

Syntax

Definition:

```
FUNCTION F_VN_BitwiseOrImages : HRESULT
VAR_INPUT
    ipSrcImage1 : ITcVnImage;
    ipSrcImage2 : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▶ 390]	First source image
ipSrcImage2	ITcVnImage [▶ 390]	Second source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

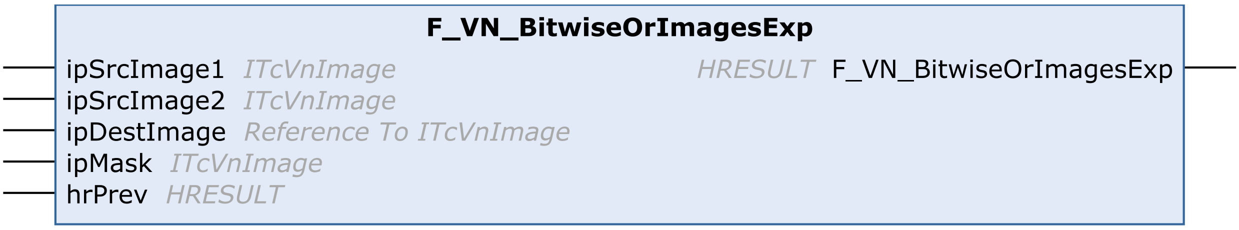
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.13 F_VN_BitwiseOrImagesExp



Element-wise application of a bit-wise OR operator to two images. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_BitwiseOrImagesExp : HRESULT
VAR_INPUT
    ipSrcImage1 : ITcVnImage;
    ipSrcImage2 : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    ipMask      : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▶ 390]	First source image
ipSrcImage2	ITcVnImage [▶ 390]	Second source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage [▶ 390]	Mask to limit the operation to specific pixel positions (mask value 0: skip pixel, mask value > 0: apply operation to pixel)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

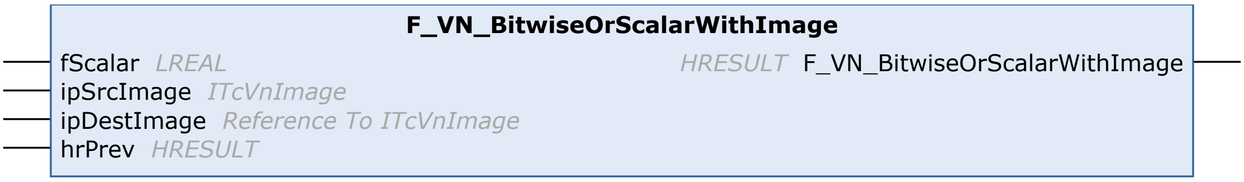
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.14 F_VN_BitwiseOrScalarWithImage



Bitwise or a scalar value with each image pixel.

Syntax

Definition:

```
FUNCTION F_VN_BitwiseOrScalarWithImage : HRESULT
VAR_INPUT
    fScalar     : LREAL;
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fScalar	LREAL	Scalar value
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

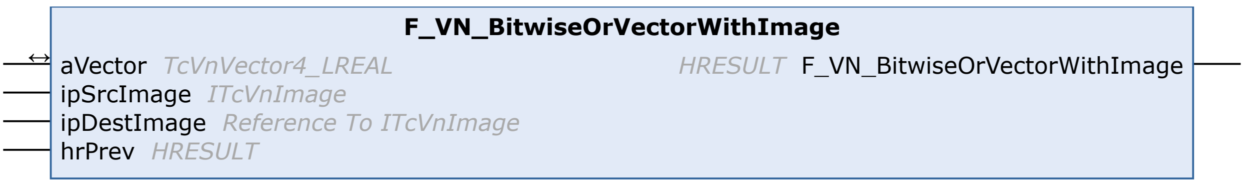
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.15 F_VN_BitwiseOrVectorWithImage



Bitwise or a vector (1 element for each image channel) with each image pixel.

Syntax

Definition:

```
FUNCTION F_VN_BitwiseOrVectorWithImage : HRESULT
VAR_IN_OUT
    aVector      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aVector	TcVnVector4_LREAL [▶ 141]	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)

Return value

HRESULT [▶ 122]

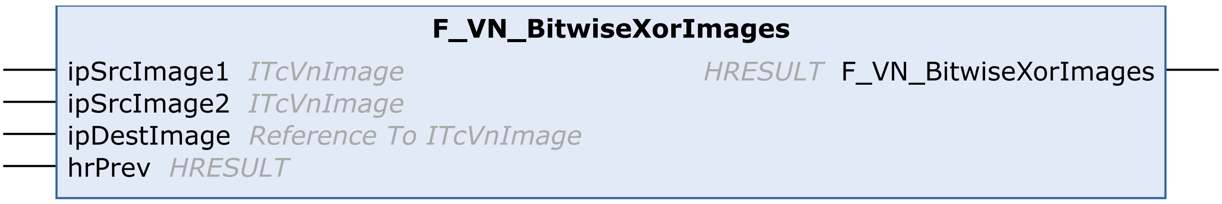
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.16 F_VN_BitwiseXorImages



Element-wise application of a bit-wise XOR operator to two images.

Syntax

Definition:

```
FUNCTION F_VN_BitwiseXorImages : HRESULT
VAR_INPUT
    ipSrcImage1 : ITcVnImage;
    ipSrcImage2 : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▶ 390]	First source image
ipSrcImage2	ITcVnImage [▶ 390]	Second source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

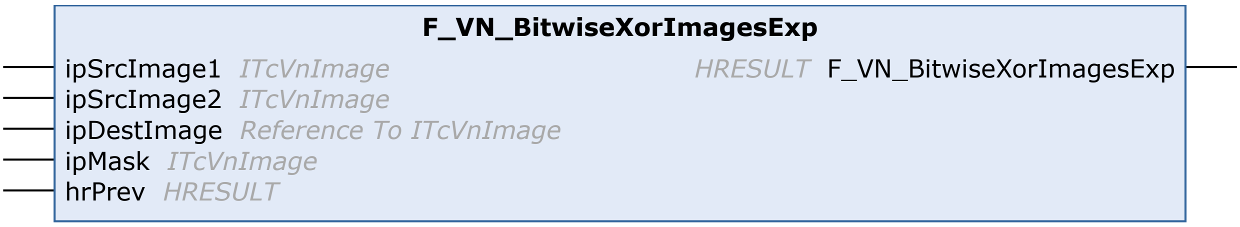
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.17 F_VN_BitwiseXorImagesExp



Element-wise application of a bit-wise XOR operator to two images. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_BitwiseXorImagesExp : HRESULT
VAR_INPUT
    ipSrcImage1 : ITcVnImage;
    ipSrcImage2 : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    ipMask      : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▶ 390]	First source image
ipSrcImage2	ITcVnImage [▶ 390]	Second source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage [▶ 390]	Mask to limit the operation to specific pixel positions (mask value 0: skip pixel, mask value > 0: apply operation to pixel)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

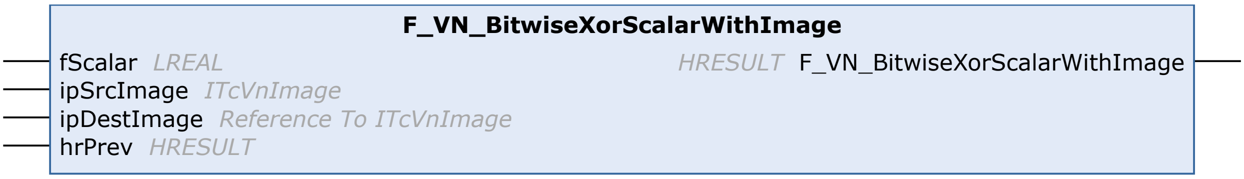
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.18 F_VN_BitwiseXorScalarWithImage



Bitwise xor a scalar value with each image pixel.

Syntax

Definition:

```
FUNCTION F_VN_BitwiseXorScalarWithImage : HRESULT
VAR_INPUT
    fScalar     : LREAL;
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fScalar	LREAL	Scalar value
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

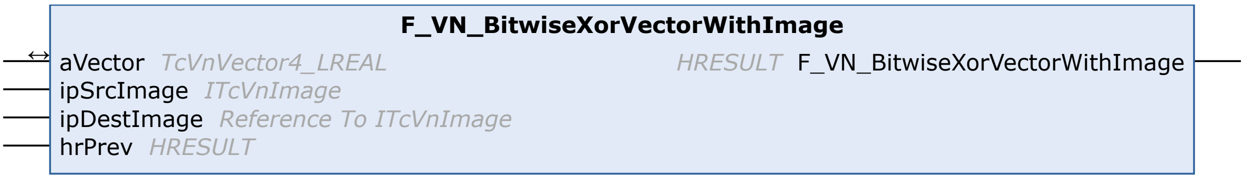
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.19 F_VN_BitwiseXorVectorWithImage



Bitwise xor a vector (1 element for each image channel) with each image pixel.

Syntax

Definition:

```
FUNCTION F_VN_BitwiseXorVectorWithImage : HRESULT
VAR_IN_OUT
    aVector     : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aVector	TcVnVector4_LREAL [▶ 141]	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)

Return value

HRESULT [▶ 122]

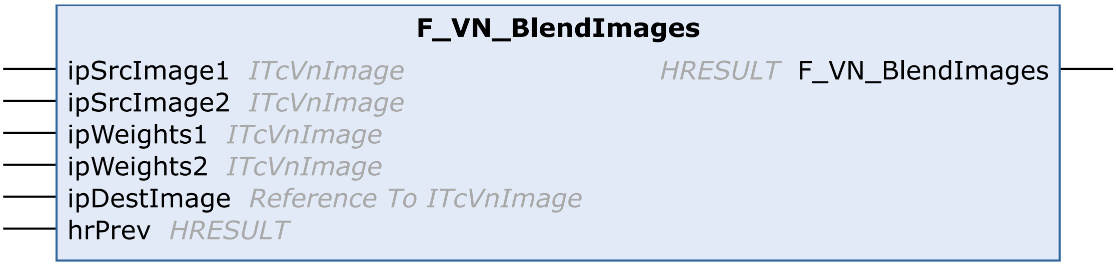
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.20 F_VN_BlendImages



Blends two images, i.e. dest = (src1 * weight1 + src2 * weight2) / (weight1 + weight2).
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_BlendImages : HRESULT
VAR_INPUT
    ipSrcImage1 : ITcVnImage;
    ipSrcImage2 : ITcVnImage;
    ipWeights1  : ITcVnImage;
    ipWeights2  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▸ 390]	First source image (ET_USINT or ET_REAL)
ipSrcImage2	ITcVnImage [▸ 390]	Second source image (same type and size as ipSrcImage1)
ipWeights1	ITcVnImage [▸ 390]	Weights for ipSrcImage1 (1 channel, ET_REAL)
ipWeights2	ITcVnImage [▸ 390]	Weights for ipSrcImage2 (1 channel, ET_REAL)
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

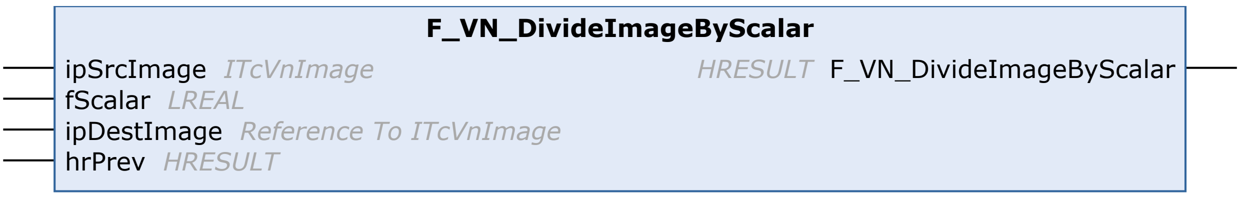
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.21 F_VN_DivideImageByScalar



Divide each image pixel by a scalar value using saturation arithmetics.

Syntax

Definition:

```
FUNCTION F_VN_DivideImageByScalar : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    fScalar     : LREAL;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
fScalar	LREAL	Scalar value
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

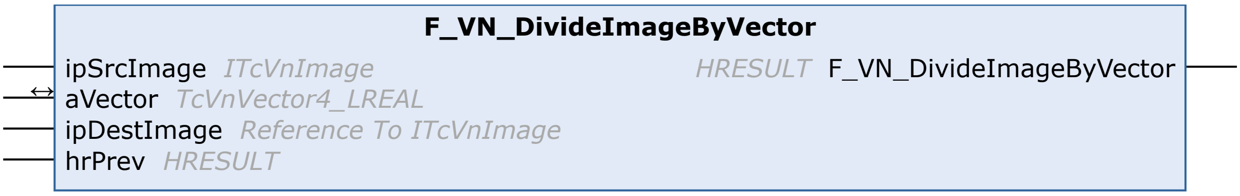
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.22 F_VN_DivideImageByVector



Divides each image pixel by a vector (1 element for each image channel) using saturation arithmetics.

Syntax

Definition:

```
FUNCTION F_VN_DivideImageByVector : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
END_VAR
VAR_IN_OUT
    aVector     : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aVector	TcVnVector4_LREAL [▸ 141]	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.23 F_VN_DivideImages

Element-wise division of two images using saturation arithmetics. (A division by zero equals zero.)

Syntax

Definition:

```
FUNCTION F_VN_DivideImages : HRESULT
VAR_INPUT
    ipSrcImage1 : ITcVnImage;
    ipSrcImage2 : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▶ 390]	First source image (dividend)
ipSrcImage2	ITcVnImage [▶ 390]	Second source image (divisor)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

在除以零的情况下，函数F_VN_DivideImages针对各元素返回零。然而，这只有在执行PLC 任务 [▶ 55]的选项**浮点异常**被停用时才有效。否则会发生除以零的错误。

注意

浮点异常

如果正在执行的PLC 任务 [▶ 55]的选项**浮点异常**启用，该功能会不必要地引起错误。因此，停用这个选项。

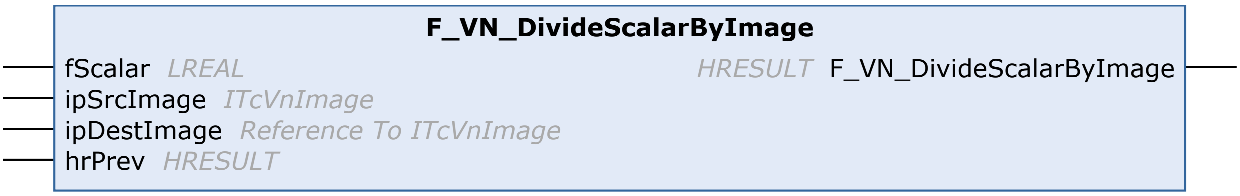
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.24 F_VN_DivideScalarByImage



Divides a scalar value by each image pixel using saturation arithmetics.

Syntax

Definition:

```
FUNCTION F_VN_DivideScalarByImage : HRESULT
VAR_INPUT
    fScalar     : LREAL;
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fScalar	LREAL	Scalar value
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

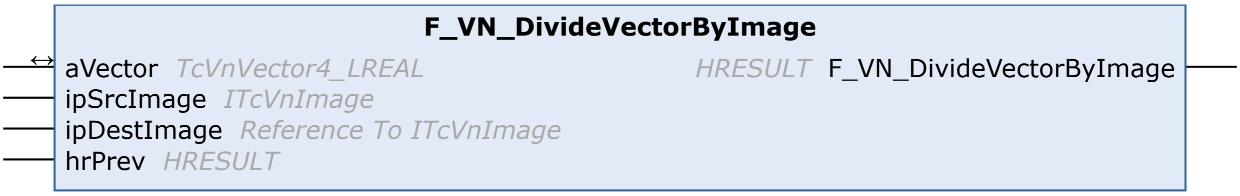
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.25 F_VN_DivideVectorByImage



Divides a vector (1 element for each image channel) by each image pixel using saturation arithmetics.

Syntax

Definition:

```
FUNCTION F_VN_DivideVectorByImage : HRESULT
VAR_IN_OUT
    aVector     : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aVector	TcVnVector4_LREAL [▶ 141]	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.26 F_VN_ElementwiseExp

Computes the natural exponent of each pixel value.

Syntax

Definition:

```
FUNCTION F_VN_ElementwiseExp : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (REAL or LREAL)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (Same type and size as ipSrcImage, an appropriate image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.27 F_VN_ElementwiseLog

Computes the natural logarithm of each pixel value.

Syntax

Definition:

```
FUNCTION F_VN_ElementwiseLog : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (REAL or LREAL)
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (Same type and size as ipSrcImage, an appropriate image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.28 F_VN_MaxImage

Element-wise maximum of two images.

Syntax

Definition:

```
FUNCTION F_VN_MaxImage : HRESULT
VAR_INPUT
    ipSrcImage1 : ITcVnImage;
    ipSrcImage2 : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▶ 390]	First source image
ipSrcImage2	ITcVnImage [▶ 390]	Second source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.29 F_VN_MaxImageExp

Element-wise maximum of two images. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_MaxImageExp : HRESULT
VAR_INPUT
    ipSrcImage1 : ITcVnImage;
    ipSrcImage2 : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    ipMask      : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▶ 390]	First source image
ipSrcImage2	ITcVnImage [▶ 390]	Second source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage [▶ 390]	Mask of type USINT
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.30 F_VN_MaxImageWithScalar

Element-wise maximum of image and scalar value.

Syntax

Definition:

```
FUNCTION F_VN_MaxImageWithScalar : HRESULT
VAR_INPUT
    fScalar     : LREAL;
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fScalar	LREAL	Scalar value
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

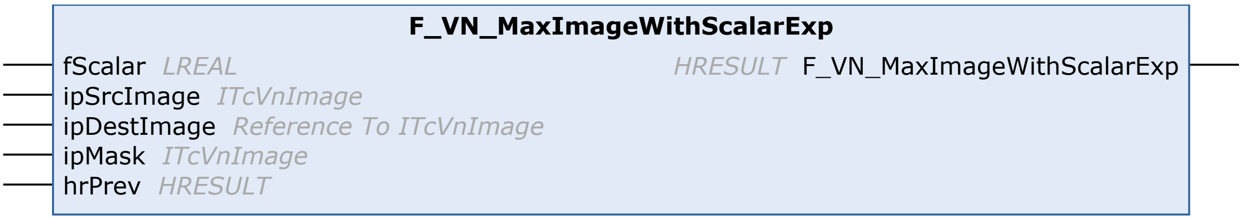
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.31 F_VN_MaxImageWithScalarExp



Element-wise maximum of image and scalar value. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_MaxImageWithScalarExp : HRESULT
VAR_INPUT
    fScalar     : LREAL;
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    ipMask      : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fScalar	LREAL	Scalar value
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage [▶ 390]	Mask of type USINT
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

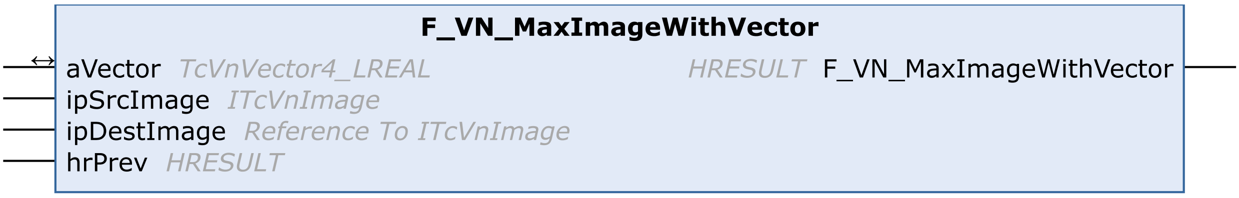
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.32 F_VN_MaxImageWithVector



Element-wise maximum of image and vector (1 element for each image channel).

Syntax

Definition:

```
FUNCTION F_VN_MaxImageWithVector : HRESULT
VAR_IN_OUT
    aVector     : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aVector	TcVnVector4_LREAL [▸ 141]	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)

Return value

HRESULT [▸ 122]

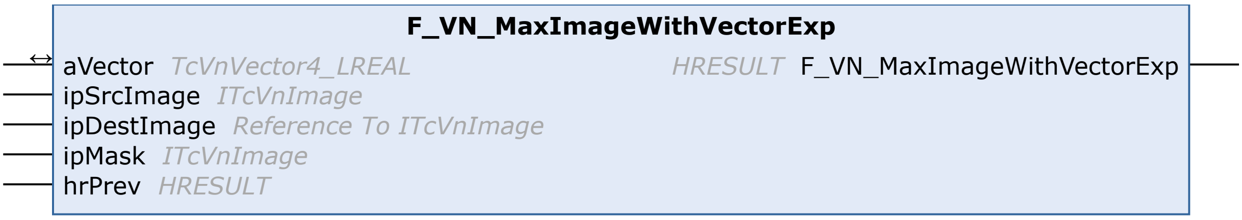
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.33 F_VN_MaxImageWithVectorExp



Element-wise maximum of image and vector (1 element for each image channel). (expert function)

Syntax

Definition:

```
FUNCTION F_VN_MaxImageWithVectorExp : HRESULT
VAR_IN_OUT
    aVector     : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    ipMask      : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage [▶ 390]	Mask of type USINT
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aVector	TcVnVector4_LREAL [▶ 141]	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.34 F_VN_MinImage

Element-wise minimum of two images.

Syntax

Definition:

```
FUNCTION F_VN_MinImage : HRESULT
VAR_INPUT
    ipSrcImage1 : ITcVnImage;
    ipSrcImage2 : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▸ 390]	First source image
ipSrcImage2	ITcVnImage [▸ 390]	Second source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

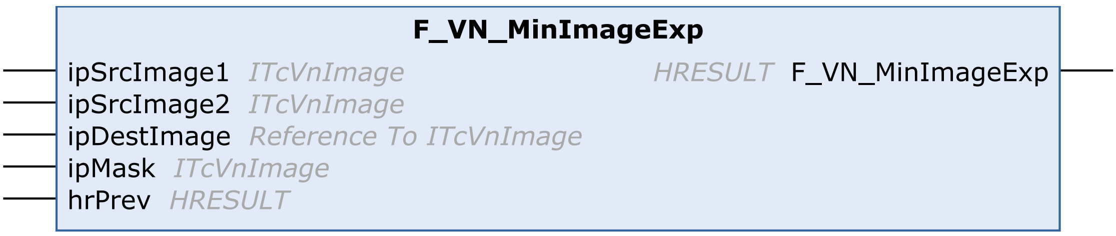
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.35 F_VN_MinImageExp



Element-wise minimum of two images. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_MinImageExp : HRESULT
VAR_INPUT
    ipSrcImage1 : ITcVnImage;
    ipSrcImage2 : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    ipMask      : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▸ 390]	First source image
ipSrcImage2	ITcVnImage [▸ 390]	Second source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage [▸ 390]	Mask of type USINT
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.36 F_VN_MinImageWithScalar

Element-wise minimum of image and scalar value.

Syntax

Definition:

```
FUNCTION F_VN_MinImageWithScalar : HRESULT
VAR_INPUT
    fScalar     : LREAL;
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fScalar	LREAL	Scalar value
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

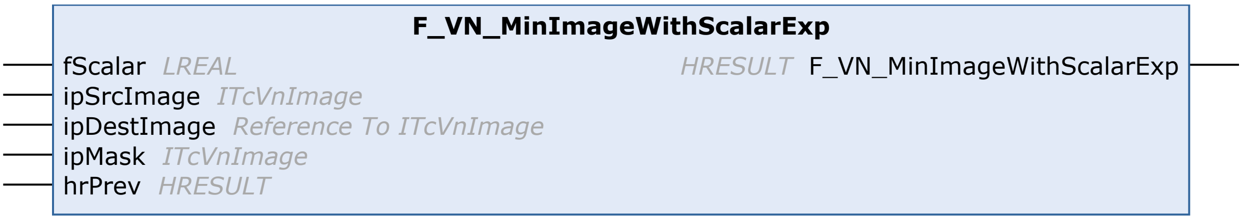
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.37 F_VN_MinImageWithScalarExp



Element-wise minimum of image and scalar value. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_MinImageWithScalarExp : HRESULT
VAR_INPUT
    fScalar     : LREAL;
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    ipMask      : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fScalar	LREAL	Scalar value
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage [▶ 390]	Mask of type USINT
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.38 F_VN_MinImageWithVector

Element-wise minimum of image and vector (1 element for each image channel).

Syntax

Definition:

```
FUNCTION F_VN_MinImageWithVector : HRESULT
VAR_IN_OUT
    aVector     : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aVector	TcVnVector4_LREAL [▸ 141]	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)

Return value

HRESULT [▸ 122]

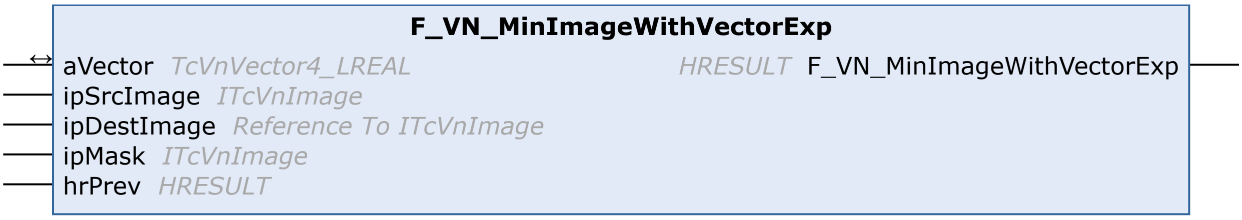
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.39 F_VN_MinImageWithVectorExp



Element-wise minimum of image and vector (1 element for each image channel). (expert function)

Syntax

Definition:

```
FUNCTION F_VN_MinImageWithVectorExp : HRESULT
VAR_IN_OUT
    aVector     : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    ipMask      : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage [▸ 390]	Mask of type USINT
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aVector	TcVnVector4_LREAL [▶ 141]	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.40 F_VN_MultiplyImages

Element-wise multiplication of two images using saturation arithmetics.

Syntax

Definition:

```
FUNCTION F_VN_MultiplyImages : HRESULT
VAR_INPUT
    ipSrcImage1 : ITcVnImage;
    ipSrcImage2 : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▶ 390]	First source image
ipSrcImage2	ITcVnImage [▶ 390]	Second source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.41 F_VN_MultiplyImageWithScalar

Multiply each image pixel by a scalar value using saturation arithmetics.

Syntax

Definition:

```
FUNCTION F_VN_MultiplyImageWithScalar : HRESULT
VAR_INPUT
    fScalar     : LREAL;
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fScalar	LREAL	Scalar value
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

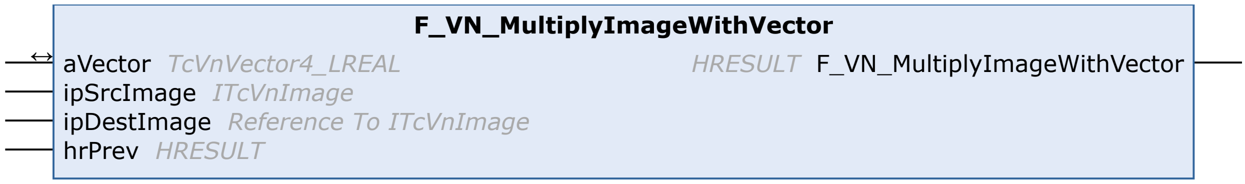
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.42 F_VN_MultiplyImageWithVector



Multiply each image pixel by a vector (1 element for each image channel) using saturation arithmetics.

Syntax

Definition:

```
FUNCTION F_VN_MultiplyImageWithVector : HRESULT
VAR_IN_OUT
    aVector     : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aVector	TcVnVector4_LREAL [▸ 141]	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.43 F_VN_SubtractImageFromScalar

Subtract each image pixel from a scalar value using saturation arithmetics.

Syntax

Definition:

```
FUNCTION F_VN_SubtractImageFromScalar : HRESULT
VAR_INPUT
    fScalar     : LREAL;
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fScalar	LREAL	Scalar value
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.44 F_VN_SubtractImageFromVector

Subtract each image pixel from a vector (1 element for each image channel) using saturation arithmetics.

Syntax

Definition:

```
FUNCTION F_VN_SubtractImageFromVector : HRESULT
VAR_IN_OUT
    aVector     : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aVector	TcVnVector4_LREAL [▸ 141]	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)

Return value

HRESULT [▸ 122]

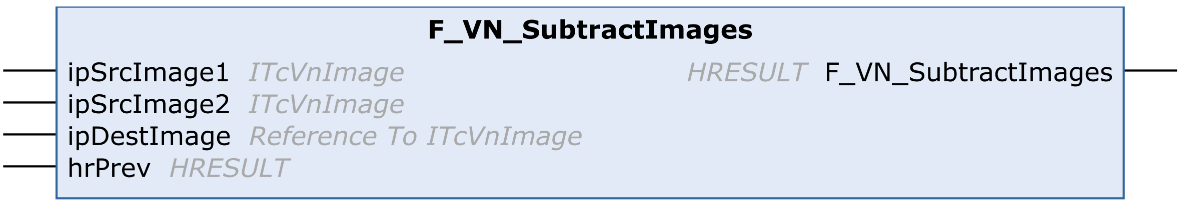
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.45 F_VN_SubtractImages



Element-wise subtraction of two images using saturation arithmetics.

Syntax

Definition:

```
FUNCTION F_VN_SubtractImages : HRESULT
VAR_INPUT
    ipSrcImage1 : ITcVnImage;
    ipSrcImage2 : ITcVnImage;
```

```
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▶ 390]	First source image (minuend)
ipSrcImage2	ITcVnImage [▶ 390]	Second source image (subtrahend)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

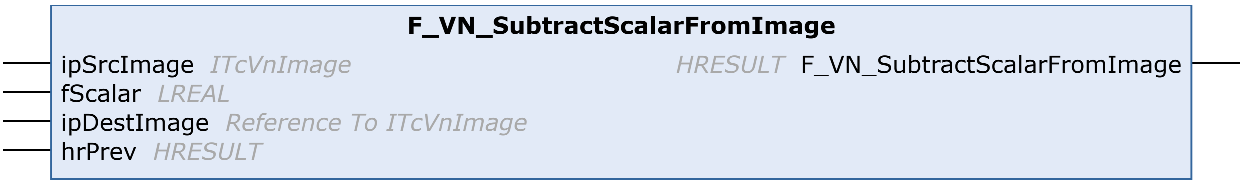
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.46 F_VN_SubtractScalarFromImage



Subtract a scalar value from each image pixel using saturation arithmetics.

Syntax

Definition:

```
FUNCTION F_VN_SubtractScalarFromImage : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    fScalar     : LREAL;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
fScalar	LREAL	Scalar value
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

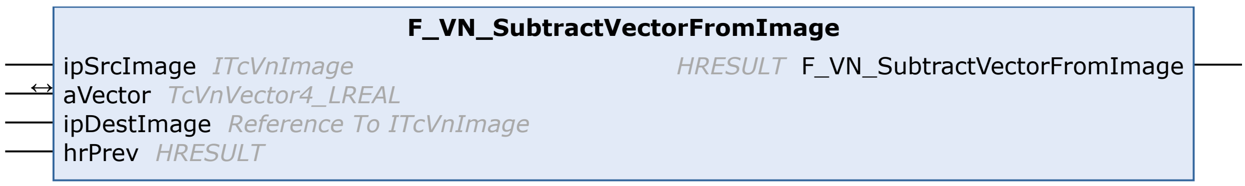
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.3.47 F_VN_SubtractVectorFromImage



Subtract a vector (1 element for each image channel) from each image pixel using saturation arithmetics.

Syntax

Definition:

```
FUNCTION F_VN_SubtractVectorFromImage : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
END_VAR
VAR_IN_OUT
    aVector     : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aVector	TcVnVector4_LREAL [▶ 141]	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4 Basic Container Operations

该组包含处理容器 [▶ 132]的功能。

函数

容器创建

- F_VN_CopyContainer [▶ 722]
- F_VN_CopyContainerElementsConditional_ITcVnContainer [▶ 722]
- F_VN_CopyContainerElementsConditional_ITcVnForwardIterator [▶ 724]
- F_VN_CreateContainer [▶ 725]
- F_VN_CreateContainerFromArray [▶ 726]
- F_VN_ExtractContainerRange [▶ 735]
- F_VN_ReserveContainerMemory [▶ 742]

容器元素的处理

- F_VN_AppendToContainer [▶ 521]
- F_VN_CheckIfEmpty [▶ 719]
- F_VN_ConvertContainerType [▶ 720]
- F_VN_EraseFromContainer [▶ 727]
- F_VN_FillContainer(Exp) [▶ 580]
- F_VN_GetAt [▶ 615]
- F_VN_GetContainer(Exp) [▶ 736]
- F_VN_GetNumberOfElements [▶ 738]
- F_VN_InsertIntoContainer [▶ 650]
- F_VN_ReverseContainer [▶ 743]
- F_VN_SetAt [▶ 684]
- F_VN_SetContainer [▶ 744]

将容器数据导出到存储区

- F_VN_ExportContainer [▶ 728]
- F_VN_ExportSubContainer [▶ 731]

- F_VN_ExportContainer_String [▶ 729]
- F_VN_ExportSubContainer_String [▶ 732]
- F_VN_ExportContainerSize [▶ 730]
- F_VN_ExportSubContainerSize [▶ 734]

迭代器的处理

- F_VN_AdvanceIterator [▶ 718]
- F_VN_CheckIfIteratorIsAtEnd [▶ 720]
- F_VN_GetForwardIterator [▶ 738]
- F_VN_GetRandomAccessIterator [▶ 739]
- F_VN_IncrementIterator [▶ 740]
- F_VN_IteratorDistance [▶ 741]
- F_VN_SetIteratorToBegin [▶ 744]

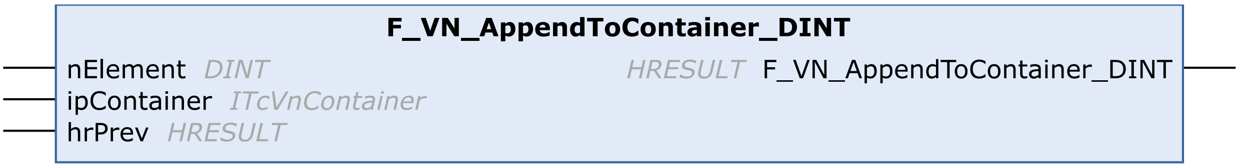
样本

容器基本操作 [▶ 2640]

6.1.4.4.1 F_VN_AppendToContainer

本章包含根据数据类型向容器中添加元素的功能。

6.1.4.4.1.1 F_VN_AppendToContainer_DINT



Append a single element of type DINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_DINT : HRESULT
VAR_INPUT
    nElement    : DINT;
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	DINT	Single element to append to ipContainer
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

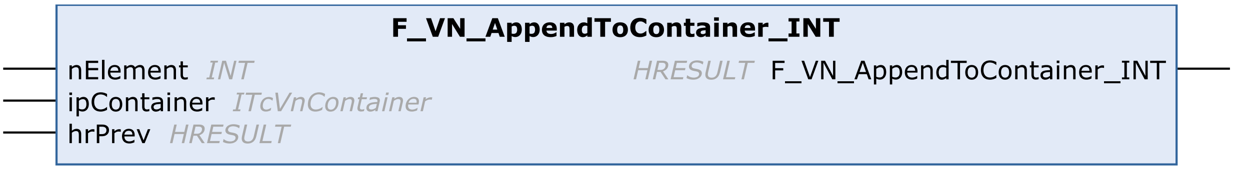
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.2 F_VN_AppendToContainer_INT



Append a single element of type INT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_INT : HRESULT
VAR_INPUT
    nElement     : INT;
    ipContainer : ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	INT	Single element to append to ipContainer
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

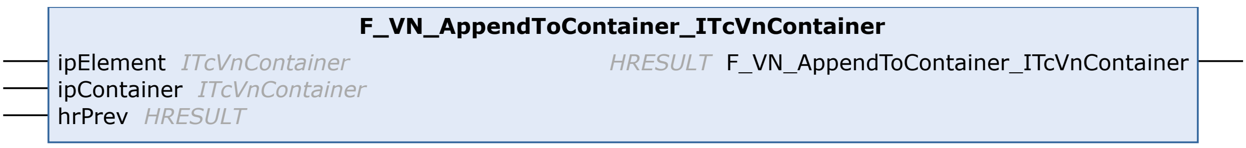
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.3 F_VN_AppendToContainer_ITcVnContainer



Append a single element to a container or concatenate containers (if ipElement has the same typeId as ipContainer).

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_ITcVnContainer : HRESULT
VAR_INPUT
    ipElement   : ITcVnContainer;
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipElement	ITcVnContainer [▶ 349]	Single element to append to ipContainer or container with several elements to be concatenated
ipContainer	ITcVnContainer [▶ 349]	Container to which the element(s) will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

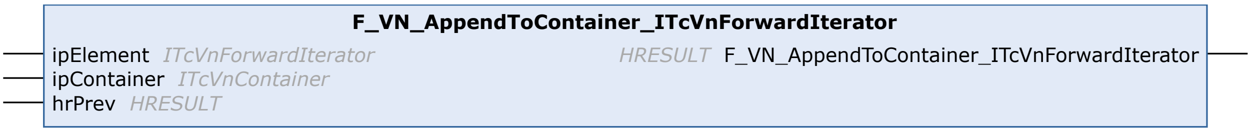
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.4 F_VN_AppendToContainer_ITcVnForwardIterator



Append a single element (represented by an iterator) to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_ITcVnForwardIterator : HRESULT
VAR_INPUT
    ipElement   : ITcVnForwardIterator;
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipElement	ITcVnForwardIterator [▶ 339]	Single element to append to ipContainer
ipContainer	ITcVnContainer [▶ 349]	Container to which the element(s) will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

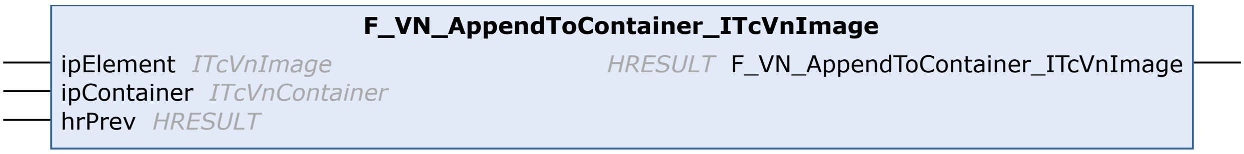
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.5 F_VN_AppendToContainer_ITcVnImage



Append a single element of type ITcVnImage to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_ITcVnImage : HRESULT
VAR_INPUT
    ipElement   : ITcVnImage;
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipElement	ITcVnImage [▶ 390]	Single element to append to ipContainer
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.6 F_VN_AppendToContainer_LREAL

Append a single element of type LREAL to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_LREAL : HRESULT
VAR_INPUT
    fElement    : LREAL;
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fElement	LREAL	Single element to append to ipContainer
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

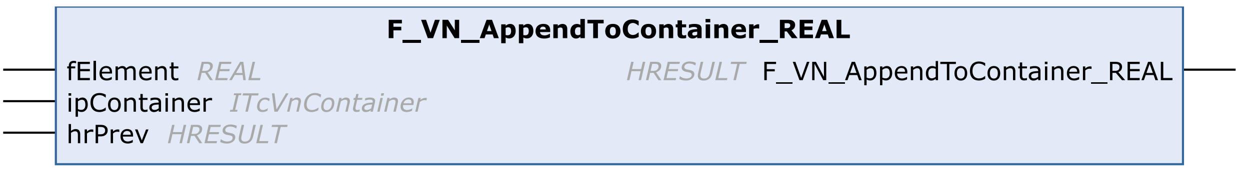
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.7 F_VN_AppendToContainer_REAL



Append a single element of type REAL to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_REAL : HRESULT
VAR_INPUT
    fElement    : REAL;
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fElement	REAL	Single element to append to ipContainer
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.8 F_VN_AppendToContainer_SINT

Append a single element of type SINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_SINT : HRESULT
VAR_INPUT
    nElement    : SINT;
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	SINT	Single element to append to ipContainer
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

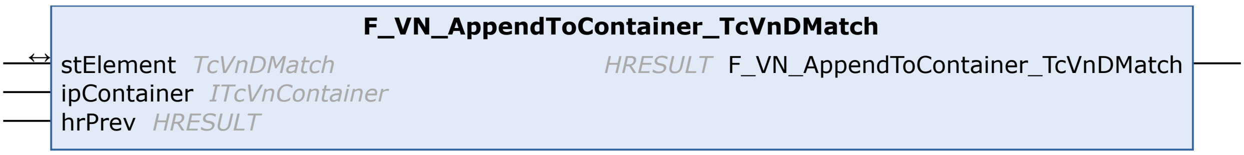
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.9 F_VN_AppendToContainer_TcVnDMatch



Append a single element of type TcVnDMatch to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnDMatch : HRESULT
VAR_IN_OUT
    stElement   : TcVnDMatch;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stElement	TcVnDMatch [▶ 210]	Single element to append to ipContainer

Return value

HRESULT [▶ 122]

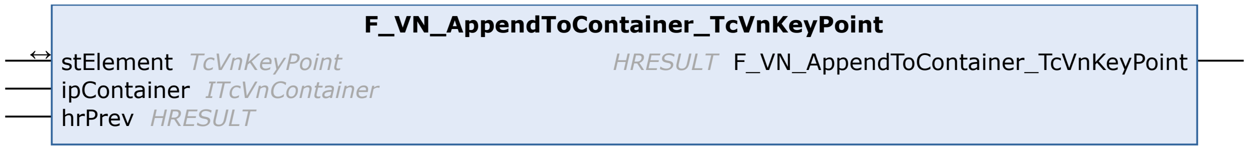
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.10 F_VN_AppendToContainer_TcVnKeyPoint



Append a single element of type TcVnKeyPoint to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnKeyPoint : HRESULT
VAR_IN_OUT
    stElement   : TcVnKeyPoint;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stElement	TcVnKeyPoint [▶ 210]	Single element to append to ipContainer

Return value

HRESULT [▶ 122]

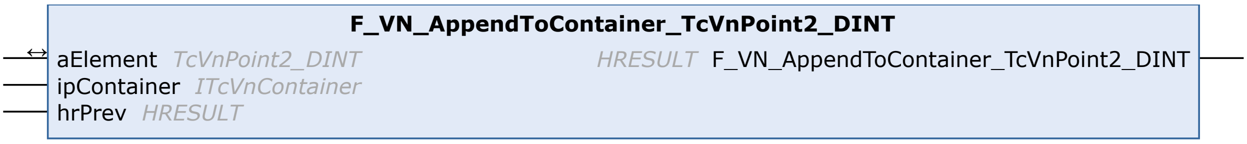
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.11 F_VN_AppendToContainer_TcVnPoint2_DINT



Append a single element of type TcVnPoint2_DINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnPoint2_DINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnPoint2_DINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container to which the element will be appended
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint2_DINT [▸ 139]	Single element to append to ipContainer

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.12 F_VN_AppendToContainer_TcVnPoint2_LREAL

F_VN_AppendToContainer_TcVnPoint2_LREAL
aElement *TcVnPoint2_LREAL* — *HRESULT* F_VN_AppendToContainer_TcVnPoint2_LREAL
ipContainer *ITcVnContainer*
hrPrev *HRESULT*

Append a single element of type TcVnPoint2_LREAL to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnPoint2_LREAL : HRESULT
VAR_IN_OUT
    aElement    : TcVnPoint2_LREAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint2_LREAL [▶ 139]	Single element to append to ipContainer

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.13 F_VN_AppendToContainer_TcVnPoint2_REAL

F_VN_AppendToContainer_TcVnPoint2_REAL
aElement *TcVnPoint2_REAL* — *HRESULT* F_VN_AppendToContainer_TcVnPoint2_REAL
ipContainer *ITcVnContainer*
hrPrev *HRESULT*

Append a single element of type TcVnPoint2_REAL to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnPoint2_REAL : HRESULT
VAR_IN_OUT
    aElement    : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint2_REAL [▶ 139]	Single element to append to ipContainer

Return value

HRESULT [▶ 122]

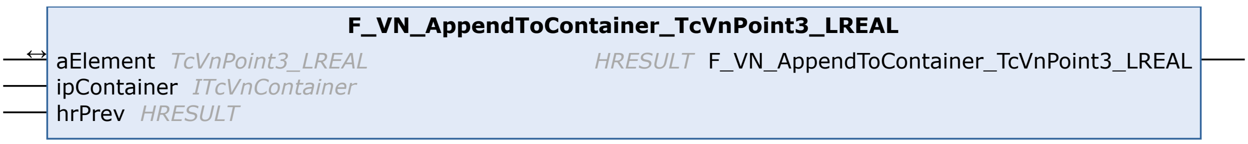
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.14 F_VN_AppendToContainer_TcVnPoint3_LREAL



Append a single element of type TcVnPoint3_LREAL to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnPoint3_LREAL : HRESULT
VAR_IN_OUT
    aElement    : TcVnPoint3_LREAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint3_LREAL [▶ 139]	Single element to append to ipContainer

Return value

HRESULT [▶ 122]

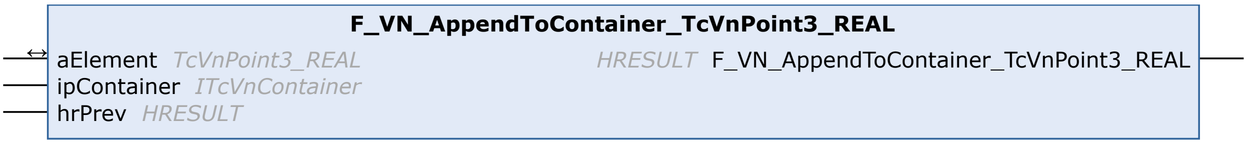
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.15 F_VN_AppendToContainer_TcVnPoint3_REAL



Append a single element of type TcVnPoint3_REAL to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnPoint3_REAL : HRESULT
VAR_IN_OUT
    aElement    : TcVnPoint3_REAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint3_REAL [▸ 139]	Single element to append to ipContainer

Return value

HRESULT [▸ 122]

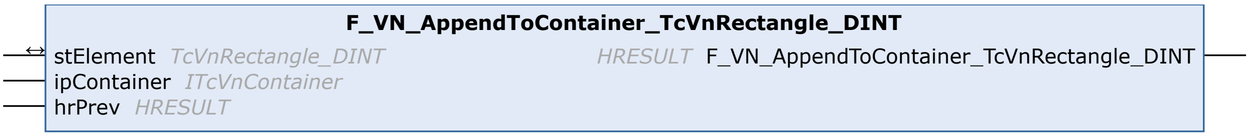
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.16 F_VN_AppendToContainer_TcVnRectangle_DINT



Append a single element of type TcVnRectangle_DINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnRectangle_DINT : HRESULT
VAR_IN_OUT
    stElement   : TcVnRectangle_DINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container to which the element will be appended
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stElement	TcVnRectangle_DINT [▸ 224]	Single element to append to ipContainer

Return value

HRESULT [▸ 122]

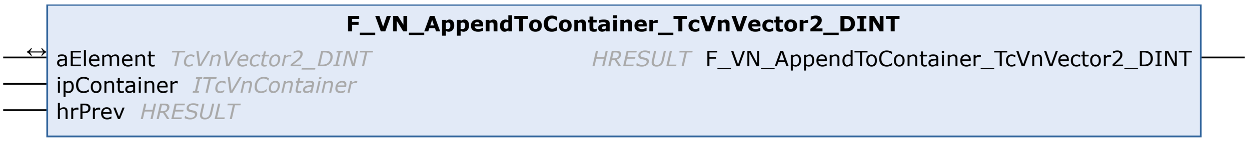
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.17 F_VN_AppendToContainer_TcVnVector2_DINT



Append a single element of type TcVnVector2_DINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnVector2_DINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector2_DINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container to which the element will be appended
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_DINT [▸ 141]	Single element to append to ipContainer

Return value

HRESULT [▸ 122]

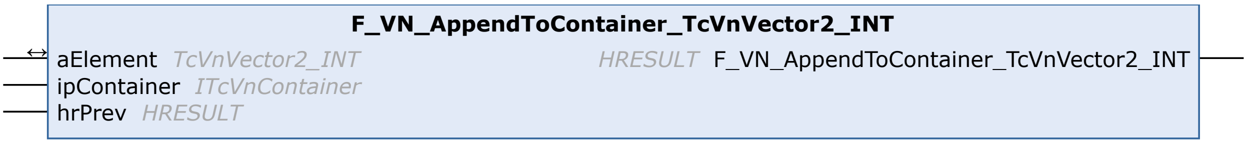
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.18 F_VN_AppendToContainer_TcVnVector2_INT



Append a single element of type TcVnVector2_INT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnVector2_INT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector2_INT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_INT [▶ 141]	Single element to append to ipContainer

Return value

HRESULT [▶ 122]

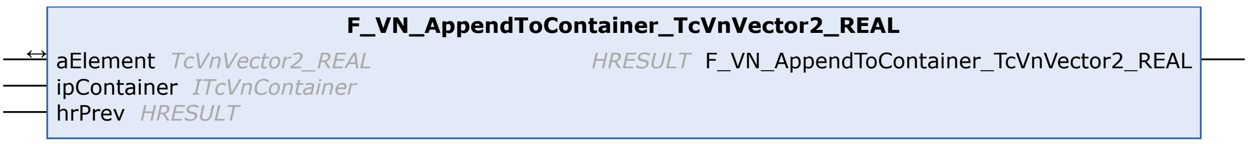
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.19 F_VN_AppendToContainer_TcVnVector2_REAL



Append a single element of type TcVnVector2_REAL to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnVector2_REAL : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector2_REAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container to which the element will be appended
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_REAL [▸ 141]	Single element to append to ipContainer

Return value

HRESULT [▸ 122]

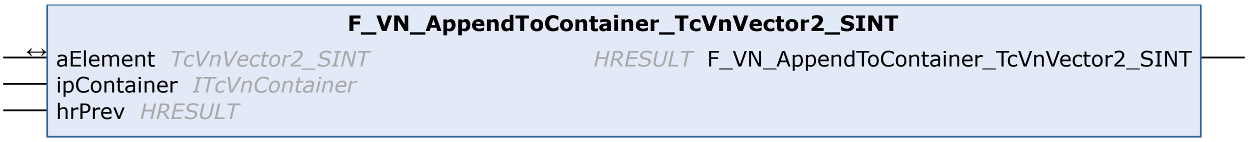
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.20 F_VN_AppendToContainer_TcVnVector2_SINT



Append a single element of type TcVnVector2_SINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnVector2_SINT : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector2_SINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_SINT [▶ 141]	Single element to append to ipContainer

Return value

HRESULT [▶ 122]

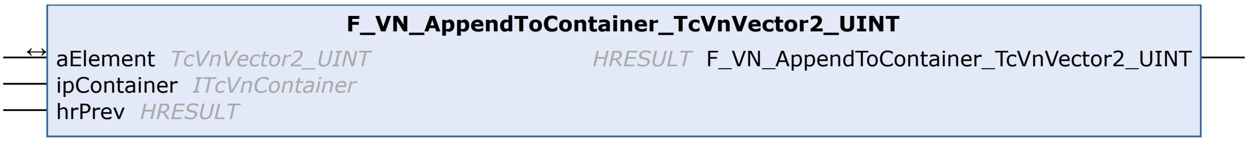
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.21 F_VN_AppendToContainer_TcVnVector2_UINT



Append a single element of type TcVnVector2_UINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnVector2_UINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector2_UINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_UINT [▸ 141]	Single element to append to ipContainer

Return value

HRESULT [▸ 122]

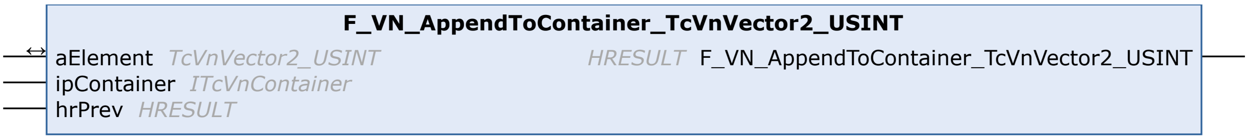
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.22 F_VN_AppendToContainer_TcVnVector2_USINT



Append a single element of type TcVnVector2_USINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnVector2_USINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector2_USINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container to which the element will be appended
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_USINT [▸ 141]	Single element to append to ipContainer

Return value

HRESULT [▸ 122]

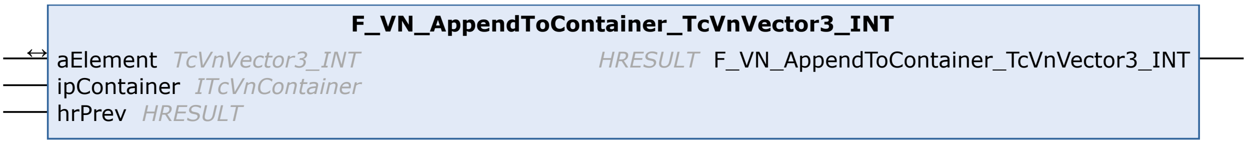
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.23 F_VN_AppendToContainer_TcVnVector3_INT



Append a single element of type TcVnVector3_INT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnVector3_INT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector3_INT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container to which the element will be appended
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_INT [▸ 141]	Single element to append to ipContainer

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.24 F_VN_AppendToContainer_TcVnVector3_REAL

F_VN_AppendToContainer_TcVnVector3_REAL	
↔ aElement *TcVnVector3_REAL*	*HRESULT* F_VN_AppendToContainer_TcVnVector3_REAL
ipContainer *ITcVnContainer*	
hrPrev *HRESULT*	

Append a single element of type TcVnVector3_REAL to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnVector3_REAL : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector3_REAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_REAL [▶ 141]	Single element to append to ipContainer

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.25 F_VN_AppendToContainer_TcVnVector3_SINT

F_VN_AppendToContainer_TcVnVector3_SINT	
↔ aElement *TcVnVector3_SINT*	*HRESULT* F_VN_AppendToContainer_TcVnVector3_SINT
ipContainer *ITcVnContainer*	
hrPrev *HRESULT*	

Append a single element of type TcVnVector3_SINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnVector3_SINT : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector3_SINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_SINT [▶ 141]	Single element to append to ipContainer

Return value

HRESULT [▶ 122]

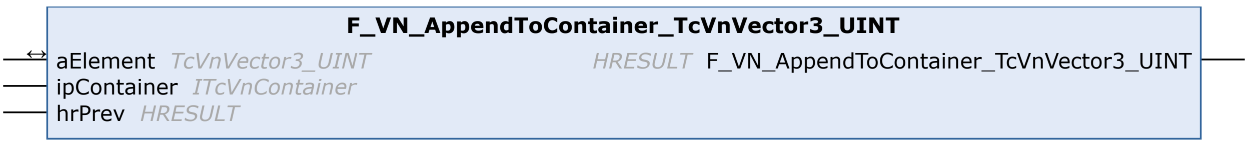
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.26 F_VN_AppendToContainer_TcVnVector3_UINT



Append a single element of type TcVnVector3_UINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnVector3_UINT : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector3_UINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_UINT [▶ 141]	Single element to append to ipContainer

Return value

HRESULT [▶ 122]

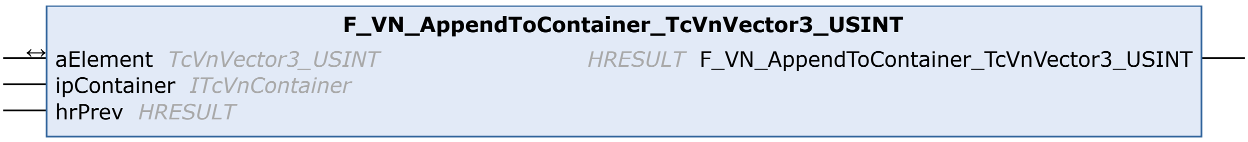
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.27 F_VN_AppendToContainer_TcVnVector3_USINT



Append a single element of type TcVnVector3_USINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnVector3_USINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector3_USINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_USINT [▶ 141]	Single element to append to ipContainer

Return value

HRESULT [▶ 122]

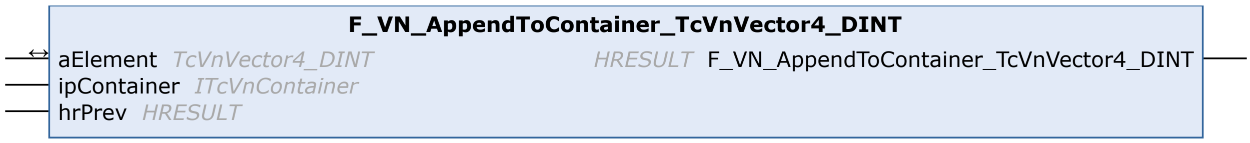
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.28 F_VN_AppendToContainer_TcVnVector4_DINT



Append a single element of type TcVnVector4_DINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnVector4_DINT : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector4_DINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_DINT [▶ 141]	Single element to append to ipContainer

Return value

HRESULT [▶ 122]

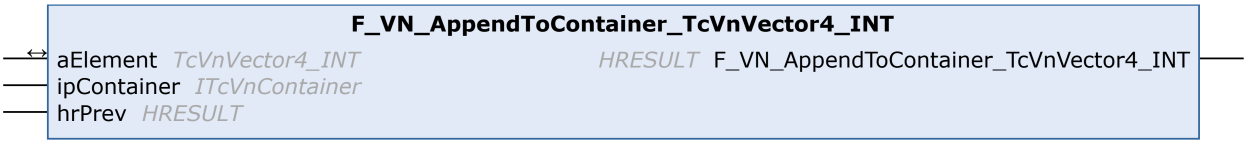
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.29 F_VN_AppendToContainer_TcVnVector4_INT



Append a single element of type TcVnVector4_INT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnVector4_INT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector4_INT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_INT [▶ 141]	Single element to append to ipContainer

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.30 F_VN_AppendToContainer_TcVnVector4_LREAL

F_VN_AppendToContainer_TcVnVector4_LREAL
aElement *TcVnVector4_LREAL* — *HRESULT* F_VN_AppendToContainer_TcVnVector4_LREAL
ipContainer *ITcVnContainer*
hrPrev *HRESULT*

Append a single element of type TcVnVector4_LREAL to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnVector4_LREAL : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_LREAL [▶ 141]	Single element to append to ipContainer

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.31 F_VN_AppendToContainer_TcVnVector4_SINT

F_VN_AppendToContainer_TcVnVector4_SINT
aElement *TcVnVector4_SINT* — *HRESULT* F_VN_AppendToContainer_TcVnVector4_SINT
ipContainer *ITcVnContainer*
hrPrev *HRESULT*

Append a single element of type TcVnVector4_SINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnVector4_SINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector4_SINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_SINT [▶ 141]	Single element to append to ipContainer

Return value

HRESULT [▶ 122]

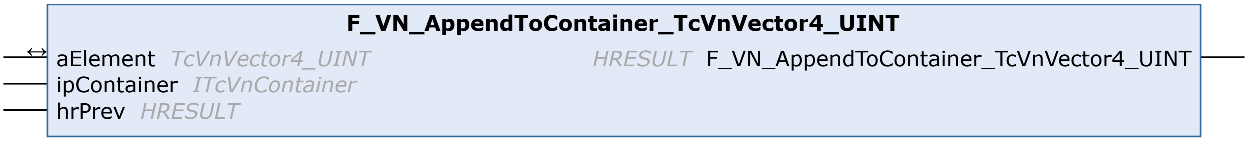
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.32 F_VN_AppendToContainer_TcVnVector4_UINT



Append a single element of type TcVnVector4_UINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnVector4_UINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector4_UINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_UINT [▶ 141]	Single element to append to ipContainer

Return value

HRESULT [▶ 122]

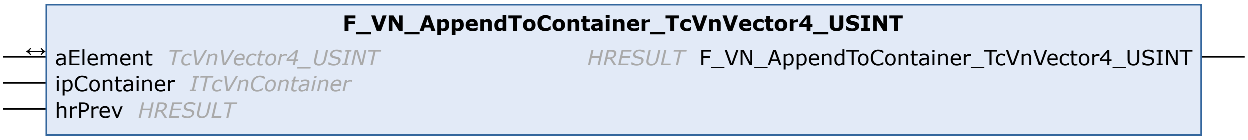
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.33 F_VN_AppendToContainer_TcVnVector4_USINT



Append a single element of type TcVnVector4_USINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_TcVnVector4_USINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector4_USINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_USINT [▶ 141]	Single element to append to ipContainer

Return value

HRESULT [▶ 122]

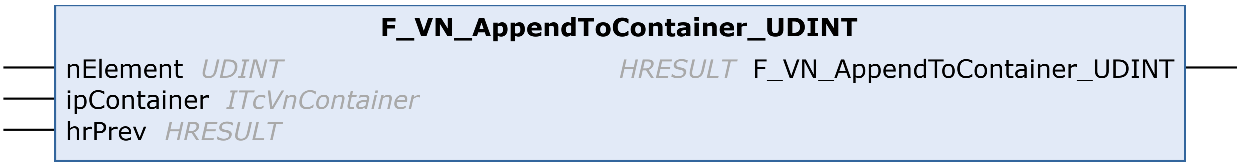
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.34 F_VN_AppendToContainer_UDINT



Append a single element of type UDINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_UDINT : HRESULT
VAR_INPUT
    nElement     : UDINT;
    ipContainer : ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	UDINT	Single element to append to ipContainer
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.35 F_VN_AppendToContainer_UINT

Append a single element of type UINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_UINT : HRESULT
VAR_INPUT
    nElement    : UINT;
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	UINT	Single element to append to ipContainer
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

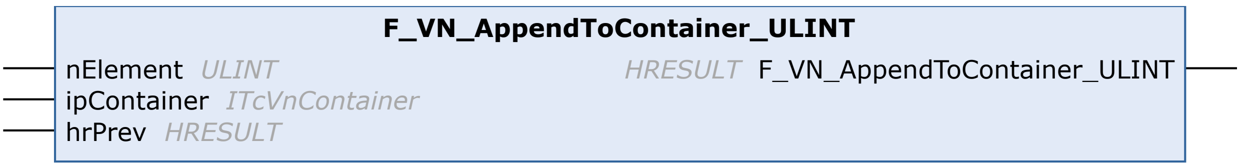
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.36 F_VN_AppendToContainer_ULINT



Append a single element of type ULINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_ULINT : HRESULT
VAR_INPUT
    nElement     : ULINT;
    ipContainer  : ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	ULINT	Single element to append to ipContainer
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

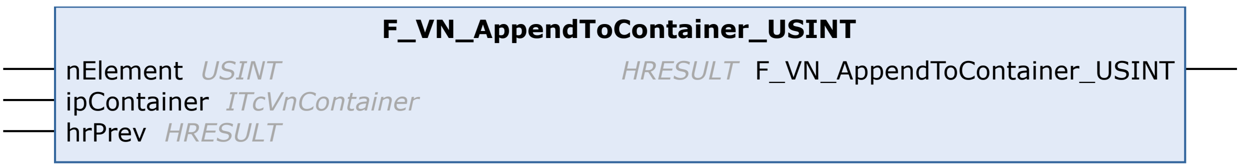
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.1.37 F_VN_AppendToContainer_USINT



Append a single element of type USINT to a container.

Syntax

Definition:

```
FUNCTION F_VN_AppendToContainer_USINT : HRESULT
VAR_INPUT
    nElement     : USINT;
    ipContainer  : ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	USINT	Single element to append to ipContainer
ipContainer	ITcVnContainer [▶ 349]	Container to which the element will be appended
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2 F_VN_FillContainer

本章包含根据数据类型用特殊值填充容器的功能。这个函数适用于包括基本元素的容器。

6.1.4.4.2.1 F_VN_FillContainer_DINT

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nValue      : DINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nValue	DINT	Value to set the container elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.2 F_VN_FillContainer_INT

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nValue      : INT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nValue	INT	Value to set the container elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

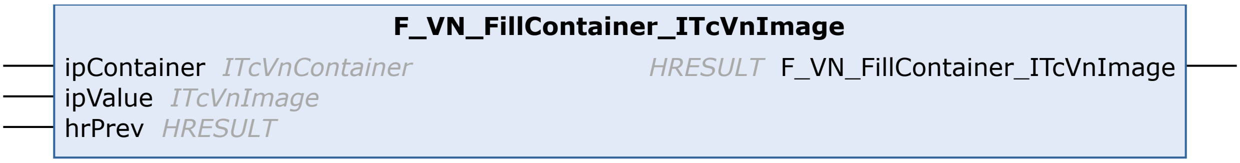
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.3 F_VN_FillContainer_ITcVnImage



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_ITcVnImage : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    ipValue     : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
ipValue	ITcVnImage [▸ 390]	Value to set the container elements
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

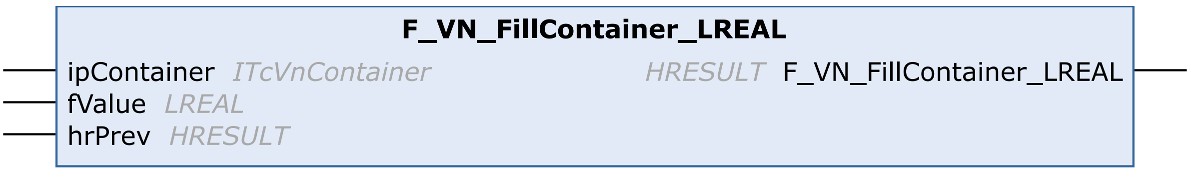
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.4 F_VN_FillContainer_LREAL



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    fValue      : LREAL;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
fValue	LREAL	Value to set the container elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.5 F_VN_FillContainer_REAL

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    fValue      : REAL;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
fValue	REAL	Value to set the container elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.6 F_VN_FillContainer_SINT

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nValue      : SINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
nValue	SINT	Value to set the container elements
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

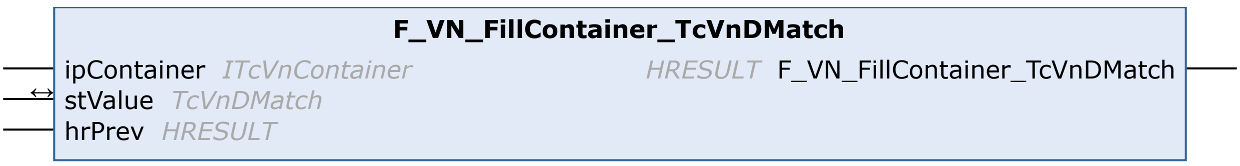
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.7 F_VN_FillContainer_TcVnDMatch



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnDMatch : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    stValue     : TcVnDMatch;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stValue	TcVnDMatch [▸ 210]	Value to set the container elements

Return value

HRESULT [▸ 122]

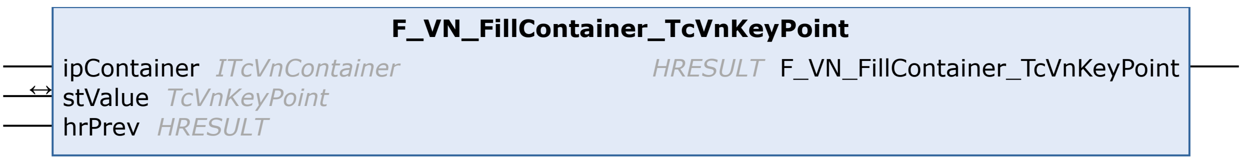
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.8 F_VN_FillContainer_TcVnKeyPoint



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnKeyPoint : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    stValue     : TcVnKeyPoint;
END_VAR
```

```
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stValue	TcVnKeyPoint [▶ 210]	Value to set the container elements

Return value

HRESULT [▶ 122]

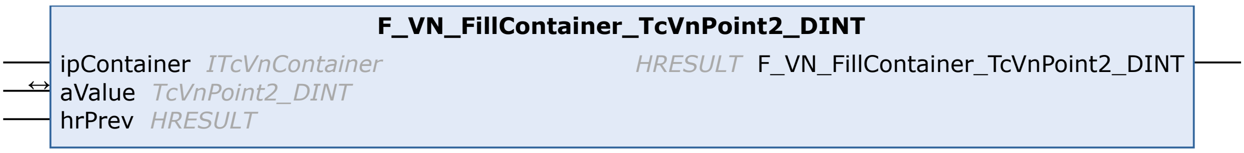
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.9 F_VN_FillContainer_TcVnPoint2_DINT



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnPoint2_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue       : TcVnPoint2_DINT;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnPoint2_DINT [▶ 139]	Value to set the container elements

Return value

HRESULT [▶ 122]

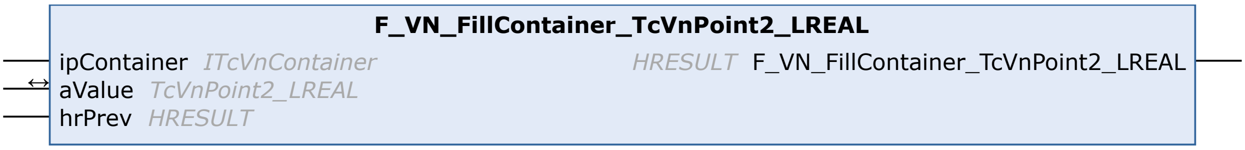
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.10 F_VN_FillContainer_TcVnPoint2_LREAL



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnPoint2_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnPoint2_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnPoint2_LREAL [▶ 139]	Value to set the container elements

Return value

HRESULT [▶ 122]

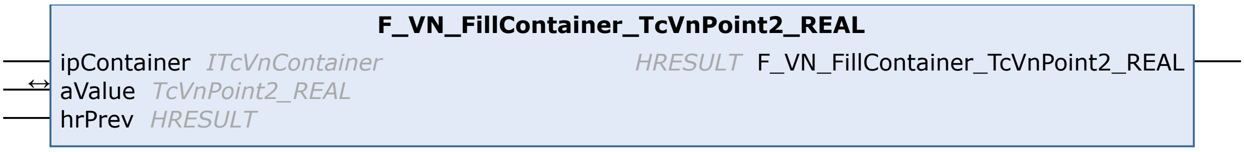
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.11 F_VN_FillContainer_TcVnPoint2_REAL



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnPoint2_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnPoint2_REAL [▶ 139]	Value to set the container elements

Return value

HRESULT [▶ 122]

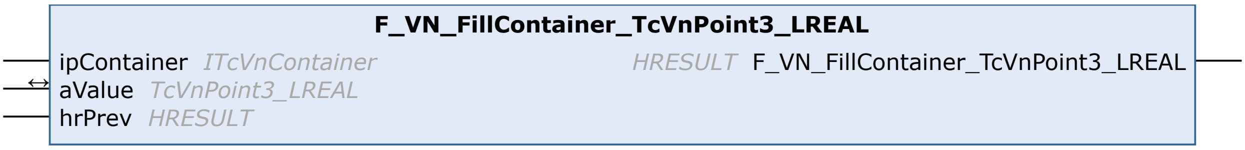
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.12 F_VN_FillContainer_TcVnPoint3_LREAL



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnPoint3_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnPoint3_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnPoint3_LREAL [▶ 139]	Value to set the container elements

Return value

HRESULT [▶ 122]

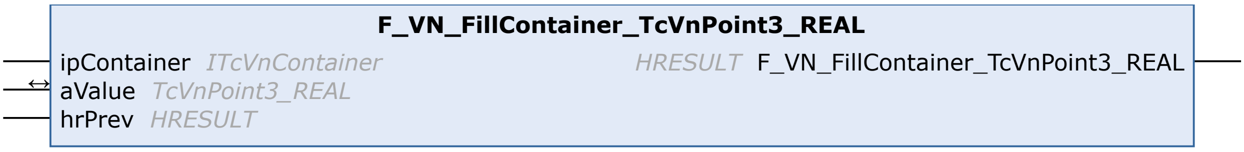
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.13 F_VN_FillContainer_TcVnPoint3_REAL



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnPoint3_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnPoint3_REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnPoint3_REAL [▸ 139]	Value to set the container elements

Return value

HRESULT [▸ 122]

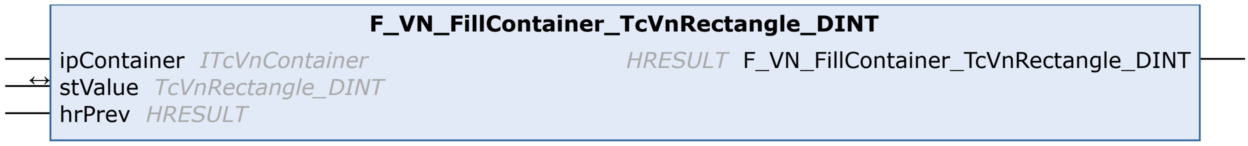
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.14 F_VN_FillContainer_TcVnRectangle_DINT



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnRectangle_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    stValue     : TcVnRectangle_DINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stValue	TcVnRectangle_DINT [▸ 224]	Value to set the container elements

Return value

HRESULT [▸ 122]

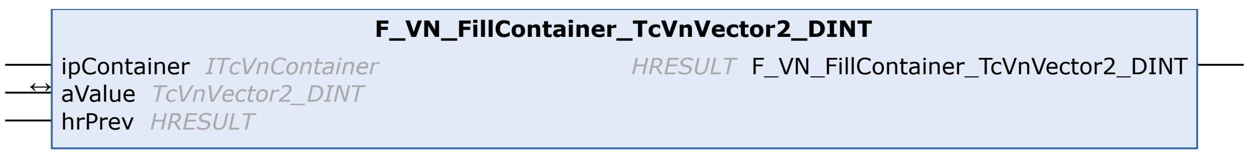
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.15 F_VN_FillContainer_TcVnVector2_DINT



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnVector2_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector2_DINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_DINT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

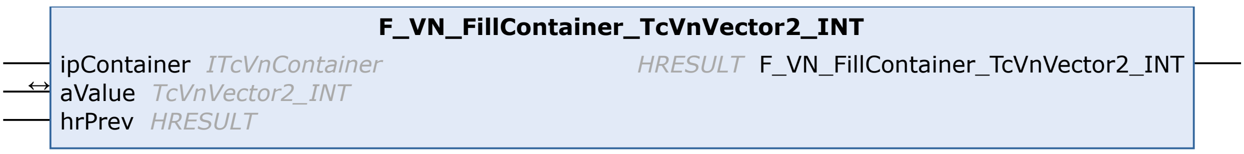
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.16 F_VN_FillContainer_TcVnVector2_INT



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnVector2_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector2_INT;
END_VAR
```

```
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_INT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

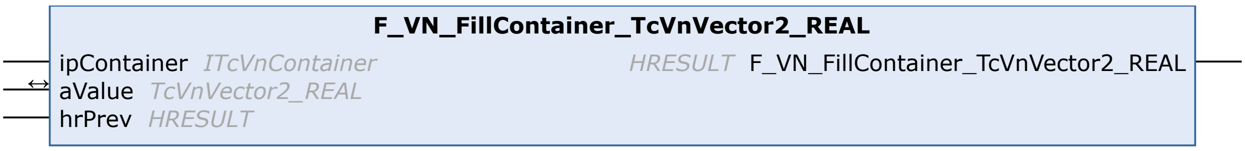
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.17 F_VN_FillContainer_TcVnVector2_REAL



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnVector2_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue       : TcVnVector2_REAL;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_REAL [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

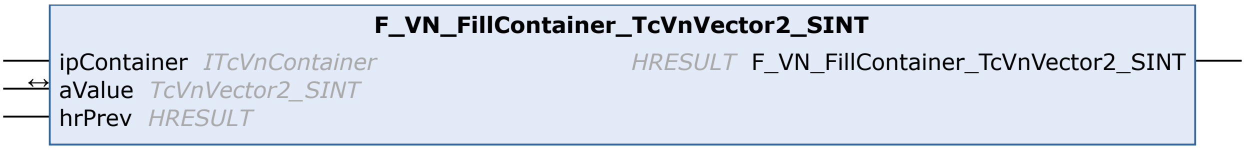
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.18 F_VN_FillContainer_TcVnVector2_SINT



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnVector2_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector2_SINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_SINT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

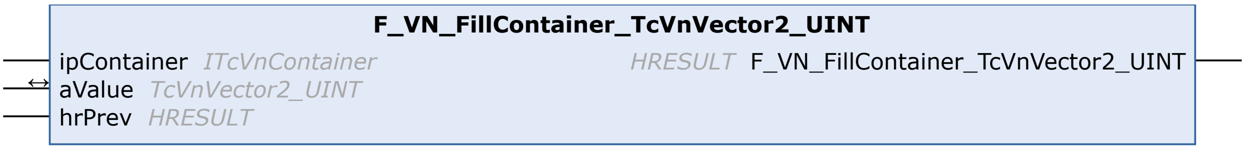
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.19 F_VN_FillContainer_TcVnVector2_UINT



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnVector2_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector2_UINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_UINT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

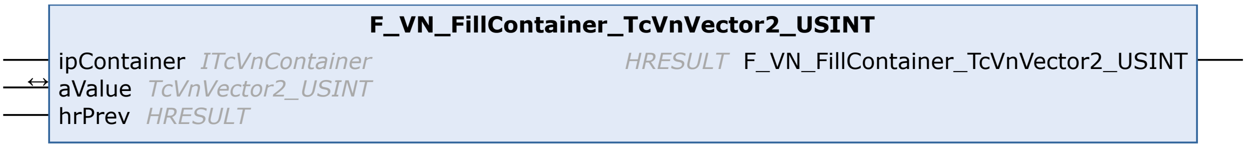
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.20 F_VN_FillContainer_TcVnVector2_USINT



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnVector2_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector2_USINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_USINT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

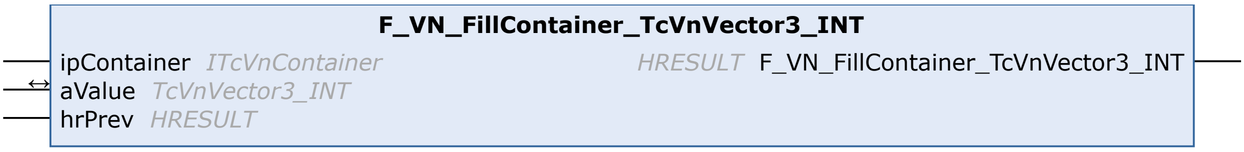
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.21 F_VN_FillContainer_TcVnVector3_INT



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnVector3_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector3_INT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector3_INT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

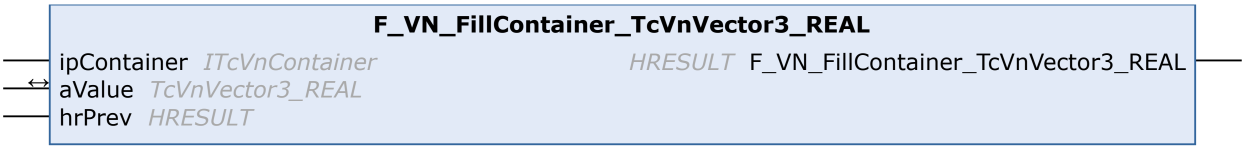
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.22 F_VN_FillContainer_TcVnVector3_REAL



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnVector3_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector3_REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector3_REAL [▸ 141]	Value to set the container elements

Return value

HRESULT [▸ 122]

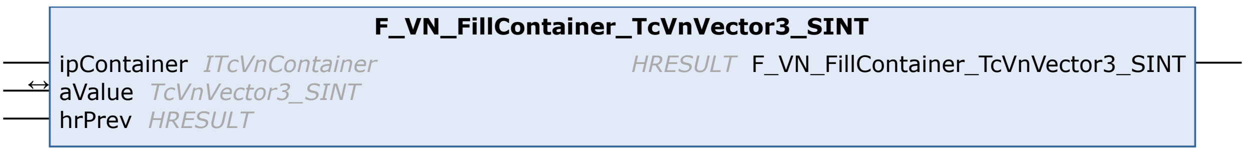
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.23 F_VN_FillContainer_TcVnVector3_SINT



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnVector3_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector3_SINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector3_SINT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

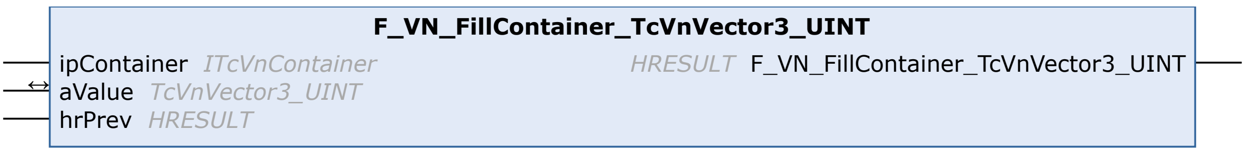
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.24 F_VN_FillContainer_TcVnVector3_UINT



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnVector3_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector3_UINT;
END_VAR
```

```
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector3_UINT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

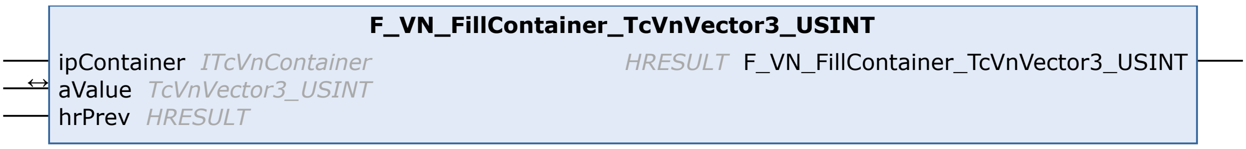
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.25 F_VN_FillContainer_TcVnVector3_USINT



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnVector3_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue       : TcVnVector3_USINT;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector3_USINT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

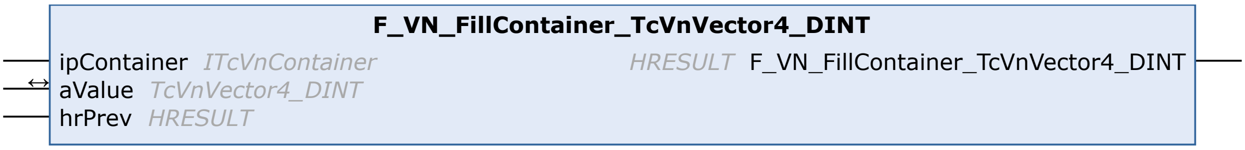
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.26 F_VN_FillContainer_TcVnVector4_DINT



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnVector4_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector4_DINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_DINT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

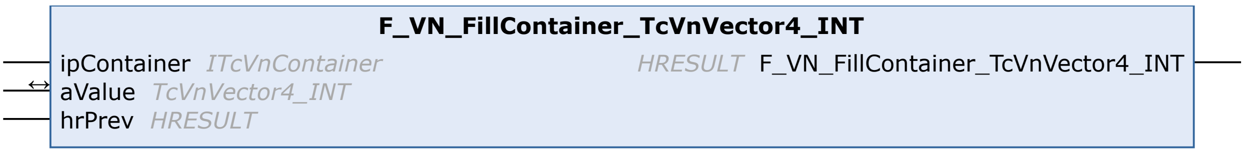
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.27 F_VN_FillContainer_TcVnVector4_INT



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnVector4_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector4_INT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_INT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

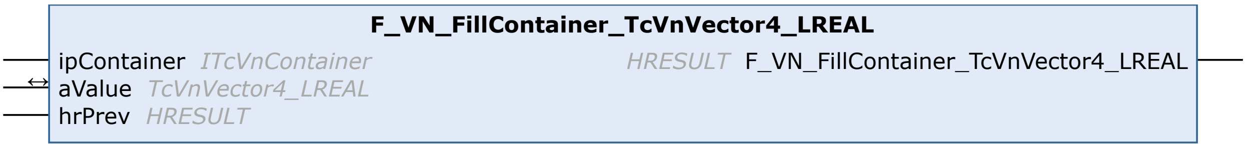
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.28 F_VN_FillContainer_TcVnVector4_LREAL



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnVector4_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_LREAL [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

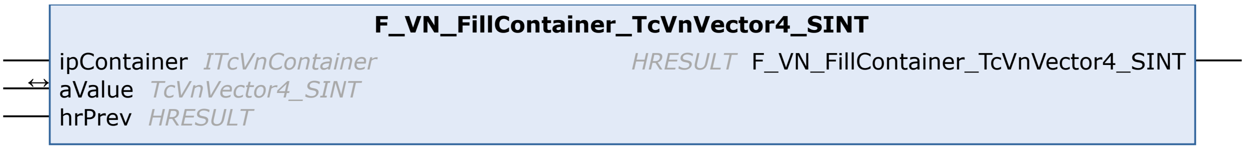
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.29 F_VN_FillContainer_TcVnVector4_SINT



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnVector4_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector4_SINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_SINT [▸ 141]	Value to set the container elements

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.30 F_VN_FillContainer_TcVnVector4_UINT

F_VN_FillContainer_TcVnVector4_UINT

ipContainer *ITcVnContainer* — *HRESULT* F_VN_FillContainer_TcVnVector4_UINT
aValue *TcVnVector4_UINT*
hrPrev *HRESULT*

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnVector4_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector4_UINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_UINT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.31 F_VN_FillContainer_TcVnVector4_USINT

F_VN_FillContainer_TcVnVector4_USINT

ipContainer *ITcVnContainer* — *HRESULT* F_VN_FillContainer_TcVnVector4_USINT
aValue *TcVnVector4_USINT*
hrPrev *HRESULT*

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_TcVnVector4_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector4_USINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_USINT [▸ 141]	Value to set the container elements

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.32 F_VN_FillContainer_UDINT

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_UDINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nValue      : UDINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
nValue	UDINT	Value to set the container elements
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.33 F_VN_FillContainer_UINT

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nValue      : UINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
nValue	UINT	Value to set the container elements
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

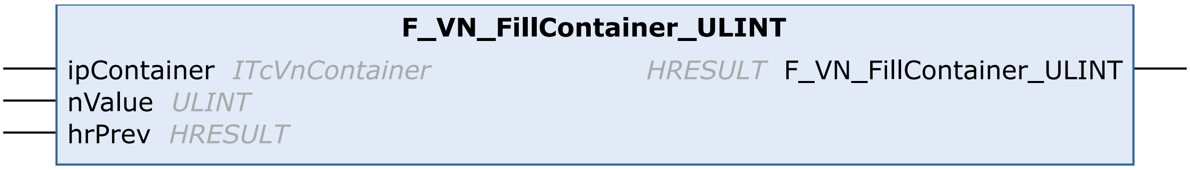
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.34 F_VN_FillContainer_ULINT



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_ULINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nValue      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nValue	ULINT	Value to set the container elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

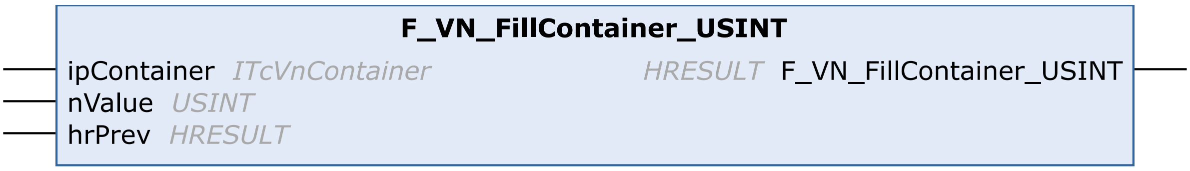
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.2.35 F_VN_FillContainer_USINT



Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_FillContainer_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nValue      : USINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nValue	USINT	Value to set the container elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3 F_VN_FillContainerExp

本章包含专家函数，根据数据类型，可用于用特定的值填充容器，即使只是部分填充。这个函数适用于包括基本元素的容器。

6.1.4.4.3.1 F_VN_FillContainerExp_DINT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nValue      : DINT;
    nStartIdx   : ULINT;
```

```
    nCount       : ULINT;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
nValue	DINT	Value to set the container elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.2 F_VN_FillContainerExp_INT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nValue      : INT;
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
nValue	INT	Value to set the container elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.3 F_VN_FillContainerExp_ITcVnImage

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_ITcVnImage : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    ipValue     : ITcVnImage;
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
ipValue	ITcVnImage [▶ 390]	Value to set the container elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.4 F_VN_FillContainerExp_LREAL

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    fValue      : LREAL;
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
fValue	LREAL	Value to set the container elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.5 F_VN_FillContainerExp_REAL

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    fValue      : REAL;
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
fValue	REAL	Value to set the container elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.6 F_VN_FillContainerExp_SINT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nValue      : SINT;
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nValue	SINT	Value to set the container elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.7 F_VN_FillContainerExp_TcVnDMatch

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnDMatch : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    stValue     : TcVnDMatch;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stValue	TcVnDMatch [▶ 210]	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.8 F_VN_FillContainerExp_TcVnKeyPoint

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnKeyPoint : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    stValue     : TcVnKeyPoint;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stValue	TcVnKeyPoint [▶ 210]	Value to set the container elements

Return value

HRESULT [▶ 122]

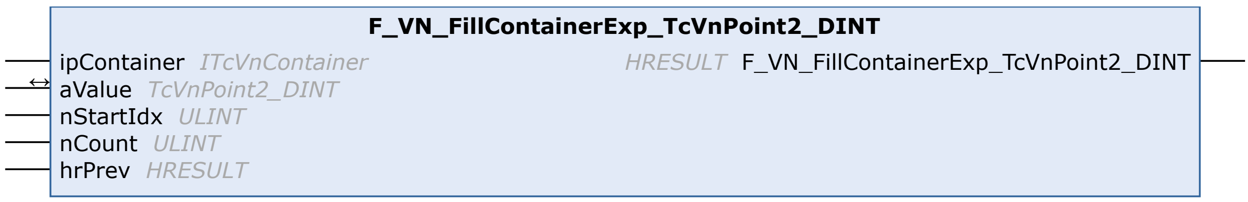
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.9 F_VN_FillContainerExp_TcVnPoint2_DINT



Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnPoint2_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnPoint2_DINT;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnPoint2_DINT [▶ 139]	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.10 F_VN_FillContainerExp_TcVnPoint2_LREAL

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnPoint2_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnPoint2_LREAL;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnPoint2_LREAL [▶ 139]	Value to set the container elements

Return value

HRESULT [▶ 122]

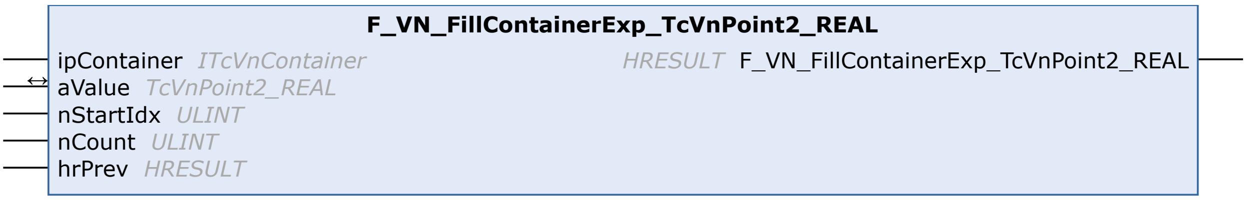
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.11 F_VN_FillContainerExp_TcVnPoint2_REAL



Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnPoint2_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnPoint2_REAL [▶ 139]	Value to set the container elements

Return value

HRESULT [▶ 122]

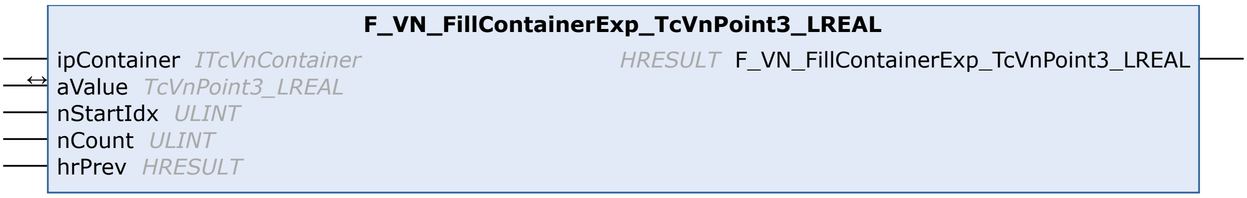
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.12 F_VN_FillContainerExp_TcVnPoint3_LREAL



Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnPoint3_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnPoint3_LREAL;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnPoint3_LREAL [▸ 139]	Value to set the container elements

Return value

HRESULT [▸ 122]

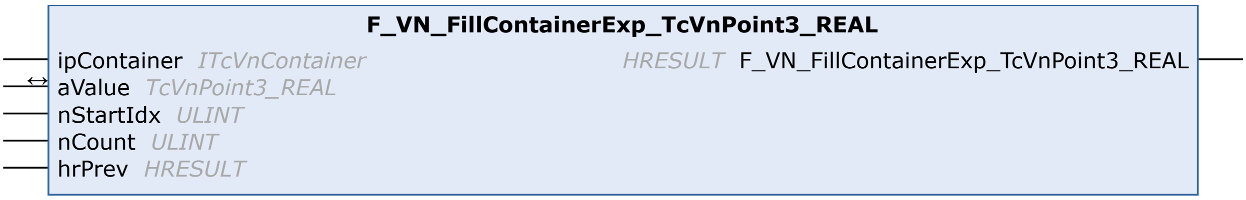
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.13 F_VN_FillContainerExp_TcVnPoint3_REAL



Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnPoint3_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnPoint3_REAL;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnPoint3_REAL [▶ 139]	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.14 F_VN_FillContainerExp_TcVnRectangle_DINT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnRectangle_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    stValue     : TcVnRectangle_DINT;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stValue	TcVnRectangle_DINT [▶ 224]	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.15 F_VN_FillContainerExp_TcVnVector2_DINT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnVector2_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector2_DINT;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_DINT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.16 F_VN_FillContainerExp_TcVnVector2_INT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnVector2_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector2_INT;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_INT [▸ 141]	Value to set the container elements

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.17 F_VN_FillContainerExp_TcVnVector2_REAL

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnVector2_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector2_REAL;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_REAL [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.18 F_VN_FillContainerExp_TcVnVector2_SINT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnVector2_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector2_SINT;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_SINT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.19 F_VN_FillContainerExp_TcVnVector2_UINT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnVector2_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector2_UINT;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_UINT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

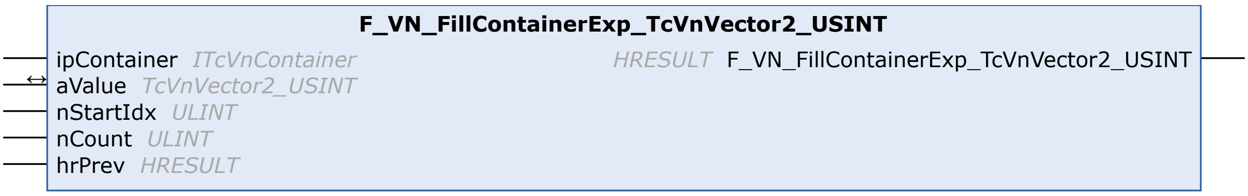
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.20 F_VN_FillContainerExp_TcVnVector2_USINT



Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnVector2_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector2_USINT;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector2_USINT [▸ 141]	Value to set the container elements

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.21 F_VN_FillContainerExp_TcVnVector3_INT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnVector3_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector3_INT;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector3_INT [▸ 141]	Value to set the container elements

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.22 F_VN_FillContainerExp_TcVnVector3_REAL

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnVector3_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector3_REAL;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector3_REAL [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.23 F_VN_FillContainerExp_TcVnVector3_SINT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnVector3_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector3_SINT;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector3_SINT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.24 F_VN_FillContainerExp_TcVnVector3_UINT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnVector3_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector3_UINT;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector3_UINT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

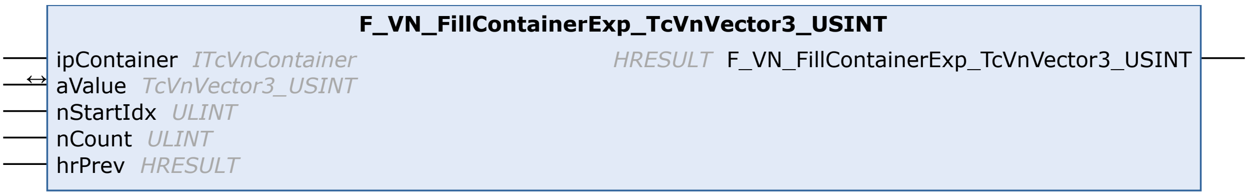
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.25 F_VN_FillContainerExp_TcVnVector3_USINT



Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnVector3_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector3_USINT;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector3_USINT [▸ 141]	Value to set the container elements

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.26 F_VN_FillContainerExp_TcVnVector4_DINT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnVector4_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector4_DINT;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_DINT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.27 F_VN_FillContainerExp_TcVnVector4_INT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnVector4_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector4_INT;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_INT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

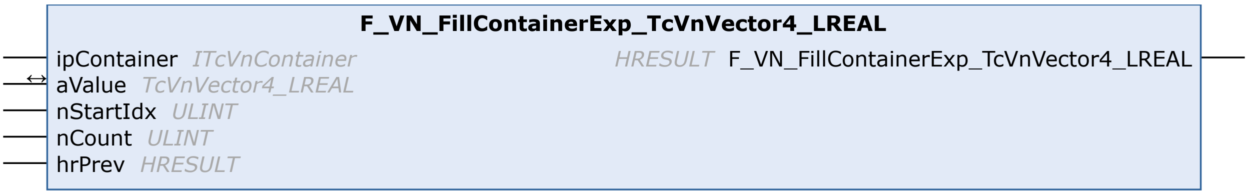
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.28 F_VN_FillContainerExp_TcVnVector4_LREAL



Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnVector4_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_LREAL [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.29 F_VN_FillContainerExp_TcVnVector4_SINT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnVector4_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector4_SINT;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_SINT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.30 F_VN_FillContainerExp_TcVnVector4_UINT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnVector4_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector4_UINT;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_UINT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.31 F_VN_FillContainerExp_TcVnVector4_USINT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_TcVnVector4_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector4_USINT;
END_VAR
VAR_INPUT
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_USINT [▶ 141]	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.32 F_VN_FillContainerExp_UDINT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_UDINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nValue      : UDINT;
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nValue	UDINT	Value to set the container elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.33 F_VN_FillContainerExp_UINT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nValue      : UINT;
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nValue	UINT	Value to set the container elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.34 F_VN_FillContainerExp_ULINT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_ULINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nValue      : ULINT;
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
nValue	ULINT	Value to set the container elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.3.35 F_VN_FillContainerExp_USINT

Partially fill the container with the specified value. Only possible for containers with basic elements. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FillContainerExp_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nValue      : USINT;
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nValue	USINT	Value to set the container elements
nStartIdx	ULINT	Start index
nCount	ULINT	Number of elements to set
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

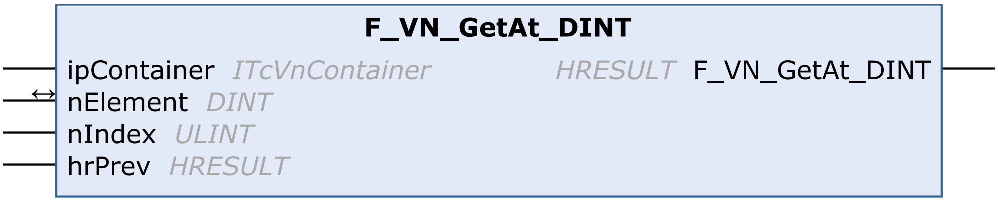
System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4 F_VN_GetAt

如果需要访问一个容器中的单个元素，应该使用F_VN_GetAt函数。如果需要连续处理容器中的所有元素，使用迭代器 [▶ 738]更好。

6.1.4.4.4.1 F_VN_GetAt_DINT



Gets the DINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nElement    : DINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with DINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nElement	DINT	Returns the element at the specified index

Return value

HRESULT [▶ 122]

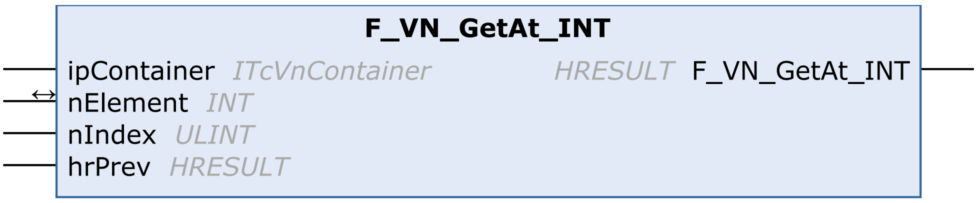
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.2 F_VN_GetAt_INT



Gets the INT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nElement    : INT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with INT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nElement	INT	Returns the element at the specified index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.3 F_VN_GetAt_ITcVnContainer

Gets the ITcVnContainer element at the specified index of the source container. (Alternatively use interface method .GetAt.)

Syntax

Definition:

```
FUNCTION F_VN_GetAt_ITcVnContainer : HRESULT
VAR_INPUT
    ipSrcContainer  : ITcVnContainer;
    ipDestContainer : Reference To ITcVnContainer;
    nIndex          : ULINT;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer	ITcVnContainer [▸ 349]	Container with ITcVnContainer elements
ipDestContainer	Reference To ITcVnContainer [▸ 349]	Returns the container at the specified index
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.4 F_VN_GetAt_ITcVnImage

Gets the ITcVnImage element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_ITcVnImage : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    ipElement   : Reference To ITcVnImage;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with ITcVnImage elements
ipElement	Reference To ITcVnImage [▸ 390]	Returns the element at the specified index
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

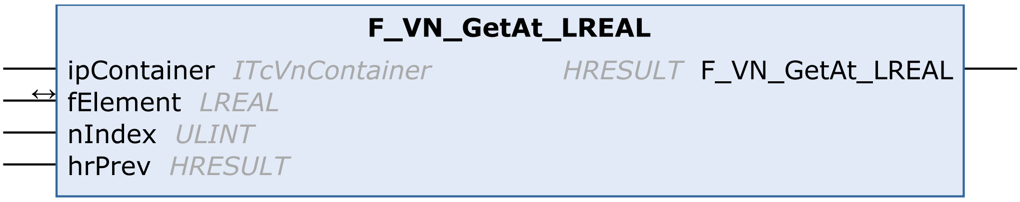
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.5 F_VN_GetAt_LREAL



Gets the LREAL element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fElement    : LREAL;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with LREAL elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fElement	LREAL	Returns the element at the specified index

Return value

HRESULT [▶ 122]

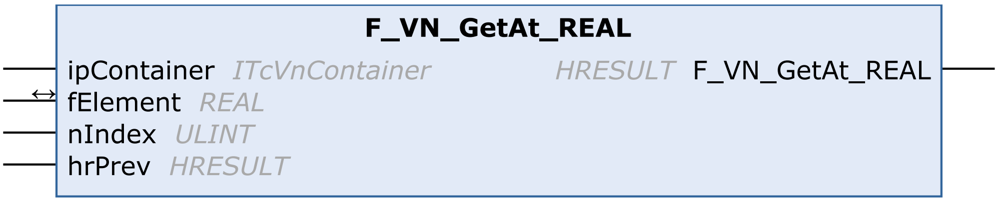
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.6 F_VN_GetAt_REAL



Gets the REAL element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fElement    : REAL;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with REAL elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fElement	REAL	Returns the element at the specified index

Return value

HRESULT [▶ 122]

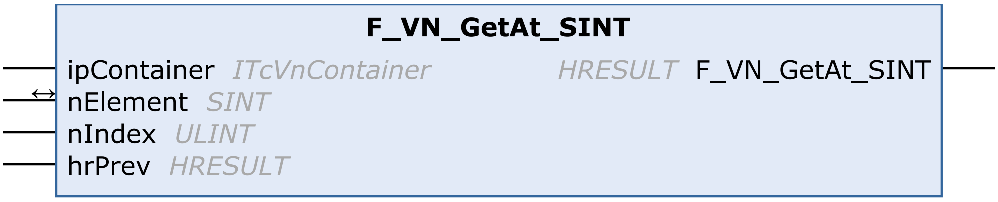
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.7 F_VN_GetAt_SINT



Gets the SINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nElement    : SINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with SINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nElement	SINT	Returns the element at the specified index

Return value

HRESULT [▶ 122]

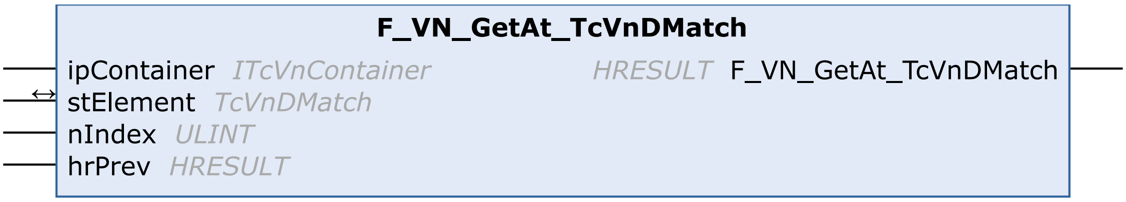
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.8 F_VN_GetAt_TcVnDMatch



Gets the TcVnDMatch element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnDMatch : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    stElement   : TcVnDMatch;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnDMatch elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stElement	TcVnDMatch [▸ 210]	Returns the element at the specified index

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.9 F_VN_GetAt_TcVnKeyPoint

Gets the TcVnKeyPoint element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnKeyPoint : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    stElement   : TcVnKeyPoint;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnKeyPoint elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stElement	TcVnKeyPoint [▶ 210]	Returns the element at the specified index

Return value

HRESULT [▶ 122]

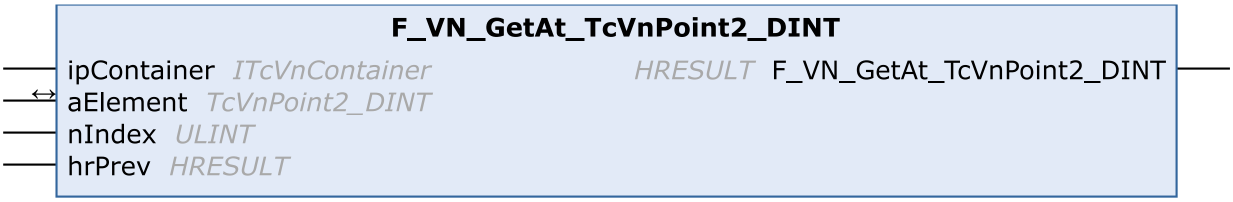
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.10 F_VN_GetAt_TcVnPoint2_DINT



Gets the TcVnPoint2_DINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnPoint2_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnPoint2_DINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnPoint2_DINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint2_DINT [▸ 139]	Returns the element at the specified index

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.11 F_VN_GetAt_TcVnPoint2_LREAL

Gets the TcVnPoint2_LREAL element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnPoint2_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnPoint2_LREAL;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnPoint2_LREAL elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint2_LREAL [▶ 139]	Returns the element at the specified index

Return value

HRESULT [▶ 122]

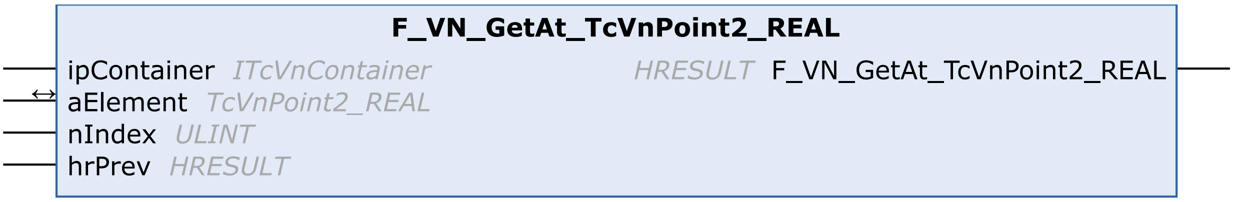
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.12 F_VN_GetAt_TcVnPoint2_REAL



Gets the TcVnPoint2_REAL element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnPoint2_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnPoint2_REAL elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint2_REAL [▸ 139]	Returns the element at the specified index

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.13 F_VN_GetAt_TcVnPoint3_LREAL

Gets the TcVnPoint3_LREAL element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnPoint3_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnPoint3_LREAL;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnPoint3_LREAL elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint3_LREAL [▶ 139]	Returns the element at the specified index

Return value

HRESULT [▶ 122]

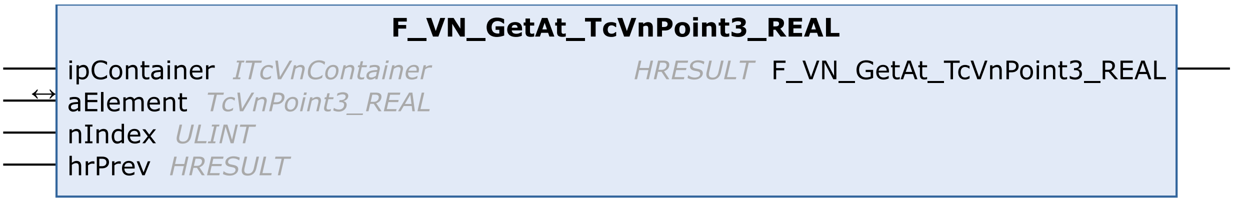
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.14 F_VN_GetAt_TcVnPoint3_REAL



Gets the TcVnPoint3_REAL element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnPoint3_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnPoint3_REAL;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnPoint3_REAL elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint3_REAL [▶ 139]	Returns the element at the specified index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.15 F_VN_GetAt_TcVnRectangle_DINT

F_VN_GetAt_TcVnRectangle_DINT
ipContainer *ITcVnContainer* — *HRESULT* F_VN_GetAt_TcVnRectangle_DINT
stElement *TcVnRectangle_DINT*
nIndex *ULINT*
hrPrev *HRESULT*

Gets the TcVnRectangle_DINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnRectangle_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    stElement   : TcVnRectangle_DINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnRectangle_DINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stElement	TcVnRectangle_DINT [▸ 224]	Returns the element at the specified index

Return value

HRESULT [▸ 122]

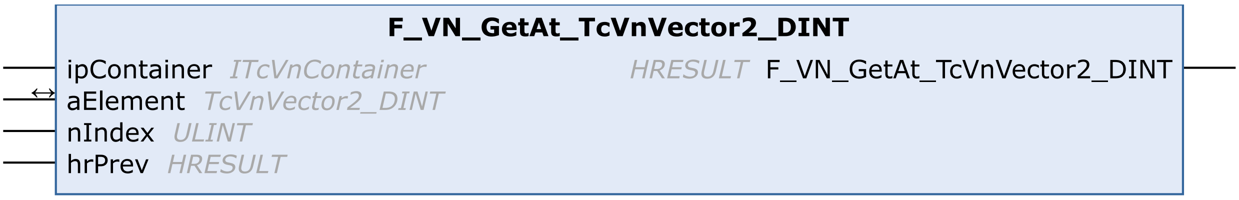
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.16 F_VN_GetAt_TcVnVector2_DINT



Gets the TcVnVector2_DINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnVector2_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnVector2_DINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector2_DINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_DINT [▶ 141]	Returns the element at the specified index

Return value

HRESULT [▶ 122]

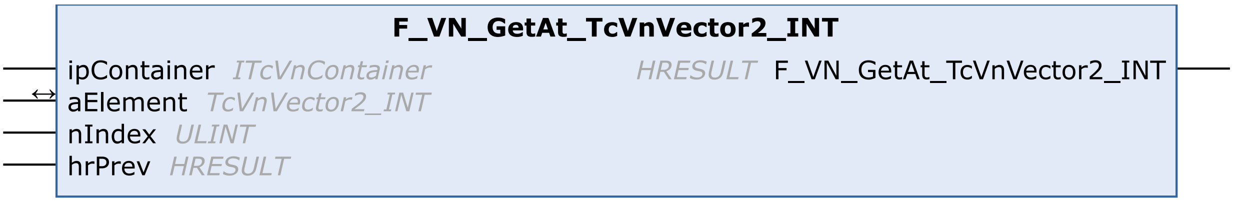
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.17 F_VN_GetAt_TcVnVector2_INT



Gets the TcVnVector2_INT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnVector2_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnVector2_INT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector2_INT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_INT [▶ 141]	Returns the element at the specified index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.18 F_VN_GetAt_TcVnVector2_REAL

Gets the TcVnVector2_REAL element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnVector2_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnVector2_REAL;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnVector2_REAL elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_REAL [▸ 141]	Returns the element at the specified index

Return value

HRESULT [▸ 122]

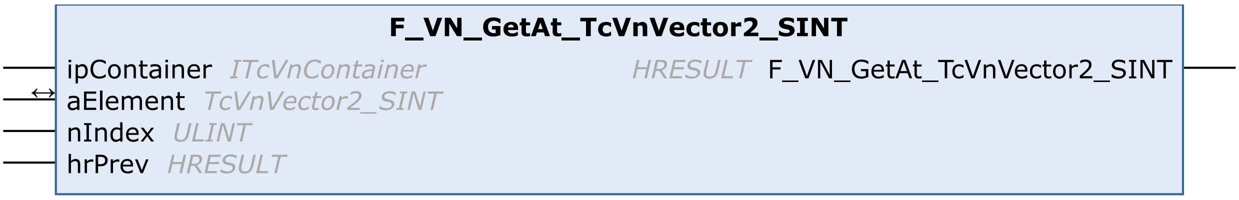
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.19 F_VN_GetAt_TcVnVector2_SINT



Gets the TcVnVector2_SINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnVector2_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnVector2_SINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnVector2_SINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_SINT [▸ 141]	Returns the element at the specified index

Return value

HRESULT [▸ 122]

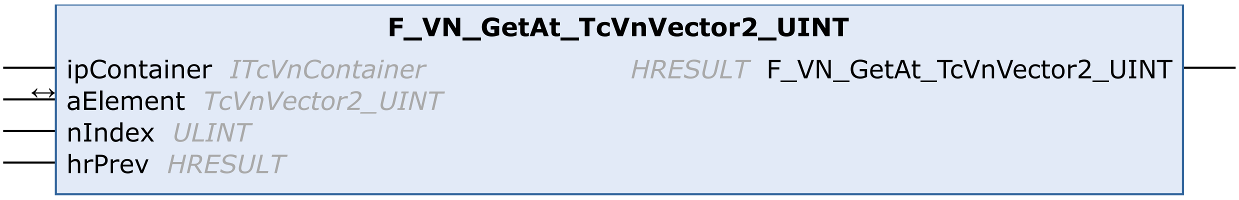
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.20 F_VN_GetAt_TcVnVector2_UINT



Gets the TcVnVector2_UINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnVector2_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnVector2_UINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector2_UINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_UINT [▶ 141]	Returns the element at the specified index

Return value

HRESULT [▶ 122]

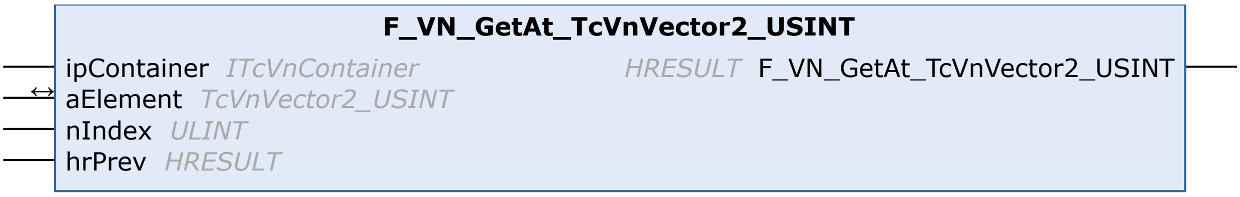
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.21 F_VN_GetAt_TcVnVector2_USINT



Gets the TcVnVector2_USINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnVector2_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnVector2_USINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnVector2_USINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_USINT [▸ 141]	Returns the element at the specified index

Return value

HRESULT [▸ 122]

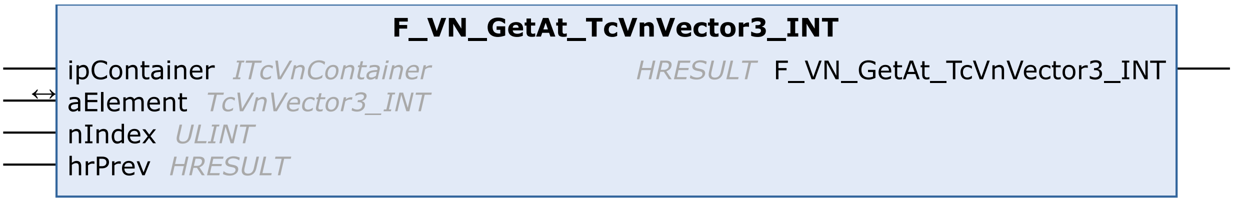
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.22 F_VN_GetAt_TcVnVector3_INT



Gets the TcVnVector3_INT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnVector3_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnVector3_INT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector3_INT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_INT [▶ 141]	Returns the element at the specified index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.23 F_VN_GetAt_TcVnVector3_REAL

Gets the TcVnVector3_REAL element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnVector3_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnVector3_REAL;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector3_REAL elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_REAL [▶ 141]	Returns the element at the specified index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.24 F_VN_GetAt_TcVnVector3_SINT

F_VN_GetAt_TcVnVector3_SINT	
ipContainer *ITcVnContainer*	*HRESULT* F_VN_GetAt_TcVnVector3_SINT
↔ aElement *TcVnVector3_SINT*	
nIndex *ULINT*	
hrPrev *HRESULT*	

Gets the TcVnVector3_SINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnVector3_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnVector3_SINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnVector3_SINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_SINT [▸ 141]	Returns the element at the specified index

Return value

HRESULT [▸ 122]

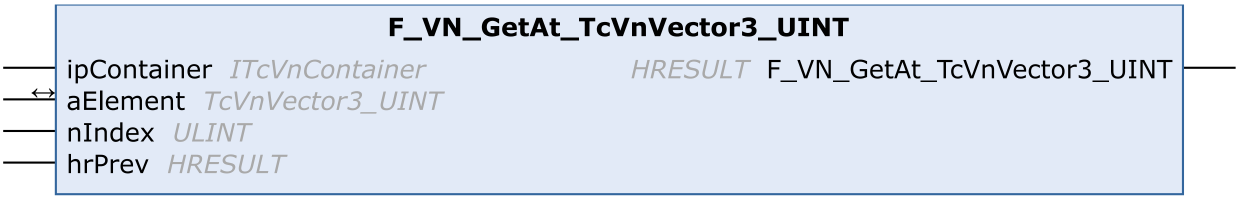
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.25 F_VN_GetAt_TcVnVector3_UINT



Gets the TcVnVector3_UINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnVector3_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnVector3_UINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnVector3_UINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_UINT [▸ 141]	Returns the element at the specified index

Return value

HRESULT [▸ 122]

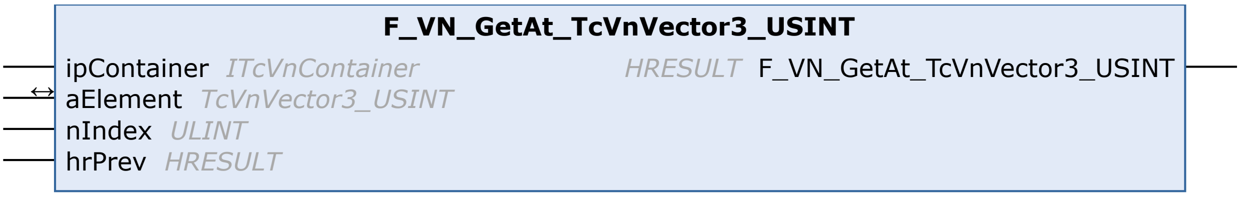
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.26 F_VN_GetAt_TcVnVector3_USINT



Gets the TcVnVector3_USINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnVector3_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnVector3_USINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector3_USINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_USINT [▶ 141]	Returns the element at the specified index

Return value

HRESULT [▶ 122]

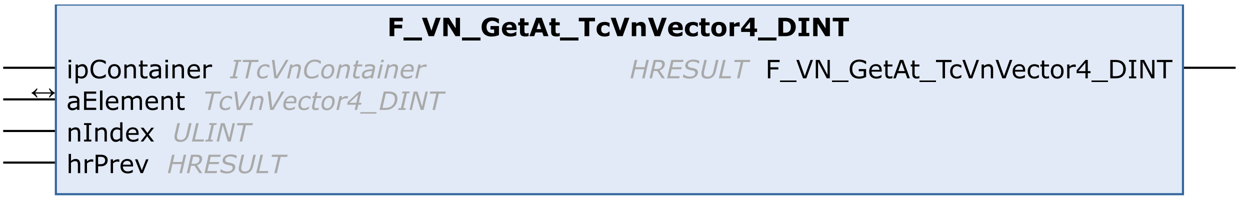
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.27 F_VN_GetAt_TcVnVector4_DINT



Gets the TcVnVector4_DINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnVector4_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnVector4_DINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector4_DINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_DINT [▶ 141]	Returns the element at the specified index

Return value

HRESULT [▶ 122]

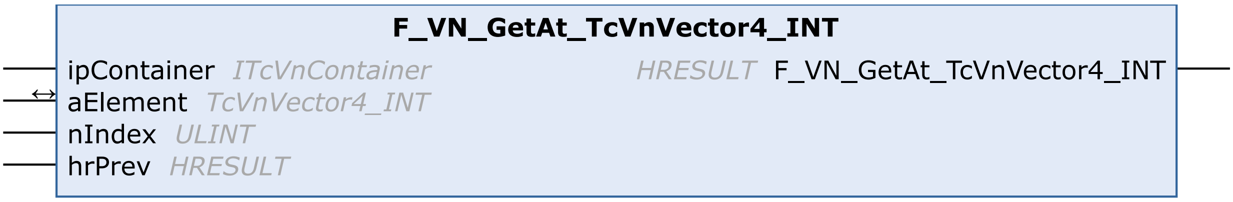
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.28 F_VN_GetAt_TcVnVector4_INT



Gets the TcVnVector4_INT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnVector4_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnVector4_INT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector4_INT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_INT [▶ 141]	Returns the element at the specified index

Return value

HRESULT [▶ 122]

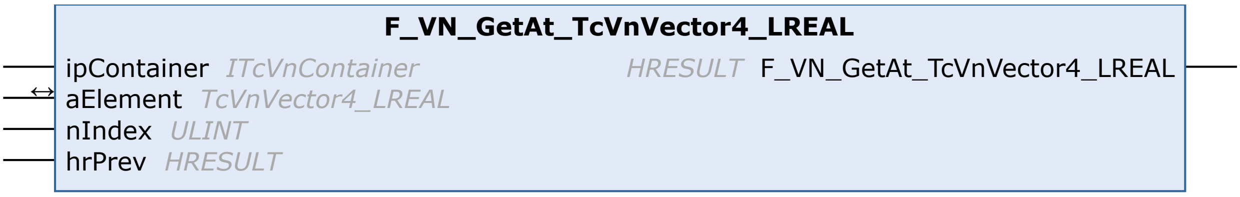
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.29 F_VN_GetAt_TcVnVector4_LREAL



Gets the TcVnVector4_LREAL element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnVector4_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector4_LREAL elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_LREAL [▶ 141]	Returns the element at the specified index

Return value

HRESULT [▶ 122]

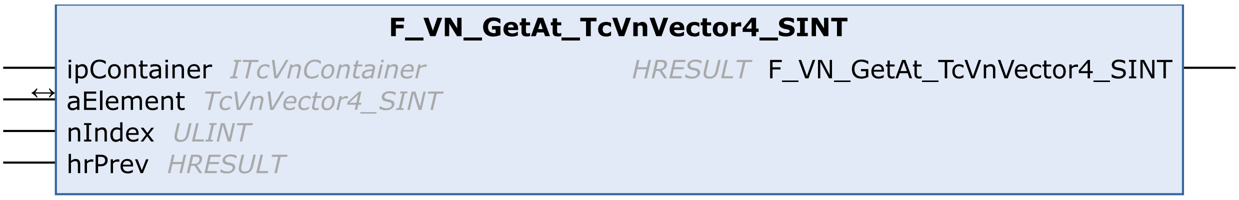
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.30 F_VN_GetAt_TcVnVector4_SINT



Gets the TcVnVector4_SINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnVector4_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnVector4_SINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnVector4_SINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_SINT [▸ 141]	Returns the element at the specified index

Return value

HRESULT [▸ 122]

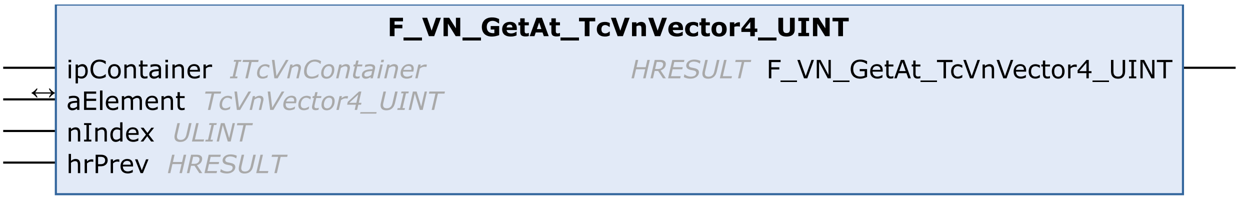
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.31 F_VN_GetAt_TcVnVector4_UINT



Gets the TcVnVector4_UINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnVector4_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnVector4_UINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector4_UINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_UINT [▶ 141]	Returns the element at the specified index

Return value

HRESULT [▶ 122]

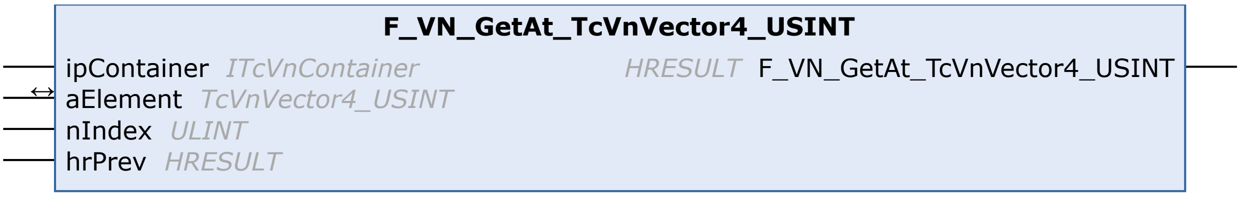
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.32 F_VN_GetAt_TcVnVector4_USINT



Gets the TcVnVector4_USINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_TcVnVector4_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aElement    : TcVnVector4_USINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector4_USINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_USINT [▶ 141]	Returns the element at the specified index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.33 F_VN_GetAt_UDINT

Gets the UDINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_UDINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nElement    : UDINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with UDINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nElement	UDINT	Returns the element at the specified index

Return value

HRESULT [▸ 122]

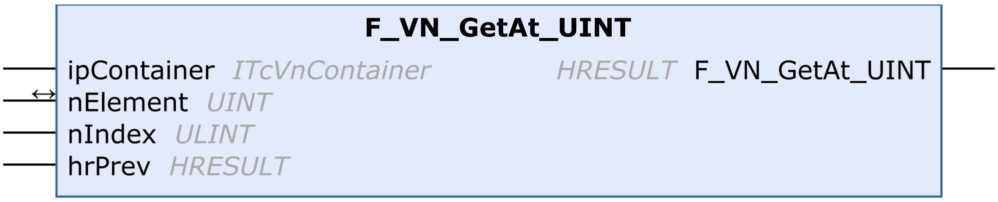
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.34 F_VN_GetAt_UINT



Gets the UINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nElement    : UINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with UINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nElement	UINT	Returns the element at the specified index

Return value

HRESULT [▸ 122]

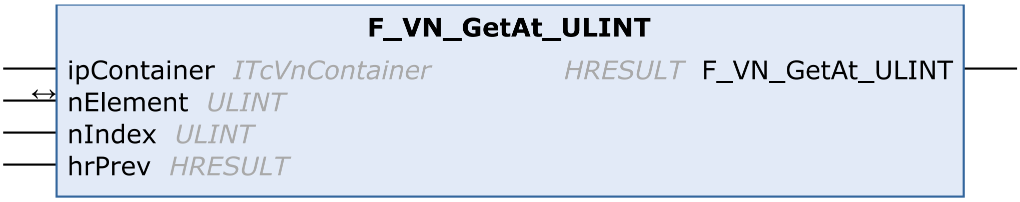
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.35 F_VN_GetAt_ULINT



Gets the ULINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_ULINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nElement    : ULINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with ULINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nElement	ULINT	Returns the element at the specified index

Return value

HRESULT [▸ 122]

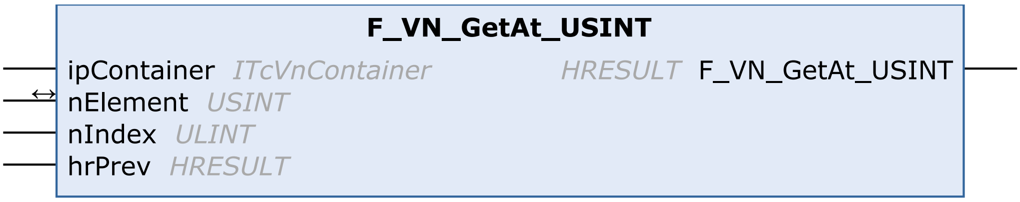
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.4.36 F_VN_GetAt_USINT



Gets the USINT element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_GetAt_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nElement    : USINT;
END_VAR
VAR_INPUT
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with USINT elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nElement	USINT	Returns the element at the specified index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5 F_VN_InsertIntoContainer

本章包含根据数据类型在特定的容器位置插入一个元素的功能。

6.1.4.4.5.1 F_VN_InsertIntoContainer_DINT

F_VN_InsertIntoContainer_DINT

nElement *DINT* — *HRESULT* F_VN_InsertIntoContainer_DINT
ipContainer *ITcVnContainer*
nIndex *ULINT*
hrPrev *HRESULT*

Insert an element of type DINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_DINT : HRESULT
VAR_INPUT
    nElement    : DINT;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	DINT	Single element to insert into ipContainer
ipContainer	ITcVnContainer [▸ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.2 F_VN_InsertIntoContainer_INT

F_VN_InsertIntoContainer_INT

nElement *INT* — *HRESULT* F_VN_InsertIntoContainer_INT
ipContainer *ITcVnContainer*
nIndex *ULINT*
hrPrev *HRESULT*

Insert an element of type INT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_INT : HRESULT
VAR_INPUT
    nElement    : INT;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	INT	Single element to insert into ipContainer
ipContainer	ITcVnContainer [▸ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

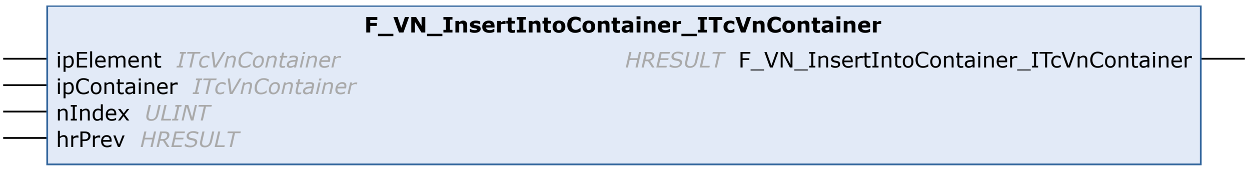
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.3 F_VN_InsertIntoContainer_ITcVnContainer



Insert an element (or a range of elements) into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_ITcVnContainer : HRESULT
VAR_INPUT
    ipElement   : ITcVnContainer;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipElement	ITcVnContainer [▶ 349]	Single element to insert into ipContainer or container with several elements
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

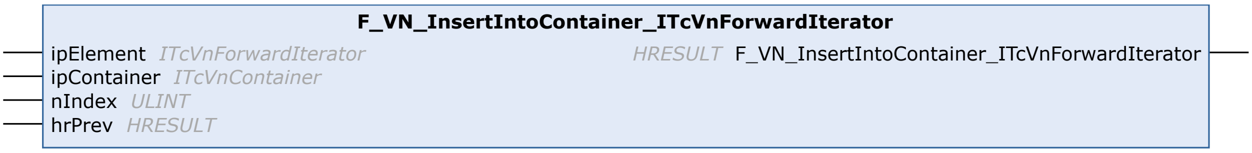
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.4 F_VN_InsertIntoContainer_ITcVnForwardIterator



Insert an element (represented by an iterator) into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_ITcVnForwardIterator : HRESULT
VAR_INPUT
    ipElement   : ITcVnForwardIterator;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipElement	ITcVnForwardIterator [▶ 339]	Single element to insert into ipContainer
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

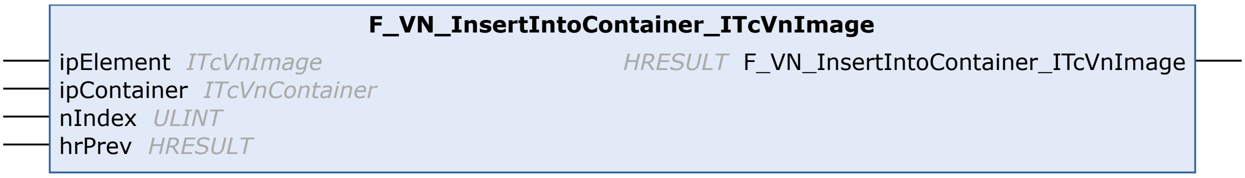
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.5 F_VN_InsertIntoContainer_ITcVnImage



Insert an element of type ITcVnImage into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_ITcVnImage : HRESULT
VAR_INPUT
    ipElement   : ITcVnImage;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipElement	ITcVnImage [▶ 390]	Single element to insert into ipContainer
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

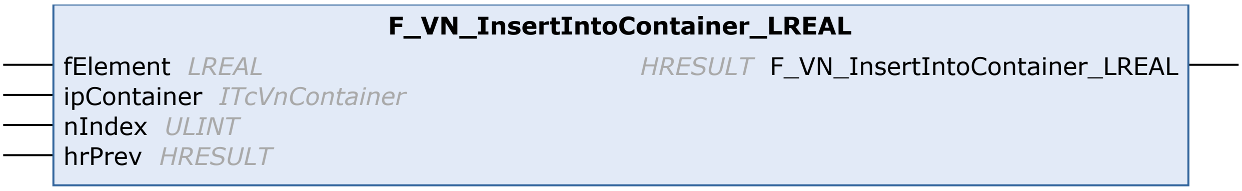
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.6 F_VN_InsertIntoContainer_LREAL



Insert an element of type LREAL into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_LREAL : HRESULT
VAR_INPUT
    fElement    : LREAL;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fElement	LREAL	Single element to insert into ipContainer
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.7 F_VN_InsertIntoContainer_REAL

Insert an element of type REAL into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_REAL : HRESULT
VAR_INPUT
    fElement     : REAL;
    ipContainer  : ITcVnContainer;
    nIndex       : ULINT;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fElement	REAL	Single element to insert into ipContainer
ipContainer	ITcVnContainer [▸ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.8 F_VN_InsertIntoContainer_SINT

Insert an element of type SINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_SINT : HRESULT
VAR_INPUT
    nElement     : SINT;
    ipContainer  : ITcVnContainer;
    nIndex       : ULINT;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	SINT	Single element to insert into ipContainer
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

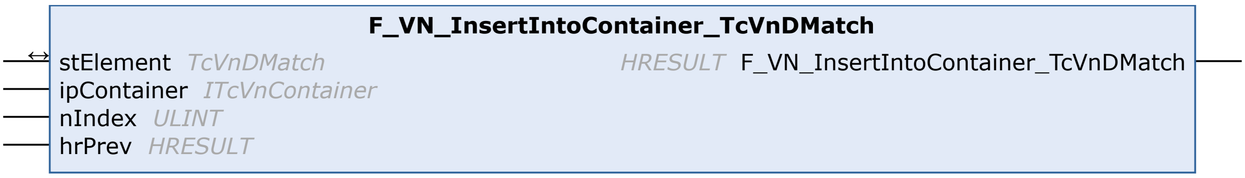
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.9 F_VN_InsertIntoContainer_TcVnDMatch



Insert an element of type TcVnDMatch into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnDMatch : HRESULT
VAR_IN_OUT
    stElement   : TcVnDMatch;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stElement	TcVnDMatch [▸ 210]	Single element to insert into ipContainer

Return value

HRESULT [▸ 122]

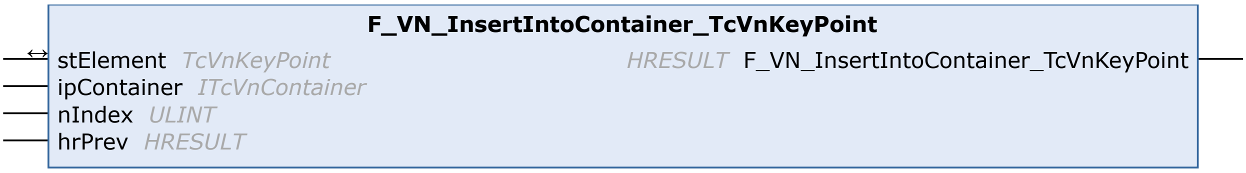
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.10 F_VN_InsertIntoContainer_TcVnKeyPoint



Insert an element of type TcVnKeyPoint into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnKeyPoint : HRESULT
VAR_IN_OUT
    stElement   : TcVnKeyPoint;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stElement	TcVnKeyPoint [▸ 210]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

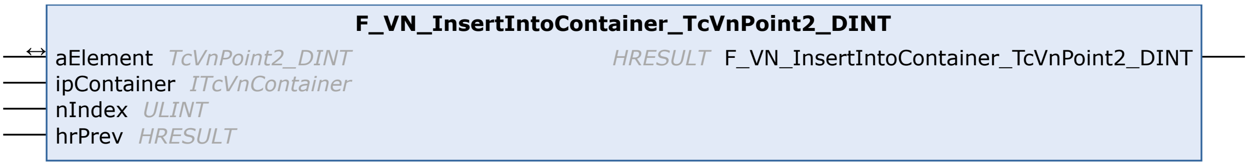
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.11 F_VN_InsertIntoContainer_TcVnPoint2_DINT



Insert an element of type TcVnPoint2_DINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnPoint2_DINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnPoint2_DINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint2_DINT [▶ 139]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

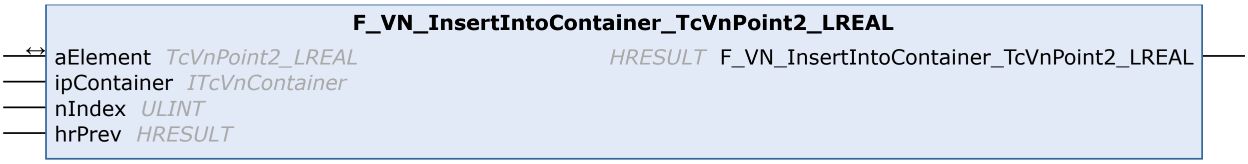
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.12 F_VN_InsertIntoContainer_TcVnPoint2_LREAL



Insert an element of type TcVnPoint2_LREAL into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnPoint2_LREAL : HRESULT
VAR_IN_OUT
    aElement    : TcVnPoint2_LREAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint2_LREAL [▶ 139]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

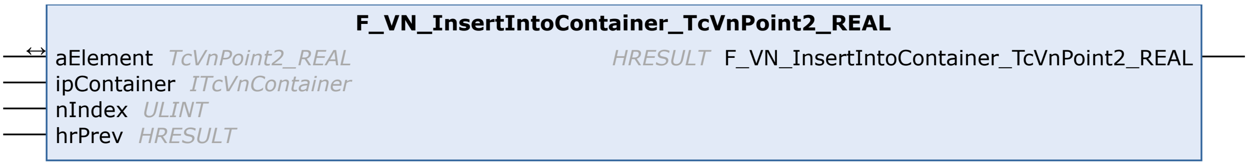
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.13 F_VN_InsertIntoContainer_TcVnPoint2_REAL



Insert an element of type TcVnPoint2_REAL into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnPoint2_REAL : HRESULT
VAR_IN_OUT
    aElement    : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint2_REAL [▶ 139]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

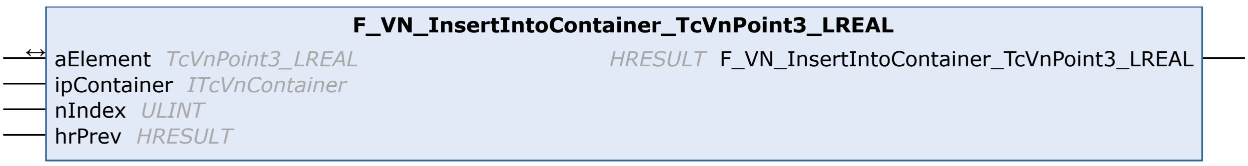
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.14 F_VN_InsertIntoContainer_TcVnPoint3_LREAL



Insert an element of type TcVnPoint3_LREAL into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnPoint3_LREAL : HRESULT
VAR_IN_OUT
    aElement     : TcVnPoint3_LREAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint3_LREAL [▶ 139]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

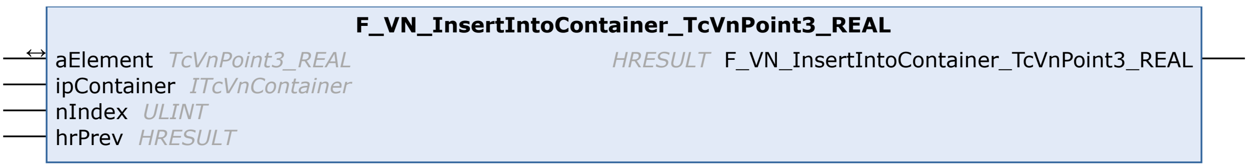
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.15 F_VN_InsertIntoContainer_TcVnPoint3_REAL



Insert an element of type TcVnPoint3_REAL into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnPoint3_REAL : HRESULT
VAR_IN_OUT
    aElement     : TcVnPoint3_REAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
```

```
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint3_REAL [▶ 139]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

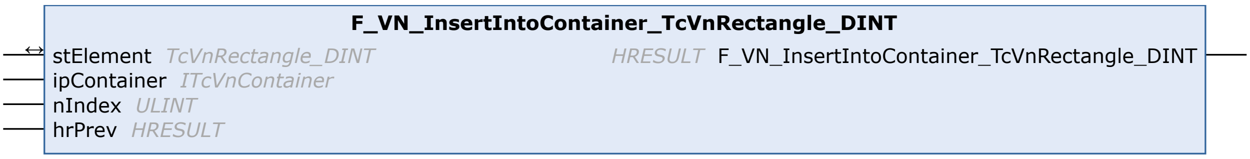
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.16 F_VN_InsertIntoContainer_TcVnRectangle_DINT



Insert an element of type TcVnRectangle_DINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnRectangle_DINT : HRESULT
VAR_IN_OUT
    stElement   : TcVnRectangle_DINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stElement	TcVnRectangle_DINT [▶ 224]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

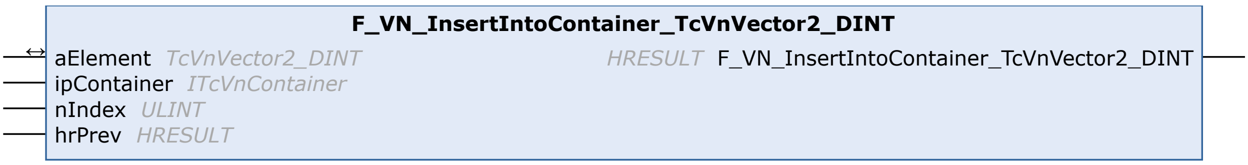
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.17 F_VN_InsertIntoContainer_TcVnVector2_DINT



Insert an element of type TcVnVector2_DINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnVector2_DINT : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector2_DINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_DINT [▶ 141]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

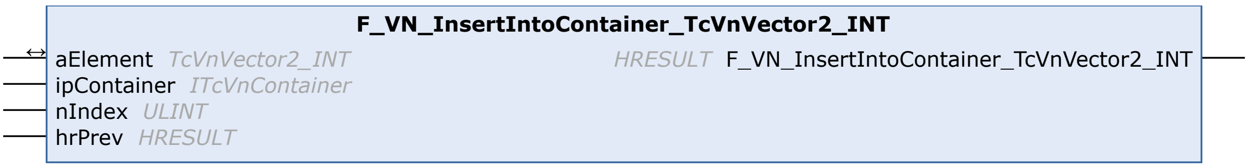
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.18 F_VN_InsertIntoContainer_TcVnVector2_INT



Insert an element of type TcVnVector2_INT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnVector2_INT : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector2_INT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_INT [▶ 141]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

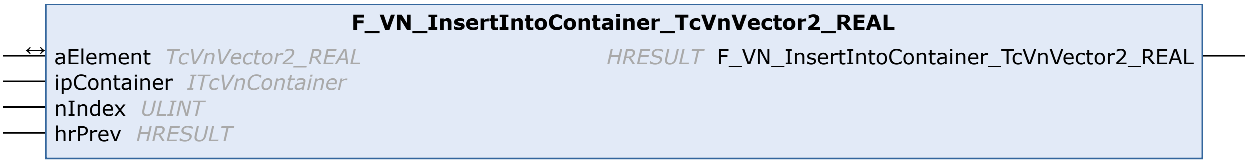
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.19 F_VN_InsertIntoContainer_TcVnVector2_REAL



Insert an element of type TcVnVector2_REAL into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnVector2_REAL : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector2_REAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_REAL [▶ 141]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

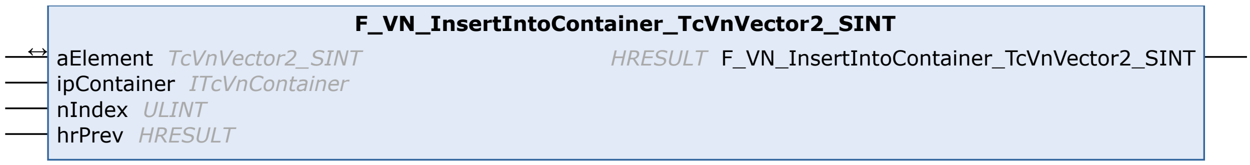
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.20 F_VN_InsertIntoContainer_TcVnVector2_SINT



Insert an element of type TcVnVector2_SINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnVector2_SINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector2_SINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_SINT [▶ 141]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

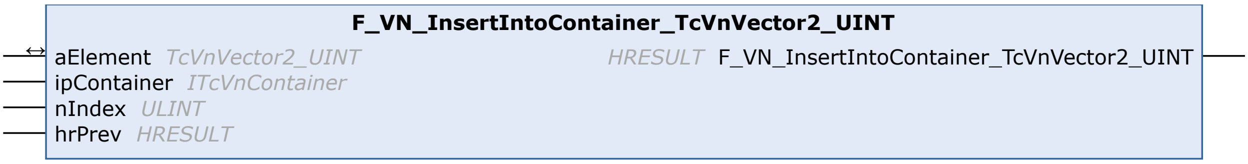
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.21 F_VN_InsertIntoContainer_TcVnVector2_UINT



Insert an element of type TcVnVector2_UINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnVector2_UINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector2_UINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_UINT [▶ 141]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

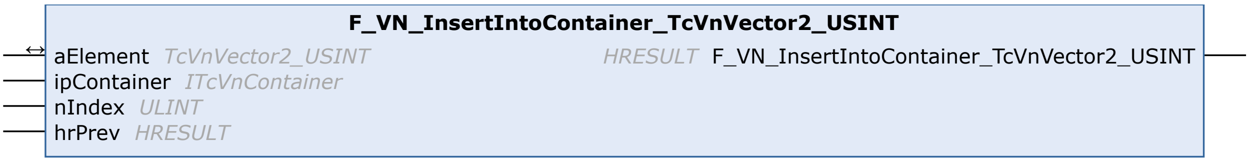
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.22 F_VN_InsertIntoContainer_TcVnVector2_USINT



Insert an element of type TcVnVector2_USINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnVector2_USINT : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector2_USINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_USINT [▶ 141]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

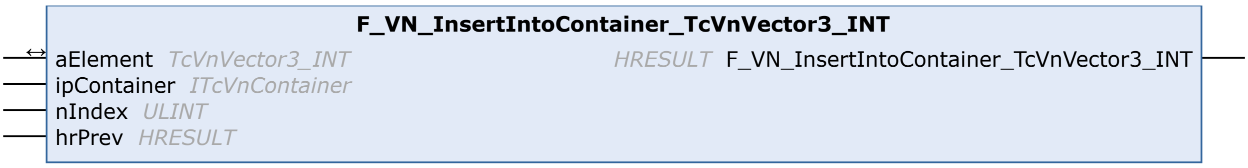
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.23 F_VN_InsertIntoContainer_TcVnVector3_INT



Insert an element of type TcVnVector3_INT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnVector3_INT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector3_INT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_INT [▶ 141]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

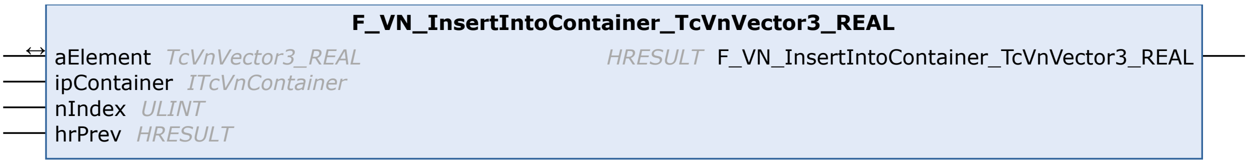
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.24 F_VN_InsertIntoContainer_TcVnVector3_REAL



Insert an element of type TcVnVector3_REAL into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnVector3_REAL : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector3_REAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_REAL [▶ 141]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

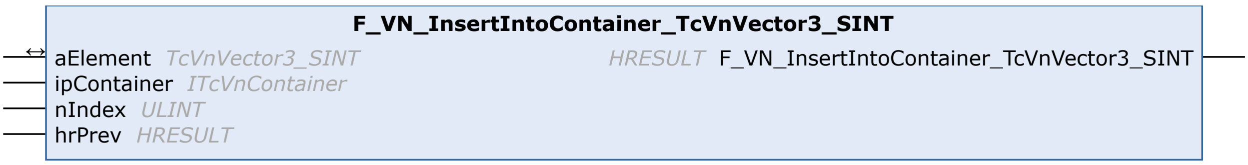
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.25 F_VN_InsertIntoContainer_TcVnVector3_SINT



Insert an element of type TcVnVector3_SINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnVector3_SINT : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector3_SINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_SINT [▶ 141]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

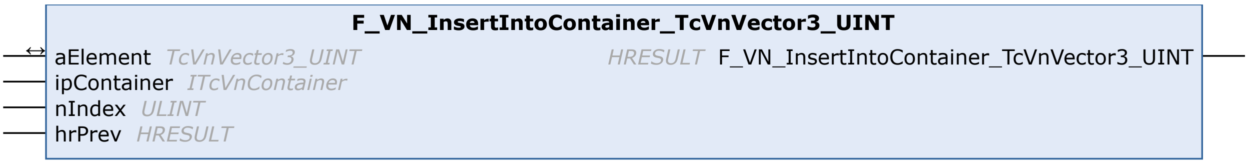
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.26 F_VN_InsertIntoContainer_TcVnVector3_UINT



Insert an element of type TcVnVector3_UINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnVector3_UINT : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector3_UINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_UINT [▶ 141]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

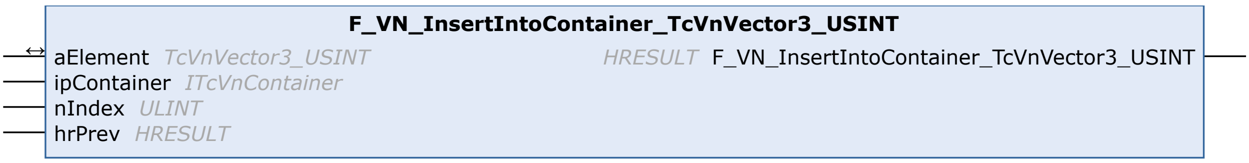
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.27 F_VN_InsertIntoContainer_TcVnVector3_USINT



Insert an element of type TcVnVector3_USINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnVector3_USINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector3_USINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_USINT [▶ 141]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

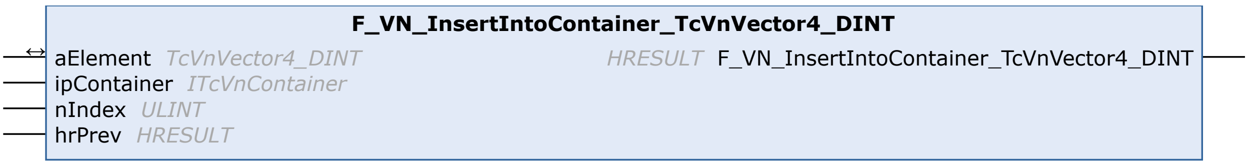
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.28 F_VN_InsertIntoContainer_TcVnVector4_DINT



Insert an element of type TcVnVector4_DINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnVector4_DINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector4_DINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_DINT [▸ 141]	Single element to insert into ipContainer

Return value

HRESULT [▸ 122]

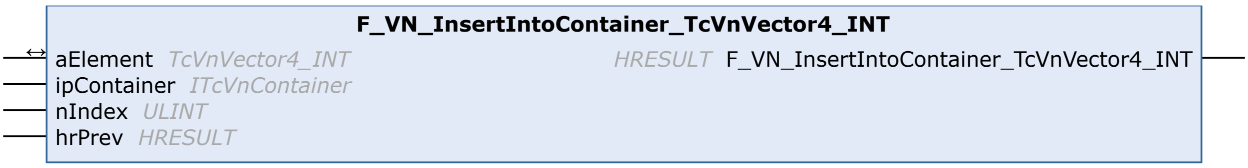
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.29 F_VN_InsertIntoContainer_TcVnVector4_INT



Insert an element of type TcVnVector4_INT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnVector4_INT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector4_INT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_INT [▶ 141]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

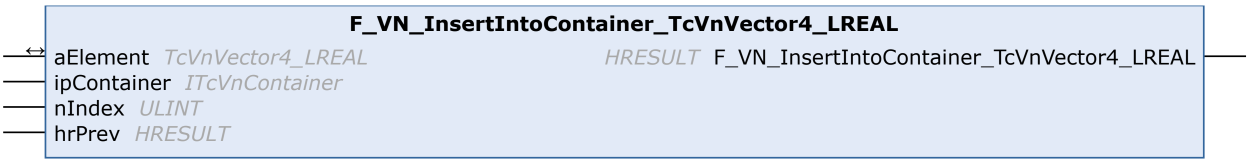
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.30 F_VN_InsertIntoContainer_TcVnVector4_LREAL



Insert an element of type TcVnVector4_LREAL into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnVector4_LREAL : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_LREAL [▸ 141]	Single element to insert into ipContainer

Return value

HRESULT [▸ 122]

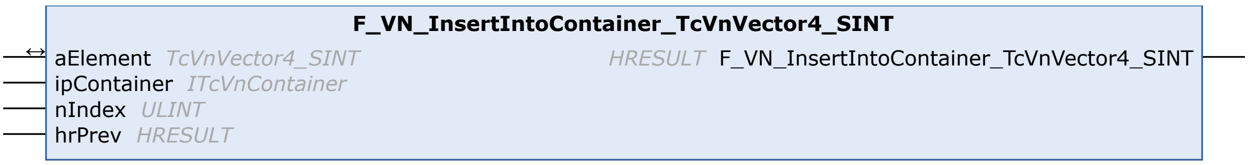
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.31 F_VN_InsertIntoContainer_TcVnVector4_SINT



Insert an element of type TcVnVector4_SINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnVector4_SINT : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector4_SINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_SINT [▶ 141]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

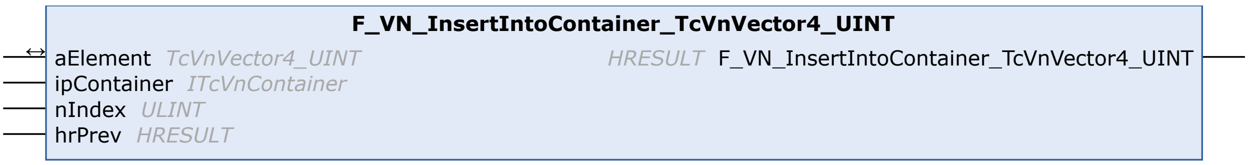
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.32 F_VN_InsertIntoContainer_TcVnVector4_UINT



Insert an element of type TcVnVector4_UINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnVector4_UINT : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector4_UINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_UINT [▶ 141]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

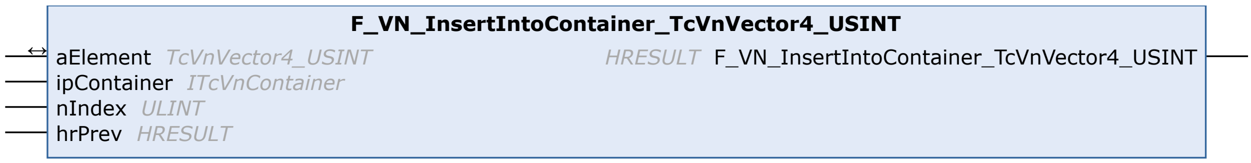
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.33 F_VN_InsertIntoContainer_TcVnVector4_USINT



Insert an element of type TcVnVector4_USINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_TcVnVector4_USINT : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector4_USINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_USINT [▶ 141]	Single element to insert into ipContainer

Return value

HRESULT [▶ 122]

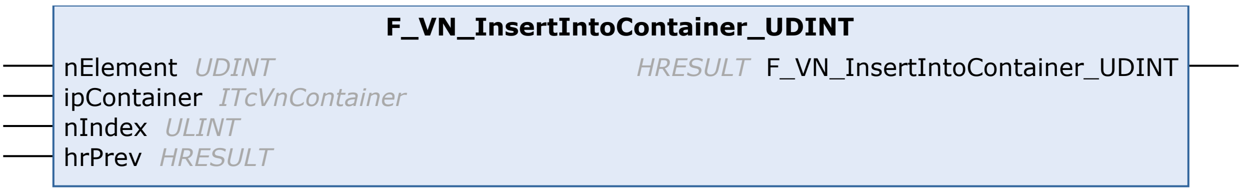
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.34 F_VN_InsertIntoContainer_UDINT



Insert an element of type UDINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_UDINT : HRESULT
VAR_INPUT
    nElement    : UDINT;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	UDINT	Single element to insert into ipContainer
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.35 F_VN_InsertIntoContainer_UINT

Insert an element of type UINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_UINT : HRESULT
VAR_INPUT
    nElement    : UINT;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	UINT	Single element to insert into ipContainer
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

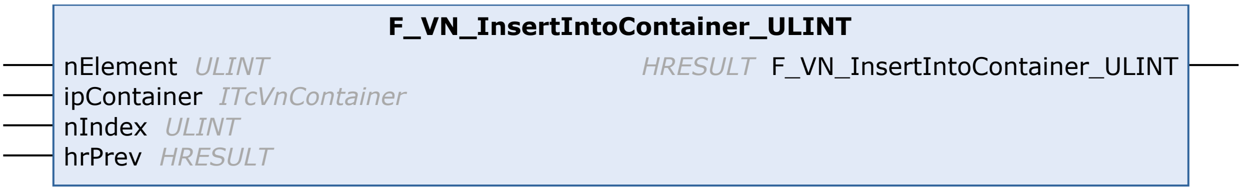
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.36 F_VN_InsertIntoContainer_ULINT



Insert an element of type ULINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_ULINT : HRESULT
VAR_INPUT
    nElement    : ULINT;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	ULINT	Single element to insert into ipContainer
ipContainer	ITcVnContainer [▶ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

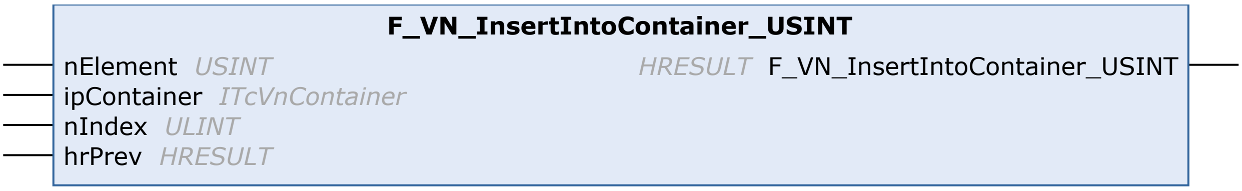
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.5.37 F_VN_InsertIntoContainer_USINT



Insert an element of type USINT into a container before the specified position.

Syntax

Definition:

```
FUNCTION F_VN_InsertIntoContainer_USINT : HRESULT
VAR_INPUT
    nElement    : USINT;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	USINT	Single element to insert into ipContainer
ipContainer	ITcVnContainer [▸ 349]	Container in which to insert the element
nIndex	ULINT	Position, before which the element will be inserted
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

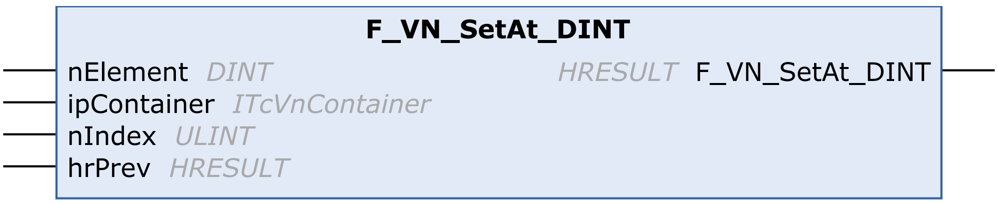
System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6 F_VN_SetAt

如果需要访问一个容器中的单个元素，应该使用F_VN_SetAt函数。如果需要连续处理容器中的所有元素，使用迭代器 [▸ 738]更好。

6.1.4.4.6.1 F_VN_SetAt_DINT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_DINT : HRESULT
VAR_INPUT
    nElement    : DINT;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	DINT	Element to set at the specified container position
ipContainer	ITcVnContainer [▶ 349]	Container with DINT elements, in which the element at position nIndex is replaced by nElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

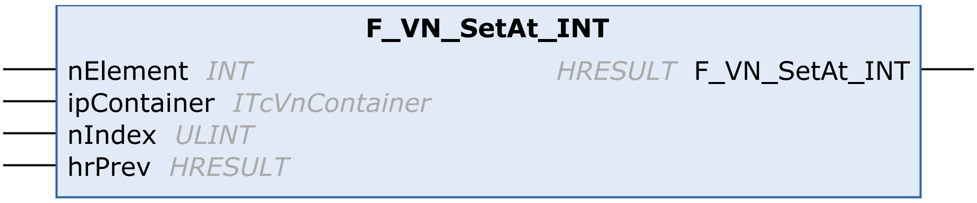
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.2 F_VN_SetAt_INT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_INT : HRESULT
VAR_INPUT
    nElement    : INT;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	INT	Element to set at the specified container position
ipContainer	ITcVnContainer [▶ 349]	Container with INT elements, in which the element at position nIndex is replaced by nElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.3 F_VN_SetAt_ITcVnContainer

Sets the element at the specified index of the container. (Alternatively use interface method .SetAt.)

Syntax

Definition:

```
FUNCTION F_VN_SetAt_ITcVnContainer : HRESULT
VAR_INPUT
    ipElement   : ITcVnContainer;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipElement	ITcVnContainer [▶ 349]	Element to set at the specified container position
ipContainer	ITcVnContainer [▶ 349]	Container with ITcVnContainer elements, in which the element at position nIndex is replaced by ipElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

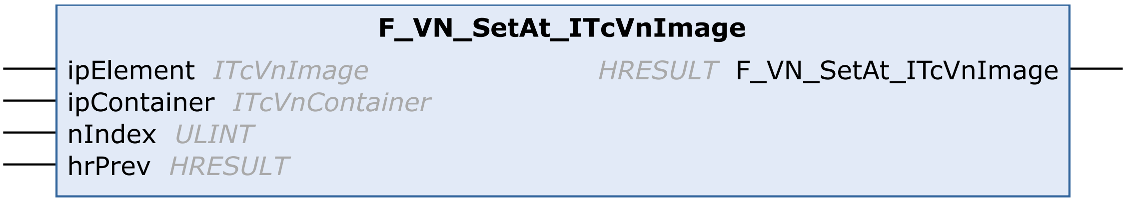
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.4 F_VN_SetAt_ITcVnImage



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_ITcVnImage : HRESULT
VAR_INPUT
    ipElement   : ITcVnImage;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipElement	ITcVnImage [▶ 390]	Element to set at the specified container position
ipContainer	ITcVnContainer [▶ 349]	Container with ITcVnImage elements, in which the element at position nIndex is replaced by ipElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

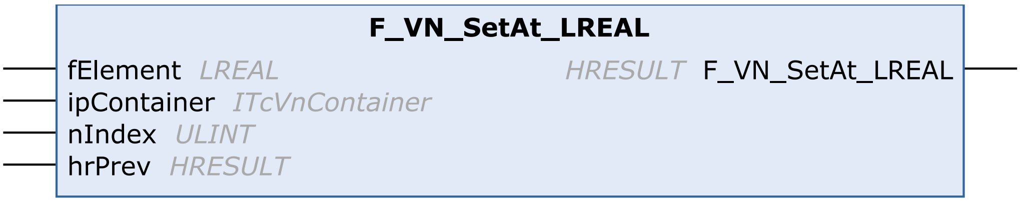
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.5 F_VN_SetAt_LREAL



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_LREAL : HRESULT
VAR_INPUT
    fElement    : LREAL;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fElement	LREAL	Element to set at the specified container position
ipContainer	ITcVnContainer [▶ 349]	Container with LREAL elements, in which the element at position nIndex is replaced by fElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

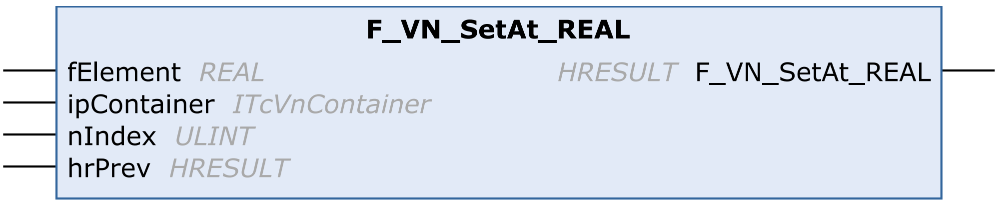
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.6 F_VN_SetAt_REAL



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_REAL : HRESULT
VAR_INPUT
    fElement     : REAL;
    ipContainer : ITcVnContainer;
    nIndex       : ULINT;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fElement	REAL	Element to set at the specified container position
ipContainer	ITcVnContainer [▸ 349]	Container with REAL elements, in which the element at position nIndex is replaced by fElement
nIndex	ULINT	Index
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

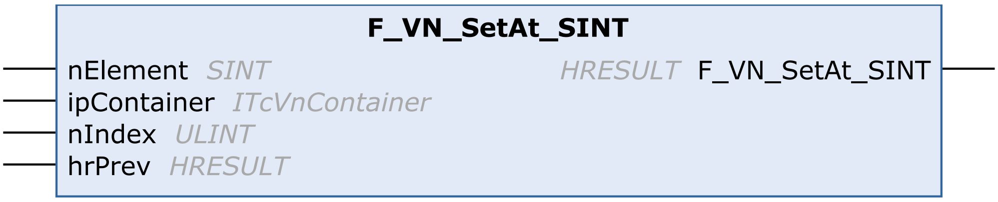
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.7 F_VN_SetAt_SINT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_SINT : HRESULT
VAR_INPUT
    nElement     : SINT;
    ipContainer : ITcVnContainer;
    nIndex       : ULINT;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	SINT	Element to set at the specified container position
ipContainer	ITcVnContainer [▶ 349]	Container with SINT elements, in which the element at position nIndex is replaced by nElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

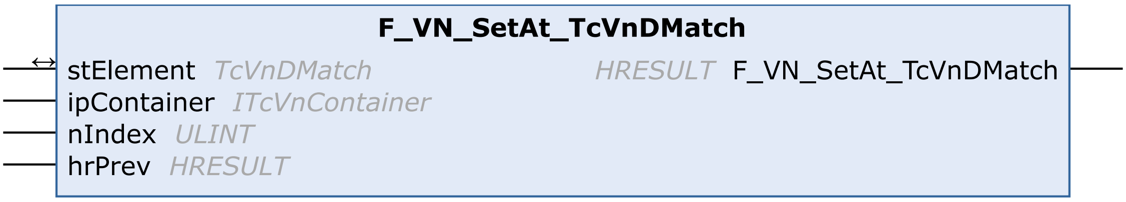
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.8 F_VN_SetAt_TcVnDMatch



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnDMatch : HRESULT
VAR_IN_OUT
    stElement   : TcVnDMatch;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnDMatch elements, in which the element at position nIndex is replaced by stElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stElement	TcVnDMatch [▶ 210]	Element to set at the specified container position

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.9 F_VN_SetAt_TcVnKeyPoint

Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnKeyPoint : HRESULT
VAR_IN_OUT
    stElement   : TcVnKeyPoint;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnKeyPoint elements, in which the element at position nIndex is replaced by stElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stElement	TcVnKeyPoint [▶ 210]	Element to set at the specified container position

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.10 F_VN_SetAt_TcVnPoint2_DINT

Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnPoint2_DINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnPoint2_DINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnPoint2_DINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint2_DINT [▶ 139]	Element to set at the specified container position

Return value

HRESULT [▶ 122]

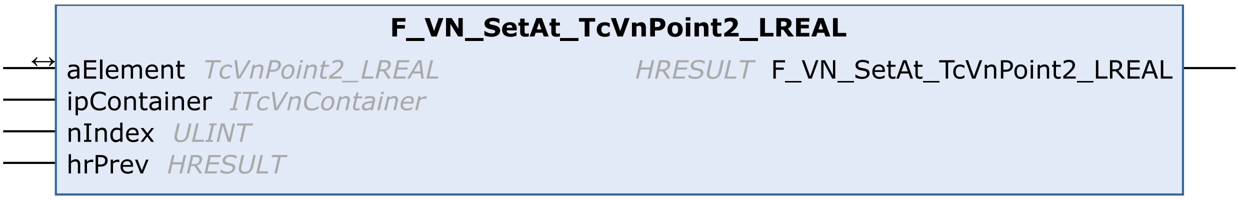
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.11 F_VN_SetAt_TcVnPoint2_LREAL



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnPoint2_LREAL : HRESULT
VAR_IN_OUT
    aElement    : TcVnPoint2_LREAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnPoint2_LREAL elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint2_LREAL [▸ 139]	Element to set at the specified container position

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.12 F_VN_SetAt_TcVnPoint2_REAL

Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnPoint2_REAL : HRESULT
VAR_IN_OUT
    aElement    : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnPoint2_REAL elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint2_REAL [▶ 139]	Element to set at the specified container position

Return value

HRESULT [▶ 122]

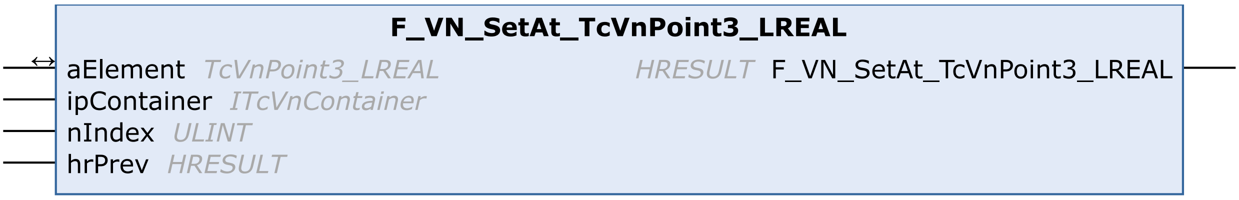
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.13 F_VN_SetAt_TcVnPoint3_LREAL



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnPoint3_LREAL : HRESULT
VAR_IN_OUT
    aElement     : TcVnPoint3_LREAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnPoint3_LREAL elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint3_LREAL [▶ 139]	Element to set at the specified container position

Return value

HRESULT [▶ 122]

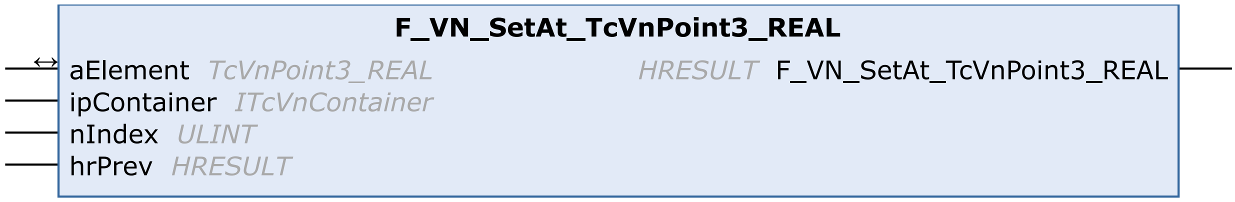
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.14 F_VN_SetAt_TcVnPoint3_REAL



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnPoint3_REAL : HRESULT
VAR_IN_OUT
    aElement     : TcVnPoint3_REAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnPoint3_REAL elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnPoint3_REAL [▶ 139]	Element to set at the specified container position

Return value

HRESULT [▶ 122]

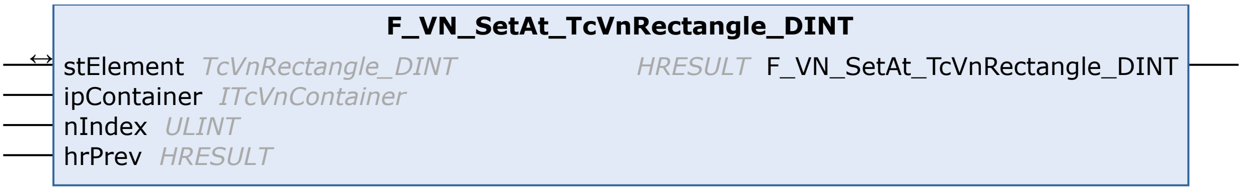
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.15 F_VN_SetAt_TcVnRectangle_DINT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnRectangle_DINT : HRESULT
VAR_IN_OUT
    stElement   : TcVnRectangle_DINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnRectangle_DINT elements, in which the element at position nIndex is replaced by stElement
nIndex	ULINT	Index
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stElement	TcVnRectangle_DINT [▸ 224]	Element to set at the specified container position

Return value

HRESULT [▸ 122]

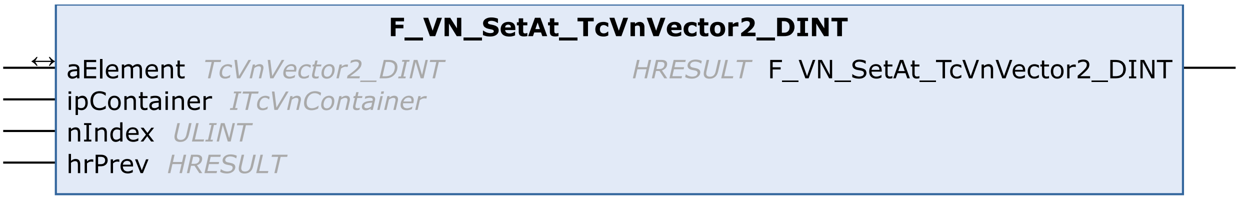
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.16 F_VN_SetAt_TcVnVector2_DINT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnVector2_DINT : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector2_DINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector2_DINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_DINT [▶ 141]	Element to set at the specified container position

Return value

HRESULT [▶ 122]

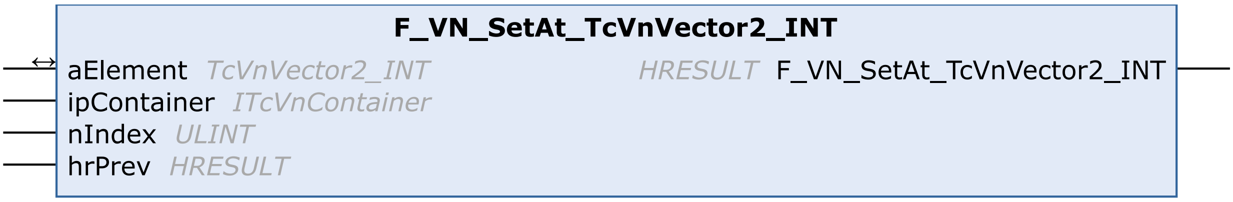
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.17 F_VN_SetAt_TcVnVector2_INT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnVector2_INT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector2_INT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector2_INT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_INT [▶ 141]	Element to set at the specified container position

Return value

HRESULT [▶ 122]

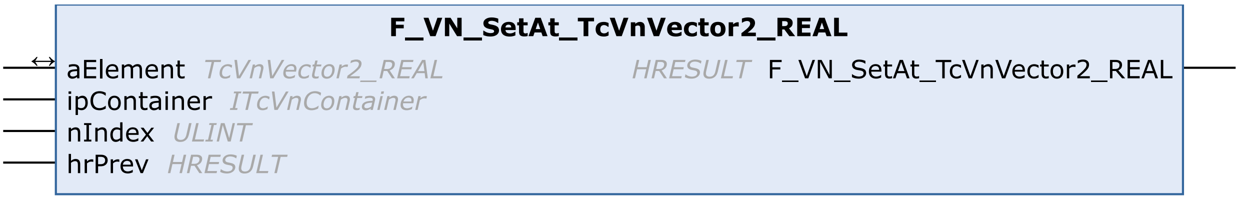
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.18 F_VN_SetAt_TcVnVector2_REAL



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnVector2_REAL : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector2_REAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector2_REAL elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_REAL [▶ 141]	Element to set at the specified container position

Return value

HRESULT [▶ 122]

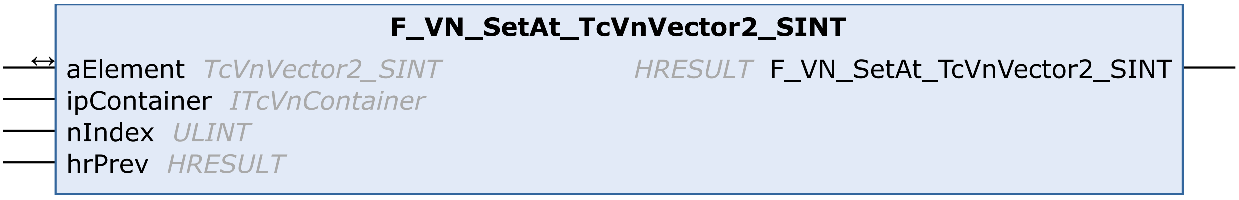
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.19 F_VN_SetAt_TcVnVector2_SINT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnVector2_SINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector2_SINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector2_SINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_SINT [▶ 141]	Element to set at the specified container position

Return value

HRESULT [▶ 122]

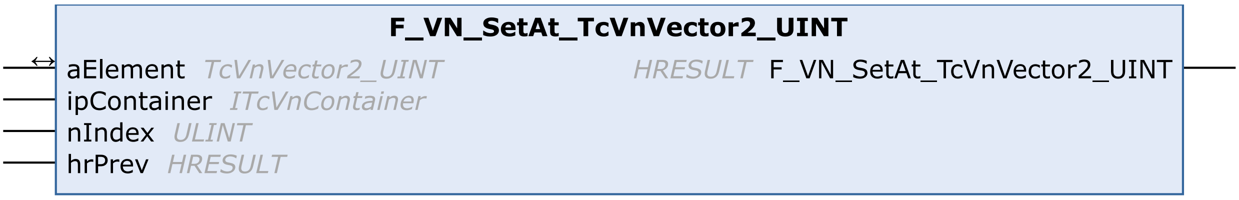
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.20 F_VN_SetAt_TcVnVector2_UINT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnVector2_UINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector2_UINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector2_UINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_UINT [▶ 141]	Element to set at the specified container position

Return value

HRESULT [▶ 122]

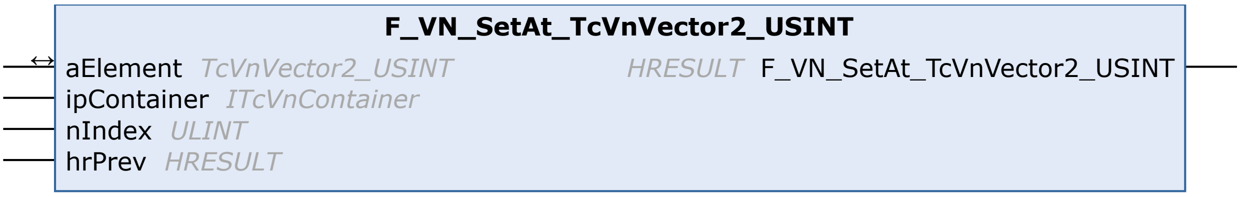
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.21 F_VN_SetAt_TcVnVector2_USINT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnVector2_USINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector2_USINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnVector2_USINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector2_USINT [▸ 141]	Element to set at the specified container position

Return value

HRESULT [▸ 122]

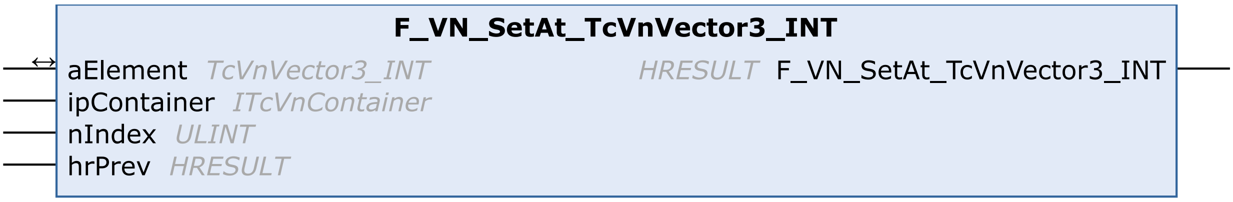
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.22 F_VN_SetAt_TcVnVector3_INT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnVector3_INT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector3_INT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector3_INT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_INT [▶ 141]	Element to set at the specified container position

Return value

HRESULT [▶ 122]

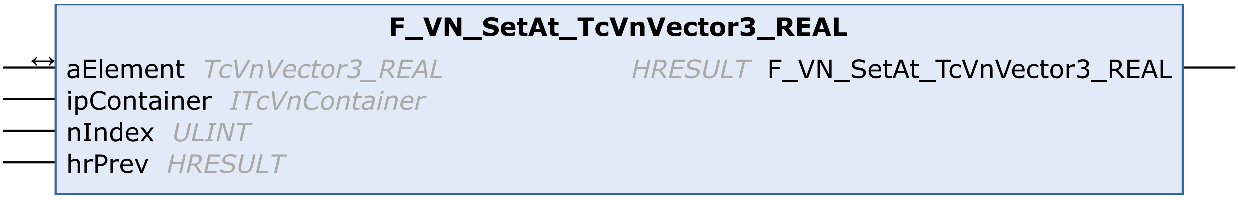
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.23 F_VN_SetAt_TcVnVector3_REAL



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnVector3_REAL : HRESULT
VAR_IN_OUT
    aElement     : TcVnVector3_REAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex       : ULINT;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnVector3_REAL elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_REAL [▸ 141]	Element to set at the specified container position

Return value

HRESULT [▸ 122]

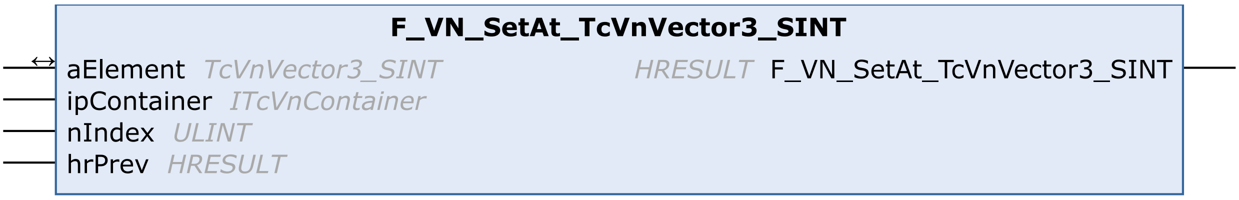
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.24 F_VN_SetAt_TcVnVector3_SINT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnVector3_SINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector3_SINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector3_SINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_SINT [▶ 141]	Element to set at the specified container position

Return value

HRESULT [▶ 122]

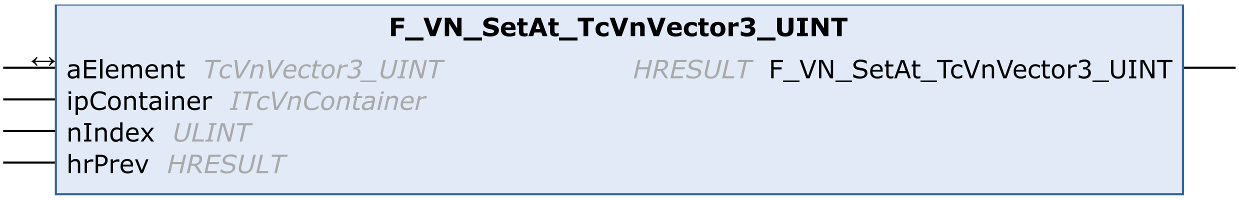
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.25 F_VN_SetAt_TcVnVector3_UINT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnVector3_UINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector3_UINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector3_UINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_UINT [▶ 141]	Element to set at the specified container position

Return value

HRESULT [▶ 122]

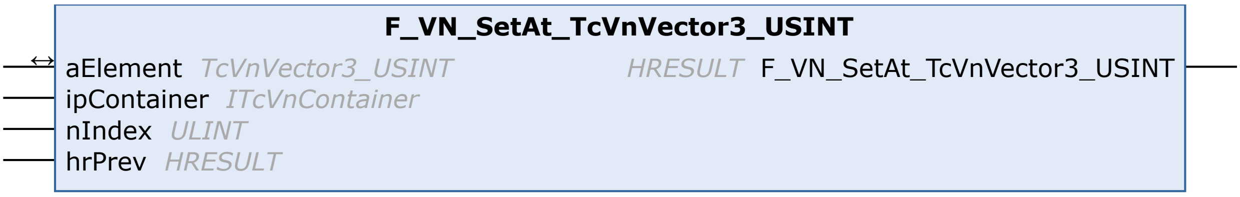
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.26 F_VN_SetAt_TcVnVector3_USINT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnVector3_USINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector3_USINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector3_USINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector3_USINT [▶ 141]	Element to set at the specified container position

Return value

HRESULT [▶ 122]

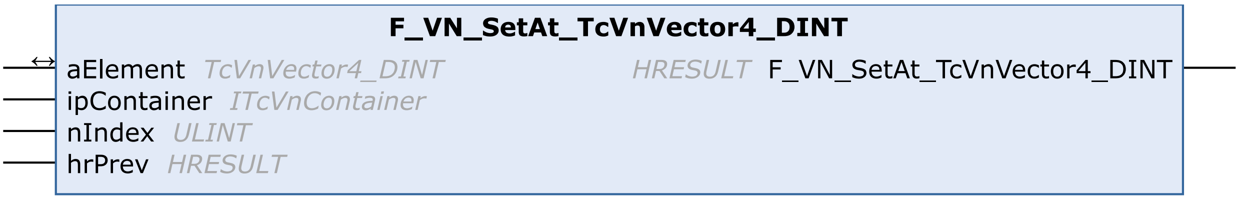
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.27 F_VN_SetAt_TcVnVector4_DINT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnVector4_DINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector4_DINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnVector4_DINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_DINT [▸ 141]	Element to set at the specified container position

Return value

HRESULT [▸ 122]

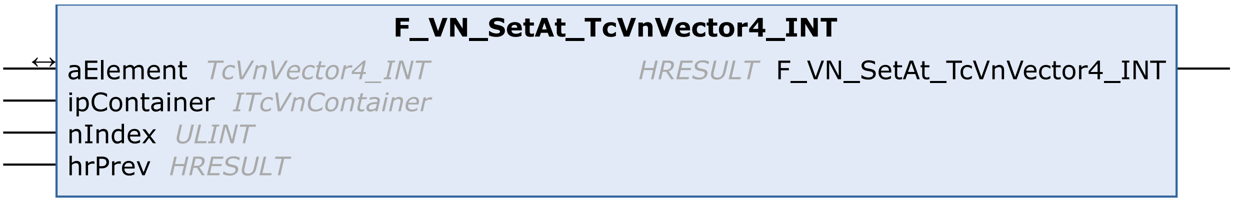
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.28 F_VN_SetAt_TcVnVector4_INT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnVector4_INT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector4_INT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector4_INT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_INT [▶ 141]	Element to set at the specified container position

Return value

HRESULT [▶ 122]

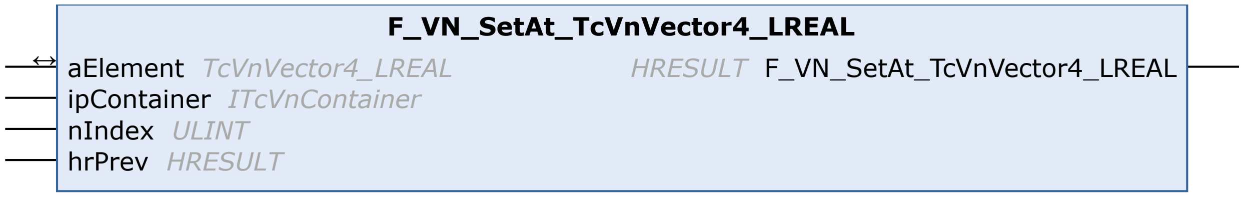
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.29 F_VN_SetAt_TcVnVector4_LREAL



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnVector4_LREAL : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector4_LREAL elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_LREAL [▶ 141]	Element to set at the specified container position

Return value

HRESULT [▶ 122]

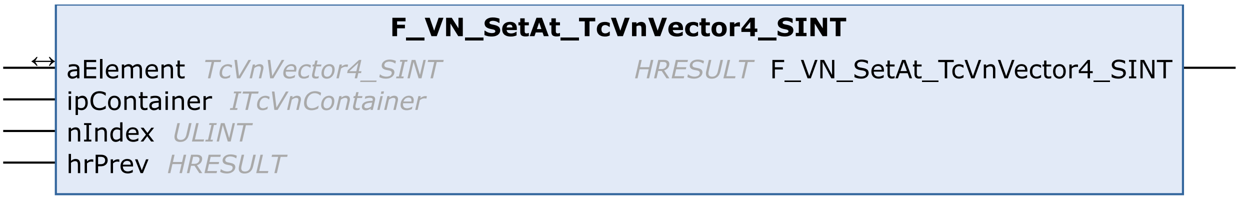
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.30 F_VN_SetAt_TcVnVector4_SINT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnVector4_SINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector4_SINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with TcVnVector4_SINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_SINT [▶ 141]	Element to set at the specified container position

Return value

HRESULT [▶ 122]

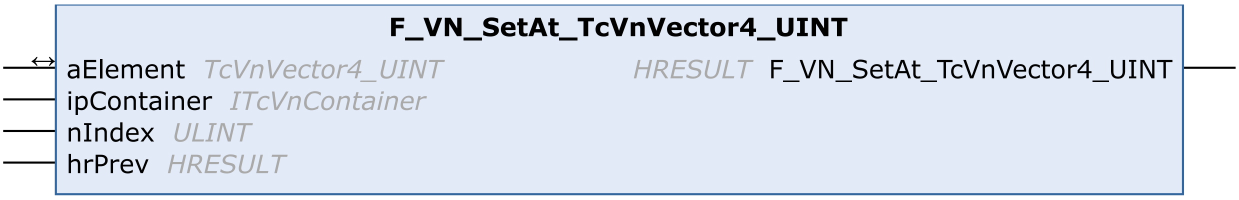
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.31 F_VN_SetAt_TcVnVector4_UINT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnVector4_UINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector4_UINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnVector4_UINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_UINT [▸ 141]	Element to set at the specified container position

Return value

HRESULT [▸ 122]

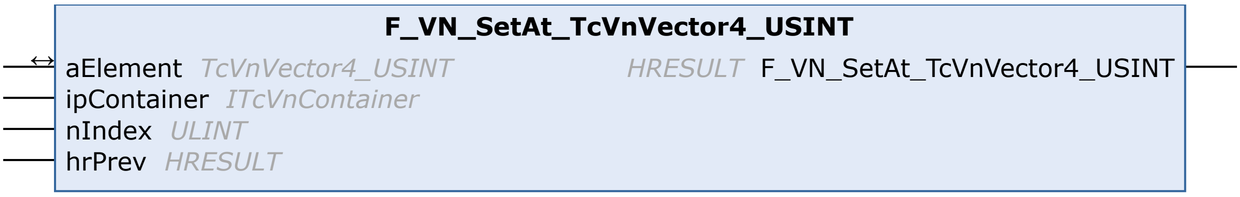
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.32 F_VN_SetAt_TcVnVector4_USINT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_TcVnVector4_USINT : HRESULT
VAR_IN_OUT
    aElement    : TcVnVector4_USINT;
END_VAR
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with TcVnVector4_USINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULINT	Index
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aElement	TcVnVector4_USINT [▸ 141]	Element to set at the specified container position

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.33 F_VN_SetAt_UDINT

Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_UDINT : HRESULT
VAR_INPUT
    nElement    : UDINT;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	UDINT	Element to set at the specified container position
ipContainer	ITcVnContainer [▶ 349]	Container with UDINT elements, in which the element at position nIndex is replaced by nElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

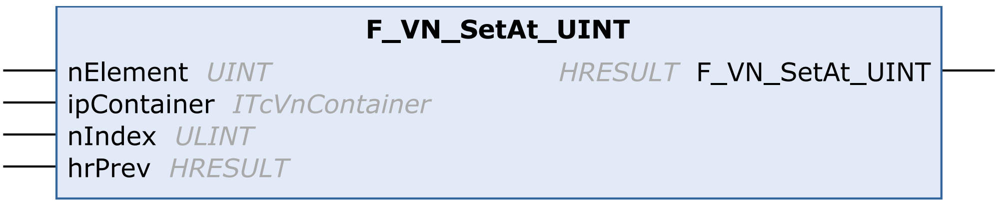
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.34 F_VN_SetAt_UINT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_UINT : HRESULT
VAR_INPUT
    nElement    : UINT;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	UINT	Element to set at the specified container position
ipContainer	ITcVnContainer [▶ 349]	Container with UINT elements, in which the element at position nIndex is replaced by nElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.35 F_VN_SetAt_ULINT

Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_ULINT : HRESULT
VAR_INPUT
    nElement    : ULINT;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	ULINT	Element to set at the specified container position
ipContainer	ITcVnContainer [▶ 349]	Container with ULINT elements, in which the element at position nIndex is replaced by nElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

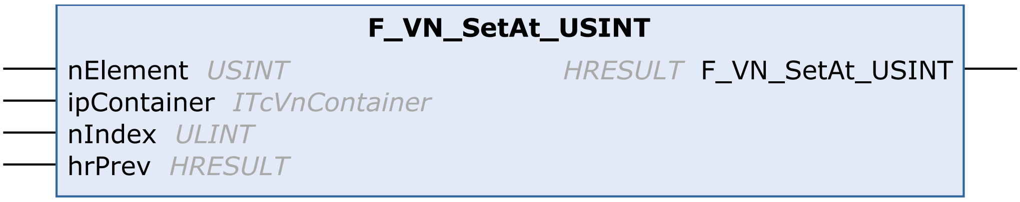
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.6.36 F_VN_SetAt_USINT



Sets the element at the specified index of the container.

Syntax

Definition:

```
FUNCTION F_VN_SetAt_USINT : HRESULT
VAR_INPUT
    nElement    : USINT;
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nElement	USINT	Element to set at the specified container position
ipContainer	ITcVnContainer [▶ 349]	Container with USINT elements, in which the element at position nIndex is replaced by nElement
nIndex	ULINT	Index
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

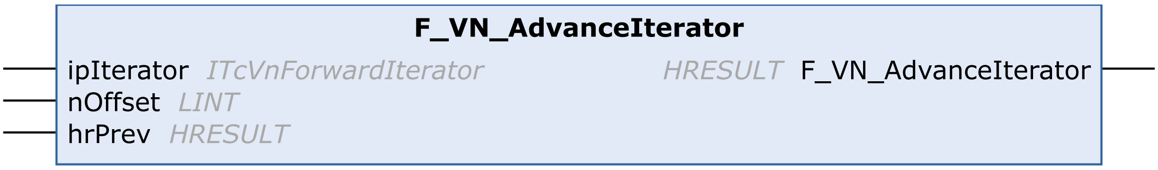
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.7 F_VN_AdvanceIterator



Advance an iterator by the specified offset.

Syntax

Definition:

```
FUNCTION F_VN_AdvanceIterator : HRESULT
VAR_INPUT
    ipIterator : ITcVnForwardIterator;
    nOffset    : LINT;
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipIterator	ITcVnForwardIterator [▶ 339]	Iterator to be advanced
nOffset	LINT	Offset, by which ipIterator is advanced
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

该函数F_VN_AdvanceIterator将一个迭代器 [▶ 337]增加或减少一定的偏移量nOffset。

如果偏移量nOffset始终为1，也可以使用更简单的函数F_VN_IncrementIterator [▶ 740]。

参数

迭代器

参数ipIterator类型为ITcVnForwardIterator。由于ITcVnBidirectionalIterator和ITcVnRandomAccessIterator从中得出，这些也可以转移到函数中。如果一个前向迭代器要被递减，内部会尝试将其转化为一个双向迭代器并进行递减。

偏移量

偏移量nOffset定义迭代器ipIterator需要递增或递减多少。如果nOffset > 0，递增；如果nOffset < 0，递减。

相关函数

- F_VN_IncrementIterator [▶ 740]
- F_VN_AdvanceIterator [▶ 718]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.8 F_VN_CheckIfEmpty

Checks if the container is empty. (Alternatively use interface method .CheckIfEmpty.)

Syntax

Definition:

```
FUNCTION F_VN_CheckIfEmpty : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.9 F_VN_CheckIfIteratorIsAtEnd

F_VN_CheckIfIteratorIsAtEnd	
ipIterator *ITcVnForwardIterator*	*HRESULT* F_VN_CheckIfIteratorIsAtEnd
hrPrev *HRESULT*	

Checks if the iterator points to the past-the-end element. (Alternatively use interface method .CheckIfEnd.)

Syntax

Definition:

```
FUNCTION F_VN_CheckIfIteratorIsAtEnd : HRESULT
VAR_INPUT
    ipIterator : ITcVnForwardIterator;
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipIterator	ITcVnForwardIterator [▸ 339]	Iterator
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.10 F_VN_ConvertContainerType

F_VN_ConvertContainerType	
ipSrcContainer *ITcVnContainer*	*HRESULT* F_VN_ConvertContainerType
ipDestContainer *Reference To ITcVnContainer*	
nDestTypeGuid *GUID*	
hrPrev *HRESULT*	

Converts a container to another type (Struct element types are not supported).

Syntax

Definition:

```
FUNCTION F_VN_ConvertContainerType : HRESULT
VAR_INPUT
    ipSrcContainer  : ITcVnContainer;
    ipDestContainer : Reference To ITcVnContainer;
    nDestTypeGuid   : GUID;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer	ITcVnContainer [▶ 349]	Source container
ipDestContainer	Reference To ITcVnContainer [▶ 349]	Returns the converted container
nDestTypeGuid	GUID	Specifies the destination container type for the conversion
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_ConvertContainerType将一个容器转换为待指定的容器类型 [▶ 139]。

全局ContainerType [▶ 139]常量可用于通过参数nTypeGuid来确定容器类型。

HRESULT

70E	INCOMPATIBLE	相同的接口指针被用作输入容器和结果容器，或者一个指定了与nDestTypeGuid中不同类型的容器被用作接口指针。

应用

例如，一个典型的用例是对一个容器进行转换，以便可以在图像中进行绘制。一个类型为ContainerType_Vector_TcVnPoint2_REAL的轮廓，除其他外由函数F_VN_LocateEllipseExp [▶ 1455]返回，可以转换为类型ContainerType_Vector_TcVnPoint2_DINT，以便能够使用函数F_VN_DrawContours [▶ 967]在图像中进行绘制：

```
hr := F_VN_LocateEllipseExp(ipImage, stEllipse, […], ipContourPoints, hr);
hr := F_VN_ConvertContainerType(
    ipSrcContainer  :=  ipContourPoints,
    ipDestContainer :=  ipContourToDraw,
    nDestTypeGuid   :=  ContainerType_Vector_TcVnPoint2_DINT,
    hr
);
hr := F_VN_DrawContours(ipContourToDraw, -1, ipImageDisp, aBlue, 5, hr);
```

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.11 F_VN_CopyContainer

Copys a container.

Syntax

Definition:

```
FUNCTION F_VN_CopyContainer : HRESULT
VAR_INPUT
    ipSrcContainer  : ITcVnContainer;
    ipDestContainer : Reference To ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContaine r	ITcVnContainer [▶ 349]	Source container
ipDestContain er	Reference To ITcVnContainer [▶ 349]	Destination container (same type ID as ipSrcContainer; Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_CopyContainer不仅复制了容器 [▶ 132]的接口指针 [▶ 119]，还对该容器进行了深度复制。

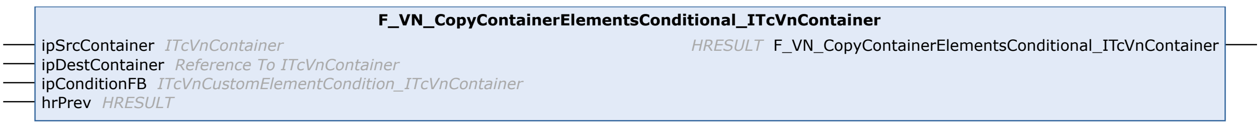
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.12 F_VN_CopyContainerElementsConditional_ITcVnContainer



Copys container elements to a new container, depending on a custom condition.

Syntax

Definition:

```
FUNCTION F_VN_CopyContainerElementsConditional_ITcVnContainer : HRESULT
VAR_INPUT
    ipSrcContainer  : ITcVnContainer;
    ipDestContainer : Reference To ITcVnContainer;
    ipConditionFB   : ITcVnCustomElementCondition_ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer	ITcVnContainer [▶ 349]	Source container
ipDestContainer	Reference To ITcVnContainer [▶ 349]	Returns a container with elements, that match the condition
ipConditionFB	ITcVnCustomElementCondition_ITcVnContainer [▶ 228]	Custom condition
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_CopyContainerElementsConditional过滤多维容器 [▶ 132]的元素。过滤条件将由功能块通过接口ITcVnCustomElementCondition_ITcVnContainer [▶ 228]指定。

参数

输入容器

输入容器ipSrcContainer必须是一个多维的容器；即该容器必须包含作为元素的子容器。因此，所有类型ContainerType_Vector_Vector_<…>的容器都可以使用。

结果容器

结果容器ipDestContainer包含所有满足ipConditionFB中条件的子容器。

过滤条件

过滤条件必须被定义为一个功能块的方法Condition。为了使这个功能块能够作为参数ipConditionFB转移到函数中，它必须实现接口ITcVnCustomElementCondition_ITcVnContainer [▶ 228]。

样本

- 容器元素选择 [▶ 2640]

相关函数

- F_VN_CopyContainerElementsConditional_ITcVnContainer [▶ 722]
- F_VN_CopyContainerElementsConditional_ITcVnForwardIterator [▶ 724]

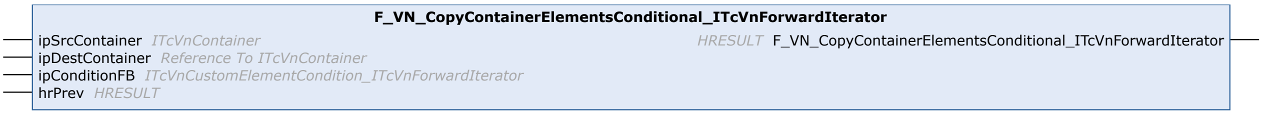
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.13 F_VN_CopyContainerElementsConditional_ITcVnForwardIterator



Copys container elements to a new container, depending on a custom condition.

Syntax

Definition:

```
FUNCTION F_VN_CopyContainerElementsConditional_ITcVnForwardIterator : HRESULT
VAR_INPUT
    ipSrcContainer  : ITcVnContainer;
    ipDestContainer : Reference To ITcVnContainer;
    ipConditionFB   : ITcVnCustomElementCondition_ITcVnForwardIterator;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer	ITcVnContainer [▶ 349]	Source container
ipDestContainer	Reference To ITcVnContainer [▶ 349]	Returns a container with elements, that match the condition
ipConditionFB	ITcVnCustomElementCondition_ITcVnForwardIterator [▶ 229]	Custom condition
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_CopyContainerElementsConditional_ITcVnForwardIterator过滤多维容器 [▶ 132]的元素。过滤条件将由功能块通过接口ITcVnCustomElementCondition_ITcVnForwardIterator [▶ 229]指定。

与F_VN_CopyContainerElementsConditional_ITcVnContainer [▶ 722]不同，子容器未转移到条件方法Condition，而是迭代器（它们各自指向子容器的第一个元素）。

参数

输入容器

输入容器ipSrcContainer必须是一个多维的容器；即该容器必须包含作为元素的子容器。因此，所有类型ContainerType_Vector_Vector_<…>的容器都可以使用。

结果容器

结果容器ipDestContainer包含所有满足ipConditionFB中条件的子容器。

过滤条件

过滤条件必须被定义为一个功能块的方法Condition。为了使这个功能块能够作为参数ipConditionFB转移到函数中，它必须实现接口ITcVnCustomElementCondition_ITcVnForwardIterator [▶ 229]。

样本

- 容器元素选择 [▶ 2640]

相关函数

- F_VN_CopyContainerElementsConditional_ITcVnContainer [▶ 722]
- F_VN_CopyContainerElementsConditional_ITcVnForwardIterator [▶ 724]

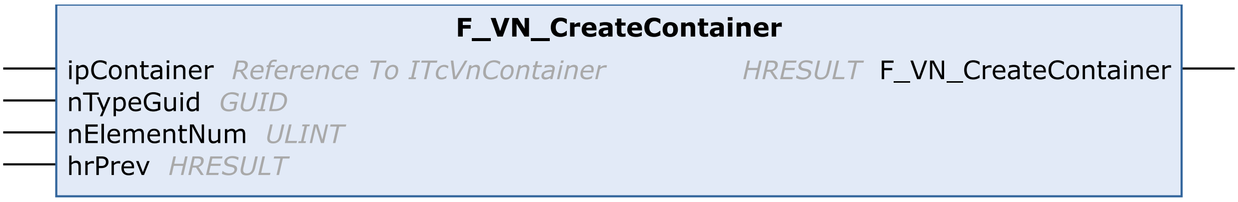
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.14 F_VN_CreateContainer



Creates a container with type GUID nTypeGuid and return its container interface.

Syntax

Definition:

```
FUNCTION F_VN_CreateContainer : HRESULT
VAR_INPUT
    ipContainer : Reference To ITcVnContainer;
    nTypeGuid   : GUID;
    nElementNum : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	Reference To ITcVnContainer [▶ 349]	Returns the created container
nTypeGuid	GUID	Type GUID of the container to be created
nElementNum	ULINT	Number of elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

全局ContainerType [▶ 139]常量可用于通过参数nTypeGuid来确定容器类型。

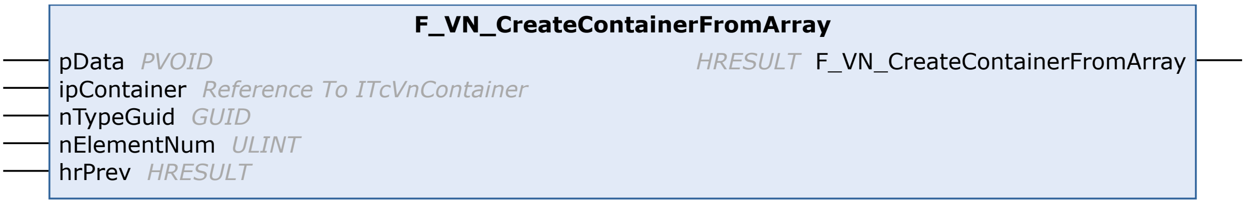
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.15 F_VN_CreateContainerFromArray



Creates a container with type GUID nTypeGuid, initialize it with provided data and return its container interface. Only supported for basic container types, i.e. not for containers of containers.

Syntax

Definition:

```
FUNCTION F_VN_CreateContainerFromArray : HRESULT
VAR_INPUT
    pData        : PVOID;
    ipContainer : Reference To ITcVnContainer;
    nTypeGuid    : GUID;
    nElementNum : ULINT;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
pData	PVOID	Pointer to the data array. Make sure that the array contains at least nElementNum elements and that the array element type matches the container element type.
ipContainer	Reference To ITcVnContainer [▶ 349]	Returns the created container
nTypeGuid	GUID	Type GUID of the container to be created
nElementNum	ULINT	Number of elements to copy from pData
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

全局ContainerType [▶ 139]常量可用于通过参数nTypeGuid来确定容器类型。

由于数组数据只是通过指针指定的，所以无法在函数中进行类型检查。因此，请注意以下警告：

⚠ 警告

内存访问

必须确保容器的类型和长度与指针处的内存区域的大小相对应。否则，这可能导致系统崩溃或无效的数据。

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.16 F_VN_EraseFromContainer

Erase elements from a container.

Syntax

Definition:

```
FUNCTION F_VN_EraseFromContainer : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nStartIdx   : ULINT;
    nCount      : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container from which to erase elements
nStartIdx	ULINT	Index of the first element to erase
nCount	ULINT	Number of elements to erase
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.17 F_VN_ExportContainer

Export the container elements into a buffer (e.g. an array). Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_ExportContainer : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    pBuffer     : PVOID;
    nBufferSize : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container with basic elements
pBuffer	PVOID	Buffer to store the container elements (Make sure to allocate enough memory! The required size in bytes can be determined using the function F_VN_ExportContainerSize)
nBufferSize	ULINT	Size of the buffer memory in bytes
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

该函数F_VN_ExportContainer将一个容器导出到一个数据缓冲区。请确保在指针的位置分配了足够的内存，以便将容器数据写入其中！所需的大小（以字节为单位）可以通过函数F_VN_ExportContainerSize [▸ 730]确定。你可以分配内存，例如通过创建一个适当大小的数组来分配。

应用

例如，包括10元素的Vector_REAL类型容器的出口如下：

```
VAR
    aArray: ARRAY [0..9] OF REAL;
    ipContainer: ITcVnContainer;
    nBufferSize: ULINT;
END_VAR

hr := F_VN_ExportContainerSize(ipContainer, nBufferSize, hr);
IF nBufferSize = SIZEOF(aArray) THEN
    hr := F_VN_ExportContainer(
        ipContainer:=ipContainer,
        pBuffer:=ADR(aArray),
        nBufferSize:=nBufferSize,
        hr
    );
END_IF
```

⚠ 警告

内存访问

必须确保容器的类型和长度与指针处的内存区域的大小相对应。否则，这可能导致系统崩溃或无效的数据。

相关函数

- F_VN_ExportContainer [▶ 728]
- F_VN_ExportSubContainer [▶ 731]
- F_VN_ExportContainer_String [▶ 729]
- F_VN_ExportSubContainer_String [▶ 732]
- F_VN_ExportContainerSize [▶ 730]
- F_VN_ExportSubContainerSize [▶ 734]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.18 F_VN_ExportContainer_String

Export the container elements into a string. Only possible for containers of type ContainerType_String_SINT.

Syntax

Definition:

```
FUNCTION F_VN_ExportContainer_String : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    sText       : STRING;
    nMaxLength  : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container of type ContainerType_String_SINT
sText	STRING	Make sure to choose a sufficient STRING size! The required size can be determined using the function VnExportContainerSize.
nMaxLength	ULINT	Maximum string length to export (including 0 termination). If the container content is longer, the string is cut off at nMaxLength - 1 and 0 termination is appended. In this case, S_FALSE is returned.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

相关函数

- F_VN_ExportContainer [▶ 728]
- F_VN_ExportSubContainer [▶ 731]
- F_VN_ExportContainer_String [▶ 729]
- F_VN_ExportSubContainer_String [▶ 732]
- F_VN_ExportContainerSize [▶ 730]
- F_VN_ExportSubContainerSize [▶ 734]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.19 F_VN_ExportContainerSize

Determine the required buffer size in bytes to store all container elements (number_of_Elements * size_per_Element). Only possible for containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_ExportContainerSize : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nBufferSize : ULINT;
END_VAR
```

```
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nBufferSize	ULINT	Output parameter containing the required buffer size

Return value

HRESULT [▶ 122]

相关函数

- F_VN_ExportContainer [▶ 728]
- F_VN_ExportSubContainer [▶ 731]
- F_VN_ExportContainer_String [▶ 729]
- F_VN_ExportSubContainer_String [▶ 732]
- F_VN_ExportContainerSize [▶ 730]
- F_VN_ExportSubContainerSize [▶ 734]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.20 F_VN_ExportSubContainer

Export the container elements of a sub-container into a buffer (e.g. an array). Only possible for 2-dimensional containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_ExportSubContainer : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    pBuffer     : PVOID;
    nBufferSize : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nIndex	ULINT	Index of the requested element
pBuffer	PVOID	Buffer to store the container elements (Make sure to allocate enough memory! The required size in bytes can be determined using the function F_VN_ExportSubContainerSize)
nBufferSize	ULINT	Size of the buffer memory in bytes
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

相关函数

- F_VN_ExportContainer [▶ 728]
- F_VN_ExportSubContainer [▶ 731]
- F_VN_ExportContainer_String [▶ 729]
- F_VN_ExportSubContainer_String [▶ 732]
- F_VN_ExportContainerSize [▶ 730]
- F_VN_ExportSubContainerSize [▶ 734]

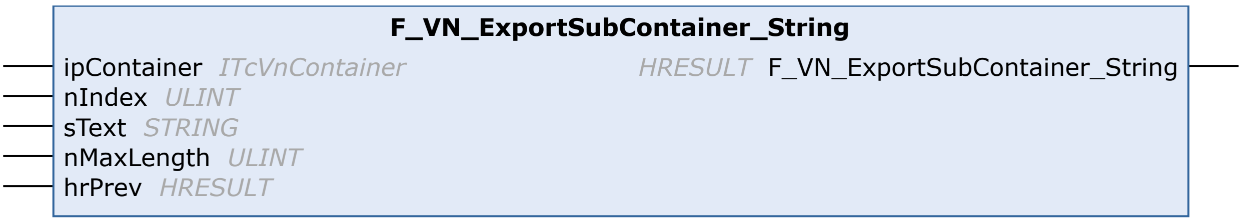
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.21 F_VN_ExportSubContainer_String



Export the container elements of a sub-container into a string. Only possible for 2-dimensional containers of type ContainerType_Vector_String_SINT.

Syntax

Definition:

```
FUNCTION F_VN_ExportSubContainer_String : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
    sText       : STRING;
    nMaxLength  : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container of type ContainerType_Vector_String_SINT
nIndex	ULINT	Index of the requested element
sText	STRING	Make sure to choose a sufficient STRING size! The required size can be determined using the function VnExportSubContainerSize.
nMaxLength	ULINT	Maximum string length to export (including 0 termination). If the container content is longer, the string is cut off at nMaxLength - 1 and 0 termination is appended. In this case, S_FALSE is returned.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

应用

例如，在代码读取 [▶ 796]的情况下，使用函数F_VN_ExportSubContainer，以获得结果文本：

```
hr := F_VN_ReadPharmaCode(ipImage, ipDecodedText, hr);
IF hr = S_OK THEN
    hr := F_VN_ExportSubContainer_String(ipDecodedText, 0, sCodeAsString, 255, hr);
    // sCodeAsString contains the result
END_IF
```

相关函数

- F_VN_ExportContainer [▶ 728]
- F_VN_ExportSubContainer [▶ 731]
- F_VN_ExportContainer_String [▶ 729]
- F_VN_ExportSubContainer_String [▶ 732]
- F_VN_ExportContainerSize [▶ 730]
- F_VN_ExportSubContainerSize [▶ 734]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.22 F_VN_ExportSubContainerSize

Determine the required buffer size in bytes to store all container elements of a sub-container (number_of_Elements * size_per_Element). Only possible for 2-dimensional containers with basic elements.

Syntax

Definition:

```
FUNCTION F_VN_ExportSubContainerSize : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nIndex      : ULINT;
END_VAR
VAR_IN_OUT
    nBufferSize : ULINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container with basic elements
nIndex	ULINT	Index of the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nBufferSize	ULINT	Output parameter containing the required buffer size

Return value

HRESULT [▶ 122]

相关函数

- F_VN_ExportContainer [▶ 728]
- F_VN_ExportSubContainer [▶ 731]
- F_VN_ExportContainer_String [▶ 729]
- F_VN_ExportSubContainer_String [▶ 732]
- F_VN_ExportContainerSize [▶ 730]
- F_VN_ExportSubContainerSize [▶ 734]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.23 F_VN_ExtractContainerRange

Copy the specified range of the source container into the destination container. (If the destination container already contains elements, they will be removed.)

Syntax

Definition:

```
FUNCTION F_VN_ExtractContainerRange : HRESULT
VAR_INPUT
    ipSrcContainer  : ITcVnContainer;
    ipDestContainer : Reference To ITcVnContainer;
    nFirstIdx       : ULINT;
    nLastIdx        : ULINT;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer	ITcVnContainer [▸ 349]	Source container
ipDestContainer	Reference To ITcVnContainer [▸ 349]	Destination container
nFirstIdx	ULINT	Index of the first element to copy
nLastIdx	ULINT	Index of the last element to copy
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

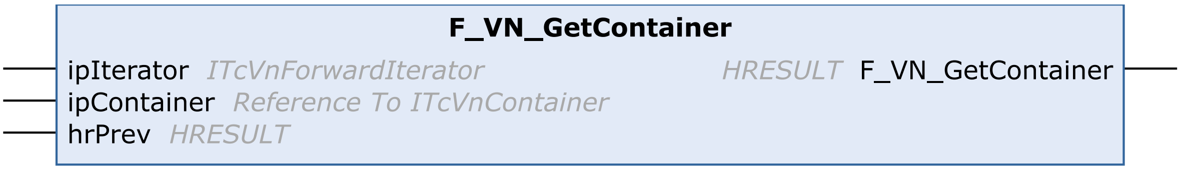
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.24 F_VN_GetContainer



Gets a pointer to the current element converted into an ITcVnContainer interface and increment its reference counter (only possible for container types). (Alternatively use interface method .GetContainer.)

Syntax

Definition:

```
FUNCTION F_VN_GetContainer : HRESULT
VAR_INPUT
    ipIterator  : ITcVnForwardIterator;
    ipContainer : Reference To ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipIterator	ITcVnForwardIterator [▸ 339]	Iterator
ipContainer	Reference To ITcVnContainer [▸ 349]	Returns the container interface
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

深度复制

这个函数创建一个容器元素的深度拷贝。这意味着，如果想要改变容器中的元素，必须把更改单独写回容器。

如果想要在访问容器元素的同时递增迭代器，专家版F_VN_GetContainerExp [▸ 737]可用于此。

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.25 F_VN_GetContainerExp

F_VN_GetContainerExp

ipIterator *ITcVnForwardIterator* — *HRESULT* F_VN_GetContainerExp
ipContainer *Reference To ITcVnContainer*
nOffset *DINT*
hrPrev *HRESULT*

Gets a pointer to the current element converted into an ITcVnContainer interface and increment its reference counter (only possible for container types). Additionally, an offset to increment or decrement the iterator afterwards can be provided. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_GetContainerExp : HRESULT
VAR_INPUT
    ipIterator  : ITcVnForwardIterator;
    ipContainer : Reference To ITcVnContainer;
    nOffset     : DINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipIterator	ITcVnForwardIterator [▶ 339]	Iterator
ipContainer	Reference To ITcVnContainer [▶ 349]	Returns the container interface
nOffset	DINT	Offset to increment (>0) or decrement (<0) the iterator afterwards
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_GetContainerExp是F_VN_GetContainer [▶ 736]的专家版本。如果在通过迭代器访问容器元素的同时还也希望递增这个迭代器，那么函数可以提供帮助。将参数nOffset设为值1 进行正常、连续递增：

```
hr := F_VN_GetContainerExp(ipIterator, ipElement, 1, hr);
```

通过标准函数，你需要两个函数调用：

```
hr := F_VN_GetContainer(ipIterator, ipElement, hr);
hr := F_VN_IncrementIterator(ipIterator, hr);
```

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.26 F_VN_GetForwardIterator

Gets a forward iterator for the container. (Alternatively use interface method .GetForwardIterator.)

Syntax

Definition:

```
FUNCTION F_VN_GetForwardIterator : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    ipIterator  : Reference To ITcVnForwardIterator;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Container
ipIterator	Reference To ITcVnForwardIterator [▸ 339]	Returns the iterator interface
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

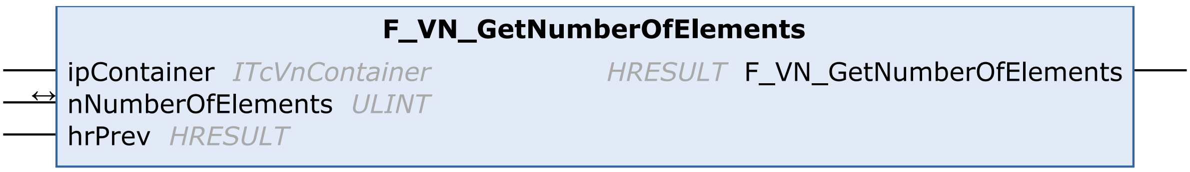
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.27 F_VN_GetNumberOfElements



Gets the number of elements in the container. (Alternatively use interface method .getElementNum.)

Syntax

Definition:

```
FUNCTION F_VN_GetNumberOfElements : HRESULT
VAR_INPUT
    ipContainer       : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nNumberOfElements : ULINT;
END_VAR
VAR_INPUT
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nNumberOfElements	ULINT	Returns the number of elements in ipContainer

Return value

HRESULT [▶ 122]

更多信息

这个函数不应用来迭代容器元素。相反，使用迭代器访问元素 [▶ 133]。特别是，不建议使用以下代码结构：

```
hr := F_VN_GetNumberOfElements(ipContainer, ipNumber, hr);
FOR i:=0 TO (ipNumber-1) DO
    // access container elements
END_FOR
```

由于ipNumber为UDINT（自然整数），所以返回值ipNumber = 0将导致下溢。

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.28 F_VN_GetRandomAccessIterator

Gets a random access iterator for the container. (Alternatively use interface method .GetRandomAccessIterator.)

Syntax

Definition:

```
FUNCTION F_VN_GetRandomAccessIterator : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    ipIterator  : Reference To ITcVnRandomAccessIterator;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container
ipIterator	Reference To ITcVnRandomAccessIterator [▶ 345]	Returns the iterator interface
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

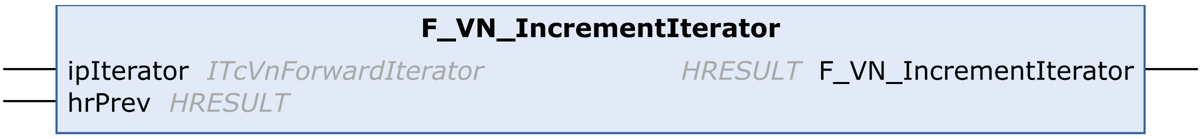
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.29 F_VN_IncrementIterator



Increment the iterator. (Alternatively use interface method .Increment.)

Syntax

Definition:

```
FUNCTION F_VN_IncrementIterator : HRESULT
VAR_INPUT
    ipIterator : ITcVnForwardIterator;
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipIterator	ITcVnForwardIterator [▶ 339]	Iterator
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_IncrementIterator将迭代器 [▶ 337]递增一个元素。相反，如果希望递减迭代器或递增几个位置，请使用函数F_VN_AdvanceIterator [▶ 718]。

参数

迭代器

参数ipIterator类型为ITcVnForwardIterator。由于ITcVnBidirectionalIterator和ITcVnRandomAccessIterator从中得出，这些也可以转移到函数中。如果一个前向迭代器要被递减，内部会尝试将其转化为一个双向迭代器并进行递减。

相关函数

- F_VN_IncrementIterator [▶ 740]
- F_VN_AdvanceIterator [▶ 718]

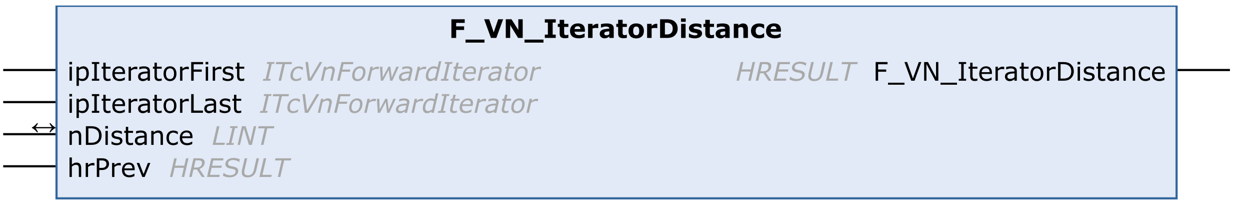
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.30 F_VN_IteratorDistance



Computes the distance between 2 iterators.

Syntax

Definition:

```
FUNCTION F_VN_IteratorDistance : HRESULT
VAR_INPUT
    ipIteratorFirst : ITcVnForwardIterator;
    ipIteratorLast  : ITcVnForwardIterator;
END_VAR
VAR_IN_OUT
    nDistance       : LINT;
```

```
END_VAR
VAR_INPUT
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipIteratorFirst	ITcVnForwardIterator [▶ 339]	First iterator
ipIteratorLast	ITcVnForwardIterator [▶ 339]	Last iterator
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nDistance	LINT	Returns the computed distance

Return value

HRESULT [▶ 122]

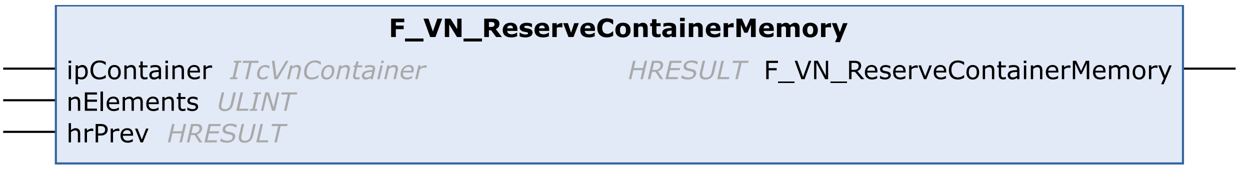
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.31 F_VN_ReserveContainerMemory



Reserve container memory (call with maximum required number of elements before manually appending elements for better performance)

Syntax

Definition:

```
FUNCTION F_VN_ReserveContainerMemory : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    nElements   : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container for which to reserve the memory
nElements	ULINT	Number of elements for which the container should reserve memory
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.32 F_VN_ReverseContainer

Reverse the container elements.

Syntax

Definition:

```
FUNCTION F_VN_ReverseContainer : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Container to be reversed
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.33 F_VN_SetContainer

F_VN_SetContainer
ipIterator *ITcVnForwardIterator* — *HRESULT* F_VN_SetContainer
ipContainer *ITcVnContainer*
hrPrev *HRESULT*

Sets the current element using an ITcVnContainer interface (only possible for container types). (Alternatively use interface method .SetContainer.)

Syntax

Definition:

```
FUNCTION F_VN_SetContainer : HRESULT
VAR_INPUT
    ipIterator  : ITcVnForwardIterator;
    ipContainer : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipIterator	ITcVnForwardIterator [▶ 339]	Iterator
ipContainer	ITcVnContainer [▶ 349]	Container interface of which the content is to be assigned to the current element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.4.34 F_VN_SetIteratorToBegin

F_VN_SetIteratorToBegin
ipIterator *ITcVnForwardIterator* — *HRESULT* F_VN_SetIteratorToBegin
hrPrev *HRESULT*

Sets the iterator to the first element of the container. (Alternatively use interface method .SetToBegin.)

Syntax

Definition:

```
FUNCTION F_VN_SetIteratorToBegin : HRESULT
VAR_INPUT
    ipIterator : ITcVnForwardIterator;
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipIterator	ITcVnForwardIterator [▸ 339]	Iterator
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5 Basic Image Operations

该组包含处理图像 [▸ 130]的函数。

函数

图像创建

- F_VN_ConvertElementType(Exp) [▸ 747]
- F_VN_CreateAssociatedImage [▸ 754]
- F_VN_CreateEmptyImage [▸ 755]
- F_VN_CreateImage [▸ 755]
- F_VN_CreateImageAndSetPixels [▸ 756]
- F_VN_CreateImageFromArray [▸ 758]

图像信息

- F_VN_GetImageHeight [▸ 768]
- F_VN_GetImageInfo [▸ 769]
- F_VN_GetImageWidth [▸ 770]
- F_VN_GetPixelFormat [▸ 772]

复制图像

- F_VN_CopyImage [▸ 749]

- F_VN_CopyImageRegion [▶ 750]
- F_VN_CopyImageRegionToRegion [▶ 752]

像素操作

- F_VN_GetPixel [▶ 771]
- F_VN_SetPixel [▶ 777]
- F_VN_SetPixels（Exp） [▶ 778]

感兴趣区域

- F_VN_GetRoi [▶ 773]
- F_VN_ResetRoi [▶ 775]
- F_VN_SetRoi [▶ 780]
- F_VN_SetRoi_TcVnRectangle_DINT [▶ 781]
- F_VN_SetRoi_TcVnRectangle_UDINT [▶ 782]

将图像内容导出到内存区域

- F_VN_ExportImage [▶ 759]
- F_VN_ExportImageAsBmp（Exp） [▶ 759]
- F_VN_ExportImageSize [▶ 763]
- F_VN_ExportImageAsBmpSize（Exp） [▶ 761]

融合图像

- F_VN_FuseImages [▶ 764]
- F_VN_FuseImagesArray [▶ 766]

处理图像通道

- F_VN_CombineImageChannels [▶ 746]
- F_VN_GetImageChannel [▶ 768]
- F_VN_MixImageChannels [▶ 774]
- F_VN_SetImageChannel [▶ 776]
- F_VN_SplitImageChannels [▶ 783]

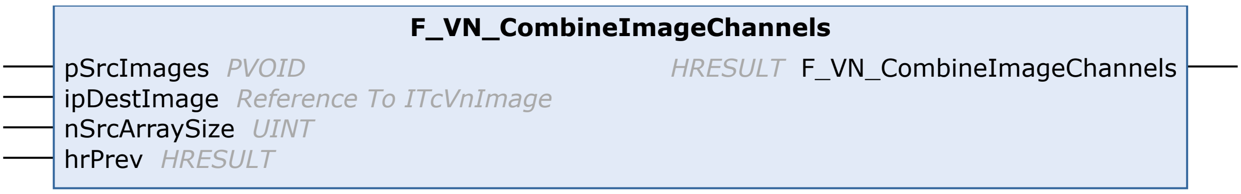
显示图像

- F_VN_CopyIntoDisplayableImage [▶ 753]
- F_VN_TransformIntoDisplayableImage（Exp） [▶ 784]

样本

基本图像操作 [▶ 2642]

6.1.4.5.1 F_VN_CombineImageChannels



Combines several single-channel images into one multi-channel image.

Syntax

Definition:

```
FUNCTION F_VN_CombineImageChannels : HRESULT
VAR_INPUT
    pSrcImages    : PVOID;
    ipDestImage   : Reference To ITcVnImage;
    nSrcArraySize : UINT;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
pSrcImages	PVOID	Pointer to an array of single-channel source images
ipDestImage	Reference To ITcVnImage [▶ 390]	Multi-channel destination image (An appropriate destination image will be created if required.)
nSrcArraySize	UINT	pSrcImages array size
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

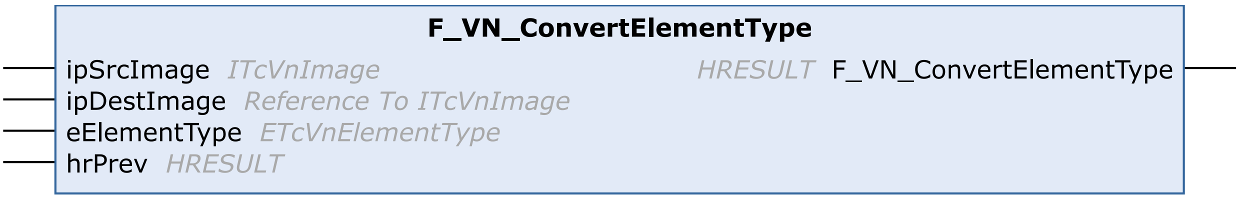
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.2 F_VN_ConvertElementType



Converts an image to another element type.

Syntax

Definition:

```
FUNCTION F_VN_ConvertElementType : HRESULT
VAR_INPUT
    ipSrcImage   : ITcVnImage;
    ipDestImage  : Reference To ITcVnImage;
    eElementType : ETcVnElementType;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
eElementType	ETcVnElementType [▶ 178]	Desired element type of the destination image
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.3 F_VN_ConvertElementTypeExp

Converts an image to another element type. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ConvertElementTypeExp : HRESULT
VAR_INPUT
    ipSrcImage   : ITcVnImage;
    ipDestImage  : Reference To ITcVnImage;
    eElementType : ETcVnElementType;
    fScaleFactor : LREAL;
    fDelta       : LREAL;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
eElementType	ETcVnElementType [▸ 178]	Desired element type of the destination image
fScaleFactor	LREAL	Scale factor for the pixel values
fDelta	LREAL	Value that is added to the scaled pixel values
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.4 F_VN_CopyImage

Creates a deep copy of an image.

Syntax

Definition:

```
FUNCTION F_VN_CopyImage : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_CopyImage创建输入图像ipSrcImage的深度拷贝。另外，函数F_VN_CopyImageRegion [▶ 750]可以用来创建图像区域的副本。

> **计算时间优化**
> 关于计算时间的优化，检查每个图像副本的必要性。

相关函数

- F_VN_CopyImage [▶ 749]用于复制整个图像
- F_VN_CopyImageRegion [▶ 750]用于复制一个图像区域
- F_VN_CopyImageRegionToRegion [▶ 752]用于将一个图像区域复制到某个位置

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.5 F_VN_CopyImageRegion

Deep copy the specified region of interest into a new image.

Syntax

Definition:

```
FUNCTION F_VN_CopyImageRegion : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    nX          : UDINT;
    nY          : UDINT;
    nWidth      : UDINT;
    nHeight     : UDINT;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
nX	UDINT	Left boundary (inclusive 0-based index)
nY	UDINT	Upper boundary (inclusive 0-based index)
nWidth	UDINT	ROI width
nHeight	UDINT	ROI height
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_CopyImageRegion创建输入图像ipSrcImage图像区域的深度拷贝。图像区域由矩形指定位置和大小，如同ROI [▶ 137]。

计算时间优化

关于计算时间的优化，检查每个图像副本的必要性。

如果不需要复制，而只需要减少图像区域的大小，则可以设置感兴趣区域 [▶ 137]。

应用

例如，复制尺寸[240, 120]且左上角位于[50, 50]的图像区域如下所示：

```
hr := F_VN_CopyImageRegion(
    ipSrcImage  :=  ipImageIn,
    nX          :=  50,
    nY          :=  50,
    nWidth      :=  240,
    nHeight     :=  120,
    ipDestImage :=  ipImageWork,
    hrPrev      :=  hr,
);
```

相关函数

- F_VN_CopyImage [▶ 749]用于复制整个图像
- F_VN_CopyImageRegion [▶ 750]用于复制一个图像区域
- F_VN_CopyImageRegionToRegion [▶ 752]用于将一个图像区域复制到某个位置

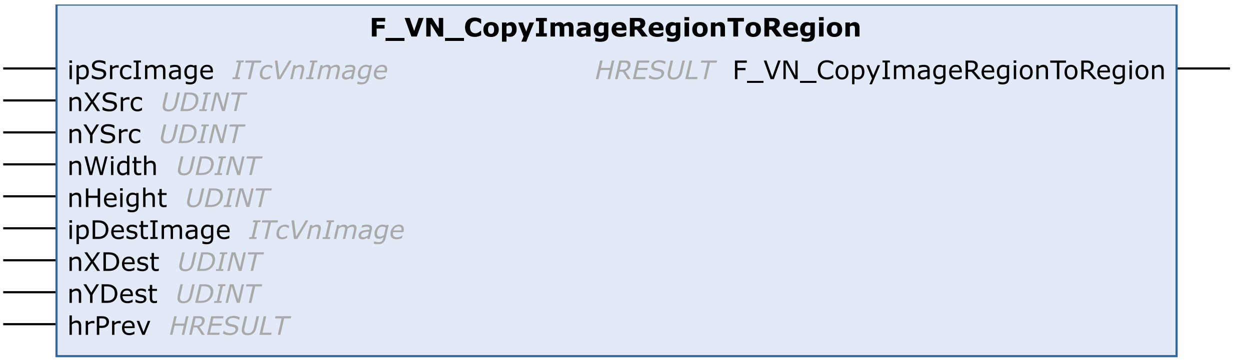
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.6 F_VN_CopyImageRegionToRegion



Copy an image region into another image region.

Syntax

Definition:

```
FUNCTION F_VN_CopyImageRegionToRegion : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    nXSrc       : UDINT;
    nYSrc       : UDINT;
    nWidth      : UDINT;
    nHeight     : UDINT;
    ipDestImage : ITcVnImage;
    nXDest      : UDINT;
    nYDest      : UDINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
nXSrc	UDINT	Left boundary in ipSrcImage (inclusive 0-based index)
nYSrc	UDINT	Upper boundary in ipSrcImage (inclusive 0-based index)
nWidth	UDINT	ROI width
nHeight	UDINT	ROI height
ipDestImage	ITcVnImage [▸ 390]	Destination image (same type as ipSrcImage)
nXDest	UDINT	Left boundary in ipDestImage (inclusive 0-based index)
nYDest	UDINT	Upper boundary in ipDestImage (inclusive 0-based index)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_CopyImageRegionToRegion将输入图像ipSrcImage的一个图像区域复制到目标图像ipDestImage的某个地方。图像区域由矩形指定位置和大小，如同ROI [▸ 137]。

应用

例如，在结果图像ipImageRes左上角的位置aPosition复制尺寸[100, 100]的图像区域如下所示：

```
hr := F_VN_CopyImageRegionToRegion(
    ipSrcImage  :=  ipImageWork,
    nXSrc       :=  aPosition[0],
    nYSrc       :=  aPosition[1],
    nWidth      :=  100,
    nHeight     :=  100,
    ipDestImage :=  ipImageRes,
    nXDest      :=  0,
    nYDest      :=  0,
    hrPrev      :=  hr
);
```

样本

- 复制图像区域 [▶ 2643]

相关函数

- F_VN_CopyImage [▶ 749]用于复制整个图像
- F_VN_CopyImageRegion [▶ 750]用于复制一个图像区域
- F_VN_CopyImageRegionToRegion [▶ 752]用于将一个图像区域复制到某个位置

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.7 F_VN_CopyIntoDisplayableImage

F_VN_CopyIntoDisplayableImage	
ipSrcImage *ITcVnImage*	*HRESULT* F_VN_CopyIntoDisplayableImage
ipDestImage *Reference To ITcVnDisplayableImage*	
hrPrev *HRESULT*	

Copys an image into a displayable image. If you do not want to use ipSrcImage after this function call, you might want to use F_VN_TransformIntoDisplayableImage instead for better performance.

Syntax

Definition:

```
FUNCTION F_VN_CopyIntoDisplayableImage : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnDisplayableImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnDisplayableImage [▶ 390]	Returns the displayable image
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

关于更多详情，请参见可显示的图像 [▶ 131]。

相关函数

- F_VN_TransformIntoDisplayableImage [▶ 784]
- F_VN_TransformIntoDisplayableImageExp [▶ 785]

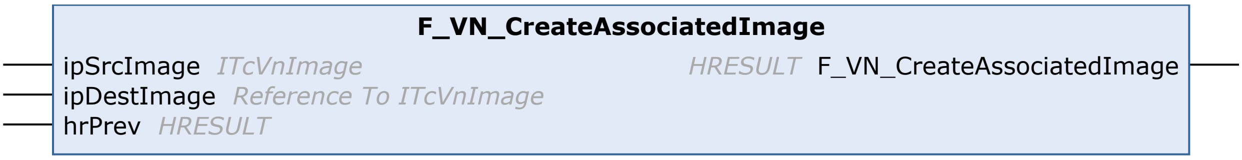
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.8 F_VN_CreateAssociatedImage



Creates a new image that shares its data with the source image. E.g. useful to work on different (disjoint) ROIs in parallel.

Syntax

Definition:

```
FUNCTION F_VN_CreateAssociatedImage : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Returns the created image (Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.9 F_VN_CreateEmptyImage

Creates an empty image without allocating any data buffer. The initial reference count is set to one if a new image interface is created and kept, otherwise.

Syntax

Definition:

```
FUNCTION F_VN_CreateEmptyImage : HRESULT
VAR_INPUT
    ipImage : Reference To ITcVnImage;
    hrPrev  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	Reference To ITcVnImage [▸ 390]	Returns the created image
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

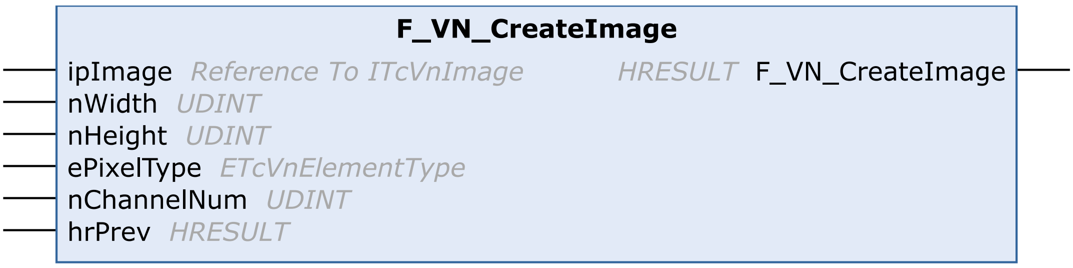
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.10 F_VN_CreateImage



Creates an image and allocate an appropriate data buffer. The initial reference count is set to one if a new image interface is created and kept, otherwise.

Syntax

Definition:

```
FUNCTION F_VN_CreateImage : HRESULT
VAR_INPUT
    ipImage     : Reference To ITcVnImage;
    nWidth      : UDINT;
    nHeight     : UDINT;
    ePixelType  : ETcVnElementType;
    nChannelNum : UDINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	Reference To ITcVnImage [▸ 390]	Returns the created image (Non-zero interface pointers are reused.)
nWidth	UDINT	Image width
nHeight	UDINT	Image height
ePixelType	ETcVnElementType [▸ 178]	Pixel type
nChannelNum	UDINT	Number of channels (1 to 255)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.11 F_VN_CreateImageAndSetPixels

Creates an image, allocates an appropriate data buffer and sets all pixels to the specified value. The initial reference count is set to one if a new image interface is created and kept, otherwise.

Syntax

Definition:

```
FUNCTION F_VN_CreateImageAndSetPixels : HRESULT
VAR_INPUT
    ipImage     : Reference To ITcVnImage;
    nWidth      : UDINT;
    nHeight     : UDINT;
    ePixelType  : ETcVnElementType;
    nChannelNum : UDINT;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	Reference To ITcVnImage [▸ 390]	Returns the created image (Non-zero interface pointers are reused.)
nWidth	UDINT	Image width
nHeight	UDINT	Image height
ePixelType	ETcVnElementType [▸ 178]	Pixel type
nChannelNum	UDINT	Number of channels (1 to 4)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_LREAL [▸ 141]	Pixel value

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.12 F_VN_CreateImageFromArray

F_VN_CreateImageFromArray

Inputs	Output
pData *PVOID*	*HRESULT* F_VN_CreateImageFromArray
ipImage *Reference To ITcVnImage*	
nWidth *UDINT*	
nHeight *UDINT*	
ePixelType *ETcVnElementType*	
nChannelNum *UDINT*	
hrPrev *HRESULT*	

Creates an image and initialize it with the provided data. The initial reference count is set to one if a new image interface is created and kept, otherwise.

Syntax

Definition:

```
FUNCTION F_VN_CreateImageFromArray : HRESULT
VAR_INPUT
    pData       : PVOID;
    ipImage     : Reference To ITcVnImage;
    nWidth      : UDINT;
    nHeight     : UDINT;
    ePixelType  : ETcVnElementType;
    nChannelNum : UDINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
pData	PVOID	Pointer to the 1D data array. Make sure that the array contains at least nWidth * nHeight * nChannelNum elements and that the array element type matches ePixelType.
ipImage	Reference To ITcVnImage [▶ 390]	Returns the created image (Non-zero interface pointers are reused.)
nWidth	UDINT	Image width
nHeight	UDINT	Image height
ePixelType	ETcVnElementType [▶ 178]	Pixel type
nChannelNum	UDINT	Number of channels (1 to 255)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.13 F_VN_ExportImage

Exports the image data into a given buffer (e.g. an array). F_VN_ExportImageSize should be called before to get the required buffer size.

Syntax

Definition:

```
FUNCTION F_VN_ExportImage : HRESULT
VAR_INPUT
    ipImage     : ITcVnImage;
    pBuffer     : PVOID;
    nBufferSize : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	Image
pBuffer	PVOID	Pointer to a buffer to store the image data (Make sure to allocate enough memory! The required size in bytes can be determined using the function F_VN_ExportImageSize)
nBufferSize	ULINT	Size of the buffer memory in bytes
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.14 F_VN_ExportImageAsBmp

Exports the image as bitmap into a given buffer (e.g. an array). F_VN_ExportImageAsBmpSize should be called before to get the required buffer size.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_ExportImageAsBmp : HRESULT
VAR_INPUT
    ipImage     : ITcVnImage;
    pBuffer     : PVOID;
    nBufferSize : ULINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▸ 390]	Image
pBuffer	PVOID	Pointer to a buffer to store the bitmap image (Make sure to allocate enough memory! The required size in bytes can be determined using the function F_VN_ExportImageAsBmpSize)
nBufferSize	ULINT	Size of the buffer memory in bytes
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

注意接口ITcVnBitmapExport [▸ 373]的转换特性。

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.15 F_VN_ExportImageAsBmpExp

Exports the image as bitmap into a given buffer (e.g. an array). F_VN_ExportImageAsBmpSizeExp should be called before to get the required buffer size. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_ExportImageAsBmpExp : HRESULT
VAR_INPUT
    ipImage     : ITcVnImage;
    pBuffer     : PVOID;
    nBufferSize : ULINT;
    nWidth      : UDINT;
    nHeight     : UDINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▸ 390]	Image
pBuffer	PVOID	Pointer to a buffer to store the bitmap image (Make sure to allocate enough memory! The required size in bytes can be determined using the function F_VN_ExportImageAsBmpSizeExp)
nBufferSize	ULINT	Size of the buffer memory in bytes
nWidth	UDINT	Desired width (or 0 to keep the original width)
nHeight	UDINT	Desired height (or 0 to keep the original height)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.16 F_VN_ExportImageAsBmpSize

Get the required buffer size (in bytes) to store the image as bitmap.

Syntax

Definition:

```
FUNCTION F_VN_ExportImageAsBmpSize : HRESULT
VAR_INPUT
    ipImage     : ITcVnImage;
END_VAR
VAR_IN_OUT
    nBufferSize : ULINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	Image
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nBufferSize	ULINT	Returns the required buffer size in bytes

Return value

HRESULT [▶ 122]

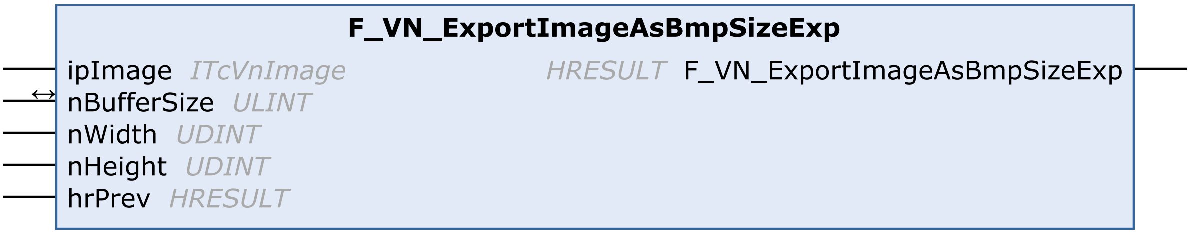
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.17 F_VN_ExportImageAsBmpSizeExp



Get the required buffer size (in bytes) to store the image as bitmap. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ExportImageAsBmpSizeExp : HRESULT
VAR_INPUT
    ipImage      : ITcVnImage;
END_VAR
VAR_IN_OUT
    nBufferSize : ULINT;
END_VAR
VAR_INPUT
    nWidth      : UDINT;
    nHeight     : UDINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	Image
nWidth	UDINT	Desired width (or 0 to keep the original width)
nHeight	UDINT	Desired height (or 0 to keep the original height)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nBufferSize	ULINT	Returns the required buffer size in bytes

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.18 F_VN_ExportImageSize

Get the required buffer size (in bytes) to store the image data.

Syntax

Definition:

```
FUNCTION F_VN_ExportImageSize : HRESULT
VAR_INPUT
    ipImage     : ITcVnImage;
END_VAR
VAR_IN_OUT
    nBufferSize : ULINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	Image
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nBufferSize	ULINT	Returns the required buffer size in bytes

Return value

HRESULT [▶ 122]

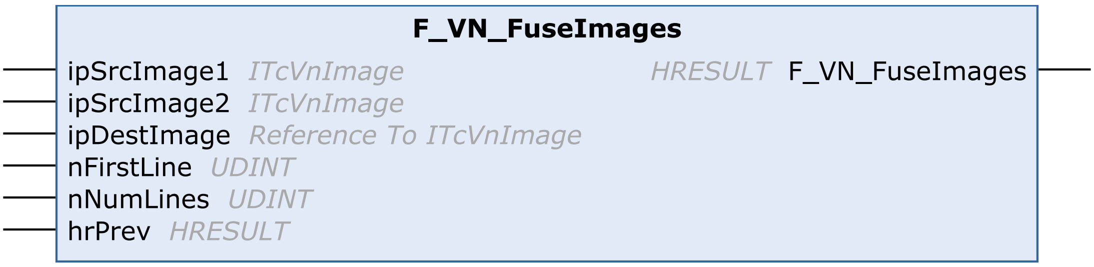
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.19 F_VN_FuseImages



Fuse 2 images vertically (intended for line scan cameras).

Syntax

Definition:

```
FUNCTION F_VN_FuseImages : HRESULT
VAR_INPUT
    ipSrcImage1 : ITcVnImage;
    ipSrcImage2 : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    nFirstLine  : UDINT;
    nNumLines   : UDINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▶ 390]	Source image 1
ipSrcImage2	ITcVnImage [▶ 390]	Source image 2
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image
nFirstLine	UDINT	Line index in ipSrcImage1, which is the first line in ipDestImage
nNumLines	UDINT	The number of lines that should be copied to ipDestImage, starting with nFirstLine in ipSrcImage1. Once the last line of ipSrcImage1 was copied, the remaining lines are copied from ipSrcImage2, starting at line index 0.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

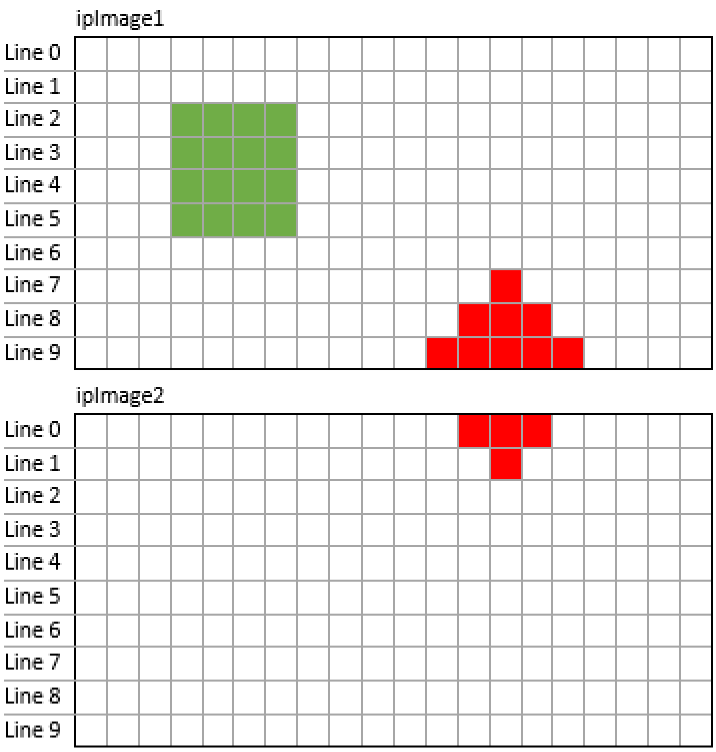
更多信息

函数F_VN_FuseImages将两个图像合并形成一个新的连续图像。这对线扫描相机来说特别有意义。

函数F_VN_FuseImagesArray [▶ 766]可用于融合最多 10 个图像。

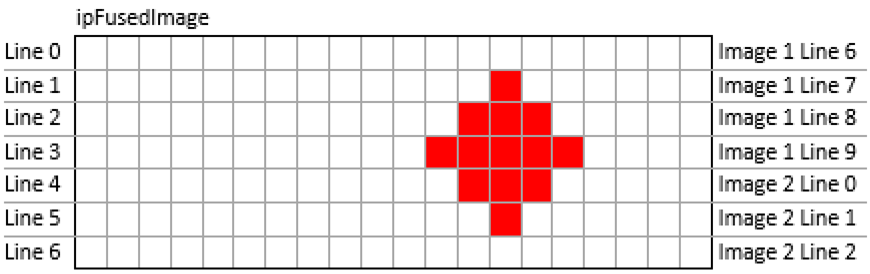
应用

线扫描相机通常将一定数量的连续捕获的线作为二维图像返回。这个二维图像可以包含整个感兴趣的物体（下图中以绿色显示），也可以包含物体的一部分（图中以红色显示）。



如需分析红色物体，可以使用函数F_VN_FuseImages以将两个物体部分融合，从而形成一个完整的图像。如果知道哪些线与对象有关，则可以将链接限制在相关线上。

```
F_VN_FuseImages(ipImage1, ipImage2, ipFusedImage, 6, 7, S_OK);
```



样本

- 融合多个图像 [▶ 2642]

相关函数

- F_VN_FuseImages [▶ 764]用于融合两个图像。
- F_VN_FuseImagesArray [▶ 766]用于融合最多 10 个图像。

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.20 F_VN_FuseImagesArray

Fuse up to 10 images vertically (intended for line scan cameras).

Syntax

Definition:

```
FUNCTION F_VN_FuseImagesArray : HRESULT
VAR_IN_OUT
    aSrcImageArray : TcVnArray10_ITcVnImage;
END_VAR
VAR_INPUT
    nValidImages    : UDINT;
    ipDestImage     : Reference To ITcVnImage;
    nFirstLine      : UDINT;
    nNumLines       : UDINT;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nValidImages	UDINT	Number of valid images in aSrcImageArray
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image
nFirstLine	UDINT	Line index in ipSrcImage1, which is the first line in ipDestImage
nNumLines	UDINT	The number of lines that should be copied to ipDestImage, starting with nFirstLine in aSrcImageArray[0]. Once the last line of aSrcImageArray[0] was copied, the remaining lines are copied from aSrcImageArray[1] (starting at line index 0) and so on.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aSrcImageArray	TcVnArray10_ITcVnImage [▶ 141]	Ten-element source image array (not all elements need to be filled, nValidImages specifies the actual amount of images)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_FuseImageArray是函数F_VN_FuseImages [▶ 764]的修改版，且可以融合最多 10 个图像而不是两个图像。

样本

- 融合多个图像 [▶ 2642]

相关函数

- F_VN_FuseImages [▶ 764]用于融合两个图像。
- F_VN_FuseImagesArray [▶ 766]用于融合最多 10 个图像。

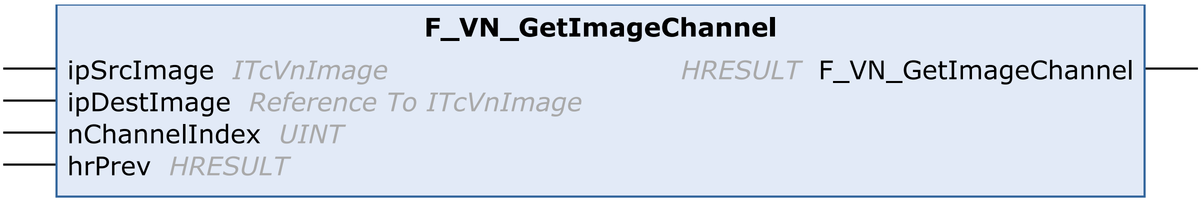
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.21 F_VN_GetImageChannel



Return the specified source image channel as a single-channel image.

Syntax

Definition:

```
FUNCTION F_VN_GetImageChannel : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipDestImage   : Reference To ITcVnImage;
    nChannelIndex : UINT;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate image will be created if required.)
nChannelIndex	UINT	Index of the requested source image channel
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.22 F_VN_GetImageHeight

Return the height of an image. (Alternatively use interface method .GetHeight.)

Syntax

Definition:

```
FUNCTION F_VN_GetImageHeight : HRESULT
VAR_INPUT
    ipImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    nHeight : UDINT;
END_VAR
VAR_INPUT
    hrPrev  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	Image
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nHeight	UDINT	Return the height of ipImage

Return value

HRESULT [▶ 122]

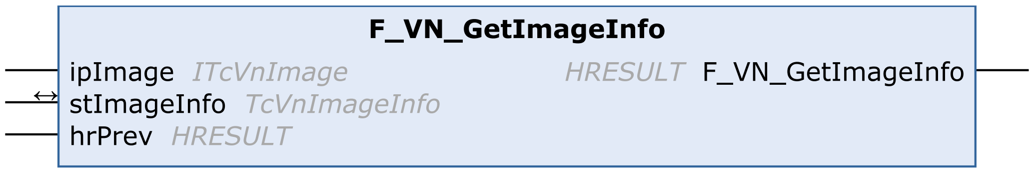
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.23 F_VN_GetImageInfo



Gets a struct containing all common meta infos of the image. (Alternatively use interface method .GetImageInfo.)

Syntax

Definition:

```
FUNCTION F_VN_GetImageInfo : HRESULT
VAR_INPUT
    ipImage     : ITcVnImage;
END_VAR
VAR_IN_OUT
    stImageInfo : TcVnImageInfo;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	Image
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stImageInfo	TcVnImageInfo [▶ 210]	Returns a struct describing the image

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.24 F_VN_GetImageWidth

Return the width of an image. (Alternatively use interface method .GetWidth.)

Syntax

Definition:

```
FUNCTION F_VN_GetImageWidth : HRESULT
VAR_INPUT
    ipImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    nWidth  : UDINT;
END_VAR
VAR_INPUT
    hrPrev  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	Image
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nWidth	UDINT	Return the width of ipImage

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.25 F_VN_GetPixel

Gets a specific pixel.

Syntax

Definition:

```
FUNCTION F_VN_GetPixel : HRESULT
VAR_INPUT
    ipSrcImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nX          : UDINT;
    nY          : UDINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
nX	UDINT	x coordinate of the pixel
nY	UDINT	y coordinate of the pixel
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_LREAL [▶ 141]	Returns the pixel value (Unused channels are set to 0.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.26 F_VN_GetPixelFormat

Gets the pixel format of an image. (Alternatively use interface method .GetPixelFormat.)

Syntax

Definition:

```
FUNCTION F_VN_GetPixelFormat : HRESULT
VAR_INPUT
    ipImage       : ITcVnImage;
END_VAR
VAR_IN_OUT
    stPixelFormat : TcVnPixelFormat;
END_VAR
VAR_INPUT
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	Image
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stPixelFormat	TcVnPixelFormat [▶ 224]	Returns a struct describing the pixel format

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.27 F_VN_GetRoi

Gets the coordinates of the region of interest (ROI) within the image.

Syntax

Definition:

```
FUNCTION F_VN_GetRoi : HRESULT
VAR_INPUT
    ipSrcImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    stRoi      : TcVnRectangle_UDINT;
END_VAR
VAR_INPUT
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stRoi	TcVnRectangle_UDINT [▸ 225]	Returns the coordinates of the region of interest

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_GetRoi表明当前在输入图像ipSrcImage上设置了哪个感兴趣区域 [▸ 137]。例如，如果需要根据整个图像指定图像中的点，或者如果需要动态放大 ROI，则需要这些信息。

应用

例如，将一个点引用到原始图像的坐标上如下所示：

```
hr := F_VN_ContourCenterOfMass(ipContour, aCenter, hr);
hr := F_VN_GetRoi(ipImageWork, stRoi, hr);
aCenter[0] := aCenter[0] + stRoi.nX;
aCenter[1] := aCenter[1] + stRoi.nY;
```

相关函数

- F_VN_SetRoi [▶ 780]用于设置感兴趣区域
- F_VN_GetRoi [▶ 773]用于检索设定的感兴趣区域
- F_VN_ResetRoi [▶ 775]用于重设整个图像的感兴趣区域
- F_VN_CopyImageRegion [▶ 751]用于复制一个图像区域

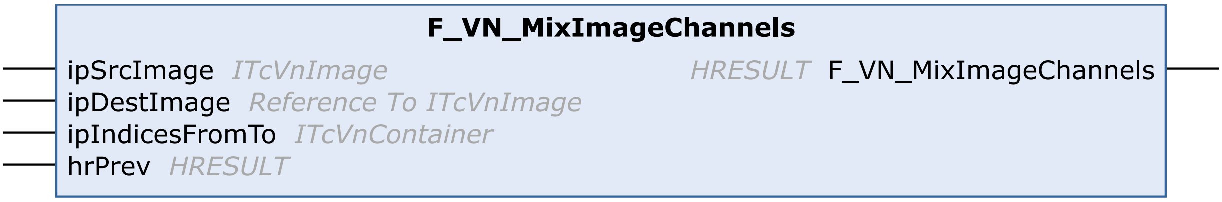
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.28 F_VN_MixImageChannels



Mix image channels by copying the specified channels of the source image into the specified channels of the destination image.

Syntax

Definition:

```
FUNCTION F_VN_MixImageChannels : HRESULT
VAR_INPUT
    ipSrcImage      : ITcVnImage;
    ipDestImage     : Reference To ITcVnImage;
    ipIndicesFromTo : ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate image will be created if required.)
ipIndicesFromTo	ITcVnContainer [▶ 349]	Index pairs (ContainerType_Vector_TcVnVector2_DINT), specifying which source channel (TcVnVector2_DINT [0]) should be copied to which destination channel (TcVnVector2_DINT [1]).
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.29 F_VN_ResetRoi

Reset the region of interest (ROI) of an image. (After this operation, the ROI is set to the entire image.)

Syntax

Definition:

```
FUNCTION F_VN_ResetRoi : HRESULT
VAR_INPUT
    ipImage : ITcVnImage;
    hrPrev  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▸ 390]	Image
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_ResetRoi重设图像的感兴趣区域 [▸ 137]。执行该函数后，图像的 ROI 在位置[0, 0]且尺寸为[ImageWidth, ImageHeight]。

应用

例如，重设 ROI 如下所示：

```
hr = F_VN_ResetRoi(ipImageWork, hr);
```

相关函数

- F_VN_SetRoi [▸ 780]用于设置感兴趣区域
- F_VN_GetRoi [▸ 773]用于检索设定的感兴趣区域
- F_VN_ResetRoi [▸ 775]用于重设整个图像的感兴趣区域

- F_VN_CopyImageRegion [▶ 751]用于复制一个图像区域

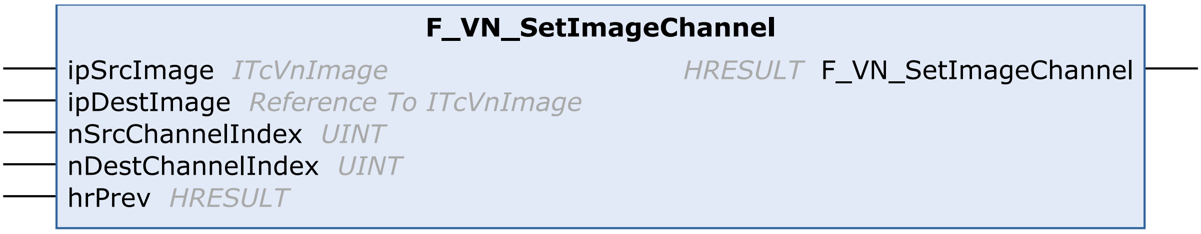
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.30 F_VN_SetImageChannel



Sets the specified destination image channel to the values of the specified source image channel.

Syntax

Definition:

```
FUNCTION F_VN_SetImageChannel : HRESULT
VAR_INPUT
    ipSrcImage         : ITcVnImage;
    ipDestImage        : Reference To ITcVnImage;
    nSrcChannelIndex   : UINT;
    nDestChannelIndex  : UINT;
    hrPrev             : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate image will be created if required.)
nSrcChannelIndex	UINT	Index of the source image channel
nDestChannelIndex	UINT	Index of the destination image channel, which will be replaced by the specified source image channel
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

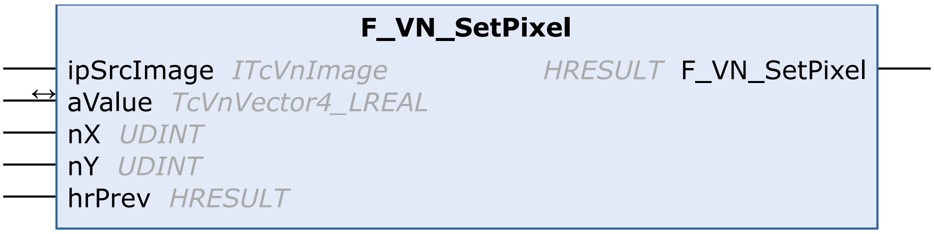
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.31 F_VN_SetPixel



Sets a specific pixel.

Syntax

Definition:

```
FUNCTION F_VN_SetPixel : HRESULT
VAR_INPUT
    ipSrcImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aValue     : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nX         : UDINT;
    nY         : UDINT;
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
nX	UDINT	x coordinate of the pixel
nY	UDINT	y coordinate of the pixel
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_LREAL [▶ 141]	The pixel value to set (additional channels are ignored.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.32 F_VN_SetPixels

Sets all pixels of an image to a given value.

Syntax

Definition:

```
FUNCTION F_VN_SetPixels : HRESULT
VAR_INPUT
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▸ 390]	Destination image
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_LREAL [▸ 141]	Pixel value

Return value

HRESULT [▸ 122]

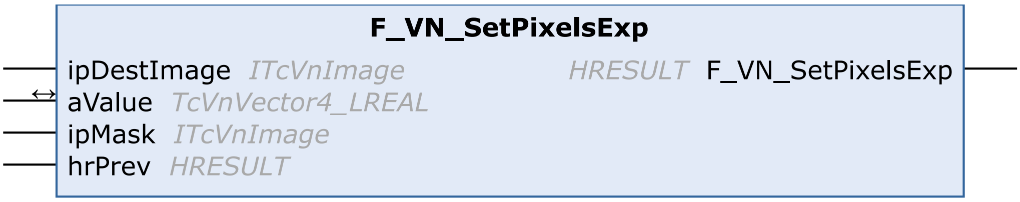
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.33 F_VN_SetPixelsExp



Sets all pixels of an image to a given value. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_SetPixelsExp : HRESULT
VAR_INPUT
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aValue      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipMask      : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▶ 390]	Destination image
ipMask	ITcVnImage [▶ 390]	Mask image
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aValue	TcVnVector4_LREAL [▶ 141]	Pixel value

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.34 F_VN_SetRoi

Sets a region of interest (ROI) within an image.

Syntax

Definition:

```
FUNCTION F_VN_SetRoi : HRESULT
VAR_INPUT
    nX          : UDINT;
    nY          : UDINT;
    nWidth      : UDINT;
    nHeight     : UDINT;
    ipDestImage : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nX	UDINT	Left boundary (inclusive 0-based index)
nY	UDINT	Upper boundary (inclusive 0-based index)
nWidth	UDINT	ROI width
nHeight	UDINT	ROI height
ipDestImage	ITcVnImage [▸ 390]	Destination image
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_SetRoi设置图像的感兴趣区域 [▸ 137]。如果不单独传输矩形参数，而是作为一个类型为TcVnRectangle_UDINT [▸ 225]的结构，可以使用函数F_VN_SetRoi_TcVnRectangle_UDINT [▸ 782]。

应用

例如，设置尺寸[240, 120]且位置[50, 50] 的 ROI 如下所示：

```
hr := F_VN_SetRoi(50, 50, 240, 120, ipImageWork, hr);
```

相关函数

- F_VN_SetRoi [▸ 780]用于设置感兴趣区域
- F_VN_GetRoi [▸ 773]用于检索设定的感兴趣区域
- F_VN_ResetRoi [▸ 775]用于重设整个图像的感兴趣区域
- F_VN_CopyImageRegion [▸ 751]用于复制一个图像区域

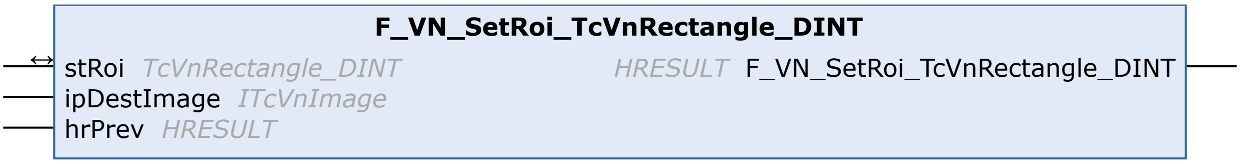
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.35 F_VN_SetRoi_TcVnRectangle_DINT



Sets a region of interest (ROI) within an image.

Syntax

Definition:

```
FUNCTION F_VN_SetRoi_TcVnRectangle_DINT : HRESULT
VAR_IN_OUT
    stRoi       : TcVnRectangle_DINT;
END_VAR
VAR_INPUT
    ipDestImage : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▶ 390]	Destination image
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stRoi	TcVnRectangle_DINT [▶ 224]	Region of interest

Return value

HRESULT [▶ 122]

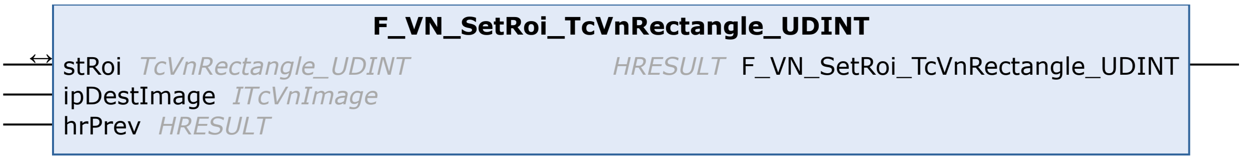
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.36 F_VN_SetRoi_TcVnRectangle_UDINT



Sets a region of interest (ROI) within an image.

Syntax

Definition:

```
FUNCTION F_VN_SetRoi_TcVnRectangle_UDINT : HRESULT
VAR_IN_OUT
    stRoi       : TcVnRectangle_UDINT;
END_VAR
VAR_INPUT
    ipDestImage : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▸ 390]	Destination image
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stRoi	TcVnRectangle_UDINT [▸ 225]	Region of interest

Return value

HRESULT [▸ 122]

更多信息

与函数 F_VN_SetRoi [▸ 780] 一样，函数 F_VN_SetRoi_TcVnRectangle_UDINT 也可设置感兴趣区域 [▸ 137]。不过，它不接受单独的矩形参数，而是将其作为 TcVnRectangle_UDINT [▸ 225] 类型的结构接受。

例如，这使得直接使用函数 F_VN_UprightBoundingRectangle [▸ 947] 的结果设置 ROI 成为可能。

应用

围绕发现的轮廓 [▸ 135]设置 ROI 的过程如下所示：

```
hr := F_VN_UprightBoundingRectangle(ipContour, stRoi, hr);
hr := F_VN_SetRoi_TcVnRectangle_UDINT(stRoi, ipImageWork, hr);
```

相关函数

- F_VN_SetRoi [▸ 780]用于设置感兴趣区域
- F_VN_GetRoi [▸ 773]用于检索设定的感兴趣区域
- F_VN_ResetRoi [▸ 775]用于重设整个图像的感兴趣区域
- F_VN_CopyImageRegion [▸ 751]用于复制一个图像区域

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.37 F_VN_SplitImageChannels

Split a multi-channel image into multiple single channel images (1 for each source image channel).

Syntax

Definition:

```
FUNCTION F_VN_SplitImageChannels : HRESULT
VAR_INPUT
    ipSrcImage      : ITcVnImage;
    pDestImages     : PVOID;
    nArraySize      : UINT;
    hrPrev          : HRESULT;
END_VAR
VAR_OUTPUT
    nImageChannels : UINT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image with multiple channels
pDestImages	PVOID	Pointer to an array of ITcVnImage (appropriate destination images will be created if required)
nArraySize	UINT	Number of pDestImages array elements (array must be >= ipSourceImage channels)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Outputs

Name	Type	Description
nImageChannels	UINT	Actual source image channels

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.38 F_VN_TransformIntoDisplayableImage

Transform an image into a displayable image. The source image will be released and zeroed while existing destination images will be released and overwritten with the function result. The source image must not be used anywhere else and the transformation is very fast. If you want to use ipSrcImage after this function call, use F_VN_CopyIntoDisplayableImage instead.

Syntax

Definition:

```
FUNCTION F_VN_TransformIntoDisplayableImage : HRESULT
VAR_INPUT
    ipSrcImage  : Reference To ITcVnImage;
    ipDestImage : Reference To ITcVnDisplayableImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	Reference To ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnDisplayableImage [▶ 390]	Returns the displayable image
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

关于更多详情，请参见可显示的图像 [▶ 131]。

注意

与 Ads Communicator 对象一起使用

请注意以下情况：待转换图像来自于 Ads Communicator 对象处于活动状态的相机对象。这时，可能会出现这样的情况：图像在物体中没有被内部释放，因此不能被转化为可显示的图像。在这种情况下，bAllowDeepCopy := TRUE时，使用F_VN_CopyIntoDisplayableImage [▶ 753]或F_VN_TransformIntoDisplayableImageExp [▶ 785]。

相关函数

- F_VN_CopyIntoDisplayableImage [▸ 753]
- F_VN_TransformIntoDisplayableImageExp [▸ 785]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.5.39 F_VN_TransformIntoDisplayableImageExp

Transform an image into a displayable image. The source image will be released and zeroed while existing destination images will be released and overwritten with the function result. If bAllowDeepCopy equals false, the source image must not be used anywhere else and the transformation is always very fast. Otherwise, if bAllowDeepCopy equals true, a deep copy of the source image might be created if required, which will result in a longer execution time. If you want to use ipSrcImage after this function call, use F_VN_CopyIntoDisplayableImage instead. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_TransformIntoDisplayableImageExp : HRESULT
VAR_INPUT
    ipSrcImage     : Reference To ITcVnImage;
    ipDestImage    : Reference To ITcVnDisplayableImage;
    bAllowDeepCopy : BOOL;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	Reference To ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnDisplayableImage [▸ 390]	Returns the displayable image
bAllowDeepCopy	BOOL	Allow deep image copies, if required
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

关于更多详情，请参见可显示的图像 [▶ 131]。

参数

深度复制的许可

深度复制的许可（TRUE）可以通过参数bAllowDeepCopy提供。如果待转换的图像来自Ads Communicator 对象 [▶ 109]已激活且因此图像可以内部引用的相机对象，那么这一点十分重要。

如果未提供许可（FALSE），也可以使用标准版本F_VN_TransformIntoDisplayableImage [▶ 784]。

相关函数

- F_VN_CopyIntoDisplayableImage [▶ 753]
- F_VN_TransformIntoDisplayableImage [▶ 784]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.6 Code Quality

还请参阅有关此

- HRESULT [▶ 122]
- ADS 返回代码 [▶ 2753]
- F_VN_GradeBarcode [▶ 786]
- F_VN_GradeDataMatrixCode [▶ 790]
- F_VN_GradeQRCode [▶ 793]

6.1.4.6.1 F_VN_GradeBarcode

Grades a 1D barcode according to ISO / IEC 15416:2016. Requires uniform illumination, good focus and the image sensor plane parallel to the code plane to achieve meaningful results.

Syntax

Definition:

```
FUNCTION F_VN_GradeBarcode : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipDecodedData : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    stCodeGrades   : TcVnCodeGrades1D;
END_VAR
VAR_INPUT
```

```
    eBarcodeType    : ETcVnBarcodeType;
    hrPrev          : HRESULT;
END_VAR
VAR_OUTPUT
    fAngleDeg       : REAL;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image, only containing the horizontally aligned barcode region (USINT, 1 channel, including quiet zones on left and right, excluding Text or empty space on more than 10% total height on top or bottom)
ipDecodedData	Reference To ITcVnContainer [▸ 349]	Returns the decoded data (ContainerType_String_SINT)
eBarcodeType	ETcVnBarcodeType [▸ 143]	Type of the barcode (supported: CODE39, CODE128, EAN8, EAN13, ITF, UPCA, UPCE)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stCodeGrades	TcVnCodeGrades1D [▸ 208]	Returns the code grades in the range from 0 (very bad) to 4 (very good)

Outputs

Name	Type	Description
fAngleDeg	REAL	Returns the approximate clockwise rotation angle in degree

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Code Quality

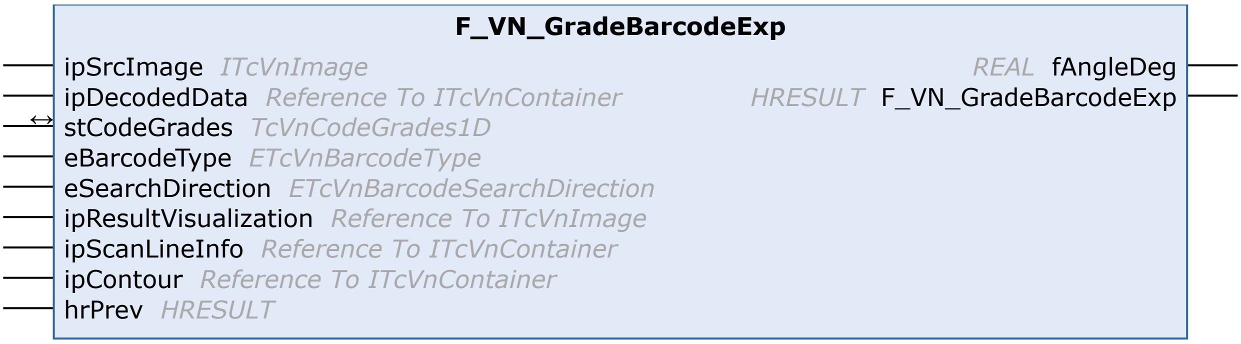
System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

还请参阅有关此

F_VN_GradeBarcodeExp [▸ 788]

6.1.4.6.2 F_VN_GradeBarcodeExp



Grades a 1D barcode according to ISO / IEC 15416:2016. Requires uniform illumination, good focus and the image sensor plane parallel to the code plane to achieve meaningful results. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_GradeBarcodeExp : HRESULT
VAR_INPUT
    ipSrcImage              : ITcVnImage;
    ipDecodedData           : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    stCodeGrades            : TcVnCodeGrades1D;
END_VAR
VAR_INPUT
    eBarcodeType            : ETcVnBarcodeType;
    eSearchDirection        : ETcVnBarcodeSearchDirection;
    ipResultVisualization   : Reference To ITcVnImage;
    ipScanLineInfo          : Reference To ITcVnContainer;
    ipContour               : Reference To ITcVnContainer;
    hrPrev                  : HRESULT;
END_VAR
VAR_OUTPUT
    fAngleDeg               : REAL;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image, only containing the horizontally aligned barcode region (USINT, 1 channel, including quiet zones on left and right, excluding Text or empty space on more than 10% total height on top or bottom)
ipDecodedData	Reference To ITcVnContainer [▶ 349]	Returns the decoded data (ContainerType_String_SINT)
eBarcodeType	ETcVnBarcodeType [▶ 143]	Type of the barcode (supported: CODE39, CODE128, EAN8, EAN13, ITF, UPCA, UPCE)
eSearchDirection	ETcVnBarcodeSearchDirection [▶ 143]	Barcode search direction (BSD_ANY first tries horizontal, then vertical)
ipResultVisualization	Reference To ITcVnImage [▶ 390]	Returns a visualization of the scan lines (optional, set to 0 if not required). Grades are color coded: >= 3 as green, >= 2 as light blue, >= 1 as orange, < 1 as red
ipScanLineInfo	Reference To ITcVnContainer [▶ 349]	Returns additional scan line info for further evaluation, e.g. to use PlotIntensityProfile for visualization (optional, set to 0 if not required; ContainerType_Vector_TcVnVector3_REAL, containing [row index, overall grade, threshold])
ipContour	Reference To ITcVnContainer [▶ 349]	Returns the 4 approximate code corner points, including the quiet zone (optional, set to 0 if not required; ContainerType_Vector_TcVnPoint2_DINT)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stCodeGrades	TcVnCodeGrades1D [▶ 208]	Returns the code grades in the range from 0 (very bad) to 4 (very good)

Outputs

Name	Type	Description
fAngleDeg	REAL	Returns the approximate clockwise rotation angle in degree

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Code Quality

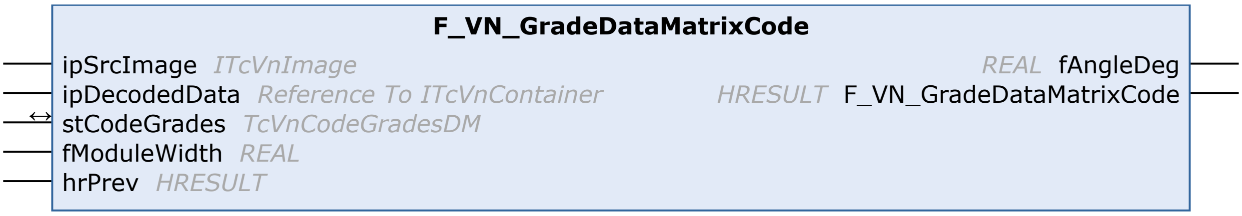
System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

还请参阅有关此

F_VN_GradeBarcode [▶ 786]
Contour Analysis [▶ 913]
F_VN_PlotIntensityProfile [▶ 1007]
F_VN_DrawContours [▶ 967]

6.1.4.6.3 F_VN_GradeDataMatrixCode



Grades a Data Matrix code (ECC200) according to ISO / IEC 15415:2011. Requires uniform illumination, good focus and the image sensor plane parallel to the code plane to achieve meaningful results.

Syntax

Definition:

```
FUNCTION F_VN_GradeDataMatrixCode : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipDecodedData : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    stCodeGrades  : TcVnCodeGradesDM;
END_VAR
VAR_INPUT
    fModuleWidth  : REAL;
    hrPrev        : HRESULT;
END_VAR
VAR_OUTPUT
    fAngleDeg     : REAL;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image, only containing the Data Matrix code region (USINT, 1 channel, including quiet zone)
ipDecodedData	Reference To ITcVnContainer [▶ 349]	Returns the decoded data (ContainerType_String_SINT)
fModuleWidth	REAL	Minimum module width of the code in the image (in pixels). Must be at least 3, preferably 5 - 8.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stCodeGrades	TcVnCodeGradesDM [▶ 208]	Returns the code grades in the range from 0 (very bad) to 4 (very good)

Outputs

Name	Type	Description
fAngleDeg	REAL	Returns the clockwise rotation angle in degree

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Code Quality

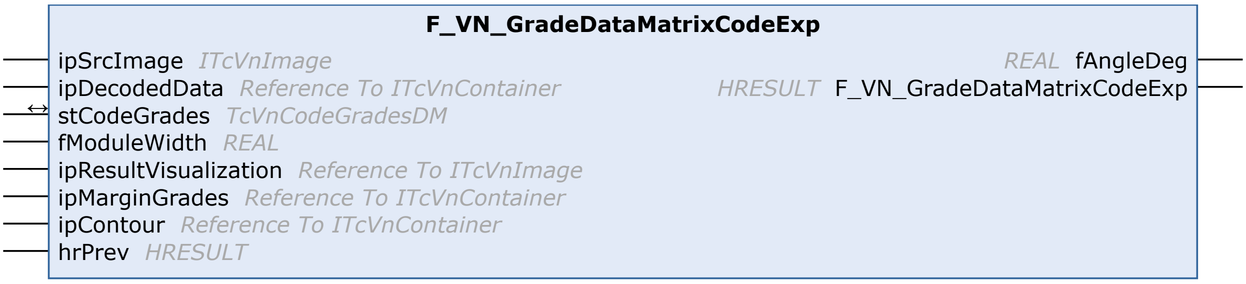
System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

还请参阅有关此

F_VN_GradeDataMatrixCodeExp [▶ 791]

6.1.4.6.4 F_VN_GradeDataMatrixCodeExp



Grades a Data Matrix code (ECC200) according to ISO / IEC 15415:2011. Requires uniform illumination, good focus and the image sensor plane parallel to the code plane to achieve meaningful results. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_GradeDataMatrixCodeExp : HRESULT
VAR_INPUT
    ipSrcImage              : ITcVnImage;
    ipDecodedData           : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    stCodeGrades            : TcVnCodeGradesDM;
END_VAR
VAR_INPUT
    fModuleWidth            : REAL;
    ipResultVisualization : Reference To ITcVnImage;
    ipMarginGrades          : Reference To ITcVnContainer;
    ipContour               : Reference To ITcVnContainer;
    hrPrev                  : HRESULT;
END_VAR
VAR_OUTPUT
    fAngleDeg               : REAL;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image, only containing the Data Matrix code region (USINT, 1 channel, including quiet zone)
ipDecodedData	Reference To ITcVnContainer [▶ 349]	Returns the decoded data (ContainerType_String_SINT)
fModuleWidth	REAL	Minimum module width of the code in the image (in pixels). Must be at least 3, preferably 5 - 8.
ipResultVisualization	Reference To ITcVnImage [▶ 390]	Returns a visualization of the scan grid (optional, set to 0 if not required). Module margin grades are color coded: >= 3 as green, >= 2 as light blue, >= 1 as orange, < 1 as red
ipMarginGrades	Reference To ITcVnContainer [▶ 349]	Returns the individual margin grade for each module, e.g. to create a custom visualization (optional, set to 0 if not required; ContainerType_Vector_TcVnVector3_REAL, containing [x-position, y-position, grade])
ipContour	Reference To ITcVnContainer [▶ 349]	Returns the 4 code corner points, excluding the quiet zone (optional, set to 0 if not required; ContainerType_Vector_TcVnPoint2_DINT)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stCodeGrades	TcVnCodeGradesDM [▶ 208]	Returns the code grades in the range from 0 (very bad) to 4 (very good)

Outputs

Name	Type	Description
fAngleDeg	REAL	Returns the clockwise rotation angle in degree

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Code Quality

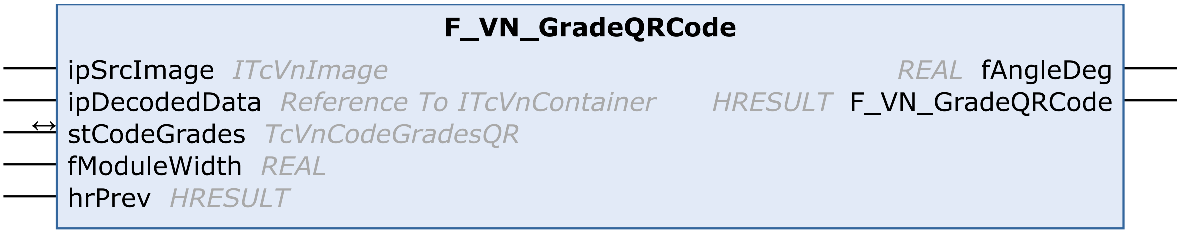
System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

还请参阅有关此

- F_VN_GradeDataMatrixCode [▶ 790]
- F_VN_DrawContours [▶ 967]
- Contour Analysis [▶ 913]

6.1.4.6.5 F_VN_GradeQRCode



Grades a QR code according to ISO / IEC 15415:2011. Requires uniform illumination, good focus and the image sensor plane parallel to the code plane to achieve meaningful results.

Syntax

Definition:

```
FUNCTION F_VN_GradeQRCode : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipDecodedData : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    stCodeGrades  : TcVnCodeGradesQR;
END_VAR
VAR_INPUT
    fModuleWidth  : REAL;
    hrPrev        : HRESULT;
END_VAR
VAR_OUTPUT
    fAngleDeg     : REAL;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image, only containing the QR code region (USINT, 1 channel, including quiet zone)
ipDecodedData	Reference To ITcVnContainer [▶ 349]	Returns the decoded data (ContainerType_String_SINT)
fModuleWidth	REAL	Minimum module width of the code in the image (in pixels). Must be at least 3, preferably 5 - 8.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stCodeGrades	TcVnCodeGradesQR [▶ 209]	Returns the code grades in the range from 0 (very bad) to 4 (very good)

Outputs

Name	Type	Description
fAngleDeg	REAL	Returns the clockwise rotation angle in degree

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Code Quality

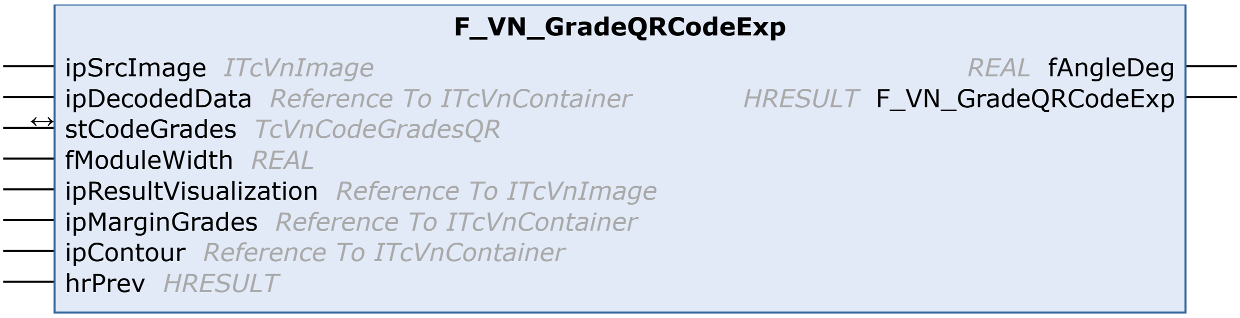
System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

还请参阅有关此

F_VN_GradeQRCodeExp [▸ 794]

6.1.4.6.6 F_VN_GradeQRCodeExp



Grades a QR code according to ISO / IEC 15415:2011. Requires uniform illumination, good focus and the image sensor plane parallel to the code plane to achieve meaningful results. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_GradeQRCodeExp : HRESULT
VAR_INPUT
    ipSrcImage              : ITcVnImage;
    ipDecodedData           : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    stCodeGrades            : TcVnCodeGradesQR;
END_VAR
VAR_INPUT
    fModuleWidth            : REAL;
    ipResultVisualization : Reference To ITcVnImage;
    ipMarginGrades          : Reference To ITcVnContainer;
    ipContour               : Reference To ITcVnContainer;
    hrPrev                  : HRESULT;
END_VAR
VAR_OUTPUT
    fAngleDeg               : REAL;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image, only containing the QR code region (USINT, 1 channel, including quiet zone)
ipDecodedData	Reference To ITcVnContainer [▸ 349]	Returns the decoded data (ContainerType_String_SINT)
fModuleWidth	REAL	Minimum module width of the code in the image (in pixels). Must be at least 3, preferably 5 - 8.
ipResultVisualization	Reference To ITcVnImage [▸ 390]	Returns a visualization of the scan grid (optional, set to 0 if not required). Module margin grades are color coded: >= 3 as green, >= 2 as light blue, >= 1 as orange, < 1 as red
ipMarginGrades	Reference To ITcVnContainer [▸ 349]	Returns the individual margin grade for each module, e.g. to create a custom visualization (optional, set to 0 if not required; ContainerType_Vector_TcVnVector3_REAL, containing [x-position, y-position, grade])
ipContour	Reference To ITcVnContainer [▸ 349]	Returns the 4 code corner points, excluding the quiet zone (optional, set to 0 if not required; ContainerType_Vector_TcVnPoint2_DINT)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stCodeGrades	TcVnCodeGradesQR [▸ 209]	Returns the code grades in the range from 0 (very bad) to 4 (very good)

Outputs

Name	Type	Description
fAngleDeg	REAL	Returns the clockwise rotation angle in degree

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Code Quality

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

还请参阅有关此

- F_VN_DrawContours [▸ 967]
- Contour Analysis [▸ 913]
- F_VN_GradeQRCode [▸ 793]

6.1.4.7 Code Reading

该组包含检测和读取一维和二维代码的函数。

函数

一维代码

- F_VN_ReadBarcode(Exp) [▶ 799]
  - CodaBar
  - Code39
  - Code93
  - Code128
  - EAN8
  - EAN13
  - ITF
  - UPCA
  - UPCE
  - Code39Extended
- F_VN_ReadPharmaCode(Exp) [▶ 810]

二维代码

- F_VN_ReadDataMatrixCode(Exp) [▶ 805]
- F_VN_ReadQRCode(Exp) [▶ 815]

代码要求

一般来说，模块大小至少需要三个像素；建议超过三个像素，以获得更稳定的结果。图像应该清晰且具有很高的对比度。此外，建议代码方向为大约 0° 或 90°，以便使代码与图像轴同步。

HRESULT 的解释

对于读码功能，HRESULT [▶ 122]返回值的使用如下：

代码	名称	描述
16#000	S_OK	函数已成功执行且所有预期代码已找到（对于标准函数至少一个代码，而对于专家函数，则有nCodeNumber）。
16#001	S_FALSE	函数已成功执行且所有预期代码已找到（对于标准函数至少一个代码，而对于专家函数，则有nCodeNumber）。
16#256	S_WATCHDOGTIMEOUT	函数被看门狗中止。一些读码功能可以返回部分结果（见单个函数描述）。
16#7xx	所有错误代码	函数未成功执行。另请参见：ADS 返回代码 [▶ 2753]

因此，通过SUCCEEDED()进行的常规检查不足以推断出找到的代码，从而推断出ipDecodedData中的现有数据。以下查询可用于此目的：

```
IF hr = S_OK THEN
    // Export Code into String
    hr := F_VN_ExportSubContainer_String(ipDecodedCode, 0, sCodeAsString, 255, hr);
    // Use sCodeAsString
ELSIF SUCCEEDED(hr) THEN
    // Process partial results
ELSE
    // Error handling
END_IF
```

二维代码的搜索策略

对于二维代码，搜索算法被应用于输入图像的不同变体，以便尽可能多地识别代码。图像的反转和镜像都被用来生成各种版本。对哪个变体进行测试的决定被称为搜索策略。为了更好地控制函数的运行时间，可以对这种搜索策略进行配置。枚举ETcVn2dCodeSearchStrategy [▶ 141]可用于此目的。通过枚举，可以为每个可能的图像转换（反转和镜像）指定是否应用，以及是否要首先检查原始图像或转换后的图像的代码。

函数参数eSearchStrategy正是为相应的读码专家函数定义了这种搜索策略。下面的表格显示了不同的组合：

表 1: 反转

电子搜索策略	原始图像	反转的图像
TCVN_CSS_ONLY_NOT_INVERTED	1.	-
TCVN_CSS_FIRST_NOT_INVERTED	1.	2.
TCVN_CSS_ONLY_INVERTED	-	1.
TCVN_CSS_FIRST_INVERTED	2.	1.

表 2: 镜像

电子搜索策略	原始图像	镜像图像
TCVN_CSS_ONLY_NOT_FLIPPED	1.	-
TCVN_CSS_FIRST_NOT_FLIPPED	1.	2.
TCVN_CSS_ONLY_FLIPPED	-	1.
TCVN_CSS_FIRST_FLIPPED	2.	1.

可以为每一种类型的转化定义一个搜索策略。不同的搜索策略可以通过以下方式链接：

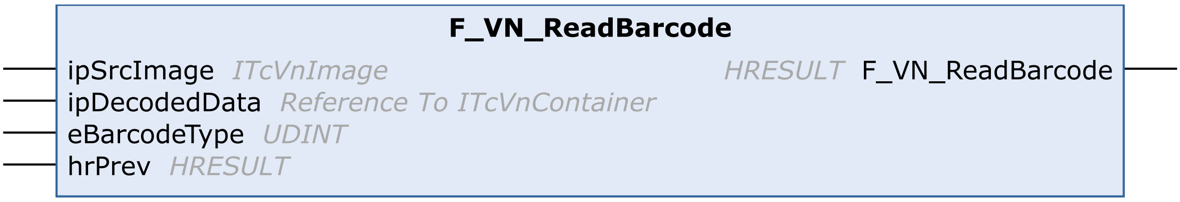
```
eSearchStrategy := TCVN_CSS_ONLY_FLIPPED + TCVN_CSS_FIRST_INVERTED;
```

TCVN_CSS_DEFAULT选项可用于选择默认设置，而默认设置因代码类型不同而不同。

无法实现搜索策略的链接

不允许将同一转化类型的几个搜索策略联系在一起。只有一个反转的策略和一个镜像的策略可以相互联系。

6.1.4.7.1 F_VN_ReadBarcode



Detect and interpret a 1d barcode within the provided image. Can be canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_ReadBarcode : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipDecodedData : Reference To ITcVnContainer;
    eBarcodeType   : UDINT;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (USINT elements, 1 channel or 3 channel (3 channel input is expected to be RGB and internally converted to Gray))
ipDecodedData	Reference To ITcVnContainer [▸ 349]	Returns the decoded code (ContainerType_Vector_String_SINT)
eBarcodeType	UDINT	Types of barcode to search for (ETcVnBarcodeType)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_ReadBarcode检测图像中的条形码并进行读取。

算法

在图像中，该函数首先搜索水平排列的代码，然后搜索垂直排列的代码。

参数

输入图像

输入图像ipSrcImage必须为元素类型USINT（8 位）的 1 通道或 3 通道图像。3 通道图像将被解释为 RGB 并在内部转换为灰度图像。

阅读数据

从代码中读取的数据在容器ipDecodedData中以字符串形式返回。容器类型是ContainerType_Vector_String_SINT。

条形码类型

参数eBarcodeType定义了要在图像中搜索的条形码类型。枚举ETcVnBarcodeType [▸ 143]中支持所有条码类型。可以选择一个以上的类型。建议将选择限制在最终应用中实际出现的类型，以改善运行时间。

```
eBarcodeType := TCVN_BT_EAN13 + TCVN_BT_CODE128;
```

使用TCVN_BT_ANY搜索所有可用的条形码类型。

专家参数

专家版F_VN_ReadBarcodeExp [▸ 799]包含额外的参数。

从容器中读取数据

如果成功找到并解码了代码，代码内容将在容器ipDecodedData中返回，并可以使用函数F_VN_ExportSubContainer_String [▸ 732]导出为一个字符串。

```
hr := F_VN_ExportSubContainer_String( ipDecodedData, 0, sText, nMaxLength, hr);
```

样本

- EAN-13 条形码读取 [▸ 2645]
- 读码过程中的结果评估 [▸ 2653]

相关函数

- F_VN_ReadBarcode [▶ 797]
- F_VN_ReadPharmaCode [▶ 808]
- F_VN_ReadDataMatrixCode [▶ 803]
- F_VN_ReadQRCode [▶ 813]

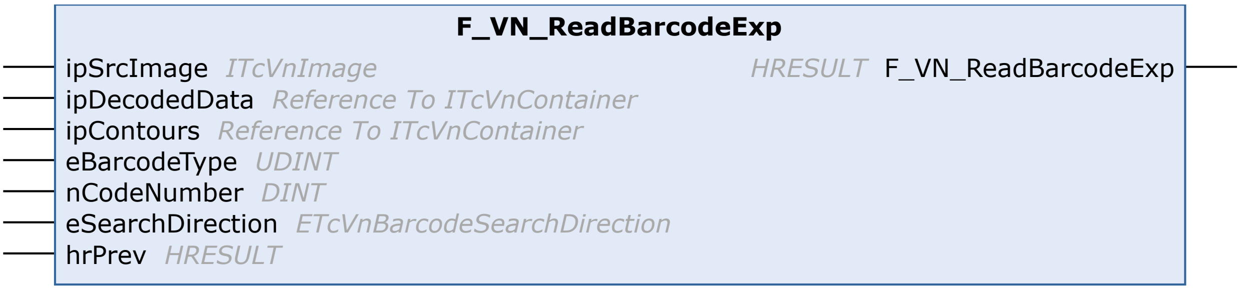
Required License

TC3 Vision Code Reading

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.7.2 F_VN_ReadBarcodeExp



Detect and interpret a 1d barcode within the provided image. Can be canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_ReadBarcodeExp : HRESULT
VAR_INPUT
    ipSrcImage        : ITcVnImage;
    ipDecodedData     : Reference To ITcVnContainer;
    ipContours        : Reference To ITcVnContainer;
    eBarcodeType      : UDINT;
    nCodeNumber       : DINT;
    eSearchDirection  : ETcVnBarcodeSearchDirection;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT elements, 1 channel or 3 channel (3 channel input is expected to be RGB and internally converted to Gray))
ipDecodedData	Reference To ITcVnContainer [▶ 349]	Returns the decoded code (ContainerType_Vector_String_SINT)
ipContours	Reference To ITcVnContainer [▶ 349]	Returns the code positions as contours (optional, set to 0 if not required; ContainerType_Vector_Vector_TcVnPoint2_DINT)
eBarcodeType	UDINT	Types of barcode to search for (ETcVnBarcodeType)
nCodeNumber	DINT	Number of 1d barcode that should be detected within the provided image (currently only 1 supported).
eSearchDirection	ETcVnBarcodeSearchDirection [▶ 143]	Barcode search direction.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

该函数F_VN_ReadBarcodeExp是F_VN_ReadBarcode [▶ 797]的专家版本。它包含用于返回代码位置和设置搜索方向的额外参数。

参数

输入图像

输入图像ipSrcImage必须为元素类型USINT（8 位）的 1 通道或 3 通道图像。3 通道图像将被解释为 RGB 并在内部转换为灰度图像。

阅读数据

从代码中读取的数据在容器ipDecodedData中以字符串形式返回。容器类型是ContainerType_Vector_String_SINT。

条形码类型

参数eBarcodeType定义了要在图像中搜索的条形码类型。枚举ETcVnBarcodeType [▶ 143]中支持所有条码类型。可以选择一个以上的类型。建议将选择限制在最终应用中实际出现的类型，以改善运行时间。

```
eBarcodeType := TCVN_BT_EAN13 + TCVN_BT_CODE128;
```

使用TCVN_BT_ANY搜索所有可用的条形码类型。

代码位置

参数ipContour返回找到的代码位置，作为轮廓。

编码数量

参数nCodeNumber定义了需要搜索多少个代码。

不支持搜索一个以上的条形码
目前不支持在一张图像中搜索多个条形码。因此，参数 nCodeNumber 必须设置为 1。

搜索方向

搜索方向可以通过eSearchDirection指定：

- TCVN_BSD_ANY首先在水平方向搜索，然后在垂直方向搜索。
- TCVN_BSD_HORIZONTAL只在水平方向上搜索。
- TCVN_BSD_VERTICAL只在垂直方向上搜索。

样本

- EAN-13 条形码读取 [▸ 2645]
- 读码过程中的结果评估 [▸ 2653]

相关函数

- F_VN_ReadBarcode [▸ 797]
- F_VN_ReadPharmaCode [▸ 808]
- F_VN_ReadDataMatrixCode [▸ 803]
- F_VN_ReadQRCode [▸ 813]

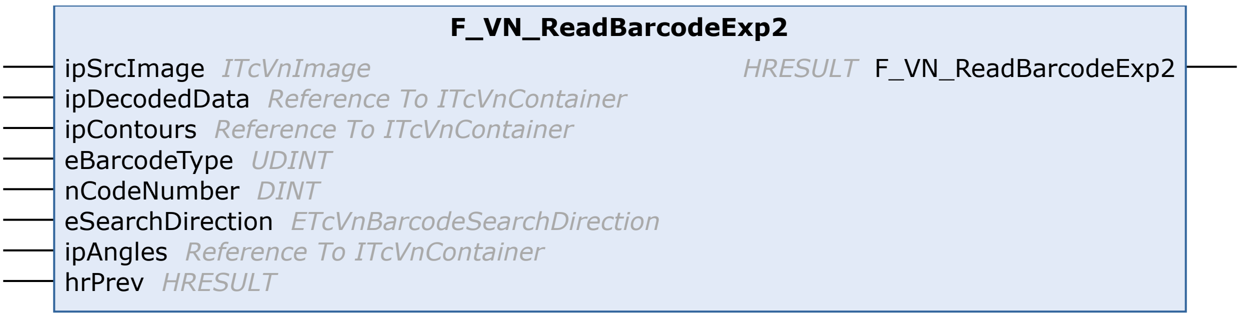
Required License

TC3 Vision Code Reading

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.7.3 F_VN_ReadBarcodeExp2



Detect and interpret a 1d barcode within the provided image. Can be canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_ReadBarcodeExp2 : HRESULT
VAR_INPUT
    ipSrcImage       : ITcVnImage;
    ipDecodedData    : Reference To ITcVnContainer;
    ipContours       : Reference To ITcVnContainer;
    eBarcodeType     : UDINT;
    nCodeNumber      : DINT;
    eSearchDirection : ETcVnBarcodeSearchDirection;
```

```
    ipAngles           : Reference To ITcVnContainer;
    hrPrev             : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT elements, 1 channel or 3 channel (3 channel input is expected to be RGB and internally converted to Gray))
ipDecodedData	Reference To ITcVnContainer [▶ 349]	Returns the decoded code (ContainerType_Vector_String_SINT)
ipContours	Reference To ITcVnContainer [▶ 349]	Returns the code positions as contours (optional, set to 0 if not required; ContainerType_Vector_Vector_TcVnPoint2_DINT)
eBarcodeType	UDINT	Types of barcode to search for (ETcVnBarcodeType)
nCodeNumber	DINT	Number of 1d barcode that should be detected within the provided image (currently only 1 supported).
eSearchDirection	ETcVnBarcodeSearchDirection [▶ 143]	Barcode search direction.
ipAngles	Reference To ITcVnContainer [▶ 349]	Returns the clockwise code rotation angles (optional, set to 0 if not required; ContainerType_Vector_REAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

该函数F_VN_ReadBarcodeExp是F_VN_ReadBarcode [▶ 797]的专家版本。它包含用于返回代码位置和设置搜索方向的额外参数。

参数

输入图像

输入图像ipSrcImage必须为元素类型USINT（8 位）的 1 通道或 3 通道图像。3 通道图像将被解释为 RGB 并在内部转换为灰度图像。

阅读数据

从代码中读取的数据在容器ipDecodedData中以字符串形式返回。容器类型是ContainerType_Vector_String_SINT。

条形码类型

参数eBarcodeType定义了要在图像中搜索的条形码类型。枚举ETcVnBarcodeType [▶ 143]中支持所有条码类型。可以选择一个以上的类型。建议将选择限制在最终应用中实际出现的类型，以改善运行时间。

```
eBarcodeType := TCVN_BT_EAN13 + TCVN_BT_CODE128;
```

使用TCVN_BT_ANY搜索所有可用的条形码类型。

代码位置

参数ipContour返回找到的代码位置，作为轮廓。

编码数量

参数nCodeNumber定义了需要搜索多少个代码。

> **不支持搜索一个以上的条形码**
> 目前不支持在一张图像中搜索多个条形码。因此，参数 nCodeNumber 必须设置为 1。

搜索方向

搜索方向可以通过eSearchDirection指定：

- TCVN_BSD_ANY首先在水平方向搜索，然后在垂直方向搜索。
- TCVN_BSD_HORIZONTAL只在水平方向上搜索。
- TCVN_BSD_VERTICAL只在垂直方向上搜索。

样本

- EAN-13 条形码读取 [▶ 2645]
- 读码过程中的结果评估 [▶ 2653]

相关函数

- F_VN_ReadBarcode [▶ 797]
- F_VN_ReadPharmaCode [▶ 808]
- F_VN_ReadDataMatrixCode [▶ 803]
- F_VN_ReadQRCode [▶ 813]

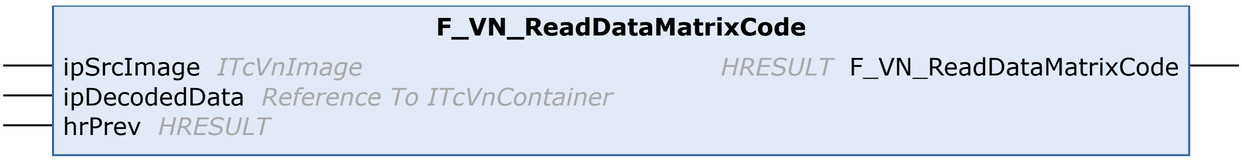
Required License

TC3 Vision Code Reading

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.7.4 F_VN_ReadDataMatrixCode



Detect and interpret a data matrix code (ECC200) within the provided image.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_ReadDataMatrixCode : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipDecodedData : Reference To ITcVnContainer;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT elements, 1 or 3 channels)
ipDecodedData	Reference To ITcVnContainer [▶ 349]	Returns the decoded code (ContainerType_Vector_String_SINT)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

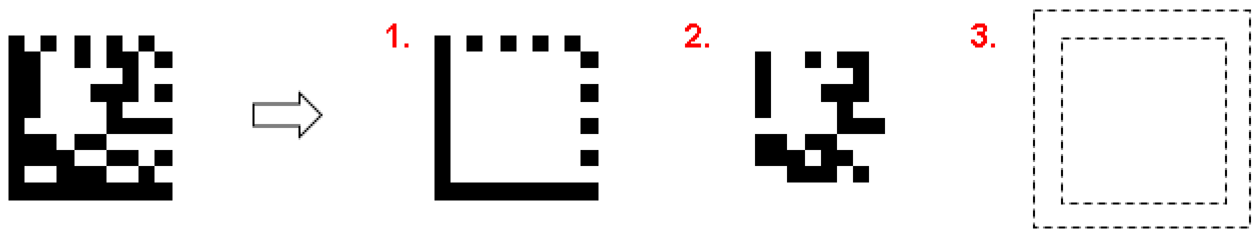
更多信息

符合 IEC 16022 标准的数据矩阵条形码是一种二维码，便于以最小所需空间存储大量数据。因此，它通常用于组件和产品标记。

函数 F_VN_ReadDataMatrixCode 只能识别和解码图像中的 ECC 200 代码类型。这以每图像一个代码为基础。如果要在一个图像中识别和解码多个数据矩阵代码，必须使用函数 F_VN_ReadDataMatrixCodeExp [▶ 805]。

数据矩阵代码结构

数据矩阵代码由矩形或圆点组成一个正方形或长方形符号，而后者又可分为三个区域：



1. 寻像模板
   实线定位器 (L) 用于定位和对齐代码，而模板线定位器 (P) 则用于确定单元大小和排列。
2. 数据区
   数据区由寻像模板包围。
3. 静区
   静区是代码背景色中寻像模板周围至少 1.5 个单元宽的区域。

寻像模板或静音区中的缺陷/损坏可能会导致无法检测到代码。

错误纠正

数据矩阵代码 ECC 200（ECC = 错误纠正代码）是普遍接受的版本，因为它的纠正效果更好。除了实际数据外，它们还根据里德所罗门代码进行冗余编码。该版本的数据矩阵代码可通过该函数解码。如有损坏，可执行错误纠正。

代码搜索

函数 F_VN_ReadDataMatrixCode 从图像中心线开始搜索图像中的数据矩阵代码。

从容器中读取数据

如果成功找到并解码了代码，代码内容将在容器ipDecodedData中返回，并可以使用函数 F_VN_ExportSubContainer_String [▶ 732]导出为一个字符串。

```
hr := F_VN_ExportSubContainer_String( ipDecodedData, 0, sText, nMaxLength, hr);
```

HRESULT

与所有 TwinCAT Vision API 函数一样，该函数也返回一个HRESULT [▶ 122]，以表明执行是否成功。在成功的情况下，在随后对结果的处理中，可以区分以下成功代码。为了区分这些情况，HRESULT变量可以直接与S_OK或S_FALSE进行比较。

代码	名称	描述
16#000	S_OK	该功能成功执行。 在图像中发现了指定数量的代码。
16#001	S_FALSE	该功能成功执行。 发现的代码比图片中显示的少。
16#256	S_WATCHDOGTIMEOUT	该功能被看门狗 [▶ 127]中止。

示例

- 数据矩阵码读取 [▶ 2648]
- 读码过程中的结果评估 [▶ 2653]

相关函数

- F_VN_ReadDataMatrixCodeExp [▶ 805]
- F_VN_ReadBarcode [▶ 797] 或 F_VN_ReadBarcodeExp [▶ 799]
- F_VN_ReadPharmaCode [▶ 808] 或 F_VN_ReadPharmaCodeExp [▶ 810]
- F_VN_ReadQRCode [▶ 813] 或 F_VN_ReadQRCodeExp [▶ 815]

Required License

TC3 Vision Code Reading

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.7.5 F_VN_ReadDataMatrixCodeExp

Detect and interpret a data matrix code (ECC200) within the provided image.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_ReadDataMatrixCodeExp : HRESULT
VAR_INPUT
    ipSrcImage      : ITcVnImage;
    ipDecodedData   : Reference To ITcVnContainer;
    ipContours      : Reference To ITcVnContainer;
    nCodeNumber     : DINT;
    eSearchStrategy : UDINT;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (USINT elements, 1 or 3 channels)
ipDecodedData	Reference To ITcVnContainer [▸ 349]	Returns the decoded code (ContainerType_Vector_String_SINT)
ipContours	Reference To ITcVnContainer [▸ 349]	Returns the code positions as contours (optional, set to 0 if not required; ContainerType_Vector_Vector_TcVnPoint2_DINT)
nCodeNumber	DINT	Number of data matrix codes that should be detected within the provided image. (If set to -1, it tries to detect all data matrix codes.)
eSearchStrategy	UDINT	Used search strategy (ETcVn2dCodeSearchStrategy)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_ReadDataMatrixCodeExp 对应于函数F_VN_ReadDataMatrixCode [▸ 803]，由以下参数扩展。

ipContours

参数ipContour返回找到的代码位置，作为轮廓。

nCodeNumber

参数nCodeNumber定义了需要搜索多少个代码。在*-1*，搜索继续进行，直到在图像中没有进一步发现代码或看门狗中止功能。

电子搜索策略

参数eSearchStrategy定义了代码搜索的搜索策略。标准选项TCVN_CSS_DEFAULT导致搜索策略TCVN_CSS_FIRST_NOT_FLIPPED。关于搜索策略的描述，见代码阅读搜索策略 [▸ 797]。

搜索策略

目前，这个功能只支持对镜像的搜索策略的设置。如果在原始图像中没有找到所有预期的代码，反转总是自动进行。

样本

- 数据矩阵码读取 [▸ 2648]
- 读码过程中的结果评估 [▸ 2653]

相关函数

- F_VN_ReadDataMatrixCode [▸ 803]
- F_VN_ReadBarcode [▸ 797]或F_VN_ReadBarcodeExp [▸ 799]
- F_VN_ReadPharmaCode [▸ 808]或F_VN_ReadPharmaCodeExp [▸ 810]
- F_VN_ReadQRCode [▸ 813]或 F_VN_ReadQRCodeExp [▸ 815]

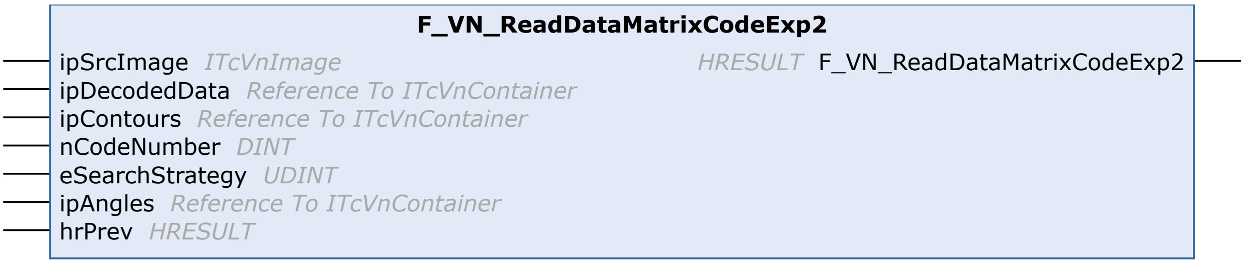
Required License

TC3 Vision Code Reading

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.7.6 F_VN_ReadDataMatrixCodeExp2



Detect and interpret a data matrix code (ECC200) within the provided image.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_ReadDataMatrixCodeExp2 : HRESULT
VAR_INPUT
    ipSrcImage       : ITcVnImage;
    ipDecodedData    : Reference To ITcVnContainer;
    ipContours       : Reference To ITcVnContainer;
    nCodeNumber      : DINT;
    eSearchStrategy  : UDINT;
    ipAngles         : Reference To ITcVnContainer;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT elements, 1 or 3 channels)
ipDecodedData	Reference To ITcVnContainer [▶ 349]	Returns the decoded code (ContainerType_Vector_String_SINT)
ipContours	Reference To ITcVnContainer [▶ 349]	Returns the code positions as contours (optional, set to 0 if not required; ContainerType_Vector_Vector_TcVnPoint2_DINT)
nCodeNumber	DINT	Number of data matrix codes that should be detected within the provided image. (If set to -1, it tries to detect all data matrix codes.)
eSearchStrategy	UDINT	Used search strategy (ETcVn2dCodeSearchStrategy)
ipAngles	Reference To ITcVnContainer [▶ 349]	Returns the clockwise code rotation angles (optional, set to 0 if not required; ContainerType_Vector_REAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_ReadDataMatrixCodeExp 对应于函数F_VN_ReadDataMatrixCode [▶ 803]，由以下参数扩展。

ipContours

参数ipContour返回找到的代码位置，作为轮廓。

nCodeNumber

参数nCodeNumber定义了需要搜索多少个代码。在*-1*，搜索继续进行，直到在图像中没有进一步发现代码或看门狗中止功能。

电子搜索策略

参数eSearchStrategy定义了代码搜索的搜索策略。标准选项TCVN_CSS_DEFAULT导致搜索策略TCVN_CSS_FIRST_NOT_FLIPPED。关于搜索策略的描述，见代码阅读搜索策略 [▶ 797]。

搜索策略

目前，这个功能只支持对镜像的搜索策略的设置。如果在原始图像中没有找到所有预期的代码，反转总是自动进行。

样本

- 数据矩阵码读取 [▶ 2648]
- 读码过程中的结果评估 [▶ 2653]

相关函数

- F_VN_ReadDataMatrixCode [▶ 803]
- F_VN_ReadBarcode [▶ 797]或F_VN_ReadBarcodeExp [▶ 799]
- F_VN_ReadPharmaCode [▶ 808]或F_VN_ReadPharmaCodeExp [▶ 810]
- F_VN_ReadQRCode [▶ 813]或 F_VN_ReadQRCodeExp [▶ 815]

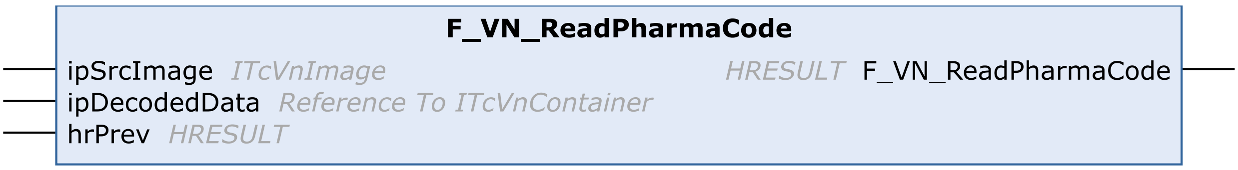
Required License

TC3 Vision Code Reading

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.7.7 F_VN_ReadPharmaCode



Detect and interpret a pharma code within the provided image. Can be canceled by Watchdog. Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_ReadPharmaCode : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipDecodedData : Reference To ITcVnContainer;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel (all element types) or 3 channel with elements of type TCVN_ET_USINT, TCVN_ET_UINT or TCVN_ET_REAL (3 channel input is expected to be RGB and internally converted to Gray))
ipDecodedData	Reference To ITcVnContainer [▶ 349]	Returns the decoded code (ContainerType_Vector_String_SINT)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

该函数F_VN_ReadPharmaCode检测并读取输入图像ipSrcImage中的药品编码。药品编码是一种条形码标准。它的设计具有良好的可读性，但只能对 3 至 131070 范围内的整数进行编码。这个标准的目的是高读取速度，减少读错。

不要混用药品编码

药品编码标准不应该与意大利或瑞士的药品编码相混淆。这些只指定具有 Code39 表示的药物识别号码。

药监码结构

药监码由粗细条（深）和它们之间的间隙（浅）组成。为获得良好可读性，建议图像中的代码要有足够的大小和对比度。根据下表进行解码：

位置	7	6	5	4	3	2	1	0
细线	128	64	32	16	8	4	2	1
粗线	256	128	64	32	16	8	4	2

有关药监码规格的详细信息可在线查阅：来自 Laetus 的药物代码指南

参数

输入图像

输入图像ipSrcImage必须有 1 或 3 个通道。3 通道图像必须是类型USINT、UINT或REAL，并且必须是在 RGB 色彩空间中。它们在内部被转换为灰度图像。

读取数据 （返回值）

从代码中读取的数据在容器ipDecodedData中以字符串形式返回。容器类型是 ContainerType_Vector_String_SINT。

专家参数

专家版F_VN_ReadPharmaCodeExp [▶ 810]包含额外的参数。

从容器中读取数据

如果成功找到并解码了代码，代码内容将在容器ipDecodedData中返回，并可以使用函数 F_VN_ExportSubContainer_String [▶ 732]导出为一个字符串。

```
hr := F_VN_ExportSubContainer_String( ipDecodedData, 0, sText, nMaxLength, hr);
```

样本

- 药品编码读取 [▸ 2649]
- 读码过程中的结果评估 [▸ 2653]

相关函数

- F_VN_ReadPharmaCode [▸ 808]用于读取药品编码。
- F_VN_ReadBarcode [▸ 797]用于读取条形码。
- F_VN_ReadDataMatrixCode [▸ 803]用于读取数据矩阵代码。
- F_VN_ReadQRCode [▸ 813]用于读取 QR 码。

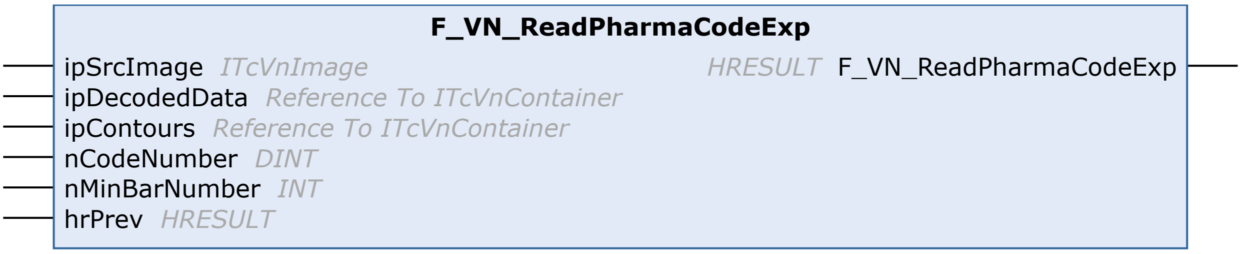
Required License

TC3 Vision Code Reading

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.7.8 F_VN_ReadPharmaCodeExp



Detect and interpret a pharma code within the provided image. Can be canceled by Watchdog. Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_ReadPharmaCodeExp : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipDecodedData : Reference To ITcVnContainer;
    ipContours     : Reference To ITcVnContainer;
    nCodeNumber    : DINT;
    nMinBarNumber : INT;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel (all element types) or 3 channel with elements of type TCVN_ET_USINT, TCVN_ET_UINT or TCVN_ET_REAL (3 channel input is expected to be RGB and internally converted to Gray))
ipDecodedData	Reference To ITcVnContainer [▶ 349]	Returns the decoded code (ContainerType_Vector_String_SINT)
ipContours	Reference To ITcVnContainer [▶ 349]	Returns the code positions as contours (optional, set to 0 if not required; ContainerType_Vector_Vector_TcVnPoint2_DINT)
nCodeNumber	DINT	Number of pharma codes that should be detected within the provided image (currently only 1 supported).
nMinBarNumber	INT	Minimal number of (dark) bars that codes must have.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

这个专家函数扩展了F_VN_ReadPharmaCode [▶ 808]函数，内容如下：

- 通过参数ipContours返回找到的代码的位置
- 指定图像中需要找到的代码数量。一旦在图像中发现nCodeNumber或更多的代码，搜索就会终止。如果找到的代码少于nCodeNumber，函数返回S_FALSE。
- 通过参数nMinBarNumber，函数找到的代码必须具有的最小暗条数。

样本

- 药品编码读取 [▶ 2649]
- 读码过程中的结果评估 [▶ 2653]

相关函数

- F_VN_ReadPharmaCode [▶ 808]
- F_VN_ReadBarcode [▶ 797]或F_VN_ReadBarcodeExp [▶ 800]
- F_VN_ReadDataMatrixCode [▶ 803]或F_VN_ReadDataMatrixCodeExp [▶ 805]
- F_VN_ReadQRCode [▶ 813]或F_VN_ReadQRCodeExp [▶ 815]

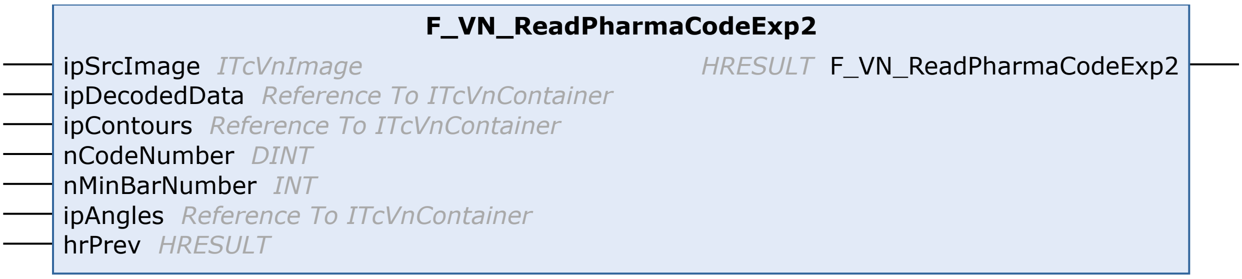
Required License

TC3 Vision Code Reading

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.7.9 F_VN_ReadPharmaCodeExp2



Detect and interpret a pharma code within the provided image. Can be canceled by Watchdog. Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_ReadPharmaCodeExp2 : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipDecodedData : Reference To ITcVnContainer;
    ipContours    : Reference To ITcVnContainer;
    nCodeNumber   : DINT;
    nMinBarNumber : INT;
    ipAngles      : Reference To ITcVnContainer;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (1 channel (all element types) or 3 channel with elements of type TCVN_ET_USINT, TCVN_ET_UINT or TCVN_ET_REAL (3 channel input is expected to be RGB and internally converted to Gray))
ipDecodedData	Reference To ITcVnContainer [▸ 349]	Returns the decoded code (ContainerType_Vector_String_SINT)
ipContours	Reference To ITcVnContainer [▸ 349]	Returns the code positions as contours (optional, set to 0 if not required; ContainerType_Vector_Vector_TcVnPoint2_DINT)
nCodeNumber	DINT	Number of pharma codes that should be detected within the provided image (currently only 1 supported).
nMinBarNumber	INT	Minimal number of (dark) bars that codes must have.
ipAngles	Reference To ITcVnContainer [▸ 349]	Returns the clockwise code rotation angles (optional, set to 0 if not required; ContainerType_Vector_REAL)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

这个专家函数扩展了F_VN_ReadPharmaCode [▸ 808]函数，内容如下：

- 通过参数ipContours返回找到的代码的位置
- 指定图像中需要找到的代码数量。一旦在图像中发现nCodeNumber或更多的代码，搜索就会终止。如果找到的代码少于nCodeNumber，函数返回S_FALSE。
- 通过参数nMinBarNumber，函数找到的代码必须具有的最小暗条数。

样本

- 药品编码读取 [▸ 2649]
- 读码过程中的结果评估 [▸ 2653]

相关函数

- F_VN_ReadPharmaCode [▸ 808]
- F_VN_ReadBarcode [▸ 797]或F_VN_ReadBarcodeExp [▸ 800]
- F_VN_ReadDataMatrixCode [▸ 803]或F_VN_ReadDataMatrixCodeExp [▸ 805]
- F_VN_ReadQRCode [▸ 813]或F_VN_ReadQRCodeExp [▸ 815]

Required License

TC3 Vision Code Reading

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.7.10 F_VN_ReadQRCode

Detect and interpret a QR code within the provided image. Can be canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_ReadQRCode : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipDecodedData : Reference To ITcVnContainer;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (USINT elements, 1 channel or 3 channel (3 channel input is expected to be RGB and internally converted to Gray))
ipDecodedData	Reference To ITcVnContainer [▸ 349]	Returns the decoded code (ContainerType_Vector_String_SINT)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

根据 IEC18004 标准，QR 码（快速响应码）是一种二维代码，在德国的移动标签领域特别常见。由于可以对汉字/假名字符进行编码，QR 码也已在亚洲的行业中得到确立。

F_VN_ReadQRCode 函数可以识别并解码图像中的这种代码类型。这是以每张图片一个代码为基础。

QR 码结构

1.		位置标记
2.		对齐标记
3.		定时排队
4.		版本信息
5.		数据格式
6.		数据领域
7.		安静区，至少 4 个单元宽

缺陷/损坏可以妨碍检测和解码。

代码搜索

函数 F_VN_ReadQRCode 从左上到右下逐行搜索图像中的 QR 码。

从容器中读取数据

如果成功找到并解码了代码，代码内容将在容器ipDecodedData中返回，并可以使用函数F_VN_ExportSubContainer_String [▸ 732]导出为一个字符串。

```
hr := F_VN_ExportSubContainer_String( ipDecodedData, 0, sText, nMaxLength, hr);
```

HRESULT

与所有 TwinCAT Vision API 函数一样，该函数也返回一个HRESULT [▸ 122]，以表明执行是否成功。在成功的情况下，在随后对结果的处理中，可以区分以下成功代码。为了区分这些情况，HRESULT变量可以直接与S_OK或S_FALSE进行比较。

代码	名称	描述
16#000	S_OK	该功能成功执行。 在图像中发现了指定数量的代码。
16#001	S_FALSE	该功能成功执行。 发现的代码比图片中显示的少。
16#256	S_WATCHDOGTIMEOUT	该功能被看门狗 [▸ 127]中止。

样本

- QR 码读取 [▸ 2651]
- 读码过程中的结果评估 [▸ 2653]

相关函数

- F_VN_ReadQRCodeExp [▸ 815]
- F_VN_ReadBarcode [▸ 797]或F_VN_ReadBarcodeExp [▸ 799]
- F_VN_ReadDataMatrixCode [▸ 803]或F_VN_ReadDataMatrixCodeExp [▸ 805]
- F_VN_ReadPharmaCode [▸ 808]或F_VN_ReadPharmaCodeExp [▸ 810]

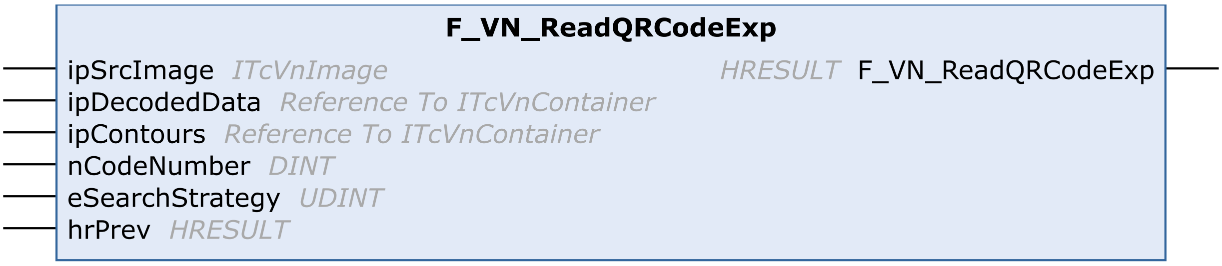
Required License

TC3 Vision Code Reading

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.7.11 F_VN_ReadQRCodeExp



Detect and interpret a QR code within the provided image. Can be canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_ReadQRCodeExp : HRESULT
VAR_INPUT
    ipSrcImage      : ITcVnImage;
    ipDecodedData   : Reference To ITcVnContainer;
    ipContours      : Reference To ITcVnContainer;
    nCodeNumber     : DINT;
    eSearchStrategy : UDINT;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT elements, 1 channel or 3 channel (3 channel input is expected to be RGB and internally converted to Gray))
ipDecodedData	Reference To ITcVnContainer [▶ 349]	Returns the decoded code (ContainerType_Vector_String_SINT)
ipContours	Reference To ITcVnContainer [▶ 349]	Returns the code positions as contours (optional, set to 0 if not required; ContainerType_Vector_Vector_TcVnPoint2_DINT)
nCodeNumber	DINT	Number of QR codes that should be detected within the provided image (currently only 1 supported).
eSearchStrategy	UDINT	Used search strategy (ETcVn2dCodeSearchStrategy)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

注意

浮点异常

如果正在执行的PLC 任务 [▶ 55]的选项**浮点异常**启用，该功能会不必要地引起错误。因此，停用这个选项。

函数 F_VN_ReadQRCodeExp 对应于函数F_VN_ReadQRCode [▶ 813]，并由以下参数扩展。

ipContour

参数ipContour返回找到的代码位置，作为轮廓。

nCodeNumber

参数nCodeNumber定义了需要搜索多少个代码。

目前不支持搜索一个以上的 QR 码

目前不支持在一张图像中搜索多个 QR 码。因此，参数 nCodeNumber 必须设置为 1。

电子搜索策略

参数eSearchStrategy定义了代码搜索的搜索策略。标准选项TCVN_CSS_DEFAULT导致搜索策略TCVN_CSS_FIRST_NOT_INVERTED和TCVN_CSS_FIRST_NOT_FLIPPED。关于搜索策略的描述，见代码阅读搜索策略 [▶ 797]。

样本

- QR 码读取 [▶ 2651]
- 读码过程中的结果评估 [▶ 2653]

相关函数

- F_VN_ReadQRCode [▶ 813]
- F_VN_ReadBarcode [▶ 797]或F_VN_ReadBarcodeExp [▶ 799]
- F_VN_ReadDataMatrixCode [▶ 803]或F_VN_ReadDataMatrixCodeExp [▶ 805]

- F_VN_ReadPharmaCode [▶ 808]或F_VN_ReadPharmaCodeExp [▶ 810]

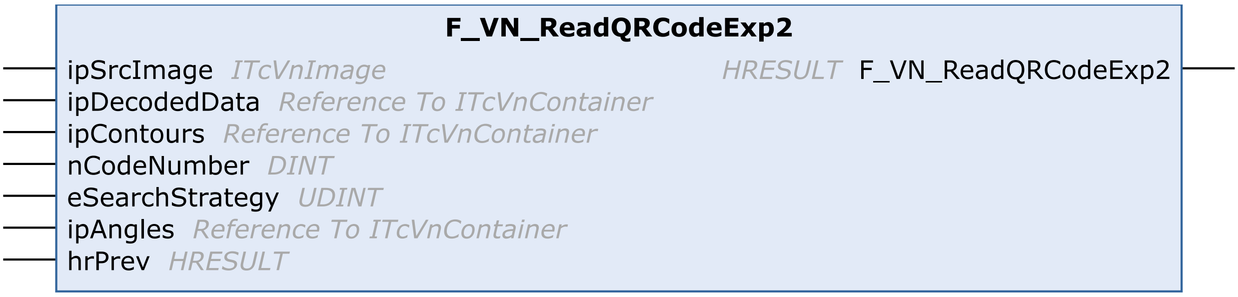
Required License

TC3 Vision Code Reading

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.7.12 F_VN_ReadQRCodeExp2



Detect and interpret a QR code within the provided image. Can be canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_ReadQRCodeExp2 : HRESULT
VAR_INPUT
    ipSrcImage      : ITcVnImage;
    ipDecodedData   : Reference To ITcVnContainer;
    ipContours      : Reference To ITcVnContainer;
    nCodeNumber     : DINT;
    eSearchStrategy : UDINT;
    ipAngles        : Reference To ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT elements, 1 channel or 3 channel (3 channel input is expected to be RGB and internally converted to Gray))
ipDecodedData	Reference To ITcVnContainer [▶ 349]	Returns the decoded code (ContainerType_Vector_String_SINT)
ipContours	Reference To ITcVnContainer [▶ 349]	Returns the code positions as contours (optional, set to 0 if not required; ContainerType_Vector_Vector_TcVnPoint2_DINT)
nCodeNumber	DINT	Number of QR codes that should be detected within the provided image (currently only 1 supported).
eSearchStrategy	UDINT	Used search strategy (ETcVn2dCodeSearchStrategy)
ipAngles	Reference To ITcVnContainer [▶ 349]	Returns the clockwise code rotation angles (optional, set to 0 if not required; ContainerType_Vector_REAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

注意

浮点异常

如果正在执行的PLC 任务 [▶ 55]的选项**浮点异常**启用，该功能会不必要地引起错误。因此，停用这个选项。

函数 F_VN_ReadQRCodeExp 对应于函数F_VN_ReadQRCode [▶ 813]，并由以下参数扩展。

ipContour

参数ipContour返回找到的代码位置，作为轮廓。

nCodeNumber

参数nCodeNumber定义了需要搜索多少个代码。

目前不支持搜索一个以上的 QR 码

目前不支持在一张图像中搜索多个 QR 码。因此，参数 nCodeNumber 必须设置为 1。

电子搜索策略

参数eSearchStrategy定义了代码搜索的搜索策略。标准选项TCVN_CSS_DEFAULT导致搜索策略TCVN_CSS_FIRST_NOT_INVERTED和TCVN_CSS_FIRST_NOT_FLIPPED。关于搜索策略的描述，见代码阅读搜索策略 [▶ 797]。

样本

- QR 码读取 [▶ 2651]
- 读码过程中的结果评估 [▶ 2653]

相关函数

- F_VN_ReadQRCode [▶ 813]
- F_VN_ReadBarcode [▶ 797]或F_VN_ReadBarcodeExp [▶ 799]
- F_VN_ReadDataMatrixCode [▶ 803]或F_VN_ReadDataMatrixCodeExp [▶ 805]
- F_VN_ReadPharmaCode [▶ 808]或F_VN_ReadPharmaCodeExp [▶ 810]

Required License

TC3 Vision Code Reading

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8 Container Statistics

该组包含对容器 [▶ 132]进行统计分析的函数。

函数

以欧几里得规范

- F_VN_ContainerAverage [▶ 900]

- F_VN_ContainerAverageVariance [▶ 905]
- F_VN_MaxElement [▶ 910]
- F_VN_MedianElement [▶ 911]
- F_VN_MinElement [▶ 912]

以元素方式

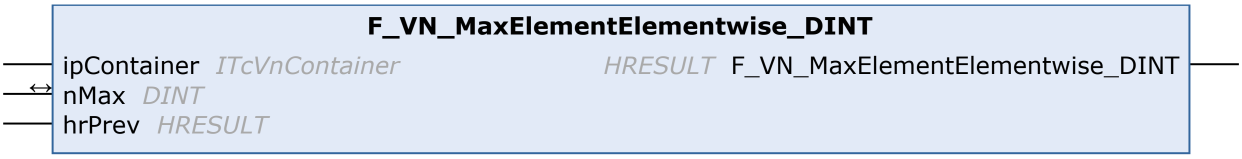
- F_VN_ContainerAverageElementwise [▶ 901]
- F_VN_ContainerAverageVarianceElementwise [▶ 906]
- F_VN_MaxElementwise [▶ 819]
- F_VN_MedianElementwise [▶ 846]
- F_VN_MinElementwise [▶ 873]

6.1.4.8.1 F_VN_MaxElementElementwise

本章包含一些函数，通过这些函数可根据每个元素的数据类型来确定最大值。这个函数适用于包括基本元素的容器。

例如，对于 TcVnPoint2_DINT，最大 X 值和最大 Y 值相互独立确定，所以这些值不一定来自于同一个点。

6.1.4.8.1.1 F_VN_MaxElementElementwise_DINT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMax        : DINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMax	DINT	Returns the requested element

Return value

HRESULT [▶ 122]

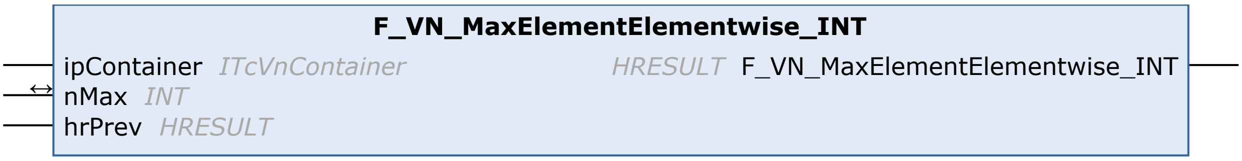
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.2 F_VN_MaxElementElementwise_INT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMax        : INT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMax	INT	Returns the requested element

Return value

HRESULT [▶ 122]

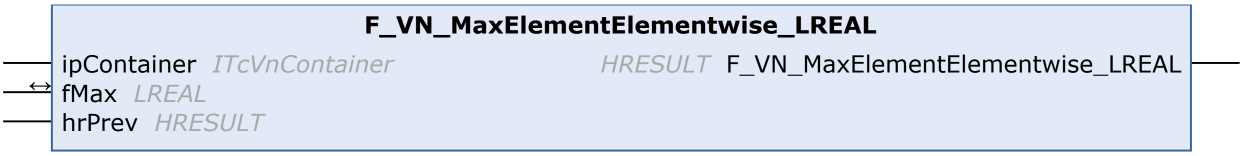
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.3 F_VN_MaxElementElementwise_LREAL



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fMax        : LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fMax	LREAL	Returns the requested element

Return value

HRESULT [▶ 122]

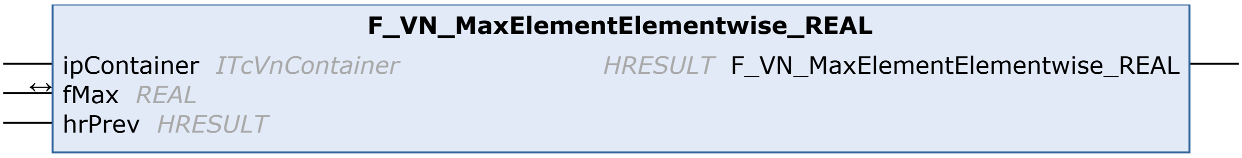
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.4 F_VN_MaxElementElementwise_REAL



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fMax        : REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fMax	REAL	Returns the requested element

Return value

HRESULT [▶ 122]

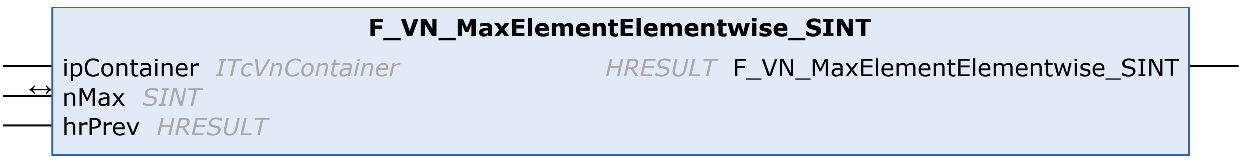
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.5 F_VN_MaxElementElementwise_SINT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMax        : SINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMax	SINT	Returns the requested element

Return value

HRESULT [▸ 122]

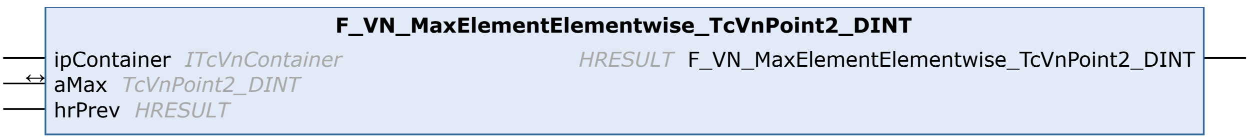
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.6 F_VN_MaxElementElementwise_TcVnPoint2_DINT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnPoint2_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnPoint2_DINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnPoint2_DINT [▶ 139]	Returns the requested element

Return value

HRESULT [▶ 122]

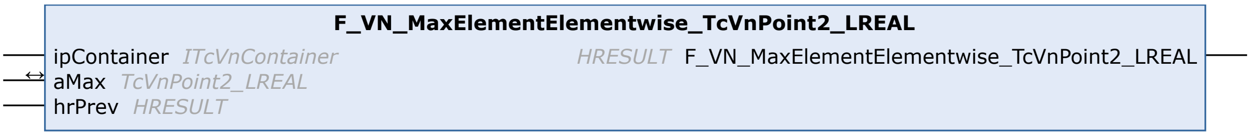
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.7 F_VN_MaxElementElementwise_TcVnPoint2_LREAL



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnPoint2_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnPoint2_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnPoint2_LREAL [▶ 139]	Returns the requested element

Return value

HRESULT [▶ 122]

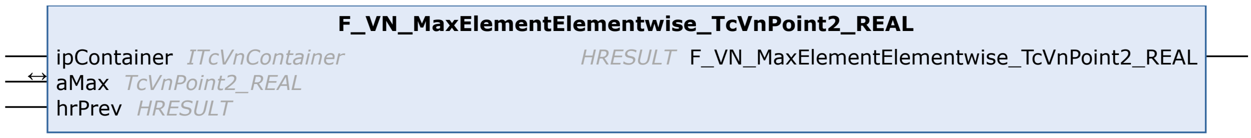
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.8 F_VN_MaxElementElementwise_TcVnPoint2_REAL



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnPoint2_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnPoint2_REAL [▶ 139]	Returns the requested element

Return value

HRESULT [▶ 122]

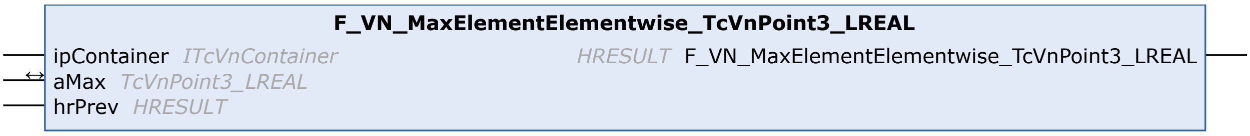
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.9 F_VN_MaxElementElementwise_TcVnPoint3_LREAL



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnPoint3_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnPoint3_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnPoint3_LREAL [▶ 139]	Returns the requested element

Return value

HRESULT [▶ 122]

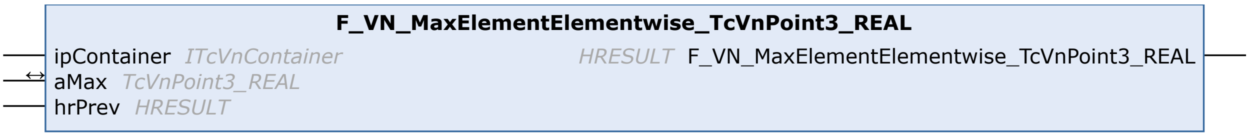
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.10 F_VN_MaxElementElementwise_TcVnPoint3_REAL



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnPoint3_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnPoint3_REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnPoint3_REAL [▶ 139]	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.11 F_VN_MaxElementElementwise_TcVnVector2_DINT

```
F_VN_MaxElementElementwise_TcVnVector2_DINT
ipContainer ITcVnContainer          HRESULT F_VN_MaxElementElementwise_TcVnVector2_DINT
aMax TcVnVector2_DINT
hrPrev HRESULT
```

Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnVector2_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnVector2_DINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnVector2_DINT [▸ 141]	Returns the requested element

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.12 F_VN_MaxElementElementwise_TcVnVector2_INT

```
F_VN_MaxElementElementwise_TcVnVector2_INT
ipContainer ITcVnContainer          HRESULT F_VN_MaxElementElementwise_TcVnVector2_INT
aMax TcVnVector2_INT
hrPrev HRESULT
```

Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnVector2_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnVector2_INT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnVector2_INT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

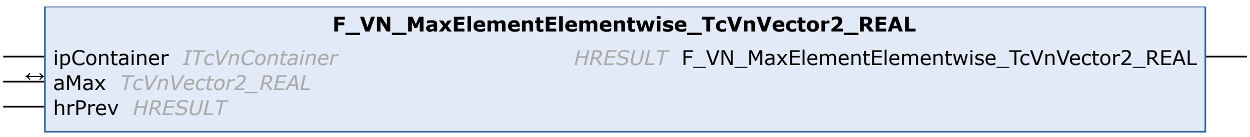
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.13 F_VN_MaxElementElementwise_TcVnVector2_REAL



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnVector2_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnVector2_REAL;
END_VAR
```

```
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnVector2_REAL [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

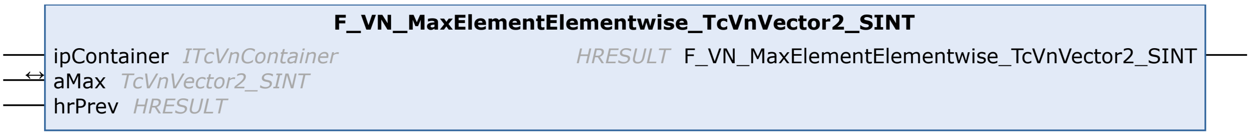
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.14 F_VN_MaxElementElementwise_TcVnVector2_SINT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnVector2_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnVector2_SINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnVector2_SINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

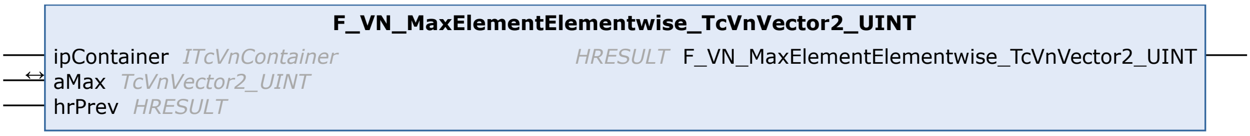
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.15 F_VN_MaxElementElementwise_TcVnVector2_UINT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnVector2_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnVector2_UINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnVector2_UINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

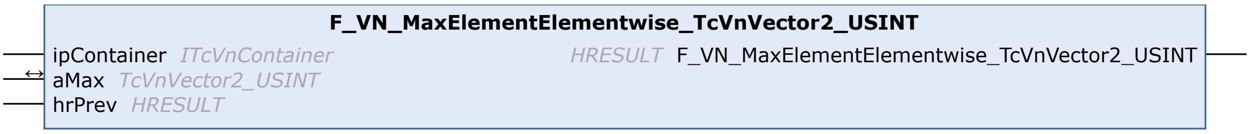
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.16 F_VN_MaxElementElementwise_TcVnVector2_USINT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnVector2_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnVector2_USINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnVector2_USINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

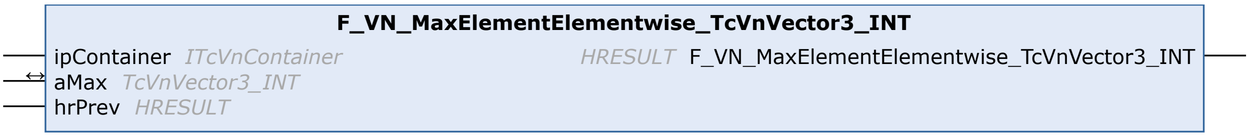
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.17 F_VN_MaxElementElementwise_TcVnVector3_INT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnVector3_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnVector3_INT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnVector3_INT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

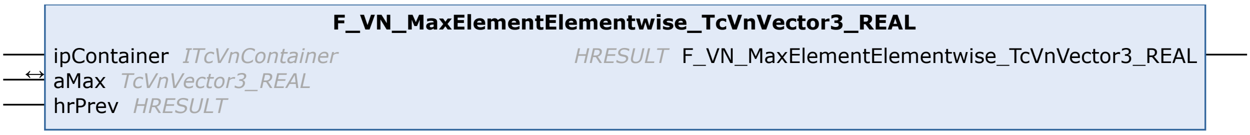
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.18 F_VN_MaxElementElementwise_TcVnVector3_REAL



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnVector3_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnVector3_REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnVector3_REAL [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

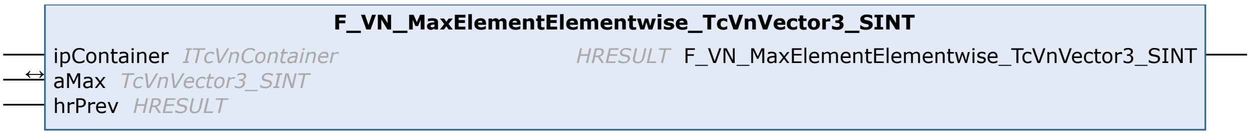
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.19 F_VN_MaxElementElementwise_TcVnVector3_SINT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnVector3_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnVector3_SINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnVector3_SINT [▸ 141]	Returns the requested element

Return value

HRESULT [▸ 122]

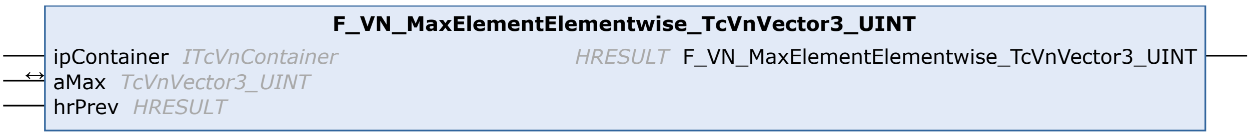
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.20 F_VN_MaxElementElementwise_TcVnVector3_UINT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnVector3_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnVector3_UINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnVector3_UINT [▸ 141]	Returns the requested element

Return value

HRESULT [▸ 122]

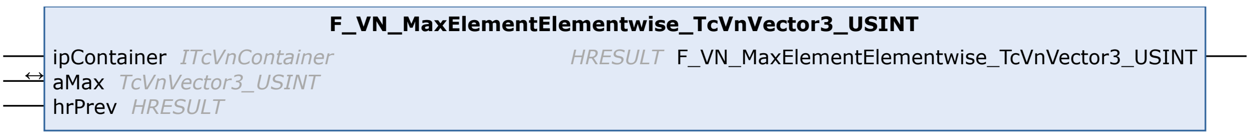
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.21 F_VN_MaxElementElementwise_TcVnVector3_USINT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnVector3_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnVector3_USINT;
END_VAR
```

```
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnVector3_USINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

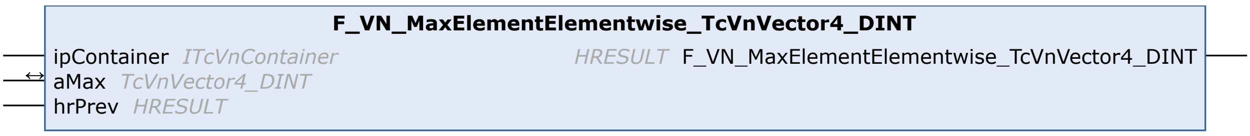
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.22 F_VN_MaxElementElementwise_TcVnVector4_DINT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnVector4_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax         : TcVnVector4_DINT;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnVector4_DINT [▸ 141]	Returns the requested element

Return value

HRESULT [▸ 122]

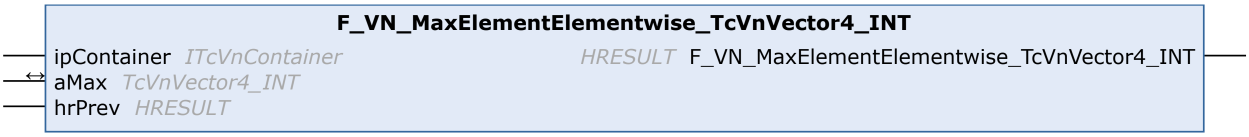
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.23 F_VN_MaxElementElementwise_TcVnVector4_INT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnVector4_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnVector4_INT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnVector4_INT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

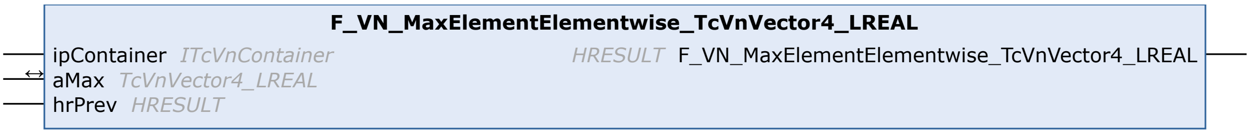
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.24 F_VN_MaxElementElementwise_TcVnVector4_LREAL



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnVector4_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnVector4_LREAL [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

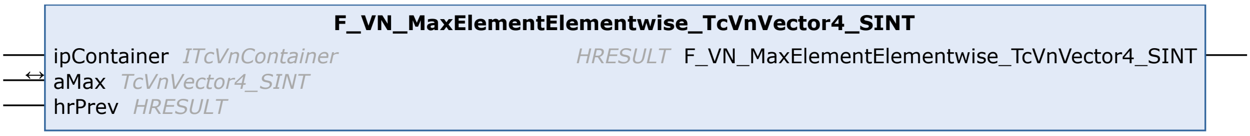
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.25 F_VN_MaxElementElementwise_TcVnVector4_SINT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnVector4_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnVector4_SINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnVector4_SINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.26 F_VN_MaxElementElementwise_TcVnVector4_UINT

F_VN_MaxElementElementwise_TcVnVector4_UINT
ipContainer *ITcVnContainer* *HRESULT* F_VN_MaxElementElementwise_TcVnVector4_UINT
aMax *TcVnVector4_UINT*
hrPrev *HRESULT*

Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnVector4_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnVector4_UINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnVector4_UINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

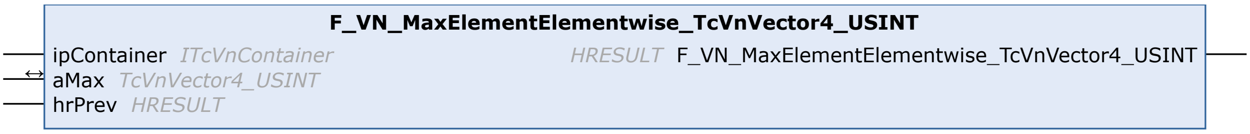
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.27 F_VN_MaxElementElementwise_TcVnVector4_USINT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_TcVnVector4_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMax        : TcVnVector4_USINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMax	TcVnVector4_USINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

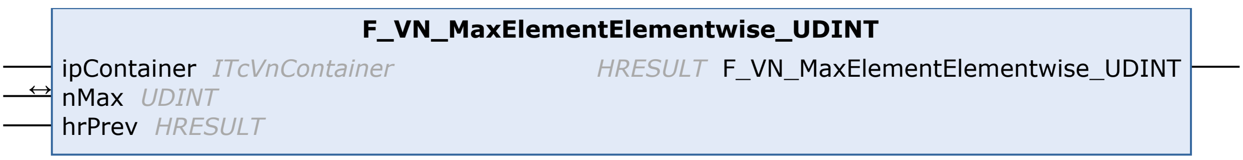
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.28 F_VN_MaxElementElementwise_UDINT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_UDINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMax        : UDINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMax	UDINT	Returns the requested element

Return value

HRESULT [▶ 122]

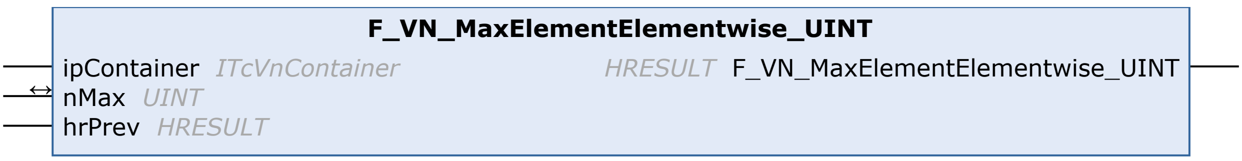
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.29 F_VN_MaxElementElementwise_UINT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMax        : UINT;
END_VAR
```

```
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMax	UINT	Returns the requested element

Return value

HRESULT [▶ 122]

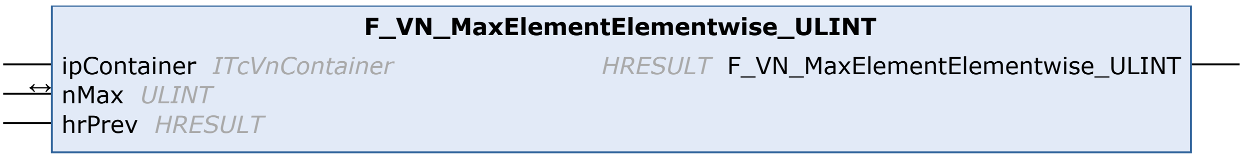
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.30 F_VN_MaxElementElementwise_ULINT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_ULINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMax        : ULINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMax	ULINT	Returns the requested element

Return value

HRESULT [▶ 122]

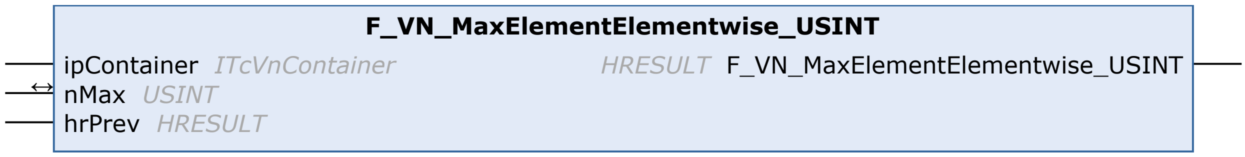
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.1.31 F_VN_MaxElementElementwise_USINT



Gets the element wise maximum container element.

Syntax

Definition:

```
FUNCTION F_VN_MaxElementElementwise_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMax        : USINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMax	USINT	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

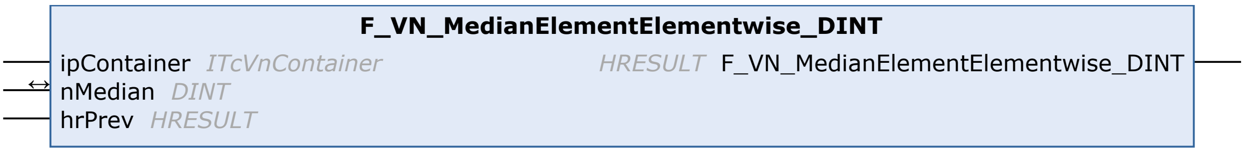
Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2 F_VN_MedianElementElementwise

本章包含根据每个元素的数据类型确定中位数的函数。这个函数适用于包括基本元素的容器。

例如，对于 TcVnPoint2_DINT 来说，中位 X 值和中位 Y 值相互独立确定，所以这些值不一定来自于同一个点。

6.1.4.8.2.1 F_VN_MedianElementElementwise_DINT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMedian     : DINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMedian	DINT	Returns the requested element

Return value

HRESULT [▸ 122]

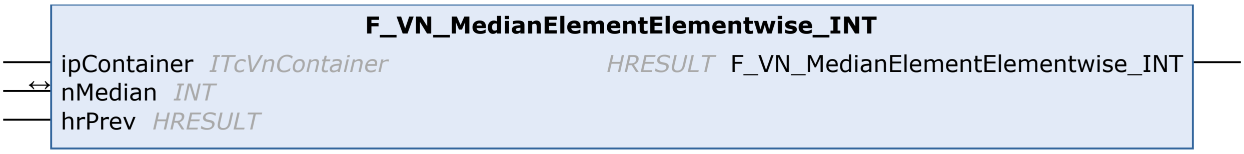
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.2 F_VN_MedianElementElementwise_INT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMedian     : INT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMedian	INT	Returns the requested element

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.3 F_VN_MedianElementElementwise_LREAL

F_VN_MedianElementElementwise_LREAL	
ipContainer *ITcVnContainer*	*HRESULT* F_VN_MedianElementElementwise_LREAL
↔ fMedian *LREAL*	
hrPrev *HRESULT*	

Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fMedian     : LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fMedian	LREAL	Returns the requested element

Return value

HRESULT [▸ 122]

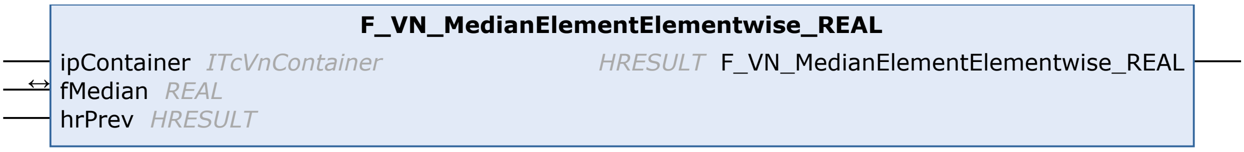
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.4 F_VN_MedianElementElementwise_REAL



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fMedian     : REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fMedian	REAL	Returns the requested element

Return value

HRESULT [▶ 122]

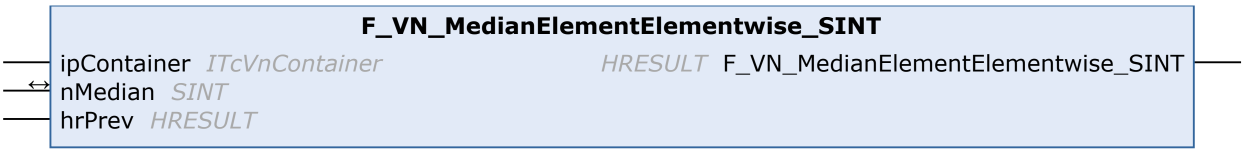
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.5 F_VN_MedianElementElementwise_SINT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMedian     : SINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMedian	SINT	Returns the requested element

Return value

HRESULT [▸ 122]

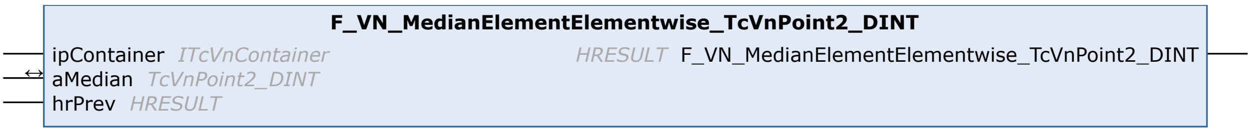
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.6 F_VN_MedianElementElementwise_TcVnPoint2_DINT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnPoint2_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnPoint2_DINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnPoint2_DINT [▶ 139]	Returns the requested element

Return value

HRESULT [▶ 122]

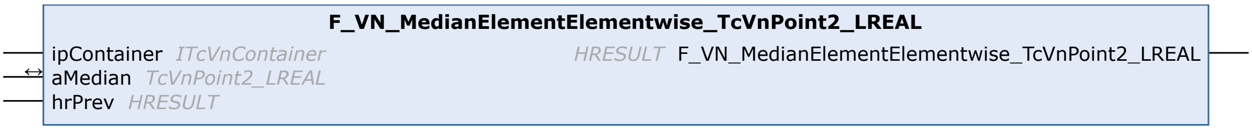
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.7 F_VN_MedianElementElementwise_TcVnPoint2_LREAL



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnPoint2_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnPoint2_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnPoint2_LREAL [▶ 139]	Returns the requested element

Return value

HRESULT [▶ 122]

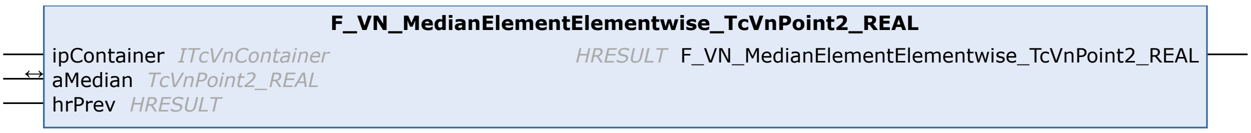
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.8 F_VN_MedianElementElementwise_TcVnPoint2_REAL



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnPoint2_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnPoint2_REAL [▶ 139]	Returns the requested element

Return value

HRESULT [▶ 122]

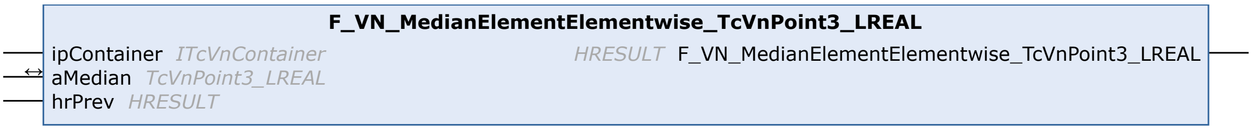
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.9 F_VN_MedianElementElementwise_TcVnPoint3_LREAL



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnPoint3_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnPoint3_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnPoint3_LREAL [▶ 139]	Returns the requested element

Return value

HRESULT [▶ 122]

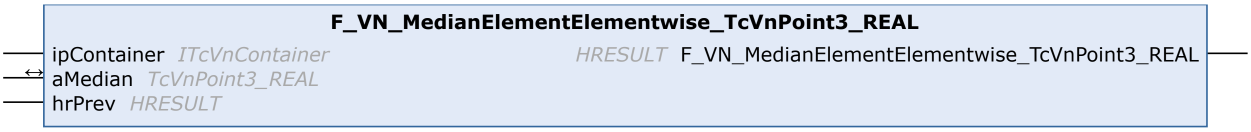
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.10 F_VN_MedianElementElementwise_TcVnPoint3_REAL



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnPoint3_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnPoint3_REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnPoint3_REAL [▶ 139]	Returns the requested element

Return value

HRESULT [▶ 122]

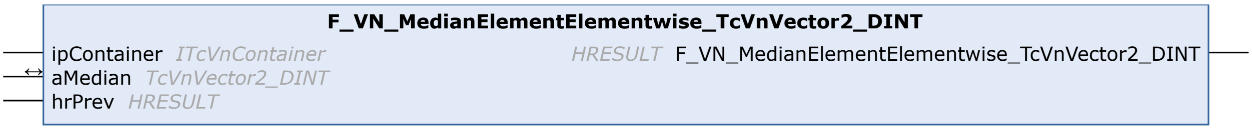
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.11 F_VN_MedianElementElementwise_TcVnVector2_DINT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnVector2_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnVector2_DINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector2_DINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

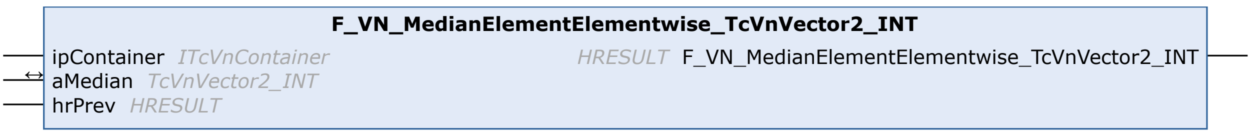
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.12 F_VN_MedianElementElementwise_TcVnVector2_INT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnVector2_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnVector2_INT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector2_INT [▸ 141]	Returns the requested element

Return value

HRESULT [▸ 122]

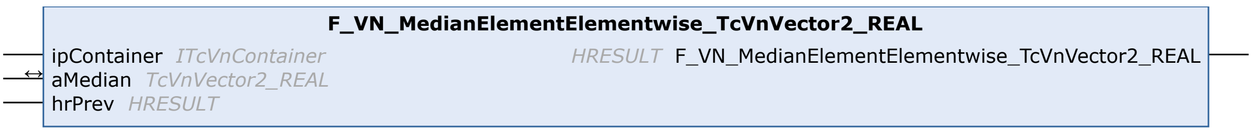
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.13 F_VN_MedianElementElementwise_TcVnVector2_REAL



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnVector2_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnVector2_REAL;
END_VAR
```

```
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector2_REAL [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

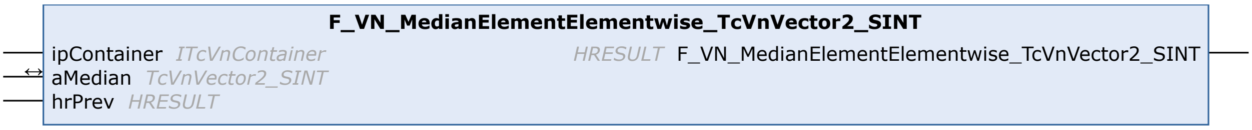
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.14 F_VN_MedianElementElementwise_TcVnVector2_SINT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnVector2_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian      : TcVnVector2_SINT;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector2_SINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

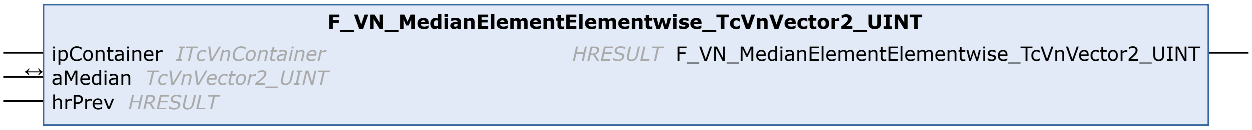
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.15 F_VN_MedianElementElementwise_TcVnVector2_UINT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnVector2_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnVector2_UINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector2_UINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

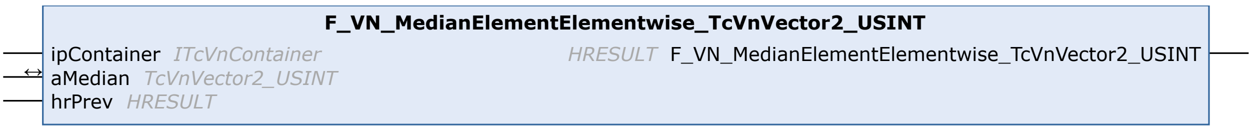
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.16 F_VN_MedianElementElementwise_TcVnVector2_USINT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnVector2_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnVector2_USINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector2_USINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

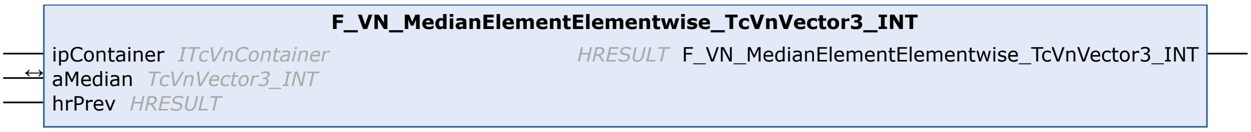
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.17 F_VN_MedianElementElementwise_TcVnVector3_INT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnVector3_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnVector3_INT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector3_INT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

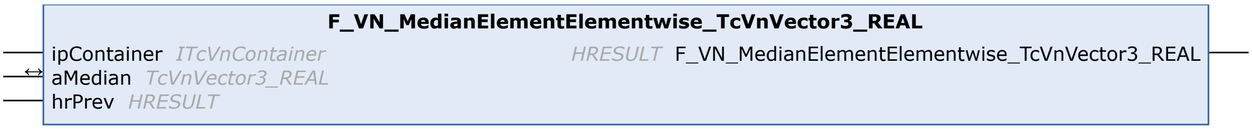
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.18 F_VN_MedianElementElementwise_TcVnVector3_REAL



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnVector3_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnVector3_REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector3_REAL [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.19 F_VN_MedianElementElementwise_TcVnVector3_SINT

F_VN_MedianElementElementwise_TcVnVector3_SINT
ipContainer *ITcVnContainer* — *HRESULT* F_VN_MedianElementElementwise_TcVnVector3_SINT
aMedian *TcVnVector3_SINT*
hrPrev *HRESULT*

Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnVector3_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnVector3_SINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector3_SINT [▸ 141]	Returns the requested element

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.20 F_VN_MedianElementElementwise_TcVnVector3_UINT

F_VN_MedianElementElementwise_TcVnVector3_UINT
ipContainer *ITcVnContainer* — *HRESULT* F_VN_MedianElementElementwise_TcVnVector3_UINT
aMedian *TcVnVector3_UINT*
hrPrev *HRESULT*

Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnVector3_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnVector3_UINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector3_UINT [▸ 141]	Returns the requested element

Return value

HRESULT [▸ 122]

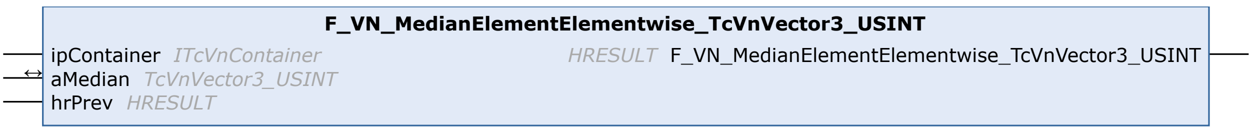
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.21 F_VN_MedianElementElementwise_TcVnVector3_USINT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnVector3_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnVector3_USINT;
END_VAR
```

```
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector3_USINT [▸ 141]	Returns the requested element

Return value

HRESULT [▸ 122]

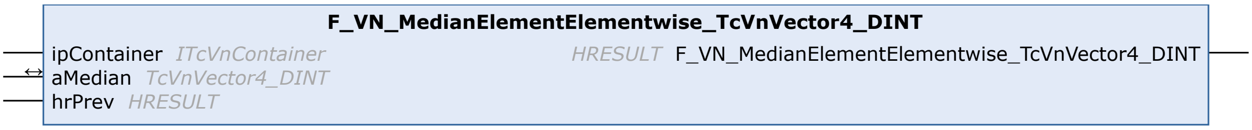
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.22 F_VN_MedianElementElementwise_TcVnVector4_DINT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnVector4_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian      : TcVnVector4_DINT;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector4_DINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

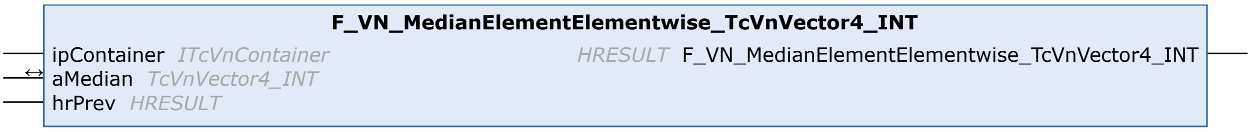
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.23 F_VN_MedianElementElementwise_TcVnVector4_INT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnVector4_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnVector4_INT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector4_INT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

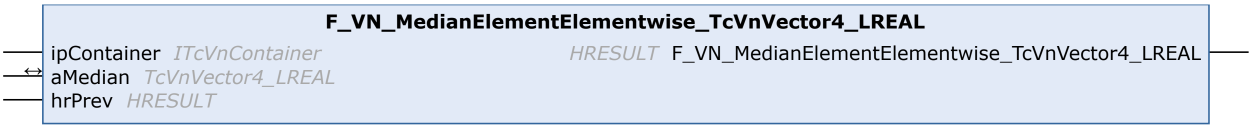
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.24 F_VN_MedianElementElementwise_TcVnVector4_LREAL



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnVector4_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector4_LREAL [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

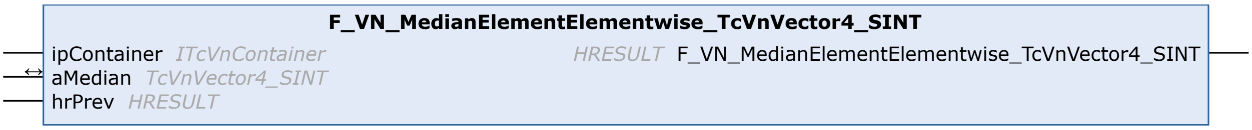
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.25 F_VN_MedianElementElementwise_TcVnVector4_SINT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnVector4_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnVector4_SINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector4_SINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

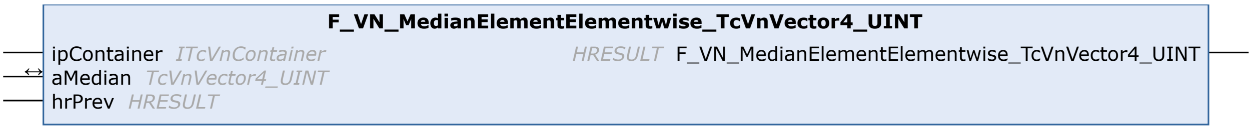
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.26 F_VN_MedianElementElementwise_TcVnVector4_UINT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnVector4_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnVector4_UINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector4_UINT [▸ 141]	Returns the requested element

Return value

HRESULT [▸ 122]

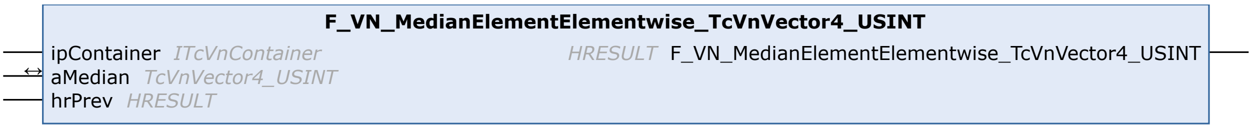
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.27 F_VN_MedianElementElementwise_TcVnVector4_USINT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_TcVnVector4_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMedian     : TcVnVector4_USINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector4_USINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

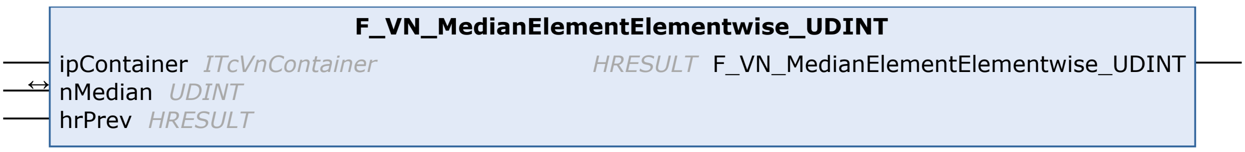
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.28 F_VN_MedianElementElementwise_UDINT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_UDINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMedian     : UDINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMedian	UDINT	Returns the requested element

Return value

HRESULT [▸ 122]

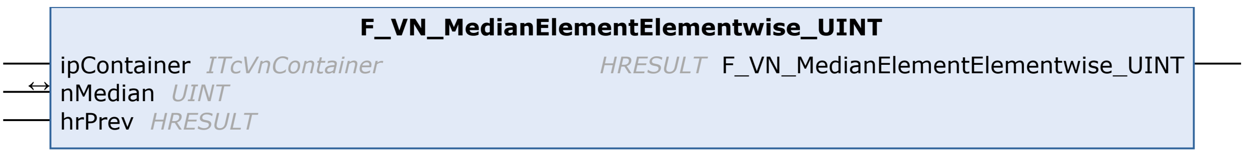
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.29 F_VN_MedianElementElementwise_UINT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMedian     : UINT;
END_VAR
```

```
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMedian	UINT	Returns the requested element

Return value

HRESULT [▶ 122]

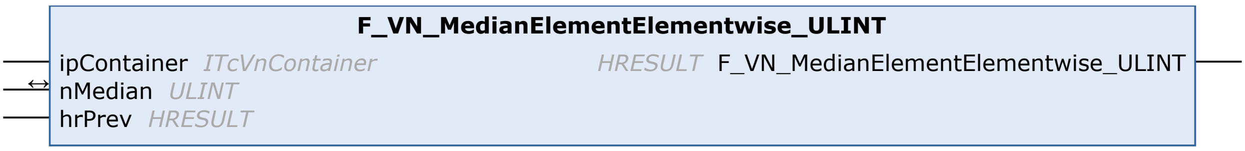
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.30 F_VN_MedianElementElementwise_ULINT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_ULINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMedian     : ULINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMedian	ULINT	Returns the requested element

Return value

HRESULT [▸ 122]

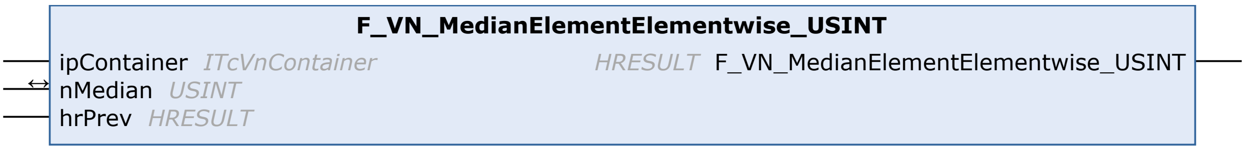
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.2.31 F_VN_MedianElementElementwise_USINT



Gets the element wise median container element.

Syntax

Definition:

```
FUNCTION F_VN_MedianElementElementwise_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMedian     : USINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMedian	USINT	Returns the requested element

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

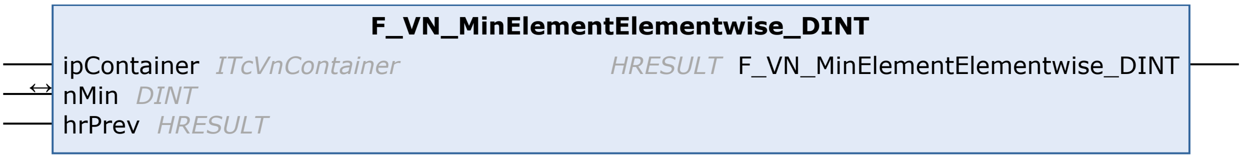
Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3 F_VN_MinElementElementwise

本章包含根据每个元素的数据类型来确定最小值的函数。这个函数适用于包括基本元素的容器。

例如，对于 TcVnPoint2_DINT，最小 X 值和最小 Y 值相互独立确定，所以这些值不一定来自于同一个点。

6.1.4.8.3.1 F_VN_MinElementElementwise_DINT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMin        : DINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMin	DINT	Returns the requested element

Return value

HRESULT [▶ 122]

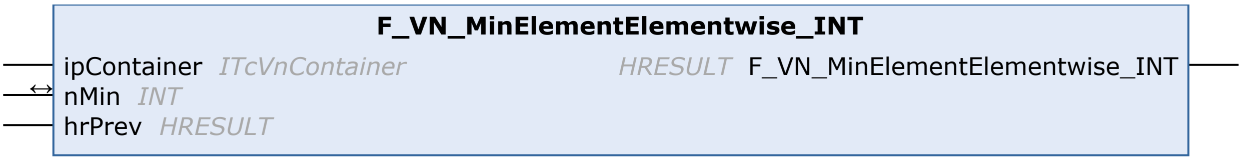
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.2 F_VN_MinElementElementwise_INT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMin        : INT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMin	INT	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.3 F_VN_MinElementElementwise_LREAL

F_VN_MinElementElementwise_LREAL	
ipContainer *ITcVnContainer*	*HRESULT* F_VN_MinElementElementwise_LREAL
↔ fMin *LREAL*	
hrPrev *HRESULT*	

Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fMin        : LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fMin	LREAL	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.4 F_VN_MinElementElementwise_REAL

F_VN_MinElementElementwise_REAL	
ipContainer *ITcVnContainer*	*HRESULT* F_VN_MinElementElementwise_REAL
↔ fMin *REAL*	
hrPrev *HRESULT*	

Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fMin        : REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fMin	REAL	Returns the requested element

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.5 F_VN_MinElementElementwise_SINT

F_VN_MinElementElementwise_SINT	
ipContainer *ITcVnContainer*	*HRESULT* F_VN_MinElementElementwise_SINT
↔ nMin *SINT*	
hrPrev *HRESULT*	

Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMin        : SINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMin	SINT	Returns the requested element

Return value

HRESULT [▶ 122]

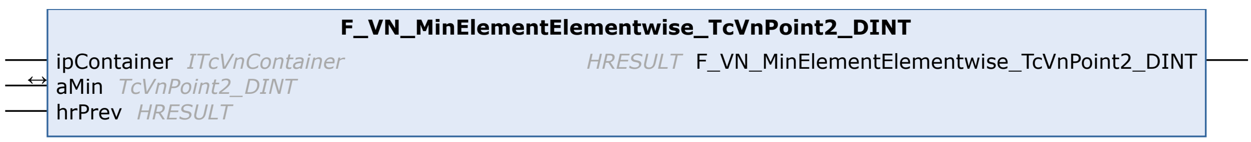
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.6 F_VN_MinElementElementwise_TcVnPoint2_DINT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnPoint2_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnPoint2_DINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnPoint2_DINT [▶ 139]	Returns the requested element

Return value

HRESULT [▶ 122]

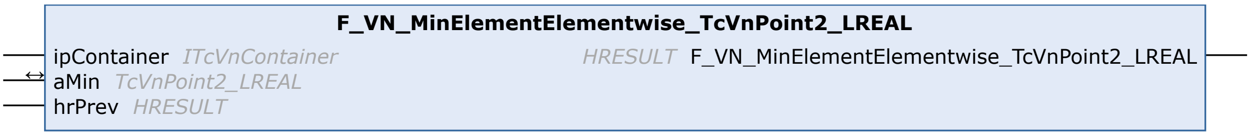
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.7 F_VN_MinElementElementwise_TcVnPoint2_LREAL



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnPoint2_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnPoint2_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnPoint2_LREAL [▶ 139]	Returns the requested element

Return value

HRESULT [▶ 122]

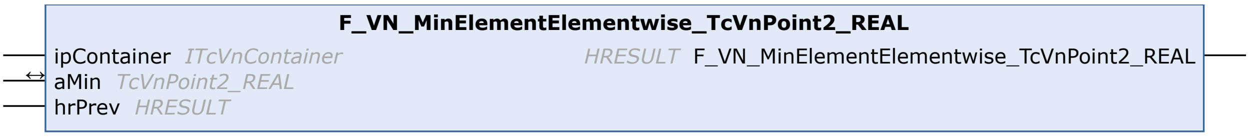
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.8 F_VN_MinElementElementwise_TcVnPoint2_REAL



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnPoint2_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnPoint2_REAL [▶ 139]	Returns the requested element

Return value

HRESULT [▶ 122]

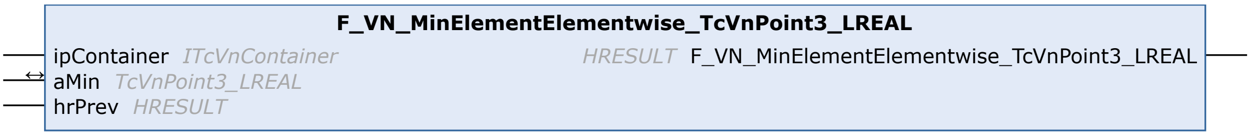
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.9 F_VN_MinElementElementwise_TcVnPoint3_LREAL



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnPoint3_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnPoint3_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnPoint3_LREAL [▶ 139]	Returns the requested element

Return value

HRESULT [▶ 122]

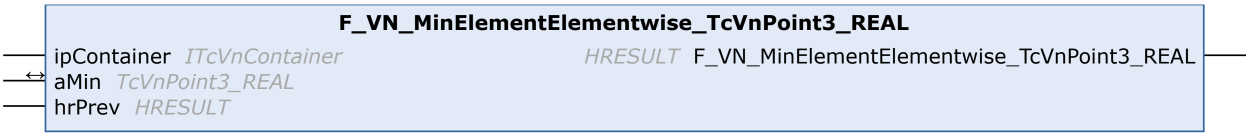
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.10 F_VN_MinElementElementwise_TcVnPoint3_REAL



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnPoint3_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnPoint3_REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnPoint3_REAL [▶ 139]	Returns the requested element

Return value

HRESULT [▶ 122]

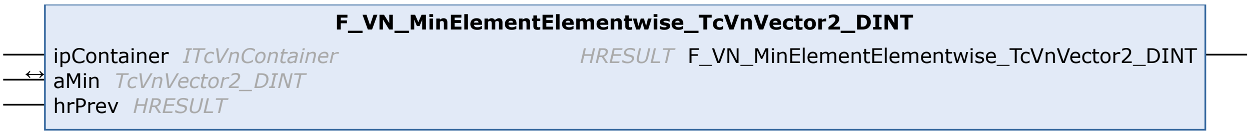
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.11 F_VN_MinElementElementwise_TcVnVector2_DINT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnVector2_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnVector2_DINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnVector2_DINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

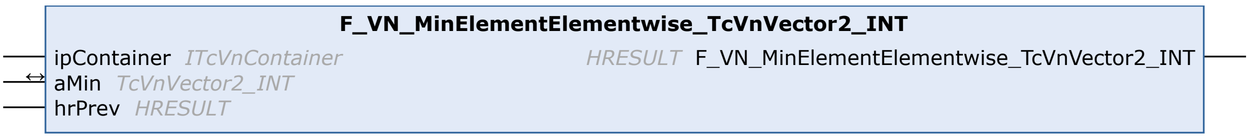
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.12 F_VN_MinElementElementwise_TcVnVector2_INT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnVector2_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnVector2_INT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnVector2_INT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

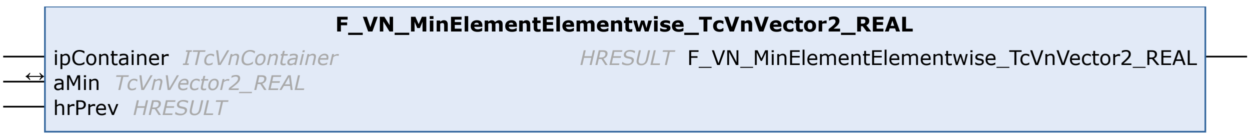
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.13 F_VN_MinElementElementwise_TcVnVector2_REAL



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnVector2_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnVector2_REAL;
END_VAR
```

```
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnVector2_REAL [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

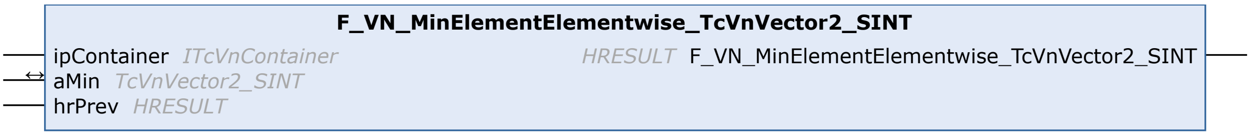
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.14 F_VN_MinElementElementwise_TcVnVector2_SINT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnVector2_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnVector2_SINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnVector2_SINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

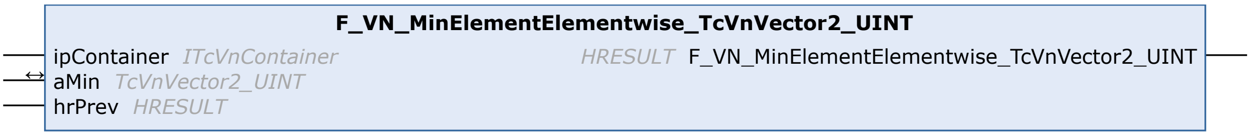
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.15 F_VN_MinElementElementwise_TcVnVector2_UINT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnVector2_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnVector2_UINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnVector2_UINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

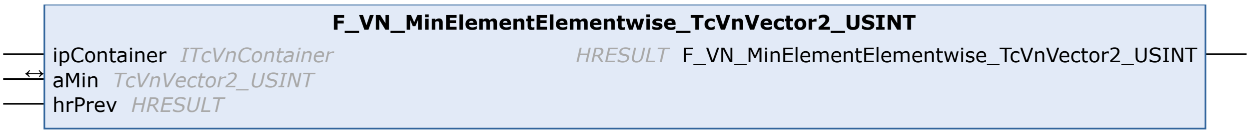
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.16 F_VN_MinElementElementwise_TcVnVector2_USINT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnVector2_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnVector2_USINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnVector2_USINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

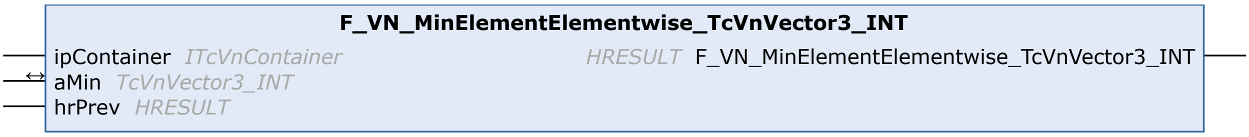
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.17 F_VN_MinElementElementwise_TcVnVector3_INT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnVector3_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnVector3_INT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnVector3_INT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

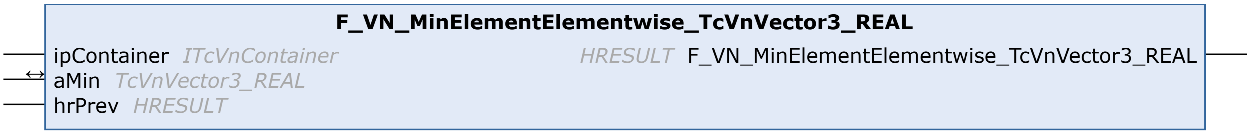
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.18 F_VN_MinElementElementwise_TcVnVector3_REAL



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnVector3_REAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnVector3_REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnVector3_REAL [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

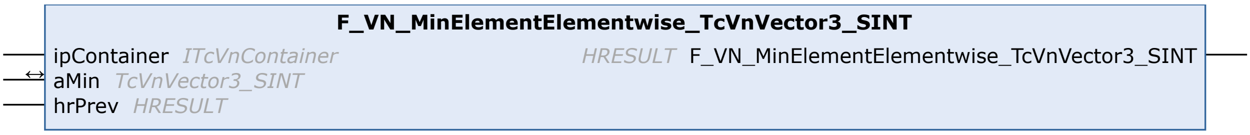
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.19 F_VN_MinElementElementwise_TcVnVector3_SINT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnVector3_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnVector3_SINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnVector3_SINT [▸ 141]	Returns the requested element

Return value

HRESULT [▸ 122]

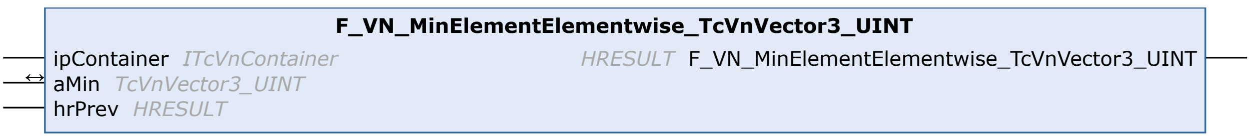
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.20 F_VN_MinElementElementwise_TcVnVector3_UINT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnVector3_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnVector3_UINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnVector3_UINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

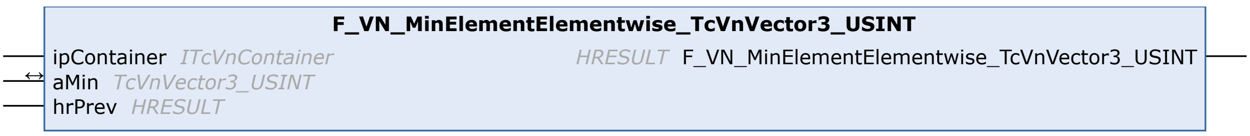
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.21 F_VN_MinElementElementwise_TcVnVector3_USINT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnVector3_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnVector3_USINT;
END_VAR
```

```
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnVector3_USINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

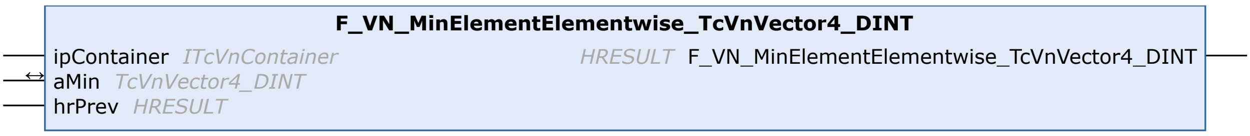
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.22 F_VN_MinElementElementwise_TcVnVector4_DINT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnVector4_DINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnVector4_DINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnVector4_DINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

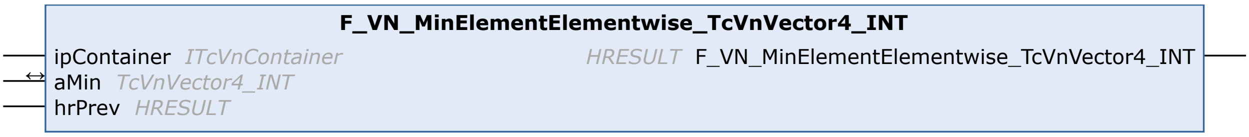
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.23 F_VN_MinElementElementwise_TcVnVector4_INT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnVector4_INT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnVector4_INT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnVector4_INT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

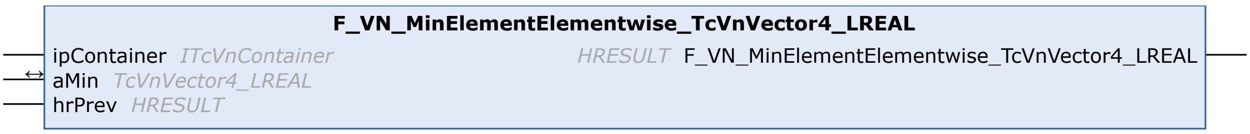
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.24 F_VN_MinElementElementwise_TcVnVector4_LREAL



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnVector4_LREAL : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnVector4_LREAL [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

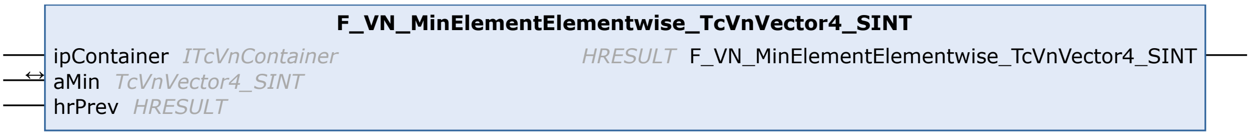
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.25 F_VN_MinElementElementwise_TcVnVector4_SINT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnVector4_SINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnVector4_SINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnVector4_SINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

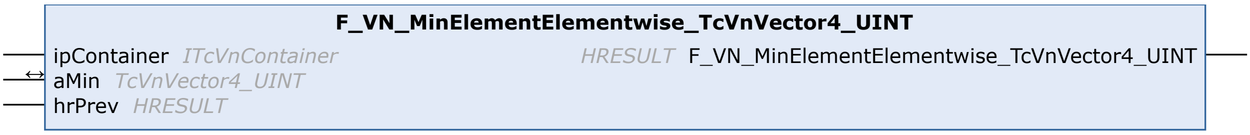
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.26 F_VN_MinElementElementwise_TcVnVector4_UINT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnVector4_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnVector4_UINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnVector4_UINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

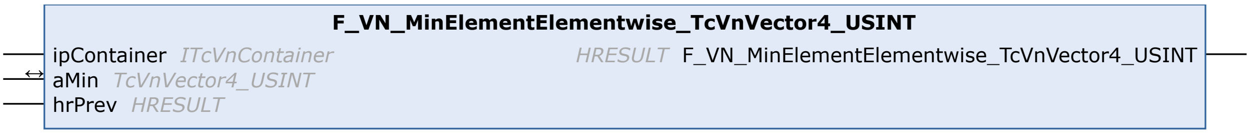
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.27 F_VN_MinElementElementwise_TcVnVector4_USINT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_TcVnVector4_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aMin        : TcVnVector4_USINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMin	TcVnVector4_USINT [▶ 141]	Returns the requested element

Return value

HRESULT [▶ 122]

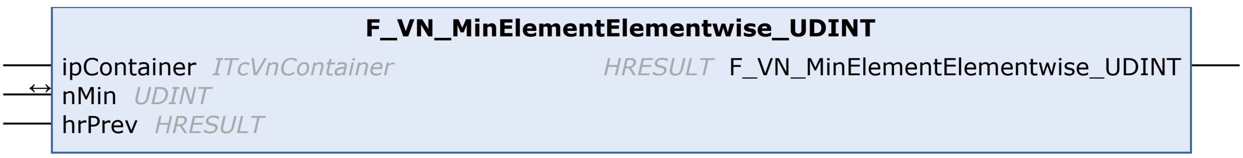
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.28 F_VN_MinElementElementwise_UDINT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_UDINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMin        : UDINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMin	UDINT	Returns the requested element

Return value

HRESULT [▶ 122]

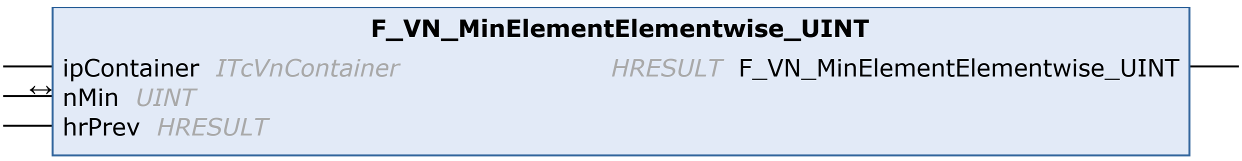
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.29 F_VN_MinElementElementwise_UINT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_UINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMin        : UINT;
END_VAR
```

```
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▸ 349]	Source container
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMin	UINT	Returns the requested element

Return value

HRESULT [▸ 122]

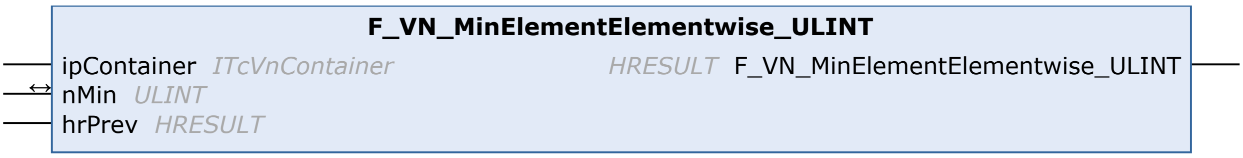
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.30 F_VN_MinElementElementwise_ULINT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_ULINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMin         : ULINT;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMin	ULINT	Returns the requested element

Return value

HRESULT [▶ 122]

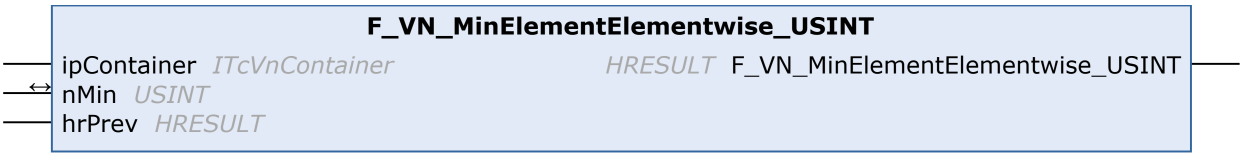
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.3.31 F_VN_MinElementElementwise_USINT



Gets the element wise minimum container element.

Syntax

Definition:

```
FUNCTION F_VN_MinElementElementwise_USINT : HRESULT
VAR_INPUT
    ipContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    nMin        : USINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer	ITcVnContainer [▶ 349]	Source container
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nMin	USINT	Returns the requested element

Return value

HRESULT [▶ 122]

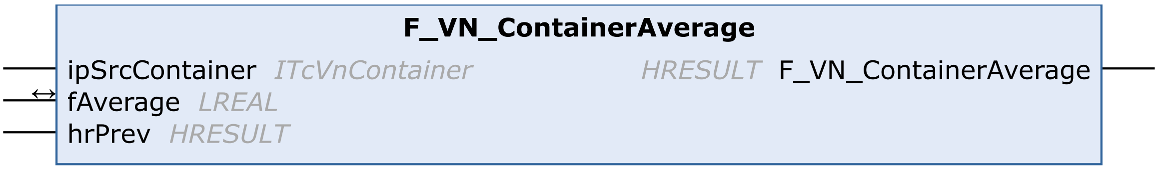
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.4 F_VN_ContainerAverage



Calculate the average of elements in a container.

Syntax

Definition:

```
FUNCTION F_VN_ContainerAverage : HRESULT
VAR_INPUT
    ipSrcContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fAverage       : LREAL;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer	ITcVnContainer [▶ 349]	Source container (Arithmetic, point and vector element types are supported. For points and vectors, the euclidean norm is used.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fAverage	LREAL	Returns the calculated average value

Return value

HRESULT [▶ 122]

更多信息

下表显示了哪些功能可用于哪种容器类型：

容器类型	可用功能
元素的容器	F_VN_ContainerAverage [▶ 900] F_VN_ContainerAverageVariance [▶ 905]
向量的容器，每个都有 2 个元素	F_VN_ContainerAverageElementwise2 [▶ 901] F_VN_ContainerAverageVarianceElementwise2 [▶ 906]
向量的容器，每个都有 3 个元素	F_VN_ContainerAverageElementwise3 [▶ 902] F_VN_ContainerAverageVarianceElementwise3 [▶ 907]
向量的容器，每个都有 4 个元素	F_VN_ContainerAverageElementwise4 [▶ 904] F_VN_ContainerAverageVarianceElementwise4 [▶ 909]

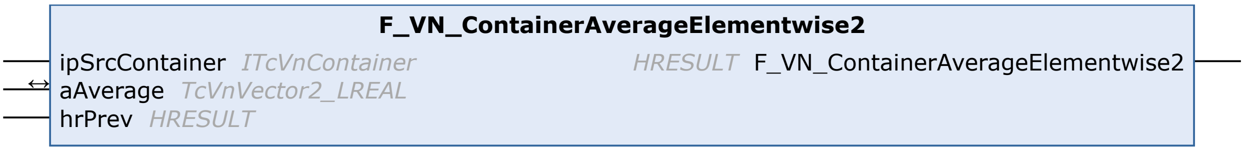
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.5 F_VN_ContainerAverageElementwise2



Calculate the elementwise average of 2D elements in a container.

Syntax

Definition:

```
FUNCTION F_VN_ContainerAverageElementwise2 : HRESULT
VAR_INPUT
    ipSrcContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aAverage       : TcVnVector2_LREAL;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer	ITcVnContainer [▶ 349]	Source container with 2D points or vectors
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aAverage	TcVnVector2_LREAL [▶ 141]	Returns the calculated average values

Return value

HRESULT [▶ 122]

更多信息

对于对每个容器元素有两个基本子元素的容器，这个函数逐个元素计算平均值。

下表显示了哪些功能可用于哪种容器类型：

容器类型	可用功能
元素的容器	F_VN_ContainerAverage [▶ 900] F_VN_ContainerAverageVariance [▶ 905]
向量的容器，每个都有 2 个元素	F_VN_ContainerAverageElementwise2 [▶ 901] F_VN_ContainerAverageVarianceElementwise2 [▶ 906]
向量的容器，每个都有 3 个元素	F_VN_ContainerAverageElementwise3 [▶ 902] F_VN_ContainerAverageVarianceElementwise3 [▶ 907]
向量的容器，每个都有 4 个元素	F_VN_ContainerAverageElementwise4 [▶ 904] F_VN_ContainerAverageVarianceElementwise4 [▶ 909]

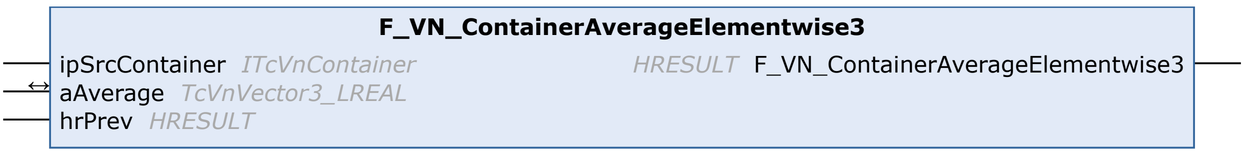
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.6 F_VN_ContainerAverageElementwise3



Calculate the elementwise average of 3D elements in a container.

Syntax

Definition:

```
FUNCTION F_VN_ContainerAverageElementwise3 : HRESULT
VAR_INPUT
    ipSrcContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aAverage       : TcVnVector3_LREAL;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer	ITcVnContainer [▶ 349]	Source container with 3D points or vectors
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aAverage	TcVnVector3_LREAL [▶ 141]	Returns the calculated average values

Return value

HRESULT [▶ 122]

更多信息

下表显示了哪些功能可用于哪种容器类型：

容器类型	可用功能
元素的容器	F_VN_ContainerAverage [▶ 900] F_VN_ContainerAverageVariance [▶ 905]
向量的容器，每个都有 2 个元素	F_VN_ContainerAverageElementwise2 [▶ 901] F_VN_ContainerAverageVarianceElementwise2 [▶ 906]
向量的容器，每个都有 3 个元素	F_VN_ContainerAverageElementwise3 [▶ 902] F_VN_ContainerAverageVarianceElementwise3 [▶ 907]
向量的容器，每个都有 4 个元素	F_VN_ContainerAverageElementwise4 [▶ 904] F_VN_ContainerAverageVarianceElementwise4 [▶ 909]

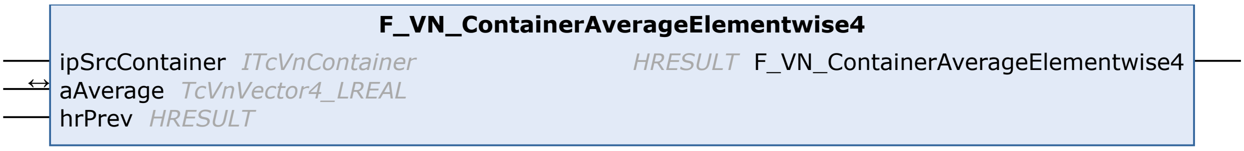
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.7 F_VN_ContainerAverageElementwise4



Calculate the elementwise average of 4D elements in a container.

Syntax

Definition:

```
FUNCTION F_VN_ContainerAverageElementwise4 : HRESULT
VAR_INPUT
    ipSrcContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aAverage       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer	ITcVnContainer [▶ 349]	Source container with 4D vectors
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aAverage	TcVnVector4_LREAL [▶ 141]	Returns the calculated average values

Return value

HRESULT [▶ 122]

更多信息

下表显示了哪些功能可用于哪种容器类型：

容器类型	可用功能
元素的容器	F_VN_ContainerAverage [▶ 900] F_VN_ContainerAverageVariance [▶ 905]
向量的容器，每个都有 2 个元素	F_VN_ContainerAverageElementwise2 [▶ 901] F_VN_ContainerAverageVarianceElementwise2 [▶ 906]
向量的容器，每个都有 3 个元素	F_VN_ContainerAverageElementwise3 [▶ 902] F_VN_ContainerAverageVarianceElementwise3 [▶ 907]
向量的容器，每个都有 4 个元素	F_VN_ContainerAverageElementwise4 [▶ 904] F_VN_ContainerAverageVarianceElementwise4 [▶ 909]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.8 F_VN_ContainerAverageVariance

Calculate the average and variance of elements in a container.

Syntax

Definition:

```
FUNCTION F_VN_ContainerAverageVariance : HRESULT
VAR_INPUT
    ipSrcContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fAverage       : LREAL;
    fVariance      : LREAL;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer	ITcVnContainer [▸ 349]	Source container (Arithmetic, point and vector element types are supported. For points and vectors, the euclidean norm is used.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fAverage	LREAL	Returns the calculated average value
fVariance	LREAL	Returns the calculated variance value

Return value

HRESULT [▸ 122]

更多信息

下表显示了哪些功能可用于哪种容器类型：

容器类型	可用功能
元素的容器	F_VN_ContainerAverage [▶ 900] F_VN_ContainerAverageVariance [▶ 905]
向量的容器，每个都有 2 个元素	F_VN_ContainerAverageElementwise2 [▶ 901] F_VN_ContainerAverageVarianceElementwise2 [▶ 906]
向量的容器，每个都有 3 个元素	F_VN_ContainerAverageElementwise3 [▶ 902] F_VN_ContainerAverageVarianceElementwise3 [▶ 907]
向量的容器，每个都有 4 个元素	F_VN_ContainerAverageElementwise4 [▶ 904] F_VN_ContainerAverageVarianceElementwise4 [▶ 909]

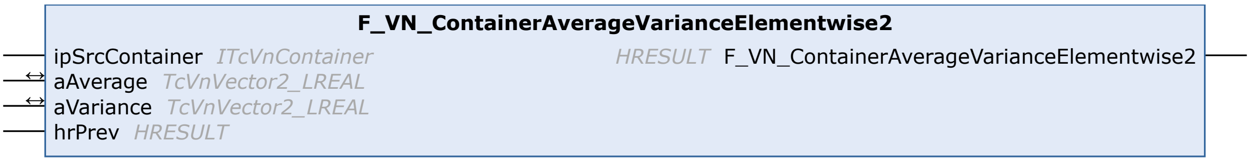
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.9 F_VN_ContainerAverageVarianceElementwise2



Calculate the elementwise average and variance of 2D elements in a container.

Syntax

Definition:

```
FUNCTION F_VN_ContainerAverageVarianceElementwise2 : HRESULT
VAR_INPUT
    ipSrcContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aAverage       : TcVnVector2_LREAL;
    aVariance      : TcVnVector2_LREAL;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer	ITcVnContainer [▶ 349]	Source container with 2D points or vectors
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aAverage	TcVnVector2_LREAL [▶ 141]	Returns the calculated average values
aVariance	TcVnVector2_LREAL [▶ 141]	Returns the calculated variance values

Return value

HRESULT [▶ 122]

更多信息

下表显示了哪些功能可用于哪种容器类型：

容器类型	可用功能
元素的容器	F_VN_ContainerAverage [▶ 900] F_VN_ContainerAverageVariance [▶ 905]
向量的容器，每个都有 2 个元素	F_VN_ContainerAverageElementwise2 [▶ 901] F_VN_ContainerAverageVarianceElementwise2 [▶ 906]
向量的容器，每个都有 3 个元素	F_VN_ContainerAverageElementwise3 [▶ 902] F_VN_ContainerAverageVarianceElementwise3 [▶ 907]
向量的容器，每个都有 4 个元素	F_VN_ContainerAverageElementwise4 [▶ 904] F_VN_ContainerAverageVarianceElementwise4 [▶ 909]

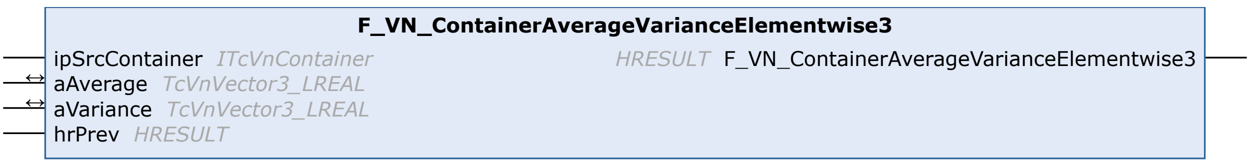
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.10 F_VN_ContainerAverageVarianceElementwise3



Calculate the elementwise average and variance of 3D elements in a container.

Syntax

Definition:

```
FUNCTION F_VN_ContainerAverageVarianceElementwise3 : HRESULT
VAR_INPUT
    ipSrcContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aAverage       : TcVnVector3_LREAL;
    aVariance      : TcVnVector3_LREAL;
```

```
END_VAR
VAR_INPUT
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer	ITcVnContainer [▶ 349]	Source container with 3D points or vectors
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aAverage	TcVnVector3_LREAL [▶ 141]	Returns the calculated average values
aVariance	TcVnVector3_LREAL [▶ 141]	Returns the calculated variance values

Return value

HRESULT [▶ 122]

更多信息

下表显示了哪些功能可用于哪种容器类型：

容器类型	可用功能
元素的容器	F_VN_ContainerAverage [▶ 900] F_VN_ContainerAverageVariance [▶ 905]
向量的容器，每个都有 2 个元素	F_VN_ContainerAverageElementwise2 [▶ 901] F_VN_ContainerAverageVarianceElementwise2 [▶ 906]
向量的容器，每个都有 3 个元素	F_VN_ContainerAverageElementwise3 [▶ 902] F_VN_ContainerAverageVarianceElementwise3 [▶ 907]
向量的容器，每个都有 4 个元素	F_VN_ContainerAverageElementwise4 [▶ 904] F_VN_ContainerAverageVarianceElementwise4 [▶ 909]

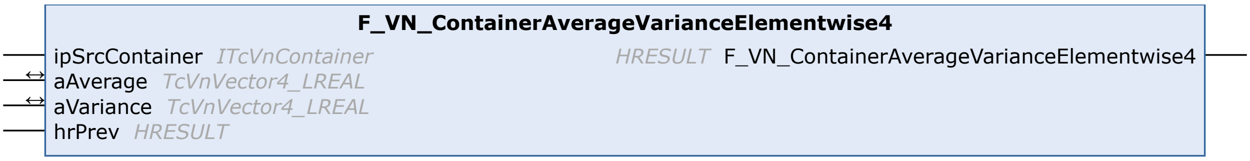
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.11 F_VN_ContainerAverageVarianceElementwise4



Calculate the elementwise average and variance of 4D elements in a container.

Syntax

Definition:

```
FUNCTION F_VN_ContainerAverageVarianceElementwise4 : HRESULT
VAR_INPUT
    ipSrcContainer : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aAverage       : TcVnVector4_LREAL;
    aVariance      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer	ITcVnContainer [▸ 349]	Source container with 4D vectors
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aAverage	TcVnVector4_LREAL [▸ 141]	Returns the calculated average values
aVariance	TcVnVector4_LREAL [▸ 141]	Returns the calculated variance values

Return value

HRESULT [▸ 122]

更多信息

下表显示了哪些功能可用于哪种容器类型：

容器类型	可用功能
元素的容器	F_VN_ContainerAverage [▶ 900] F_VN_ContainerAverageVariance [▶ 905]
向量的容器，每个都有 2 个元素	F_VN_ContainerAverageElementwise2 [▶ 901] F_VN_ContainerAverageVarianceElementwise2 [▶ 906]
向量的容器，每个都有 3 个元素	F_VN_ContainerAverageElementwise3 [▶ 902] F_VN_ContainerAverageVarianceElementwise3 [▶ 907]
向量的容器，每个都有 4 个元素	F_VN_ContainerAverageElementwise4 [▶ 904] F_VN_ContainerAverageVarianceElementwise4 [▶ 909]

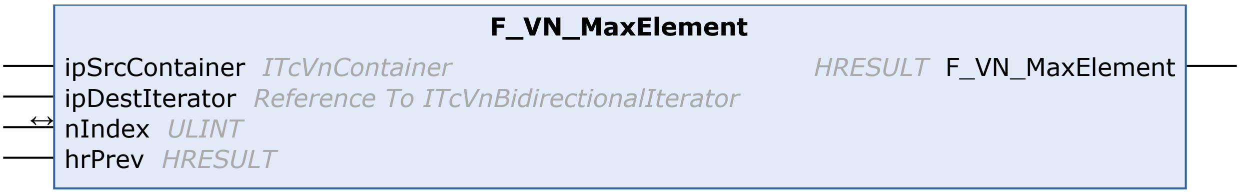
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.12 F_VN_MaxElement



Gets the maximum element of a container.

Syntax

Definition:

```
FUNCTION F_VN_MaxElement : HRESULT
VAR_INPUT
    ipSrcContainer : ITcVnContainer;
    ipDestIterator : Reference To ITcVnBidirectionalIterator;
END_VAR
VAR_IN_OUT
    nIndex         : ULINT;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer	ITcVnContainer [▸ 349]	Source container (Arithmetic, point and vector element types are supported. For points and vectors, the euclidean norm is used.)
ipDestIterator	Reference To ITcVnBidirectionalIterator [▸ 337]	Returns an iterator to the requested element
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nIndex	ULINT	Returns the position of the requested element within the container

Return value

HRESULT [▸ 122]

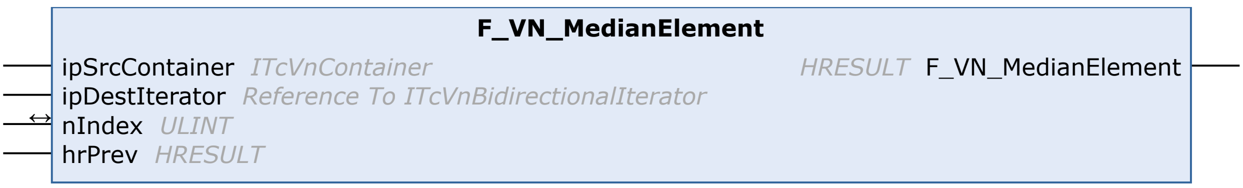
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.13 F_VN_MedianElement



Gets the median element of a container.

Syntax

Definition:

```
FUNCTION F_VN_MedianElement : HRESULT
VAR_INPUT
    ipSrcContainer : ITcVnContainer;
    ipDestIterator : Reference To ITcVnBidirectionalIterator;
END_VAR
VAR_IN_OUT
    nIndex         : ULINT;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer	ITcVnContainer [▶ 349]	Source container (Arithmetic, point and vector element types are supported. For points and vectors, the euclidean norm is used.)
ipDestIterator	Reference To ITcVnBidirectionalIterator [▶ 337]	Returns an iterator to the requested element
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nIndex	ULINT	Returns the position of the requested element within the container

Return value

HRESULT [▶ 122]

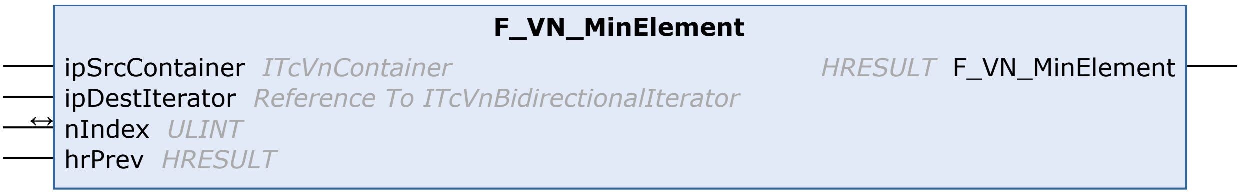
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.8.14 F_VN_MinElement



Gets the minimum element of a container.

Syntax

Definition:

```
FUNCTION F_VN_MinElement : HRESULT
VAR_INPUT
    ipSrcContainer : ITcVnContainer;
    ipDestIterator : Reference To ITcVnBidirectionalIterator;
END_VAR
VAR_IN_OUT
    nIndex         : ULINT;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContainer	ITcVnContainer [▸ 349]	Source container (Arithmetic, point and vector element types are supported. For points and vectors, the euclidean norm is used.)
ipDestIterator	Reference To ITcVnBidirectionalIterator [▸ 337]	Returns an iterator to the requested element
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nIndex	ULINT	Returns the position of the requested element within the container

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9 Contour Analysis

该组包含用于分析轮廓 [▸ 135]的函数：

函数

形状近似

- F_VN_ApproximatePolygon [▸ 914]
- F_VN_ConvexHullPoints (Exp) [▸ 929]
- F_VN_EnclosingCircle [▸ 934]
- F_VN_EnclosingRectangle [▸ 935]
- F_VN_EnclosingTriangle [▸ 936]
- F_VN_FitEllipse [▸ 937]
- F_VN_FitLine (Exp) [▸ 939]
- F_VN_UprightBoundingRectangle [▸ 947]

点

- F_VN_CheckIfPointIsInsideContour [▸ 915]
- F_VN_ContourExtremePoint [▸ 922]
- F_VN_ConvexityDefects [▸ 931]

几何特征

- F_VN_ContourArea [▶ 916]
- F_VN_ContourCenterOfMass [▶ 917]
- F_VN_ContourCircularity [▶ 918]
- F_VN_ContourConvexity [▶ 919]
- F_VN_ContourEccentricity [▶ 920]
- F_VN_ContourElongation [▶ 921]
- F_VN_ContourInertiaRatio [▶ 923]
- F_VN_ContourMoments [▶ 924]
- F_VN_ContourOrientation(Exp) [▶ 925]
- F_VN_ContourPerimeter [▶ 927]
- F_VN_ContourRoundness [▶ 928]
- F_VN_FourierDescriptors [▶ 940]

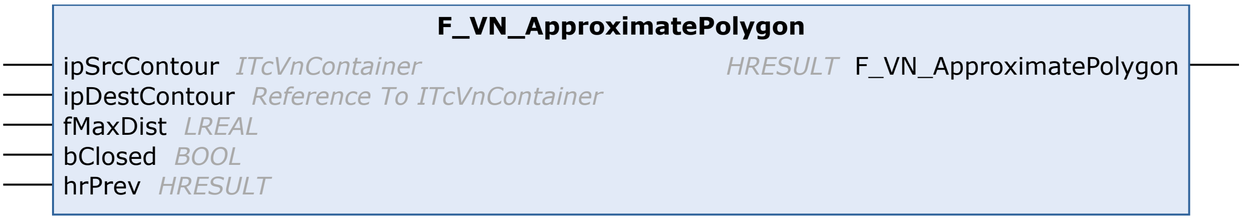
匹配

- F_VN_MatchContours(Exp) [▶ 945]
- F_VN_MatchContours1vsN(Exp) [▶ 944]

用户定义的操作

- F_VN_CustomElementWiseContainerOperation_ITcVnContainer [▶ 932]
- F_VN_CustomElementWiseContainerOperation_ITcVnForwardIterator [▶ 933]

6.1.4.9.1 F_VN_ApproximatePolygon



Approximate a contour to a simplified polygon (using the Douglas-Peucker algorithm).

Syntax

Definition:

```
FUNCTION F_VN_ApproximatePolygon : HRESULT
VAR_INPUT
    ipSrcContour  : ITcVnContainer;
    ipDestContour : Reference To ITcVnContainer;
    fMaxDist      : LREAL;
    bClosed       : BOOL;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcContour	ITcVnContainer [▶ 349]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
ipDestContour	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the approximated polygon points (same type ID as ipSrcContour; Non-zero interface pointers are reused.)
fMaxDist	LREAL	Maximum distance between the original contour and its approximation
bClosed	BOOL	Specify, if the contour is closed (first and last points connected) or not
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.2 F_VN_CheckIfPointIsInsideContour

Checks if a point is inside a contour (and optionally return the distance to it).

Syntax

Definition:

```
FUNCTION F_VN_CheckIfPointIsInsideContour : HRESULT
VAR_INPUT
    ipContour         : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aPoint            : TcVnPoint2_REAL;
    fDist             : LREAL;
END_VAR
VAR_INPUT
    bMeasureDistance : BOOL;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContour	ITcVnContainer [▸ 349]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL)
bMeasureDistance	BOOL	If true, fDist returns the distance to the nearest contour edge. Otherwise, fDist only returns -1, 0 or +1 (recommended to set to false if the distance is not required, as this is faster).
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aPoint	TcVnPoint2_REAL [▸ 139]	Point position to check
fDist	LREAL	Returns the result (> 0: the point is inside the contour; 0: the point is on the contour; < 0: the point is outside the contour)

Return value

HRESULT [▸ 122]

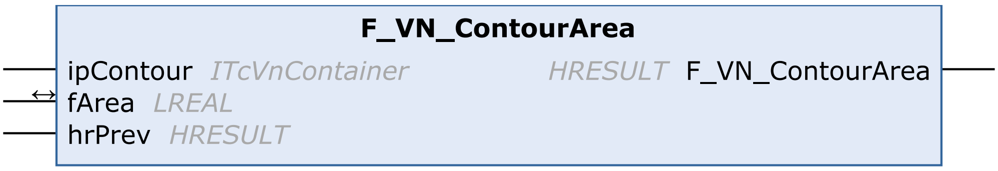
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.3 F_VN_ContourArea



Estimate the contour area using Green's formula.

Syntax

Definition:

```
FUNCTION F_VN_ContourArea : HRESULT
VAR_INPUT
    ipContour : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fArea     : LREAL;
END_VAR
VAR_INPUT
    hrPrev    : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContour	ITcVnContainer [▶ 349]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fArea	LREAL	Returns the estimated contour area (The actual contour area may differ depending on the shape of the contour.)

Return value

HRESULT [▶ 122]

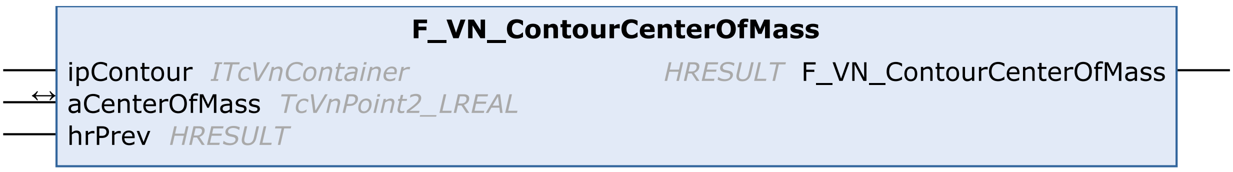
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.4 F_VN_ContourCenterOfMass



Computes the center of mass of a contour.

Syntax

Definition:

```
FUNCTION F_VN_ContourCenterOfMass : HRESULT
VAR_INPUT
    ipContour      : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aCenterOfMass : TcVnPoint2_LREAL;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContour	ITcVnContainer [▶ 349]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCenterOfMass	TcVnPoint2_LREAL [▶ 139]	Returns the center of mass of the contour

Return value

HRESULT [▶ 122]

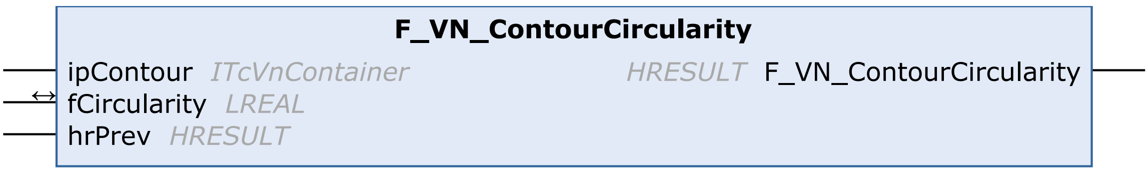
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.5 F_VN_ContourCircularity



Computes the circularity of a contour (1.0: ideal circle, 0.0: straight line).

Syntax

Definition:

```
FUNCTION F_VN_ContourCircularity : HRESULT
VAR_INPUT
    ipContour     : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fCircularity : LREAL;
END_VAR
VAR_INPUT
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContour	ITcVnContainer [▶ 349]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fCircularity	LREAL	Returns the circularity of the contour [0..1]

Return value

HRESULT [▶ 122]

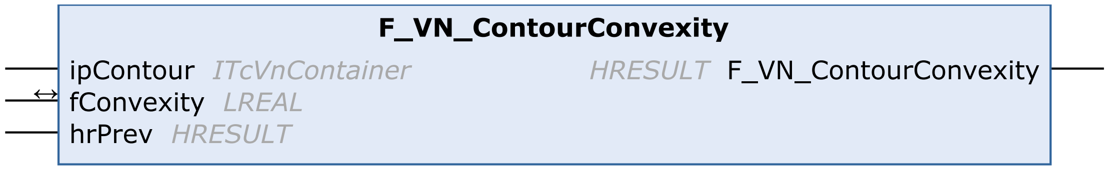
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.6 F_VN_ContourConvexity



Computes the convexity of a contour as the ratio between the contour's area and the area of the convex hull (1.0: convex shapes (i.e. rectangle, circle, ..., etc.), 0.0: zero area (i.e. a line or an empty contour)).

Syntax

Definition:

```
FUNCTION F_VN_ContourConvexity : HRESULT
VAR_INPUT
    ipContour   : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fConvexity : LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContour	ITcVnContainer [▸ 349]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fConvexity	LREAL	Returns the convexity of the contour [0..1]

Return value

HRESULT [▸ 122]

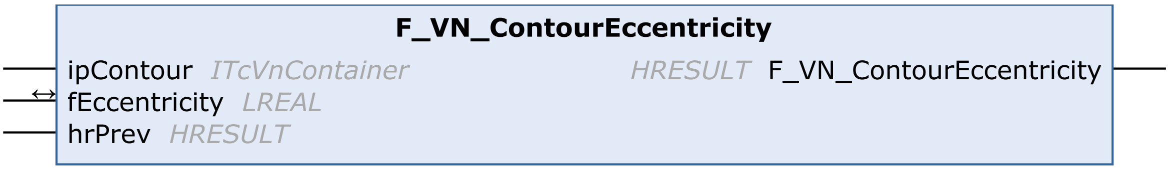
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.7 F_VN_ContourEccentricity



Computes the eccentricity of a contour (0.0: circular, 1.0: linear).

Syntax

Definition:

```
FUNCTION F_VN_ContourEccentricity : HRESULT
VAR_INPUT
    ipContour      : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fEccentricity : LREAL;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContour	ITcVnContainer [▶ 349]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fEccentricity	LREAL	Returns the eccentricity of the contour [0..1]

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.8 F_VN_ContourElongation

Computes the elongation factor of a contour.

Syntax

Definition:

```
FUNCTION F_VN_ContourElongation : HRESULT
VAR_INPUT
    ipContour    : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fElongation : LREAL;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContour	ITcVnContainer [▶ 349]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fElongation	LREAL	Returns the elongation of the contour

Return value

HRESULT [▶ 122]

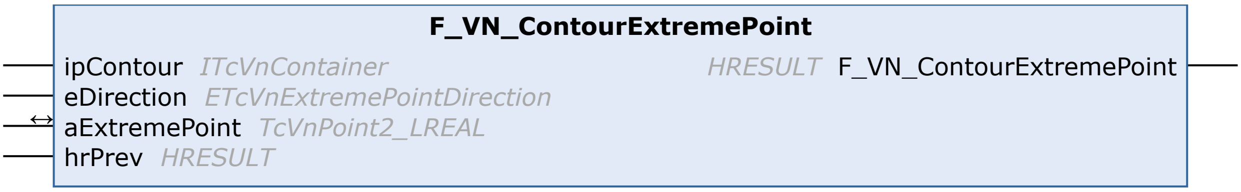
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.9 F_VN_ContourExtremePoint



Finds the contour extreme point.

Syntax

Definition:

```
FUNCTION F_VN_ContourExtremePoint : HRESULT
VAR_INPUT
    ipContour     : ITcVnContainer;
    eDirection    : ETcVnExtremePointDirection;
END_VAR
VAR_IN_OUT
    aExtremePoint : TcVnPoint2_LREAL;
END_VAR
VAR_INPUT
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContour	ITcVnContainer [▶ 349]	Source container with 2D points (TcVnPoint2_DINT or TcVnPoint2_REAL or TcVnPoint2_LREAL)
eDirection	ETcVnExtremePointDirection [▶ 179]	Selects the search direction for the extreme point
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aExtremePoint	TcVnPoint2_LREAL [▶ 139]	Returns the extreme point of the contour, according to eDirection

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.10 F_VN_ContourInertiaRatio

Computes the inertia ratio of a contour which reflects how a contour/shape is elongated (i.e. circle: 1.0, line: 0.0).

Syntax

Definition:

```
FUNCTION F_VN_ContourInertiaRatio : HRESULT
VAR_INPUT
    ipContour      : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fInertiaRatio : LREAL;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContour	ITcVnContainer [▶ 349]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fInertiaRatio	LREAL	Returns the inertia raio of the contour [0..1]

Return value

HRESULT [▶ 122]

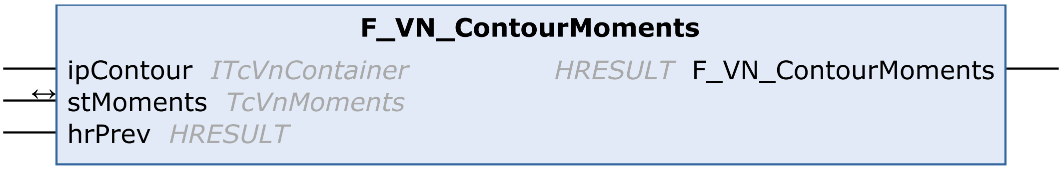
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.11 F_VN_ContourMoments



Computes the spatial moments, the central moments, and the central normalized moments of a contour up to the third order.

Syntax

Definition:

```
FUNCTION F_VN_ContourMoments : HRESULT
VAR_INPUT
    ipContour : ITcVnContainer;
END_VAR
VAR_IN_OUT
    stMoments : TcVnMoments;
END_VAR
VAR_INPUT
    hrPrev    : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContour	ITcVnContainer [▶ 349]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stMoments	TcVnMoments [▶ 211]	Returns a struct containing the moments

Return value

HRESULT [▶ 122]

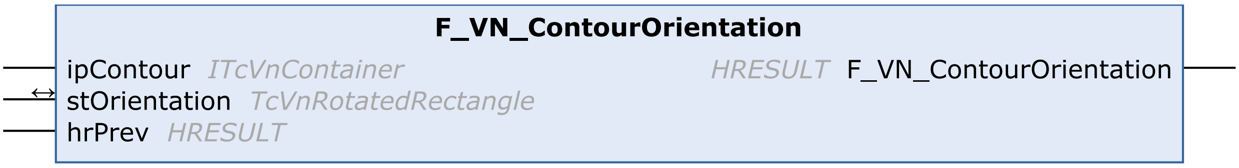
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.12 F_VN_ContourOrientation



Calculate the orientation of a contour or a set of points.

Syntax

Definition:

```
FUNCTION F_VN_ContourOrientation : HRESULT
VAR_INPUT
    ipContour     : ITcVnContainer;
END_VAR
VAR_IN_OUT
    stOrientation : TcVnRotatedRectangle;
END_VAR
VAR_INPUT
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContour	ITcVnContainer [▶ 349]	Source point set (ContainerType_Vector_TcVnPoint2_DINT, ContainerType_Vector_TcVnPoint2_REAL, or ContainerType_Vector_TcVnPoint2_LREAL. The elements of this container are the individual points.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stOrientation	TcVnRotatedRectangle [▶ 225]	Resulting rotated rectangle, containig center of mass, lenghts of axes, and rotation angle of the contour in clockwise direction.

Return value

HRESULT [▶ 122]

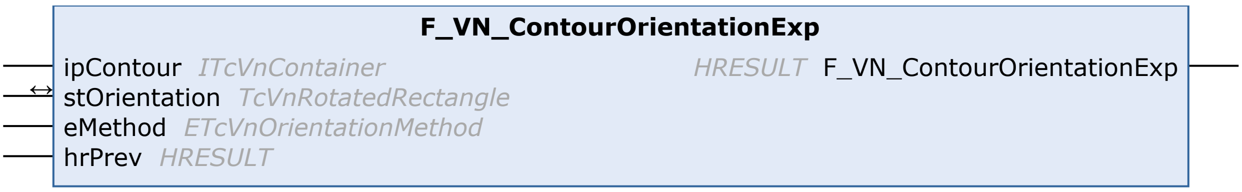
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.13 F_VN_ContourOrientationExp



Calculate the orientation of a contour or a set of points. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ContourOrientationExp : HRESULT
VAR_INPUT
    ipContour     : ITcVnContainer;
END_VAR
VAR_IN_OUT
    stOrientation : TcVnRotatedRectangle;
END_VAR
VAR_INPUT
    eMethod       : ETcVnOrientationMethod;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContour	ITcVnContainer [▶ 349]	Source point set (ContainerType_Vector_TcVnPoint2_DINT, ContainerType_Vector_TcVnPoint2_REAL, or ContainerType_Vector_TcVnPoint2_LREAL. The elements of this container are the individual points.)
eMethod	ETcVnOrientationMethod [▶ 191]	Method for calculating the orientation.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stOrientation	TcVnRotatedRectangle [▶ 225]	Resulting rotated rectangle, containig center of mass, lenghts of axes, and rotation angle of the contour in clockwise direction.

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.14 F_VN_ContourPerimeter

Computes the perimeter of a contour (curve length if the contour is not closed).

Syntax

Definition:

```
FUNCTION F_VN_ContourPerimeter : HRESULT
VAR_INPUT
    ipContour  : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fPerimeter : LREAL;
END_VAR
VAR_INPUT
    bClosed    : BOOL;
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContour	ITcVnContainer [▶ 349]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
bClosed	BOOL	Specifys, if the contour is closed (first and last points connected) or not
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fPerimeter	LREAL	Returns the perimeter of the contour

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.15 F_VN_ContourRoundness

Computes the roundness of a contour (perimeter^2 / area).

Syntax

Definition:

```
FUNCTION F_VN_ContourRoundness : HRESULT
VAR_INPUT
    ipContour  : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fRoundness : LREAL;
END_VAR
VAR_INPUT
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContour	ITcVnContainer [▸ 349]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fRoundness	LREAL	Returns the roundness of the contour

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.16 F_VN_ConvexHullPoints

Determines the convex hull of a point set.

Syntax

Definition:

```
FUNCTION F_VN_ConvexHullPoints : HRESULT
VAR_INPUT
    ipPointSet   : ITcVnContainer;
    ipConvexHull : Reference To ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipPointSet	ITcVnContainer [▶ 349]	Source point set (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the individual points.)
ipConvexHull	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the sorted convex hull points (same type ID as ipPointSet; Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.17 F_VN_ConvexHullPointsExp

Determines the convex hull of a point set. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ConvexHullPointsExp : HRESULT
VAR_INPUT
    ipPointSet   : ITcVnContainer;
    ipConvexHull : Reference To ITcVnContainer;
    bClockwise   : BOOL;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipPointSet	ITcVnContainer [▶ 349]	Source point set (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the individual points.)
ipConvexHull	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the sorted convex hull points (same type ID as ipPointSet; Non-zero interface pointers are reused.)
bClockwise	BOOL	Selects the sorting direction of the hull points, assuming cartesian coordinates.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.18 F_VN_ConvexityDefects

Finds the convexity defects of a point set.

Syntax

Definition:

```
FUNCTION F_VN_ConvexityDefects : HRESULT
VAR_INPUT
    ipContour           : ITcVnContainer;
    ipConvexityDefects : Reference To ITcVnContainer;
    hrPrev              : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContour	ITcVnContainer [▶ 349]	Source contour (ContainerType_Vector_TcVnPoint2_DINT; The elements of this container are the individual points.)
ipConvexityDefects	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the convexity defects (ContainerType_Vector_TcVnVector4_DINT with [startIndex, endIndex, furthestPointIndex, fixedPointDistance (8 fraction bits, i.e. divide by 256 to get the distance in pixels)]; Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

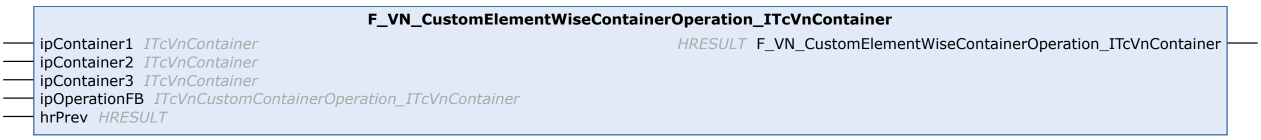
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.19 F_VN_CustomElementWiseContainerOperation_ITcVnContainer



Performs a custom, element wise operation on a set of up to 3 containers.

Syntax

Definition:

```
FUNCTION F_VN_CustomElementWiseContainerOperation_ITcVnContainer : HRESULT
VAR_INPUT
    ipContainer1  : ITcVnContainer;
    ipContainer2  : ITcVnContainer;
    ipContainer3  : ITcVnContainer;
    ipOperationFB : ITcVnCustomContainerOperation_ITcVnContainer;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer1	ITcVnContainer [▶ 349]	Container 1
ipContainer2	ITcVnContainer [▶ 349]	Container 2
ipContainer3	ITcVnContainer [▶ 349]	Container 3
ipOperationFB	ITcVnCustomContainerOperation_ITcVnContainer [▶ 226]	Custom operation on the container elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

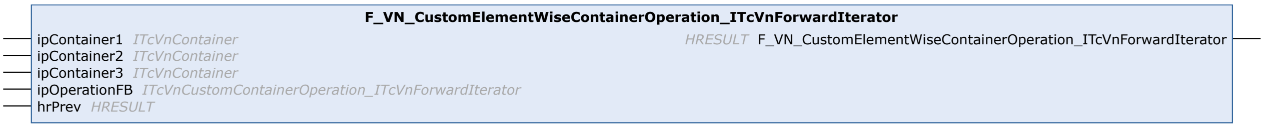
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.20 F_VN_CustomElementWiseContainerOperation_ITcVnForwardIterator



Performs a custom, element wise operation on a set of up to 3 containers.

Syntax

Definition:

```
FUNCTION F_VN_CustomElementWiseContainerOperation_ITcVnForwardIterator : HRESULT
VAR_INPUT
    ipContainer1  : ITcVnContainer;
    ipContainer2  : ITcVnContainer;
    ipContainer3  : ITcVnContainer;
    ipOperationFB : ITcVnCustomContainerOperation_ITcVnForwardIterator;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContainer1	ITcVnContainer [▸ 349]	Container 1
ipContainer2	ITcVnContainer [▸ 349]	Container 2
ipContainer3	ITcVnContainer [▸ 349]	Container 3
ipOperationFB	ITcVnCustomContainerOperation_ITcVnForwardIterator [▸ 227]	Custom operation on the elements
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

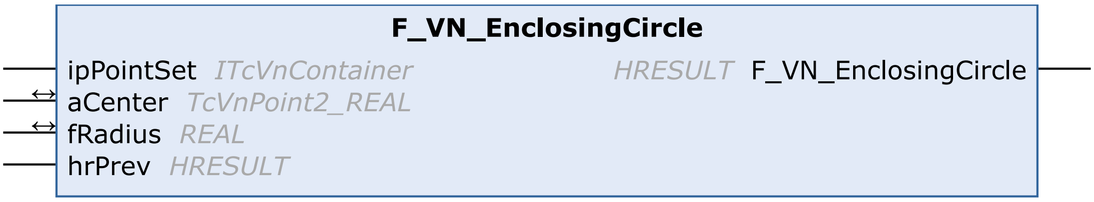
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.21 F_VN_EnclosingCircle



Searchs for a minimum area circle enclosing a set of points.

Syntax

Definition:

```
FUNCTION F_VN_EnclosingCircle : HRESULT
VAR_INPUT
    ipPointSet : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aCenter    : TcVnPoint2_REAL;
    fRadius    : REAL;
END_VAR
VAR_INPUT
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipPointSet	ITcVnContainer [▶ 349]	Source point set (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the individual points.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCenter	TcVnPoint2_REAL [▶ 139]	Returns the center of the circle
fRadius	REAL	Returns the radius of the circle

Return value

HRESULT [▶ 122]

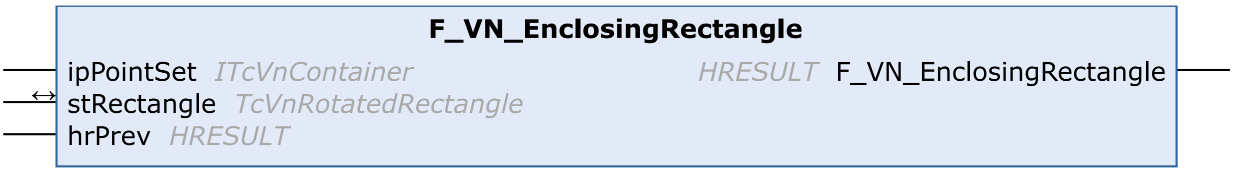
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.22 F_VN_EnclosingRectangle



Searches for a minimum area rectangle enclosing a set of points.

Syntax

Definition:

```
FUNCTION F_VN_EnclosingRectangle : HRESULT
VAR_INPUT
    ipPointSet  : ITcVnContainer;
END_VAR
VAR_IN_OUT
    stRectangle : TcVnRotatedRectangle;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipPointSet	ITcVnContainer [▸ 349]	Source point set (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the individual points.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stRectangle	TcVnRotatedRectangle [▸ 225]	Returns the determined rectangle

Return value

HRESULT [▸ 122]

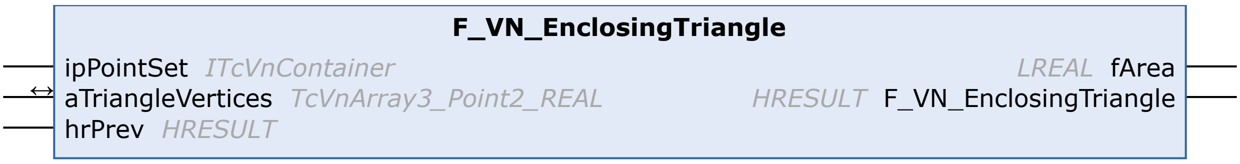
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.23 F_VN_EnclosingTriangle



Searches for a minimum area triangle enclosing a set of points.

Syntax

Definition:

```
FUNCTION F_VN_EnclosingTriangle : HRESULT
VAR_INPUT
    ipPointSet        : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aTriangleVertices : TcVnArray3_Point2_REAL;
END_VAR
VAR_INPUT
    hrPrev            : HRESULT;
END_VAR
VAR_OUTPUT
    fArea             : LREAL;
END_VAR
```

Inputs

Name	Type	Description
ipPointSet	ITcVnContainer [▶ 349]	Source point set (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the individual points.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTriangleVertices	TcVnArray3_Point2_REAL [▶ 141]	Returns the 3 triangle vertices

Outputs

Name	Type	Description
fArea	LREAL	Returns the triangle area

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.24 F_VN_FitEllipse

Fit an ellipse in a point set.

Syntax

Definition:

```
FUNCTION F_VN_FitEllipse : HRESULT
VAR_INPUT
    ipPointSet : ITcVnContainer;
END_VAR
VAR_IN_OUT
    stEllipse  : TcVnRotatedRectangle;
END_VAR
VAR_INPUT
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipPointSet	ITcVnContainer [▶ 349]	Source point set (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; At least 5 reasonable points are required! The elements of this container are the individual points.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stEllipse	TcVnRotatedRectangle [▶ 225]	Resulting ellipse, described by a rotated rectangle.

Return value

HRESULT [▶ 122]

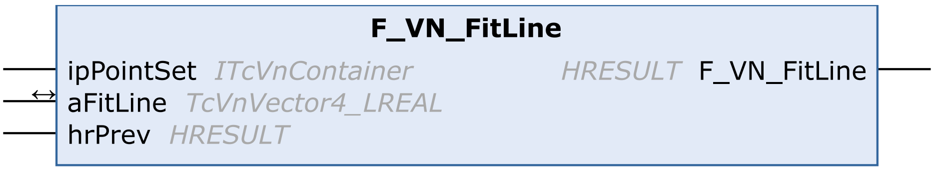
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.25 F_VN_FitLine



Fit a line into a point set.

Syntax

Definition:

```
FUNCTION F_VN_FitLine : HRESULT
VAR_INPUT
    ipPointSet : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aFitLine   : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipPointSet	ITcVnContainer [▸ 349]	Source point set (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the individual points.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aFitLine	TcVnVector4_LREAL [▸ 141]	Resulting line. The first and second element describe the x and y component of a vector collinear to the line. The third and fourth element describe the x and y component of a point on the line.

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.26 F_VN_FitLineExp

Fit a line into a point set. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FitLineExp : HRESULT
VAR_INPUT
    ipPointSet     : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aFitLine       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    eDistanceType : ETcVnDistanceType;
    fParam         : LREAL;
    fEpsRadius     : LREAL;
```

```
    fEpsAngle       : LREAL;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipPointSet	ITcVnContainer [▶ 349]	Source point set (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the individual points.)
eDistanceType	ETcVnDistanceType [▶ 175]	Distance computation method (supported: L1, L2, L12, FAIR, WELSCH, HUBER)
fParam	LREAL	Numerical parameter (c) for some eDistanceType (should be >= 0). If 0, an optimal value is chosen internally.
fEpsRadius	LREAL	Accuracy of the radius (distance of the line from the coordinate origin, should be > 0). A smaller value means higher accuracy.
fEpsAngle	LREAL	Accuracy of the angle (should be > 0). A smaller value means higher accuracy.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aFitLine	TcVnVector4_LREAL [▶ 141]	Resulting line. The first and second element describe the x and y component of a vector collinear to the line. The third and fourth element describe the x and y component of a point on the line.

Return value

HRESULT [▶ 122]

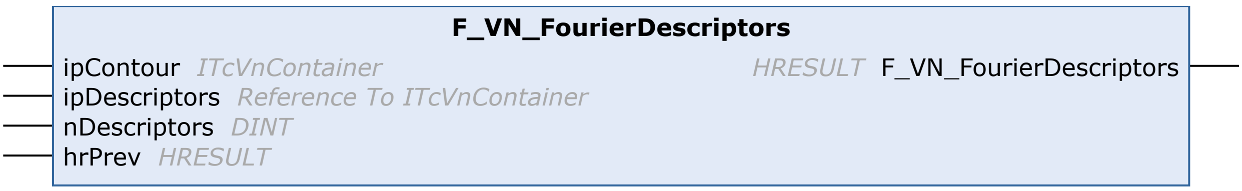
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.27 F_VN_FourierDescriptors



Computes the fourier descriptors for a closed contour.

Syntax

Definition:

```
FUNCTION F_VN_FourierDescriptors : HRESULT
VAR_INPUT
    ipContour      : ITcVnContainer;
    ipDescriptors : Reference To ITcVnContainer;
    nDescriptors  : DINT;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContour	ITcVnContainer [▸ 349]	Closed contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_LREAL. Providing the full contour is strongly recommended, i.e. use TCVN_CAM_NONE for contour detection algorithms.)
ipDescriptors	Reference To ITcVnContainer [▸ 349]	Returns the fourier descriptors for ipContour (Re0, Im0, Re1, Im1, ...; ContainerType_Vector_LREAL if ipContour is of type ContainerType_Vector_TcVnPoint2_LREAL, else ContainerType_Vector_REAL.)
nDescriptors	DINT	Specifies how many descriptors should be returned (set to -1 to return all computed descriptors, i.e. at least as many as the number of contour points)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.28 F_VN_MatchContours

Compare contours using the Hu moment invariants. In case of multiple contours in each container, the best matches are found and the average dissimilarity over all matched contours is returned.

Syntax

Definition:

```
FUNCTION F_VN_MatchContours : HRESULT
VAR_INPUT
    ipContour1         : ITcVnContainer;
    ipContour2         : ITcVnContainer;
    eComparisonMethod : ETcVnContoursMatchComparisonMethod;
```

```
END_VAR
VAR_IN_OUT
    fDissimilarity    : LREAL;
END_VAR
VAR_INPUT
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContour1	ITcVnContainer [▶ 349]	First contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour) or collection of multiple contours (ContainerType_Vector_Vector_TcVnPoint2_DINT)
ipContour2	ITcVnContainer [▶ 349]	Second contour (same type as ipContour1)
eComparisonMethod	ETcVnContoursMatchComparisonMethod [▶ 172]	Method used for comparing the Hu moment invariants of the contours
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fDissimilarity	LREAL	Returns the dissimilarity of the contours depending on the chosen comparison method

Return value

HRESULT [▶ 122]

示例

- 匹配轮廓 1vsN（手动形状） [▶ 2655]
- 匹配轮廓（提取的形状） [▶ 2659]

相关函数

- F_VN_ContourMoments [▶ 924]
- F_VN_HuMomentInvariants [▶ 1483]
- F_VN_MatchContours [▶ 941]
- F_VN_MatchContours1vsN [▶ 943]

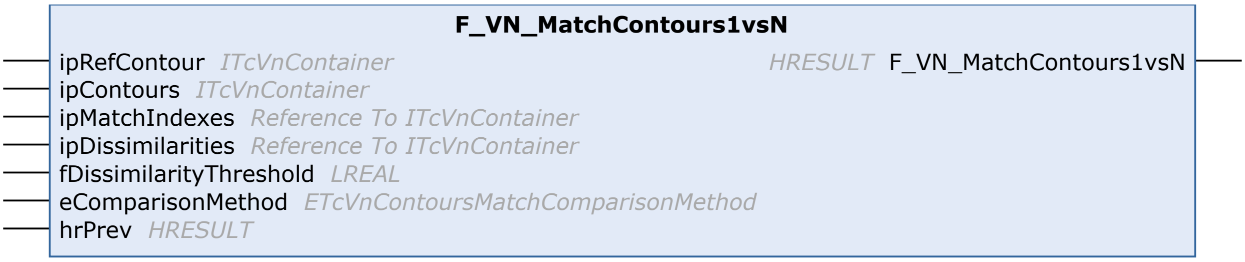
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.29 F_VN_MatchContours1vsN



Compare a reference contour with multiple other contours using the Hu moment invariants. Returns a sorted list of best matches.

Syntax

Definition:

```
FUNCTION F_VN_MatchContours1vsN : HRESULT
VAR_INPUT
    ipRefContour            : ITcVnContainer;
    ipContours              : ITcVnContainer;
    ipMatchIndexes          : Reference To ITcVnContainer;
    ipDissimilarities       : Reference To ITcVnContainer;
    fDissimilarityThreshold : LREAL;
    eComparisonMethod       : ETcVnContoursMatchComparisonMethod;
    hrPrev                  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipRefContour	ITcVnContainer [▶ 349]	Reference contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour)
ipContours	ITcVnContainer [▶ 349]	Collection of multiple contours (CTcVnContainer_Vector_Vector_TcVnPoint2_DINT)
ipMatchIndexes	Reference To ITcVnContainer [▶ 349]	Returns the indexes of the best matches (CTcVnContainer_Vector_ULINT; sorted, first element is best match, i.e. has lowest dissimilarity)
ipDissimilarities	Reference To ITcVnContainer [▶ 349]	Returns the computed dissimilarities of the best matches (CTcVnContainer_Vector_LREAL; Optional, set to 0 if not required; sorted corresponding to ipMatchIndexes)
fDissimilarityThreshold	LREAL	Neglect irrelevant matches, i.e. dissimilarity > fDissimilarityThreshold
eComparisonMethod	ETcVnContoursMatchComparisonMethod [▶ 172]	Method used for comparing the Hu moment invariants of the contours
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

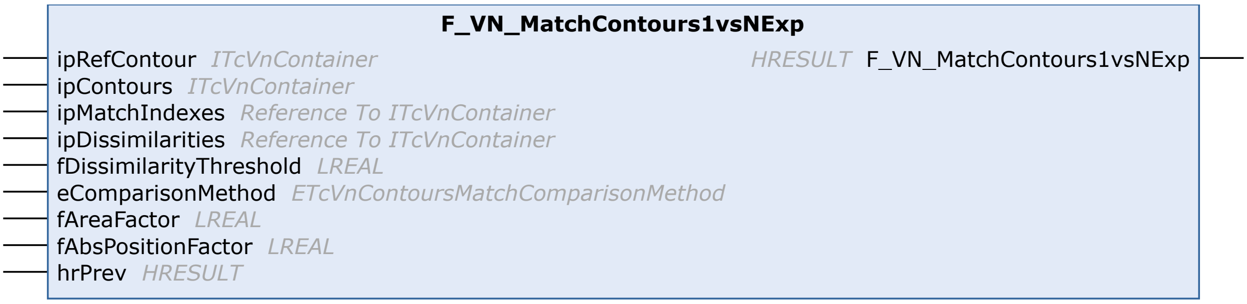
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.30 F_VN_MatchContours1vsNExp



Compare a reference contour with multiple other contours using the Hu moment invariants (and optionally further aspects). Returns a sorted list of best matches. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_MatchContours1vsNExp : HRESULT
VAR_INPUT
    ipRefContour            : ITcVnContainer;
    ipContours              : ITcVnContainer;
    ipMatchIndexes          : Reference To ITcVnContainer;
    ipDissimilarities       : Reference To ITcVnContainer;
    fDissimilarityThreshold : LREAL;
    eComparisonMethod       : ETcVnContoursMatchComparisonMethod;
    fAreaFactor             : LREAL;
    fAbsPositionFactor      : LREAL;
    hrPrev                  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipRefContour	ITcVnContainer [▶ 349]	Reference contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour)
ipContours	ITcVnContainer [▶ 349]	Collection of multiple contours (CTcVnContainer_Vector_Vector_TcVnPoint2_DINT)
ipMatchIndexes	Reference To ITcVnContainer [▶ 349]	Returns the indexes of the best matches (CTcVnContainer_Vector_ULINT; sorted, first element is best match, i.e. has lowest dissimilarity)
ipDissimilarities	Reference To ITcVnContainer [▶ 349]	Returns the computed dissimilarities of the best matches (CTcVnContainer_Vector_LREAL; Optional, set to 0 if not required; sorted corresponding to ipMatchIndexes)
fDissimilarityThreshold	LREAL	Neglect irrelevant matches, i.e. dissimilarity > fDissimilarityThreshold
eComparisonMethod	ETcVnContoursMatchComparisonMethod [▶ 172]	Method used for comparing the Hu moment invariants of the contours
fAreaFactor	LREAL	If > 0, the relative area difference between contours is scaled by this factor and added to the computed dissimilarity
fAbsPositionFactor	LREAL	If > 0, the absolute position difference between contours (i.e. the coordinates of the geometric contour centers) is scaled by this factor and added to the computed dissimilarity
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

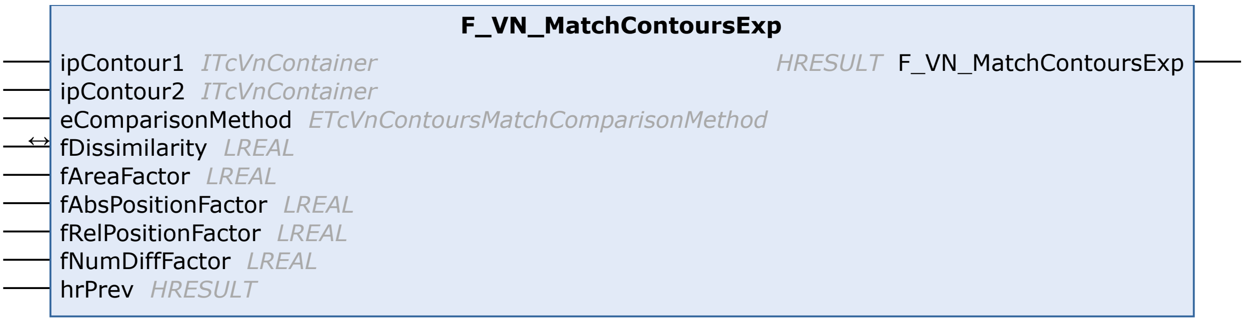
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.31 F_VN_MatchContoursExp



Compare contours using the Hu moment invariants (and optionally further aspects). In case of multiple contours in each container, the best matches are found and the average dissimilarity over all matched contours is returned. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_MatchContoursExp : HRESULT
VAR_INPUT
    ipContour1          : ITcVnContainer;
    ipContour2          : ITcVnContainer;
    eComparisonMethod   : ETcVnContoursMatchComparisonMethod;
END_VAR
VAR_IN_OUT
    fDissimilarity      : LREAL;
END_VAR
VAR_INPUT
    fAreaFactor         : LREAL;
    fAbsPositionFactor  : LREAL;
    fRelPositionFactor  : LREAL;
    fNumDiffFactor      : LREAL;
    hrPrev              : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContour1	ITcVnContainer [▸ 349]	First contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour) or collection of multiple contours (ContainerType_Vector_Vector_TcVnPoint2_DINT)
ipContour2	ITcVnContainer [▸ 349]	Second contour (same type as ipContour1)
eComparisonMethod	ETcVnContoursMatchComparisonMethod [▸ 172]	Method used for comparing the Hu moment invariants of the contours
fAreaFactor	LREAL	If > 0, the relative area difference between contours is scaled by this factor and added to the computed dissimilarity
fAbsPositionFactor	LREAL	If > 0, the absolute position difference between contours (i.e. the coordinates of the geometric contour centers) is scaled by this factor and added to the computed dissimilarity
fRelPositionFactor	LREAL	If > 0, the relative position difference between contours is scaled by this factor and added to the computed dissimilarity (only used if matching multiple contours)
fNumDiffFactor	LREAL	If > 0, the difference between the number of contours in both containers is scaled by this factor and added to the computed dissimilarity (only used if matching multiple contours)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fDissimilarity	LREAL	Returns the dissimilarity of the contours depending on the chosen comparison method

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.9.32 F_VN_UprightBoundingRectangle

Determines the upright bounding rectangle of a set of points.

Syntax

Definition:

```
FUNCTION F_VN_UprightBoundingRectangle : HRESULT
VAR_INPUT
    ipPointSet   : ITcVnContainer;
END_VAR
VAR_IN_OUT
    stRectangle : TcVnRectangle_UDINT;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipPointSet	ITcVnContainer [▶ 349]	Source point set (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the individual points.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stRectangle	TcVnRectangle_UDINT [▶ 225]	Returns the determined rectangle

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.10 Control Functions

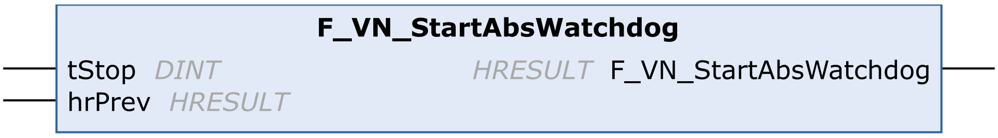
该组包含用于顺序控制的函数，例如用于处理看门狗 [▶ 127]。

函数

看门狗

- F_VN_StartAbsWatchdog(Exp) [▶ 949]
- F_VN_StartRelWatchdog(Exp) [▶ 950]
- F_VN_StopWatchdog [▶ 951]

6.1.4.10.1 F_VN_StartAbsWatchdog



Starts a cooperative watchdog given an absolute stop time.

Syntax

Definition:

```
FUNCTION F_VN_StartAbsWatchdog : HRESULT
VAR_INPUT
    tStop  : DINT;
    hrPrev : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
tStop	DINT	Stop time in us
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

关于更多信息，可查看看门狗 [▶ 127]一章。

超过 tStop 时间

请注意，当看门狗被触发时，tStop 时间会稍微超过（几微秒）。

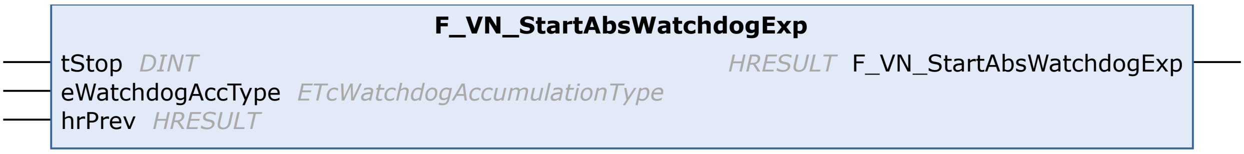
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.10.2 F_VN_StartAbsWatchdogExp



Start a cooperative watchdog given an absolute stop time. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_StartAbsWatchdogExp : HRESULT
VAR_INPUT
    tStop            : DINT;
    eWatchdogAccType : ETcWatchdogAccumulationType;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
tStop	DINT	Stop time in us
eWatchdogAccType	ETcWatchdogAccumulationType [▸ 203]	Accumalation method used for combining the results of multiple functions enclosed by this watchdog
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

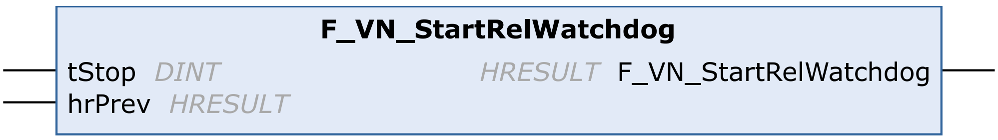
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.10.3 F_VN_StartRelWatchdog



Starts a cooperative watchdog given a stop time relative to the current time.

Syntax

Definition:

```
FUNCTION F_VN_StartRelWatchdog : HRESULT
VAR_INPUT
    tStop  : DINT;
    hrPrev : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
tStop	DINT	Stop time in us
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

关于更多信息，可查看看门狗 [▶ 127]一章。

超过 tStop 时间

请注意，当看门狗被触发时，tStop 时间会稍微超过（几微秒）。

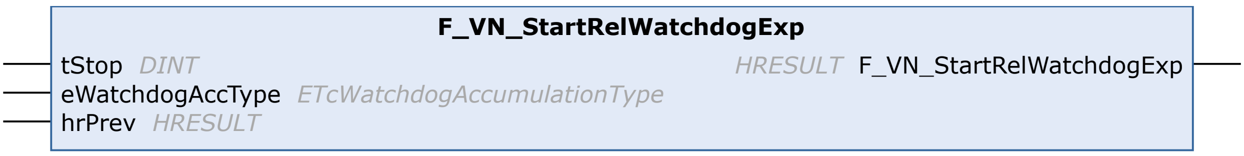
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.10.4 F_VN_StartRelWatchdogExp



Start a cooperative watchdog given a stop time relative to the current time. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_StartRelWatchdogExp : HRESULT
VAR_INPUT
    tStop            : DINT;
    eWatchdogAccType : ETcWatchdogAccumulationType;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
tStop	DINT	Stop time in us
eWatchdogAccType	ETcWatchdogAccumulationType [▶ 203]	Accumalation method used for combining the results of multiple functions enclosed by this watchdog
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.10.5 F_VN_StopWatchdog

Stops a watchdog and provide runtime information.

Syntax

Definition:

```
FUNCTION F_VN_StopWatchdog : HRESULT
VAR_INPUT
    hrStartWatchdog      : HRESULT;
END_VAR
VAR_OUTPUT
    nFunctionsMonitored : ULINT;
    nFractionProcessed  : UDINT;
    tRest               : DINT;
END_VAR
```

Inputs

Name	Type	Description
hrStartWatchdog	HRESULT [▶ 122]	HRESULT indicating the result of the function used to start the watchdog (If SUCCEEDED(hrStartWatchdog) equals false, no operation is executed.)

Outputs

Name	Type	Description
nFunctionsMonitored	ULINT	Returns the number of functions monitored
nFractionProcessed	UDINT	Returns the fraction processed accumulated over the monitored functions in percent
tRest	DINT	Returns the remaining computation time in us (may be negative)

Return value

HRESULT [▸ 122]

更多信息

输出 nFractionProcessed 必须根据看门狗的参数化进行不同的解释。这在启动看门狗时通过 ETcWatchdogAccumulationType [▸ 203]设置。

关于更多信息，可查看看门狗 [▸ 127]。

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11 Drawing

该组包含用于绘制和描述图像的功能。

处理 4 个通道的颜色

4 通道的颜色被直接接受到结果图像中。没有阿尔法混合或类似的情况发生。

函数

箭头

- F_VN_DrawArrow(Exp) [▸ 954]
- F_VN_DrawArrow(Exp)_TcVnVector4_DINT [▸ 955]

圆圈和弧形

- F_VN_DrawCircle (Exp) [▸ 959]
- F_VN_DrawCircles (Exp) [▸ 961]F_VN_DrawCircles(Exp)
- F_VN_DrawCircularArc (Exp) [▸ 963]
- F_VN_FillCircle [▸ 1001]

组件

- F_VN_DrawComponents(Exp) [▸ 965]

轮廓

- F_VN_DrawContours (Exp) [▸ 967]
- F_VN_FillContours [▸ 1002]

椭圆

- F_VN_DrawEllipse（Exp） [▶ 971]
- F_VN_FillEllipse [▶ 1003]

关键点和匹配

- F_VN_DrawKeypoints（Exp） [▶ 973]
- F_VN_DrawMatches（Exp） [▶ 985]

线

- F_VN_DrawLine（Exp） [▶ 975]
- F_VN_DrawLines（Exp） [▶ 982]
- F_VN_DrawLine(Exp)_TcVnVector4_DINT [▶ 976]
- F_VN_DrawLine(Exp)_TcVnVector4_LREAL [▶ 977]

定向

- F_VN_DrawOrientation(Exp) [▶ 987]

点

- F_VN_DrawPoint（Exp） [▶ 990]
- F_VN_DrawPoints（Exp） [▶ 992]

多边形

- F_VN_DrawPolygon(Exp) [▶ 994]
- F_VN_FillPolygon [▶ 1004]

矩形

- F_VN_DrawRectangle [▶ 996]
- F_VN_DrawRectangle_TcVnRectangle_DINT [▶ 997]
- F_VN_DrawRectangle_TcVnRectangle_UDINT [▶ 998]
- F_VN_DrawRotatedRectangle（Exp） [▶ 999]
- F_VN_FillRectangle [▶ 1005]
- F_VN_FillRotatedRectangle [▶ 1006]

强度分布

- F_VN_PlotIntensityProfile(Exp) [▶ 1007]

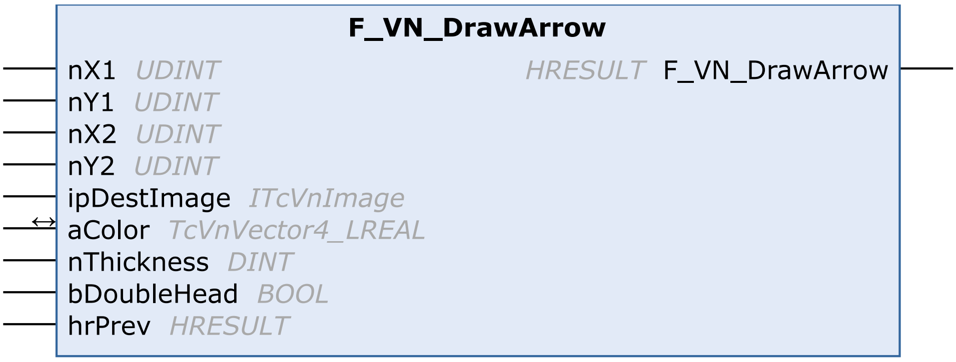
文本

- F_VN_PutLabel（Exp） [▶ 1010]
- F_VN_PutText（Exp） [▶ 1012]

绘图和填充

对于许多形状，可以选择同时绘制形状轮廓和填充形状。对此，可使用两个函数：（分别为 F_VN_Draw 和 F_VN_Fill）。此外，将轮廓厚度 nThickness 设置为 -1 会导致形状被填充。

6.1.4.11.1 F_VN_DrawArrow



Draw an arrowed line.

Syntax

Definition:

```
FUNCTION F_VN_DrawArrow : HRESULT
VAR_INPUT
    nX1         : UDINT;
    nY1         : UDINT;
    nX2         : UDINT;
    nY2         : UDINT;
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness  : DINT;
    bDoubleHead : BOOL;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nX1	UDINT	x coordinate of the start point
nY1	UDINT	y coordinate of the start point
nX2	UDINT	x coordinate of the end point
nY2	UDINT	y coordinate of the end point
ipDestImage	ITcVnImage [▸ 390]	Destination image
nThickness	DINT	Line thickness
bDoubleHead	BOOL	To draw an arrow on each side of the line.
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▸ 141]	Color

Return value

HRESULT [▸ 122]

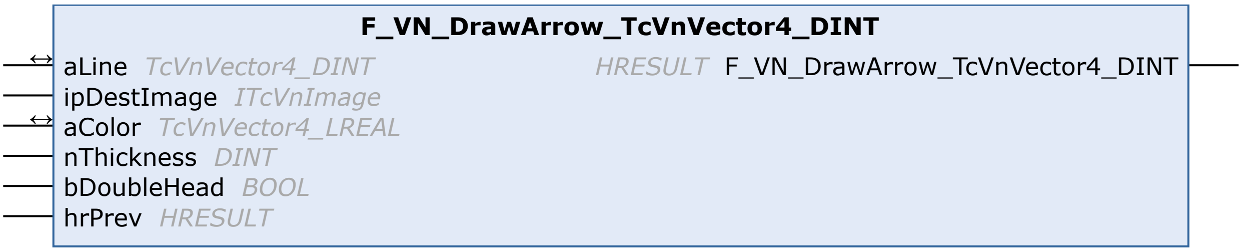
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.2 F_VN_DrawArrow_TcVnVector4_DINT



Draw an arrowed line.

Syntax

Definition:

```
FUNCTION F_VN_DrawArrow_TcVnVector4_DINT : HRESULT
VAR_IN_OUT
    aLine        : TcVnVector4_DINT;
END_VAR
VAR_INPUT
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness  : DINT;
    bDoubleHead : BOOL;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▸ 390]	Destination image
nThickness	DINT	Line thickness
bDoubleHead	BOOL	To draw an arrow on each side of the line.
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aLine	TcVnVector4_DINT [▸ 141]	The start and end point of a line segment [x1, y1, x2, y2]
aColor	TcVnVector4_LREAL [▸ 141]	Color

Return value

HRESULT [▶ 122]

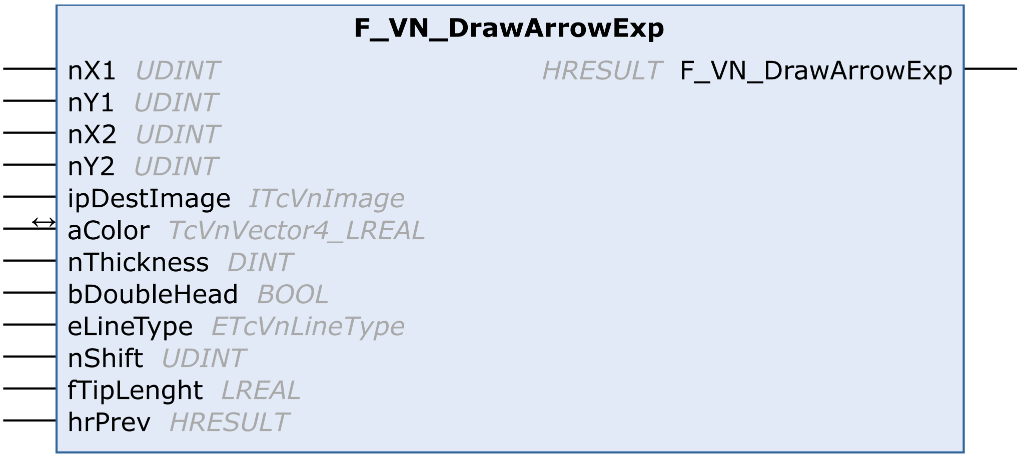
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.3 F_VN_DrawArrowExp



Draws an arrowed line. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_DrawArrowExp : HRESULT
VAR_INPUT
    nX1         : UDINT;
    nY1         : UDINT;
    nX2         : UDINT;
    nY2         : UDINT;
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness  : DINT;
    bDoubleHead : BOOL;
    eLineType   : ETcVnLineType;
    nShift      : UDINT;
    fTipLenght  : LREAL;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nX1	UDINT	x coordinate of the start point
nY1	UDINT	y coordinate of the start point
nX2	UDINT	x coordinate of the end point
nY2	UDINT	y coordinate of the end point
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness
bDoubleHead	BOOL	To draw an arrow on each side of the line.
eLineType	ETcVnLineType [▶ 187]	Line type
nShift	UDINT	Fractional bits of the coordinates (bit shift)
fTipLenght	LREAL	Lenght of the arrow's tip as a fraction of the line's lenght
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

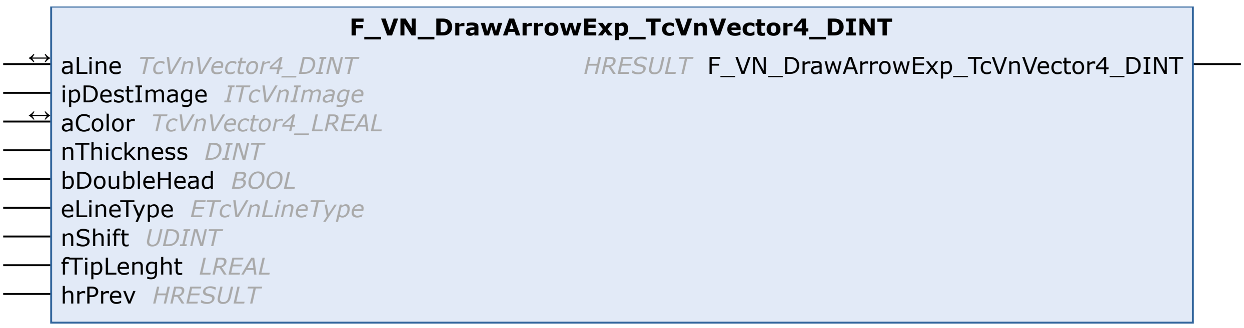
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.4 F_VN_DrawArrowExp_TcVnVector4_DINT



Draws an arrowed line. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_DrawArrowExp_TcVnVector4_DINT : HRESULT
VAR_IN_OUT
    aLine       : TcVnVector4_DINT;
END_VAR
VAR_INPUT
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness  : DINT;
    bDoubleHead : BOOL;
    eLineType   : ETcVnLineType;
    nShift      : UDINT;
    fTipLenght  : LREAL;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▸ 390]	Destination image
nThickness	DINT	Line thickness
bDoubleHead	BOOL	To draw an arrow on each side of the line.
eLineType	ETcVnLineType [▸ 187]	Line type
nShift	UDINT	Fractional bits of the coordinates (bit shift)
fTipLenght	LREAL	Lenght of the arrow's tip as a fraction of the line's lenght
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aLine	TcVnVector4_DINT [▸ 141]	The start and end point of a line segment [x1, y1, x2, y2]
aColor	TcVnVector4_LREAL [▸ 141]	Color

Return value

HRESULT [▸ 122]

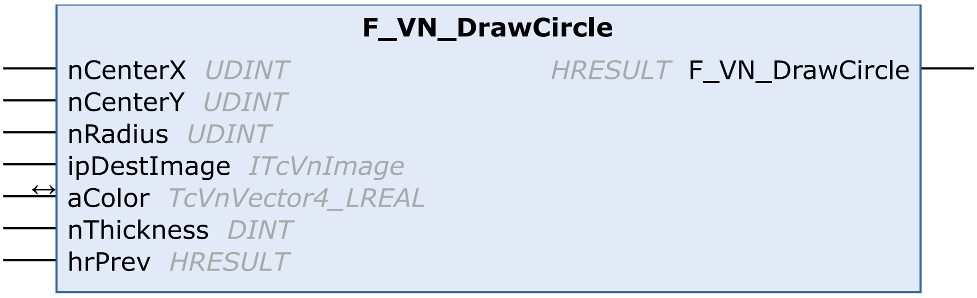
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.5 F_VN_DrawCircle



Draws a circle.

Syntax

Definition:

```
FUNCTION F_VN_DrawCircle : HRESULT
VAR_INPUT
    nCenterX    : UDINT;
    nCenterY    : UDINT;
    nRadius     : UDINT;
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness  : DINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nCenterX	UDINT	x coordinate of the center
nCenterY	UDINT	y coordinate of the center
nRadius	UDINT	Radius
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness (if negative, the circle is filled)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

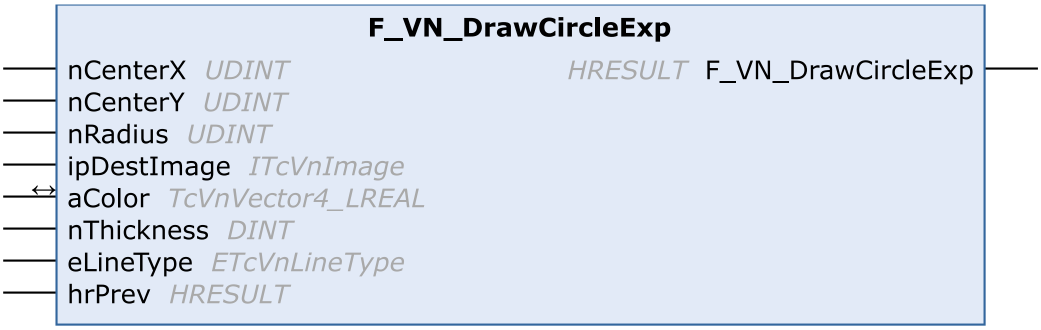
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.6 F_VN_DrawCircleExp



Draws a circle.

Syntax

Definition:

```
FUNCTION F_VN_DrawCircleExp : HRESULT
VAR_INPUT
    nCenterX    : UDINT;
    nCenterY    : UDINT;
    nRadius     : UDINT;
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness  : DINT;
    eLineType   : ETcVnLineType;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nCenterX	UDINT	x coordinate of the center
nCenterY	UDINT	y coordinate of the center
nRadius	UDINT	Radius
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness (if negative, the circle is filled)
eLineType	ETcVnLineType [▶ 187]	Line type
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

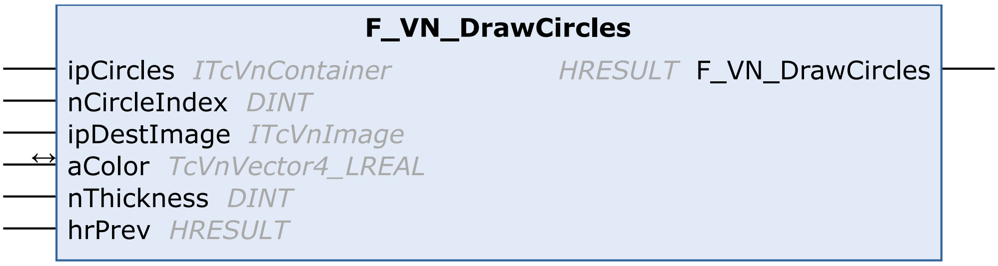
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.7 F_VN_DrawCircles



Draws circles.

Syntax

Definition:

```
FUNCTION F_VN_DrawCircles : HRESULT
VAR_INPUT
    ipCircles     : ITcVnContainer;
    nCircleIndex : DINT;
    ipDestImage  : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness   : DINT;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipCircles	ITcVnContainer [▶ 349]	Container with circles (ContainerType_Vector_TcVnVector3_REAL; Each container element contains the x coordinate of the circle center [0], the y coordinate of the circle center [1], and the radius [2].)
nCircleIndex	DINT	Index of a specific circle to be drawn (if negative, all circles within the container are drawn)
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness (if negative, the circle is filled)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

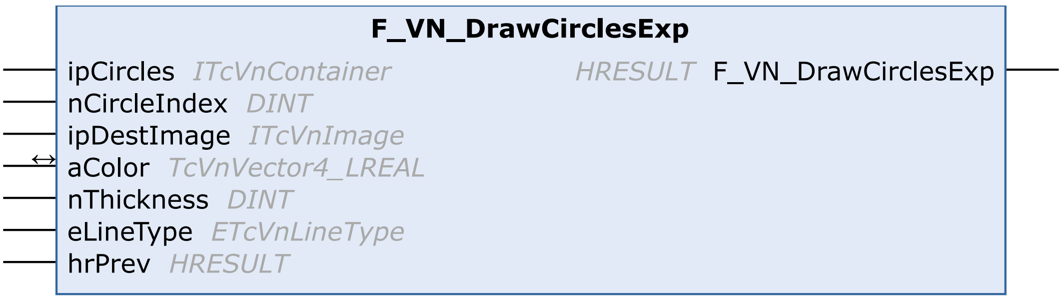
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.8 F_VN_DrawCirclesExp



Draws circles. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_DrawCirclesExp : HRESULT
VAR_INPUT
    ipCircles     : ITcVnContainer;
    nCircleIndex : DINT;
    ipDestImage  : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness   : DINT;
    eLineType    : ETcVnLineType;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipCircles	ITcVnContainer [▶ 349]	Container with circles (ContainerType_Vector_TcVnVector3_REAL; Each container element contains the x coordinate of the circle center [0], the y coordinate of the circle center [1], and the radius [2].)
nCircleIndex	DINT	Index of a specific circle to be drawn (if negative, all circles within the container are drawn)
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness (if negative, the circle is filled)
eLineType	ETcVnLineType [▶ 187]	Line type
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

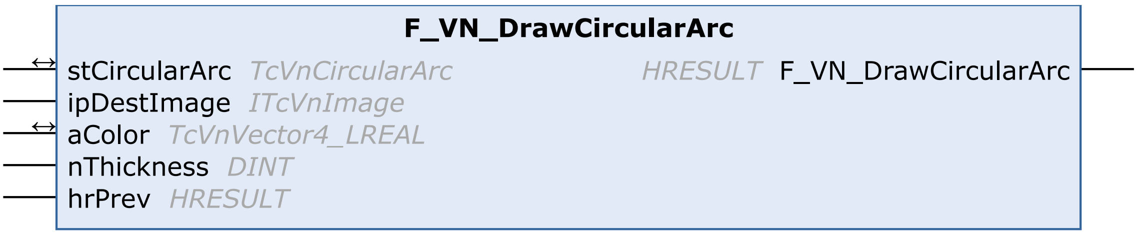
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.9 F_VN_DrawCircularArc



Draws a circular arc.

Syntax

Definition:

```
FUNCTION F_VN_DrawCircularArc : HRESULT
VAR_IN_OUT
    stCircularArc : TcVnCircularArc;
END_VAR
VAR_INPUT
    ipDestImage   : ITcVnImage;
END_VAR
```

```
VAR_IN_OUT
    aColor        : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness    : DINT;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness (if negative, the arc is filled)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stCircularArc	TcVnCircularArc [▶ 207]	Circular arc definition
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

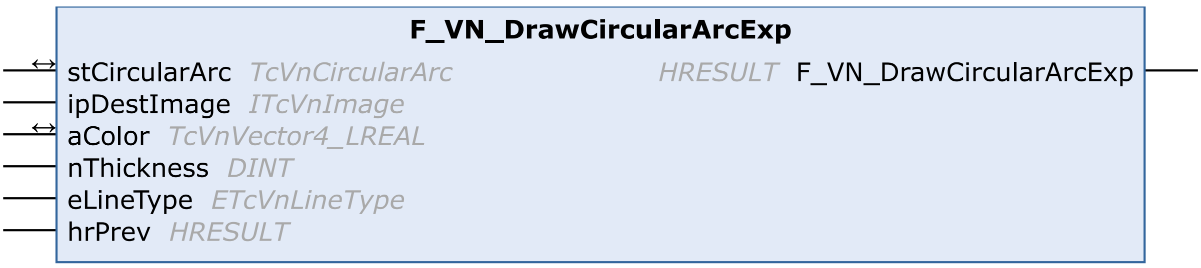
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.10 F_VN_DrawCircularArcExp



Draws a circular arc. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_DrawCircularArcExp : HRESULT
VAR_IN_OUT
    stCircularArc : TcVnCircularArc;
END_VAR
VAR_INPUT
```

```
    ipDestImage    : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor         : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness     : DINT;
    eLineType      : ETcVnLineType;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness (if negative, the arc is filled)
eLineType	ETcVnLineType [▶ 187]	Line type
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stCircularArc	TcVnCircularArc [▶ 207]	Circular arc definition
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

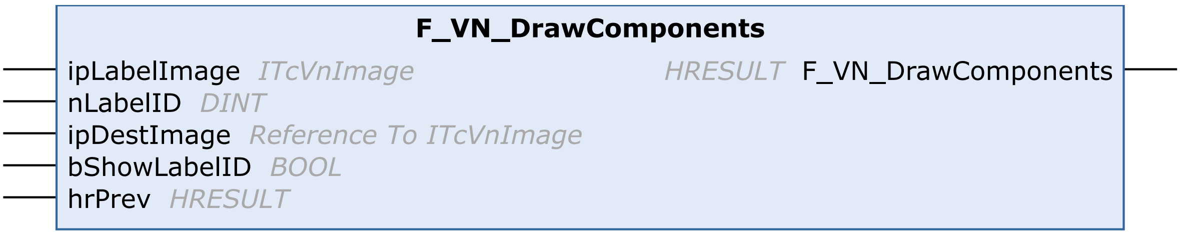
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.11 F_VN_DrawComponents



Draw the results of connected component function.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_DrawComponents : HRESULT
VAR_INPUT
    ipLabelImage : ITcVnImage;
    nLabelID     : DINT;
    ipDestImage  : Reference To ITcVnImage;
    bShowLabelID : BOOL;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipLabelImage	ITcVnImage [▸ 390]	Image contains labels for each pixel (1 channel DINT or UINT).
nLabelID	DINT	Value of a specific label to be drawn (if negative, all components within the image are drawn)
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate image will be created if required. The output image is 1 or 3 channels USINT).
bShowLabelID	BOOL	Show label ID in the center of the component on the destination image
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

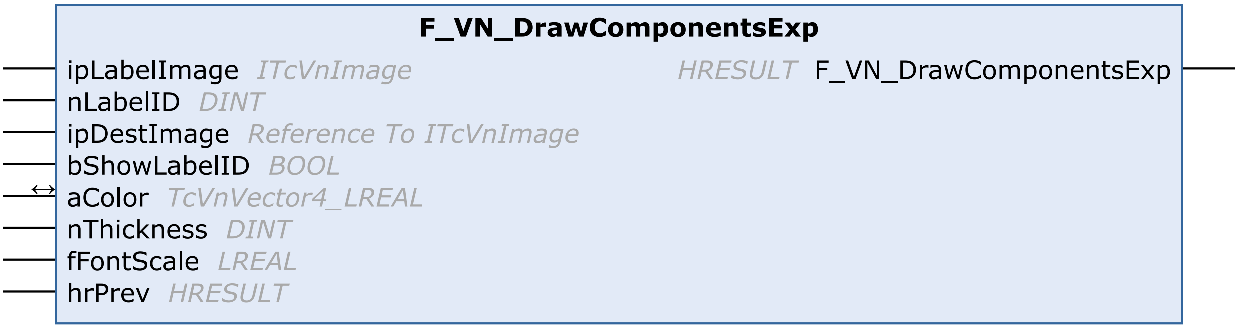
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.12 F_VN_DrawComponentsExp



Draw the results of connected component function. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_DrawComponentsExp : HRESULT
VAR_INPUT
    ipLabelImage : ITcVnImage;
    nLabelID     : DINT;
```

```
    ipDestImage   : Reference To ITcVnImage;
    bShowLabelID : BOOL;
END_VAR
VAR_IN_OUT
    aColor        : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness    : DINT;
    fFontScale    : LREAL;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipLabelImage	ITcVnImage [▶ 390]	Image contains labels for each pixel (1 channel DINT or UINT).
nLabelID	DINT	Value of a specific label to be drawn (if negative, all components within the image are drawn)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate image will be created if required. The output image is 1 or 3 channels USINT).
bShowLabelID	BOOL	Show lable ID in the center of the component on the destination image
nThickness	DINT	Text line thickness
fFontScale	LREAL	Scaling factor for the text
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▶ 141]	Color to draw the components (for {-1, -1, -1, -1}, a random color is chosen for each component)

Return value

HRESULT [▶ 122]

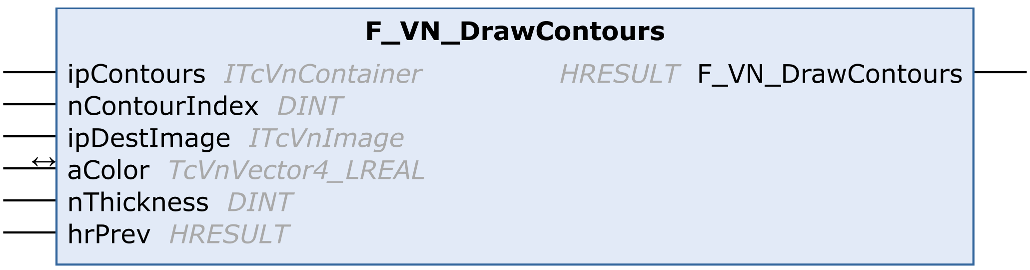
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.13 F_VN_DrawContours



Draws a single point set or multiple point sets that are interpreted as contours.

Syntax

Definition:

```
FUNCTION F_VN_DrawContours : HRESULT
VAR_INPUT
    ipContours    : ITcVnContainer;
    nContourIndex : DINT;
    ipDestImage   : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor        : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness    : DINT;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContours	ITcVnContainer [▸ 349]	Single contour (ContainerType_Vector_TcVnPoint2_DINT) or multiple contours (ContainerType_Vector_Vector_TcVnPoint2_DINT)
nContourIndex	DINT	Index of a specific contour to be drawn (if negative, all contours within the container are drawn)
ipDestImage	ITcVnImage [▸ 390]	Destination image
nThickness	DINT	Line thickness (if negative, the contours are filled)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▸ 141]	Color

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_DrawContours在目标图像中绘制轮廓。一个轮廓的描述基于存储在容器中的一组点。请注意，只能绘制整数的轮廓点，因此，容器必须是类型*ContainerType_Vector_TcVnPoint2_DINT*。如果使用浮点数描述点，F_VN_ConvertContainerType [▸ 721]可以用来转换点。

除了具有一个轮廓的容器外，具有几个轮廓的容器也可以转移到函数中。其类型为*ContainerType_Vector_Vector_TcVnPoint2_DINT*。输入变量*nContourIndex*表示要画哪条轮廓。如果需要绘制所有的轮廓，*nContourIndex*必须设置为负值。一次性绘制整个容器的轮廓（将*nContourIndex*设置为负值）比在一个循环中重复调用该函数一个接一个地绘制更有效率。

专家参数

专家版F_VN_DrawContoursExp [▸ 969]包含额外的参数。

应用

例如，在一个轮廓列表ipContours中使用厚度为 5px 的红线画出的所有轮廓如下所示：

```
hr := F_VN_DrawContours(
    ipContours      :=  ipContours,
    nContourIndex   :=  -1,
    ipDestImage     :=  ipImageRes,
    aColor          :=  aColorRed,
    nThickness      :=  5,
    hrPrev          :=  hr
);
```

返回 ContainerType_Vector_Vector_TcVnPoint2_DINT 类型轮廓的典型函数

- F_VN_DetectBlobs [▶ 1140]
- F_VN_FindContours [▶ 1147]或F_VN_FindContoursExp [▶ 1150]
- F_VN_FindContourHierarchyExp [▶ 1144]

样本

- 带看门狗监控的斑点检测 [▶ 2664]

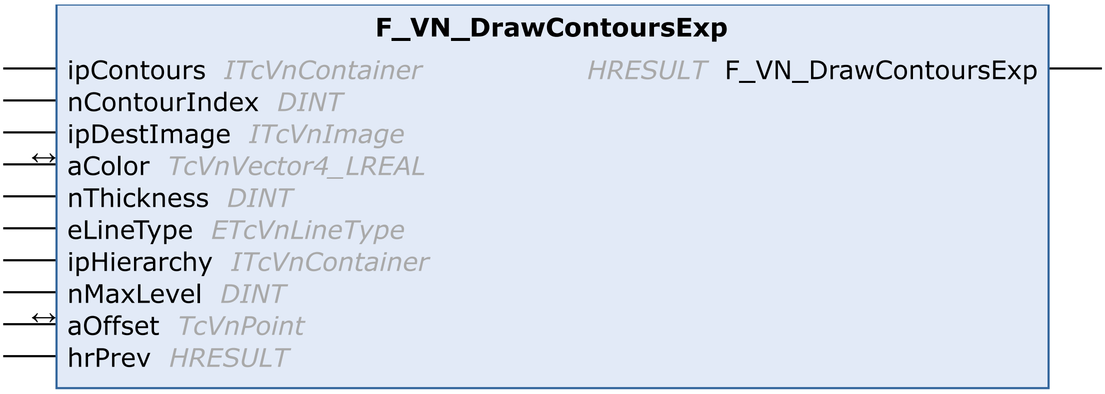
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.14 F_VN_DrawContoursExp



Draws a single point set or multiple point sets that are interpreted as contours. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_DrawContoursExp : HRESULT
VAR_INPUT
    ipContours    : ITcVnContainer;
    nContourIndex : DINT;
    ipDestImage   : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor        : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness    : DINT;
    eLineType     : ETcVnLineType;
    ipHierarchy   : ITcVnContainer;
    nMaxLevel     : DINT;
END_VAR
VAR_IN_OUT
    aOffset       : TcVnPoint;
```

```
END_VAR
VAR_INPUT
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContours	ITcVnContainer [▶ 349]	Single contour (ContainerType_Vector_TcVnPoint2_DINT) or multiple contours (ContainerType_Vector_Vector_TcVnPoint2_DINT)
nContourIndex	DINT	Index of a specific contour to be drawn (if negative, all contours within the container are drawn)
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness (if negative, the contours are filled)
eLineType	ETcVnLineType [▶ 187]	Line type
ipHierarchy	ITcVnContainer [▶ 349]	Contour hierarchy (ContainerType_Vector_TcVnVector4_DINT)
nMaxLevel	DINT	Maximum level of contours to be drawn
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▶ 141]	Color
aOffset	TcVnPoint [▶ 139]	Offset by which every contour point is shifted

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_DrawContoursExp 对应于函数F_VN_DrawContours [▶ 967]，由额外的输入变量扩展。

因此，除了找到的轮廓外，还可以考虑函数F_VN_FindContourHierarchyExp [▶ 1143]返回的层次结构描述。绘制轮廓的层次结构通过ipHierarchy和输入变量nMaxLevel指定。如果 0 被转移到ipHierarchy，那么层次结构就被忽略了。

aOffset可以用来绘制所有在 x 和 y 方向上有统一偏移的轮廓点。这一点很有意义，例如，如果轮廓是在一个 ROI 中搜索的，但要画到原始图像中。

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.15 F_VN_DrawEllipse

Draws an ellipse.

Syntax

Definition:

```
FUNCTION F_VN_DrawEllipse : HRESULT
VAR_IN_OUT
    stEllipse   : TcVnRotatedRectangle;
END_VAR
VAR_INPUT
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness  : DINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▸ 390]	Destination image
nThickness	DINT	Line thickness (if negative, the ellipse is filled)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stEllipse	TcVnRotatedRectangle [▸ 225]	Ellipse to be drawn (rotation angle in degrees)
aColor	TcVnVector4_LREAL [▸ 141]	Color

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.16 F_VN_DrawEllipseExp

Draws an ellipse. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_DrawEllipseExp : HRESULT
VAR_IN_OUT
    stEllipse   : TcVnRotatedRectangle;
END_VAR
VAR_INPUT
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness  : DINT;
    eLineType   : ETcVnLineType;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▸ 390]	Destination image
nThickness	DINT	Line thickness (if negative, the ellipse is filled)
eLineType	ETcVnLineType [▸ 187]	Line type
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stEllipse	TcVnRotatedRectangle [▸ 225]	Ellipse to be drawn (rotation angle in degrees)
aColor	TcVnVector4_LREAL [▸ 141]	Color

Return value

HRESULT [▸ 122]

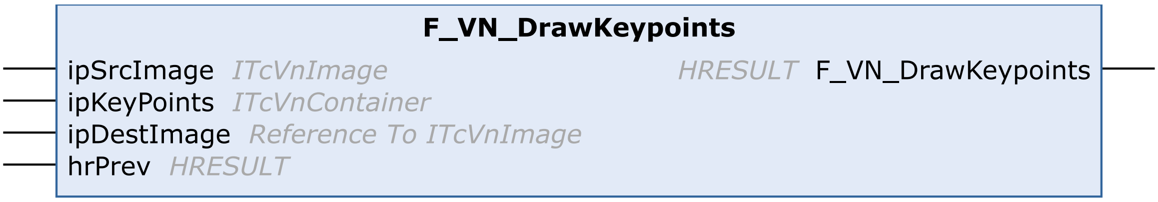
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.17 F_VN_DrawKeypoints



Draws the keypoints for visualization purpose.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_DrawKeypoints : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipKeyPoints : ITcVnContainer;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (elements of type USINT)
ipKeyPoints	ITcVnContainer [▶ 349]	Container with the keypoints (ContainerType_Vector_TcVnKeyPoint)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate color image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

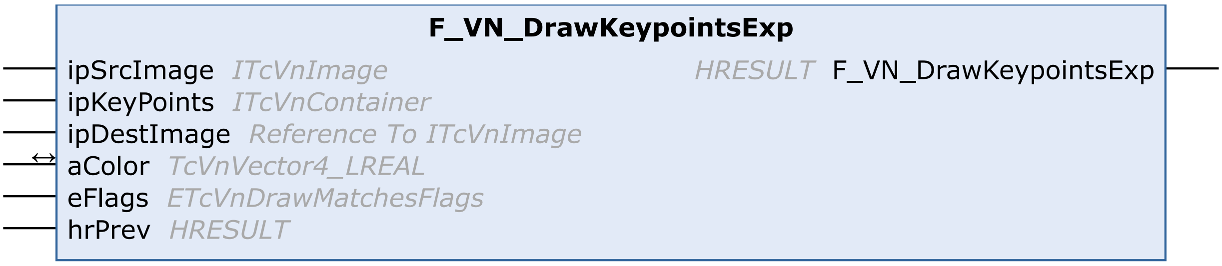
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.18 F_VN_DrawKeypointsExp



Draws the keypoints for visualization purpose. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_DrawKeypointsExp : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipKeyPoints : ITcVnContainer;
    ipDestImage : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    eFlags      : ETcVnDrawMatchesFlags;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (elements of type USINT)
ipKeyPoints	ITcVnContainer [▶ 349]	Container with the keypoints (ContainerType_Vector_TcVnKeyPoint)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate color image will be created if required.)
eFlags	ETcVnDrawMatchesFlags [▶ 175]	A combination of flags to support overdrawing an existing destination image and/or drawing additional (Rich-)Keypoint information (size and orientation)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▶ 141]	Color to draw the keypoints (for {-1, -1, -1, -1}, a random color is chosen for each point)

Return value

HRESULT [▶ 122]

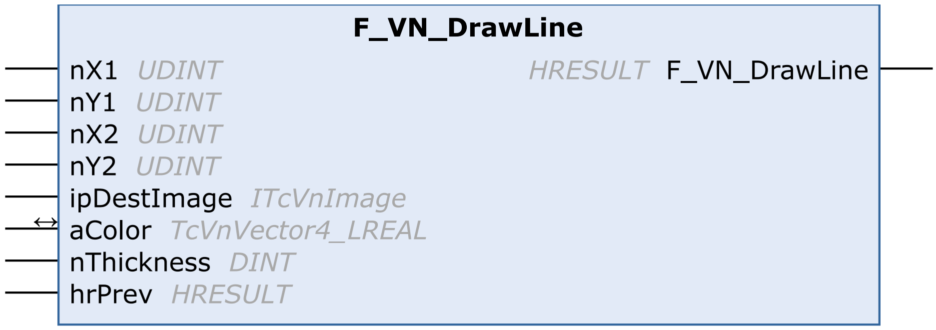
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.19 F_VN_DrawLine



Draws a line.

Syntax

Definition:

```
FUNCTION F_VN_DrawLine : HRESULT
VAR_INPUT
    nX1          : UDINT;
    nY1          : UDINT;
    nX2          : UDINT;
    nY2          : UDINT;
    ipDestImage  : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness   : DINT;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nX1	UDINT	x coordinate of the start point
nY1	UDINT	y coordinate of the start point
nX2	UDINT	x coordinate of the end point
nY2	UDINT	y coordinate of the end point
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

HRESULT

16#734	OUTOFRANGE	部分线条位于图像区域之外。使用函数F_VN_ClipLineToBoundary_ITcVnImage [▶ 1476]以确保指定的线适合在图像区域内。

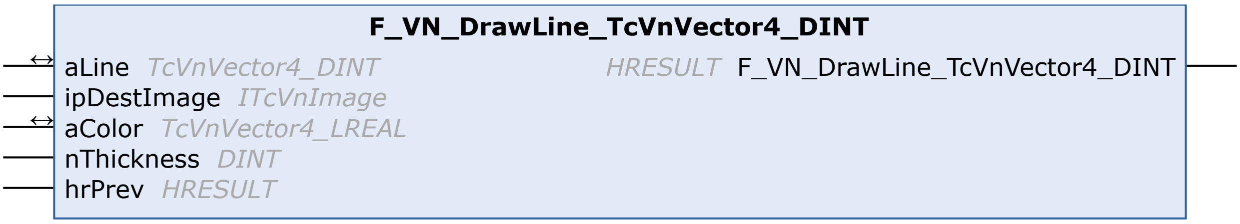
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.20 F_VN_DrawLine_TcVnVector4_DINT



Draws a line.

Syntax

Definition:

```
FUNCTION F_VN_DrawLine_TcVnVector4_DINT : HRESULT
VAR_IN_OUT
    aLine        : TcVnVector4_DINT;
END_VAR
VAR_INPUT
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness  : DINT;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aLine	TcVnVector4_DINT [▶ 141]	The start and end point of a line segment [x1, y1, x2, y2]
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

HRESULT

16#734	OUTOFRANGE	部分线条位于图像区域之外。使用函数F_VN_ClipLineToBoundary_ITcVnImage [▶ 1476]以确保指定的线适合在图像区域内。

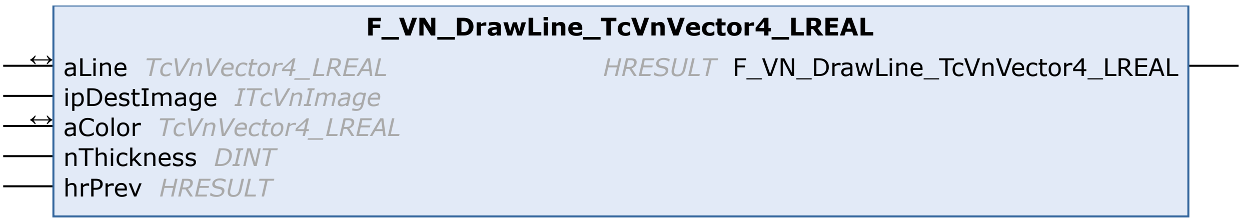
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.21 F_VN_DrawLine_TcVnVector4_LREAL



Draws a line.

Syntax

Definition:

```
FUNCTION F_VN_DrawLine_TcVnVector4_LREAL : HRESULT
VAR_IN_OUT
    aLine       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness  : DINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aLine	TcVnVector4_LREAL [▶ 141]	The first and second element describe the x and y component of the direction vector. The third and fourth element describe the x and y component of the position vector.
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

注意

浮点异常

如果正在执行的PLC 任务 [▶ 55]的选项**浮点异常**启用，该功能会不必要地引起错误。因此，停用这个选项。

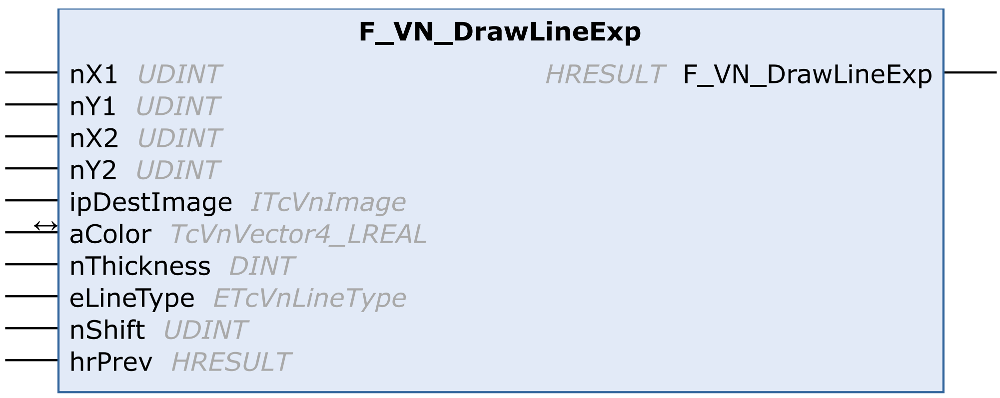
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.22 F_VN_DrawLineExp



Draws a line. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_DrawLineExp : HRESULT
VAR_INPUT
    nX1         : UDINT;
    nY1         : UDINT;
    nX2         : UDINT;
    nY2         : UDINT;
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness  : DINT;
    eLineType   : ETcVnLineType;
    nShift      : UDINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nX1	UDINT	x coordinate of the start point
nY1	UDINT	y coordinate of the start point
nX2	UDINT	x coordinate of the end point
nY2	UDINT	y coordinate of the end point
ipDestImage	ITcVnImage [▸ 390]	Destination image
nThickness	DINT	Line thickness
eLineType	ETcVnLineType [▸ 187]	Line type
nShift	UDINT	Fractional bits of the coordinates (bit shift)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▸ 141]	Color

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.23 F_VN_DrawLineExp_TcVnVector4_DINT

F_VN_DrawLineExp_TcVnVector4_DINT	
aLine *TcVnVector4_DINT*	*HRESULT* F_VN_DrawLineExp_TcVnVector4_DINT
ipDestImage *ITcVnImage*	
aColor *TcVnVector4_LREAL*	
nThickness *DINT*	
eLineType *ETcVnLineType*	
nShift *UDINT*	
hrPrev *HRESULT*	

Draws a line. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_DrawLineExp_TcVnVector4_DINT : HRESULT
VAR_IN_OUT
    aLine       : TcVnVector4_DINT;
END_VAR
VAR_INPUT
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness  : DINT;
    eLineType   : ETcVnLineType;
    nShift      : UDINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▸ 390]	Destination image
nThickness	DINT	Line thickness
eLineType	ETcVnLineType [▸ 187]	Line type
nShift	UDINT	Fractional bits of the coordinates (bit shift)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aLine	TcVnVector4_DINT [▸ 141]	The start and end point of a line segment [x1, y1, x2, y2]
aColor	TcVnVector4_LREAL [▸ 141]	Color

Return value

HRESULT [▸ 122]

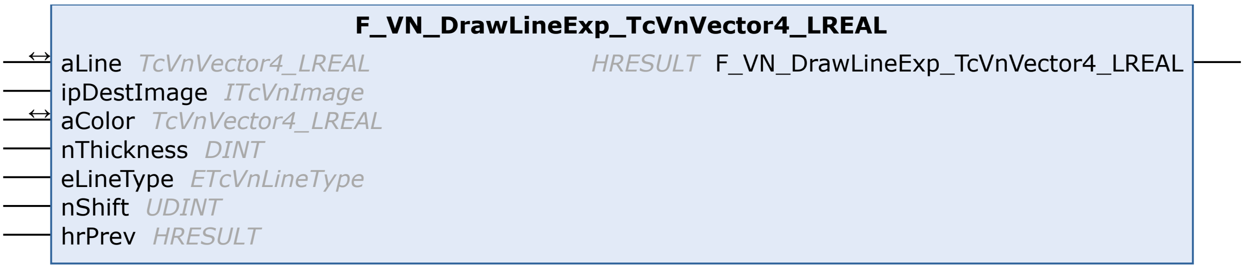
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.24 F_VN_DrawLineExp_TcVnVector4_LREAL



Draws a line. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_DrawLineExp_TcVnVector4_LREAL : HRESULT
VAR_IN_OUT
    aLine       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness  : DINT;
    eLineType   : ETcVnLineType;
    nShift      : UDINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness
eLineType	ETcVnLineType [▶ 187]	Line type
nShift	UDINT	Fractional bits of the coordinates (bit shift)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aLine	TcVnVector4_LREAL [▶ 141]	The first and second element describe the x and y component of the direction vector. The third and fourth element describe the x and y component of the position vector.
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

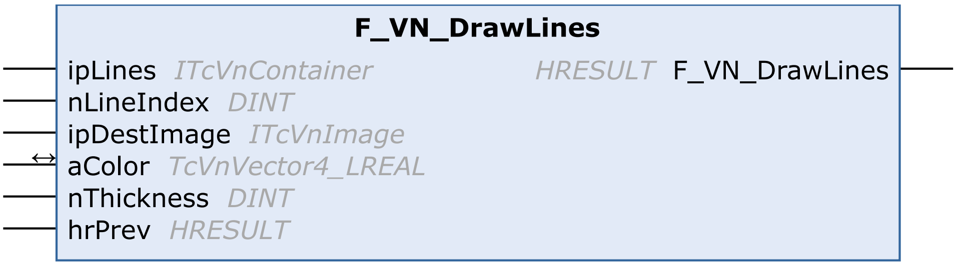
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.25 F_VN_DrawLines



Draws lines.

Syntax

Definition:

```
FUNCTION F_VN_DrawLines : HRESULT
VAR_INPUT
    ipLines      : ITcVnContainer;
    nLineIndex   : DINT;
    ipDestImage  : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness   : DINT;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipLines	ITcVnContainer [▶ 349]	Container with line descriptions (ContainerType_Vector_TcVnVector2_REAL: the distance from the origin [0] in pixels and the rotation angle [1] in radians. ContainerType_Vector_TcVnVector4_LREAL: the first and second element describe the x and y component of a vector collinear to the line. The third and fourth element describe the x and y component of a point on the line. ContainerType_Vector_TcVnVector4_DINT: start and end point [x1, y1, x2, y2])
nLineIndex	DINT	Index of a specific line to be drawn (if negative, all lines within the container are drawn)
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

HRESULT

16#734	OUTOFRANGE	部分线条位于图像区域之外。使用函数F_VN_ClipLineToBoundary_ITcVnImage [▶ 1476]以确保指定的线适合在图像区域内。

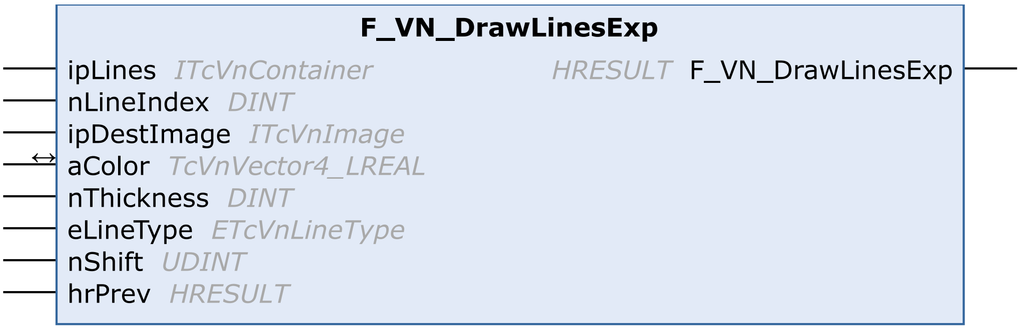
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.26 F_VN_DrawLinesExp



Draws lines. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_DrawLinesExp : HRESULT
VAR_INPUT
    ipLines      : ITcVnContainer;
    nLineIndex   : DINT;
    ipDestImage  : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness   : DINT;
    eLineType    : ETcVnLineType;
    nShift       : UDINT;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipLines	ITcVnContainer [▸ 349]	Container with line descriptions (ContainerType_Vector_TcVnVector2_REAL: the distance from the origin [0] in pixels and the rotation angle [1] in radians. ContainerType_Vector_TcVnVector4_LREAL: the first and second element describe the x and y component of a vector collinear to the line. The third and fourth element describe the x and y component of a point on the line. ContainerType_Vector_TcVnVector4_DINT: start and end point [x1, y1, x2, y2])
nLineIndex	DINT	Index of a specific line to be drawn (if negative, all lines within the container are drawn)
ipDestImage	ITcVnImage [▸ 390]	Destination image
nThickness	DINT	Line thickness
eLineType	ETcVnLineType [▸ 187]	Line type
nShift	UDINT	Fractional bits of the coordinates (bit shift)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▸ 141]	Color

Return value

HRESULT [▸ 122]

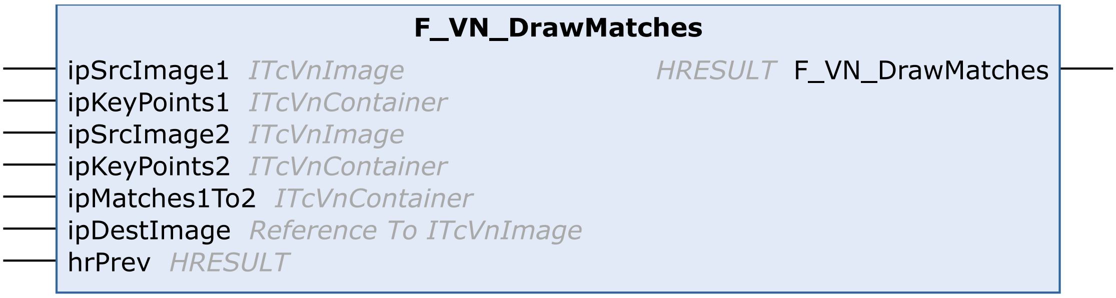
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.27 F_VN_DrawMatches



Draws the keypoints and matches for visualization purpose.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_DrawMatches : HRESULT
VAR_INPUT
    ipSrcImage1   : ITcVnImage;
    ipKeyPoints1  : ITcVnContainer;
    ipSrcImage2   : ITcVnImage;
    ipKeyPoints2  : ITcVnContainer;
    ipMatches1To2 : ITcVnContainer;
    ipDestImage   : Reference To ITcVnImage;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▸ 390]	Source image 1 (elements of type USINT)
ipKeyPoints1	ITcVnContainer [▸ 349]	Container with the keypoints 1 (ContainerType_Vector_TcVnKeyPoint)
ipSrcImage2	ITcVnImage [▸ 390]	Source image 2 (elements of type USINT, must have the same number of channels as ipSrcImage1)
ipKeyPoints2	ITcVnContainer [▸ 349]	Container with the keypoints 2 (ContainerType_Vector_TcVnKeyPoint)
ipMatches1To2	ITcVnContainer [▸ 349]	Container with the descriptor matches (ContainerType_Vector_TcVnDMatch)
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate color image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

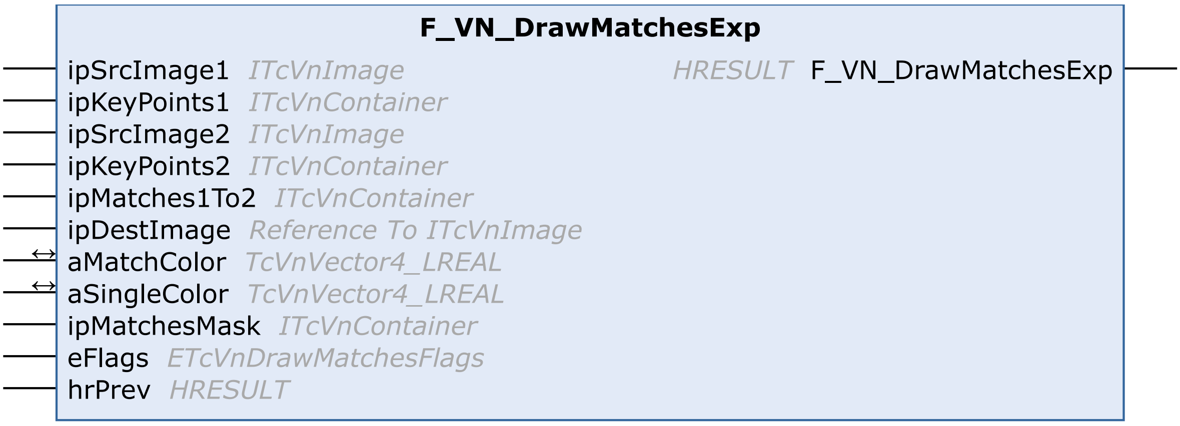
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.28 F_VN_DrawMatchesExp



Draws the keypoints and matches for visualization purpose. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_DrawMatchesExp : HRESULT
VAR_INPUT
    ipSrcImage1    : ITcVnImage;
    ipKeyPoints1   : ITcVnContainer;
    ipSrcImage2    : ITcVnImage;
    ipKeyPoints2   : ITcVnContainer;
    ipMatches1To2  : ITcVnContainer;
    ipDestImage    : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aMatchColor    : TcVnVector4_LREAL;
    aSingleColor   : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipMatchesMask : ITcVnContainer;
    eFlags         : ETcVnDrawMatchesFlags;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage1	ITcVnImage [▶ 390]	Source image 1 (elements of type USINT)
ipKeyPoints1	ITcVnContainer [▶ 349]	Container with the keypoints 1 (ContainerType_Vector_TcVnKeyPoint)
ipSrcImage2	ITcVnImage [▶ 390]	Source image 2 (elements of type USINT, must have the same number of channels as ipSrcImage1)
ipKeyPoints2	ITcVnContainer [▶ 349]	Container with the keypoints 2 (ContainerType_Vector_TcVnKeyPoint)
ipMatches1To2	ITcVnContainer [▶ 349]	Container with the descriptor matches (ContainerType_Vector_TcVnDMatch)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate color image will be created if required.)
ipMatchesMask	ITcVnContainer [▶ 349]	Mask to select the matches to be drawn (ContainerType_Vector_SINT; Set to 0 if all matches should be drawn.)
eFlags	ETcVnDrawMatchesFlags [▶ 175]	A combination of flags to support overdrawing an existing destination image and/or drawing additional (Rich-)Keypoint information (size and orientation) and/or skipping single keypoints
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMatchColor	TcVnVector4_LREAL [▶ 141]	Color to draw the matches (for {-1, -1, -1, -1}, a random color is chosen for each point and line)
aSingleColor	TcVnVector4_LREAL [▶ 141]	Color to draw the single keypoints (for {-1, -1, -1, -1}, a random color is chosen for each point)

Return value

HRESULT [▶ 122]

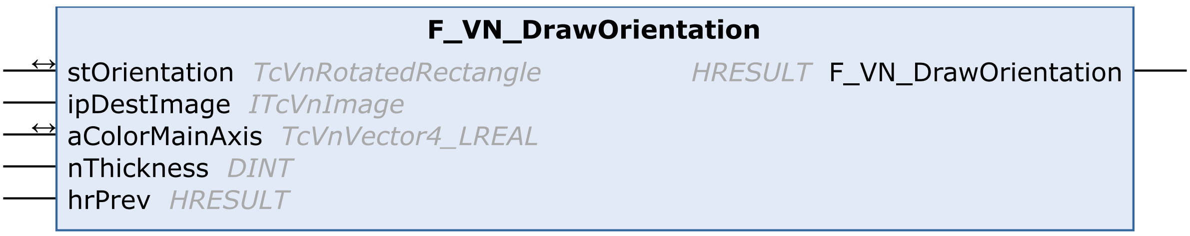
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.29 F_VN_DrawOrientation



Draw the main axis of a set of points/contour based on a rotated rectangle.

Syntax

Definition:

```
FUNCTION F_VN_DrawOrientation : HRESULT
VAR_IN_OUT
    stOrientation   : TcVnRotatedRectangle;
END_VAR
VAR_INPUT
    ipDestImage     : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColorMainAxis : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness      : DINT;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stOrientation	TcVnRotatedRectangle [▶ 225]	Rotated rectangle, containig center of mass, lenghts of axes, and rotation angle in clockwise direction
aColorMainAxis	TcVnVector4_LREAL [▶ 141]	Color of the major axis

Return value

HRESULT [▶ 122]

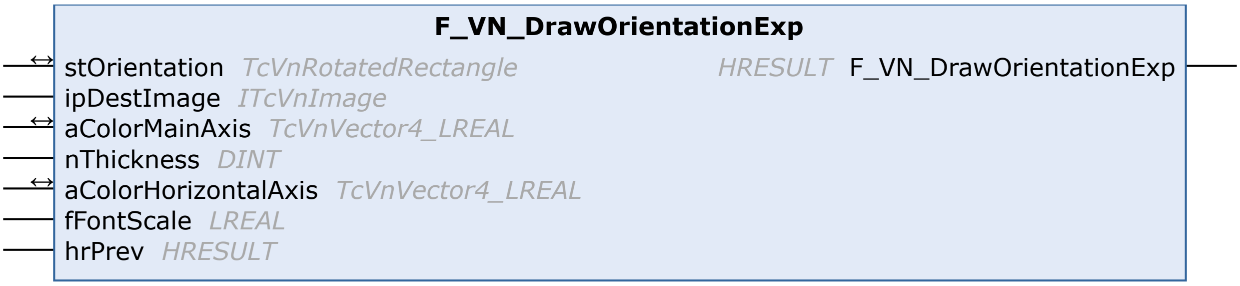
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.30 F_VN_DrawOrientationExp



Draw the main axis, the horizontal axis and the angle of a set of points/contour based on a rotated rectangle. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_DrawOrientationExp : HRESULT
VAR_IN_OUT
    stOrientation         : TcVnRotatedRectangle;
END_VAR
VAR_INPUT
    ipDestImage           : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColorMainAxis        : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness            : DINT;
END_VAR
VAR_IN_OUT
    aColorHorizontalAxis : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    fFontScale            : LREAL;
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▸ 390]	Destination image
nThickness	DINT	Line thickness
fFontScale	LREAL	Scaling factor for the text
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stOrientation	TcVnRotatedRectangle [▸ 225]	Rotated rectangle, containig center of mass, lenghts of axes, and rotation angle in clockwise direction
aColorMainAxis	TcVnVector4_LREAL [▸ 141]	Color of the major axis
aColorHorizontalAxis	TcVnVector4_LREAL [▸ 141]	Color of the horizontal axis

Return value

HRESULT [▸ 122]

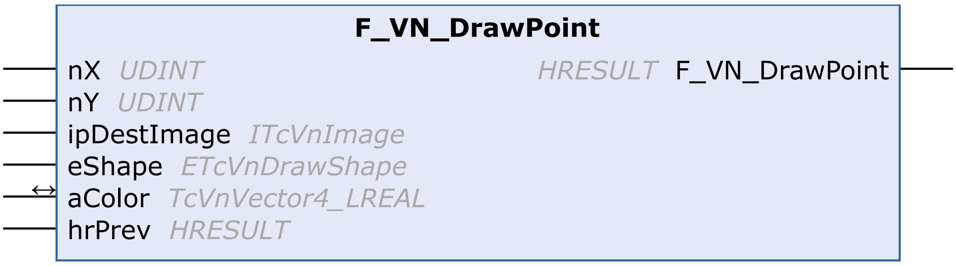
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.31 F_VN_DrawPoint



Draws a point.

Syntax

Definition:

```
FUNCTION F_VN_DrawPoint : HRESULT
VAR_INPUT
    nX          : UDINT;
    nY          : UDINT;
    ipDestImage : ITcVnImage;
    eShape      : ETcVnDrawShape;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nX	UDINT	x coordinate of the point
nY	UDINT	y coordinate of the point
ipDestImage	ITcVnImage [▸ 390]	Destination image
eShape	ETcVnDrawShape [▸ 176]	Point shape
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▸ 141]	Color (for {-1, -1, -1, -1}, a random color is chosen)

Return value

HRESULT [▸ 122]

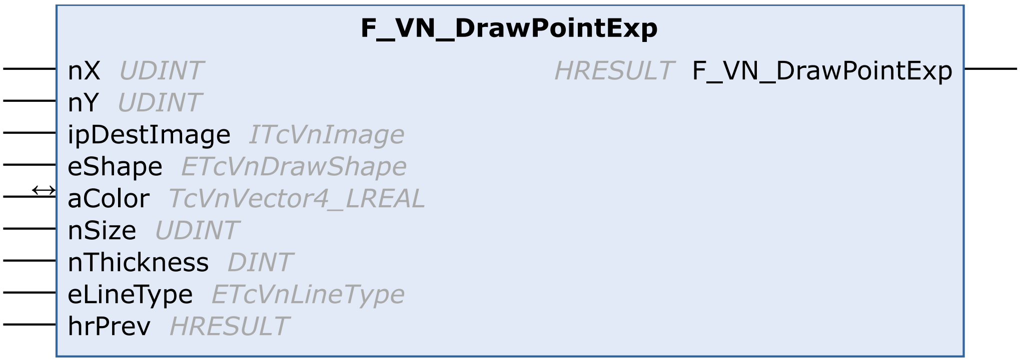
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.32 F_VN_DrawPointExp



Draw a point. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_DrawPointExp : HRESULT
VAR_INPUT
    nX          : UDINT;
    nY          : UDINT;
    ipDestImage : ITcVnImage;
    eShape      : ETcVnDrawShape;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nSize       : UDINT;
    nThickness  : DINT;
    eLineType   : ETcVnLineType;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nX	UDINT	x coordinate of the point
nY	UDINT	y coordinate of the point
ipDestImage	ITcVnImage [▶ 390]	Destination image
eShape	ETcVnDrawShape [▶ 176]	Point shape
nSize	UDINT	Size of the shape (half width)
nThickness	DINT	Line thickness (if negative, the shape is filled if it is closed)
eLineType	ETcVnLineType [▶ 187]	Line type
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▶ 141]	Color (for {-1, -1, -1, -1}, a random color is chosen)

Return value

HRESULT [▶ 122]

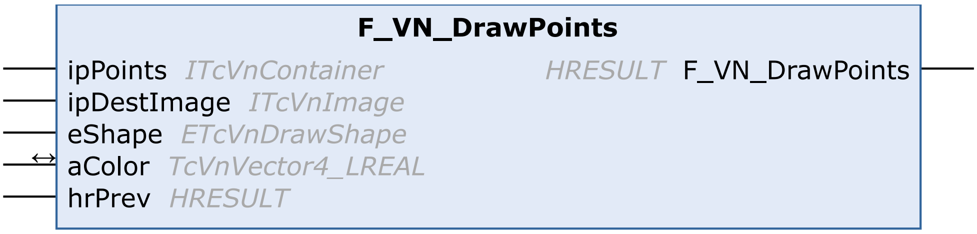
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.33 F_VN_DrawPoints



Draw a collection of points.

Syntax

Definition:

```
FUNCTION F_VN_DrawPoints : HRESULT
VAR_INPUT
    ipPoints     : ITcVnContainer;
    ipDestImage  : ITcVnImage;
    eShape       : ETcVnDrawShape;
END_VAR
VAR_IN_OUT
    aColor       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipPoints	ITcVnContainer [▶ 349]	Container with TcVnPoint2_DINT or TcVnPoint2_REAL elements
ipDestImage	ITcVnImage [▶ 390]	Destination image
eShape	ETcVnDrawShape [▶ 176]	Point shape
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▶ 141]	Color (for {-1, -1, -1, -1}, a random color is chosen for each point)

Return value

HRESULT [▶ 122]

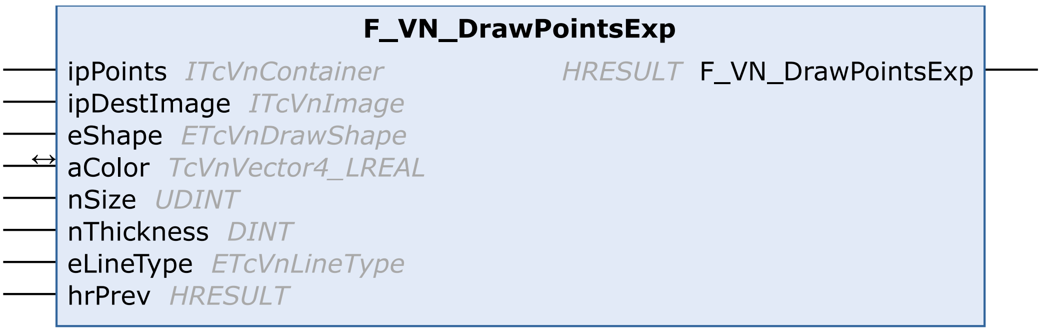
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.34 F_VN_DrawPointsExp



Draw a collection of points. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_DrawPointsExp : HRESULT
VAR_INPUT
    ipPoints     : ITcVnContainer;
    ipDestImage : ITcVnImage;
    eShape       : ETcVnDrawShape;
END_VAR
VAR_IN_OUT
    aColor       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nSize        : UDINT;
    nThickness   : DINT;
    eLineType    : ETcVnLineType;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipPoints	ITcVnContainer [▶ 349]	Container with TcVnPoint2_DINT or TcVnPoint2_REAL elements
ipDestImage	ITcVnImage [▶ 390]	Destination image
eShape	ETcVnDrawShape [▶ 176]	Point shape
nSize	UDINT	Size of the shape (half width)
nThickness	DINT	Line thickness (if negative, the shape is filled if it is closed)
eLineType	ETcVnLineType [▶ 187]	Line type
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▶ 141]	Color (for {-1, -1, -1, -1}, a random color is chosen for each point)

Return value

HRESULT [▶ 122]

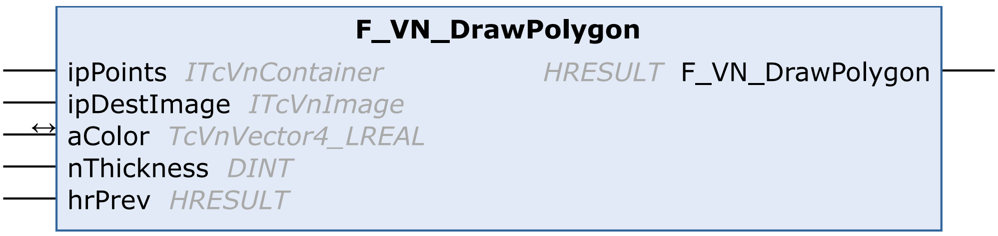
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.35 F_VN_DrawPolygon



Draw a pologon using collection of points.

Syntax

Definition:

```
FUNCTION F_VN_DrawPolygon : HRESULT
VAR_INPUT
    ipPoints    : ITcVnContainer;
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
```

```
    nThickness  : DINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipPoints	ITcVnContainer [▶ 349]	Container with TcVnPoint2_DINT, TcVnPoint2_REAL, or TcVnPoint2_LREAL elements
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness (if negative, the polygon is filled)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

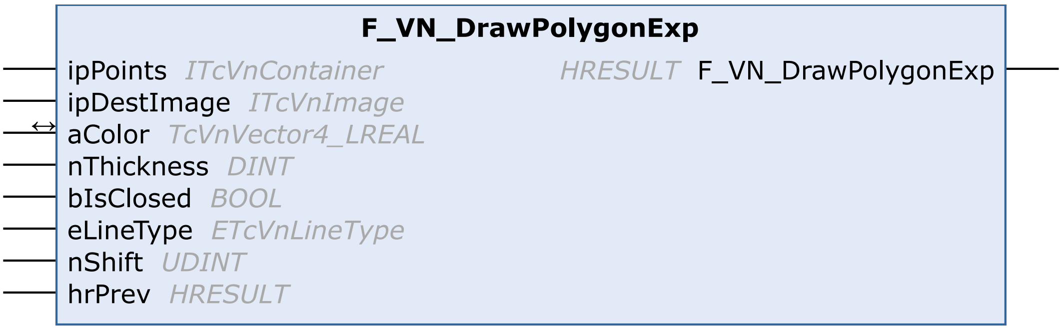
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.36 F_VN_DrawPolygonExp



Draw a pologon using collection of points. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_DrawPolygonExp : HRESULT
VAR_INPUT
    ipPoints    : ITcVnContainer;
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
```

```
END_VAR
VAR_INPUT
    nThickness  : DINT;
    bIsClosed   : BOOL;
    eLineType   : ETcVnLineType;
    nShift      : UDINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipPoints	ITcVnContainer [▶ 349]	Container with TcVnPoint2_DINT, TcVnPoint2_REAL, or TcVnPoint2_LREAL elements
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness (if negative, the shape is filled)
bIsClosed	BOOL	If it is true, the function draws a line from the last point to the first point to close the polygon.
eLineType	ETcVnLineType [▶ 187]	Line type
nShift	UDINT	Fractional bits of the coordinates (bit shift)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

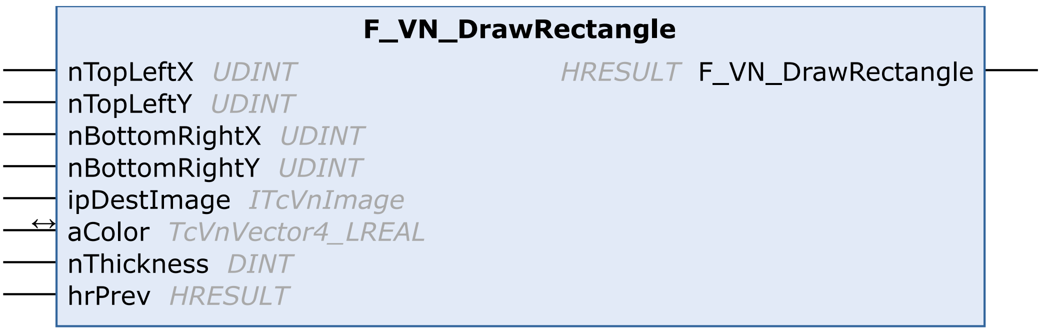
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.37 F_VN_DrawRectangle



Draw a rectangle.

Syntax

Definition:

```
FUNCTION F_VN_DrawRectangle : HRESULT
VAR_INPUT
    nTopLeftX      : UDINT;
    nTopLeftY      : UDINT;
    nBottomRightX  : UDINT;
    nBottomRightY  : UDINT;
    ipDestImage    : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor         : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness     : DINT;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nTopLeftX	UDINT	x coordinate of the top left corner
nTopLeftY	UDINT	y coordinate of the top left corner
nBottomRightX	UDINT	x coordinate of the bottom right corner
nBottomRightY	UDINT	y coordinate of the bottom right corner
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness (if negative, the rectangle is filled)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

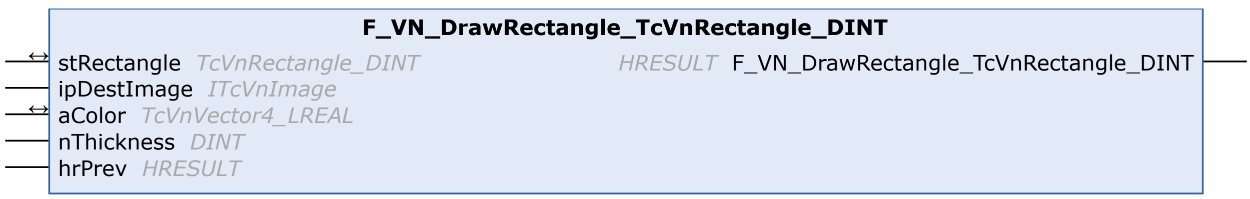
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.38 F_VN_DrawRectangle_TcVnRectangle_DINT



Draw a rectangle.

Syntax

Definition:

```
FUNCTION F_VN_DrawRectangle_TcVnRectangle_DINT : HRESULT
VAR_IN_OUT
    stRectangle : TcVnRectangle_DINT;
END_VAR
VAR_INPUT
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness  : DINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▸ 390]	Destination image
nThickness	DINT	Line thickness (if negative, the rectangle is filled)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stRectangle	TcVnRectangle_DINT [▸ 224]	Rectangle to be drawn
aColor	TcVnVector4_LREAL [▸ 141]	Color

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.39 F_VN_DrawRectangle_TcVnRectangle_UDINT

Draw a rectangle.

Syntax

Definition:

```
FUNCTION F_VN_DrawRectangle_TcVnRectangle_UDINT : HRESULT
VAR_IN_OUT
    stRectangle : TcVnRectangle_UDINT;
END_VAR
VAR_INPUT
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness  : DINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness (if negative, the rectangle is filled)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stRectangle	TcVnRectangle_UDINT [▶ 225]	Rectangle to be drawn
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

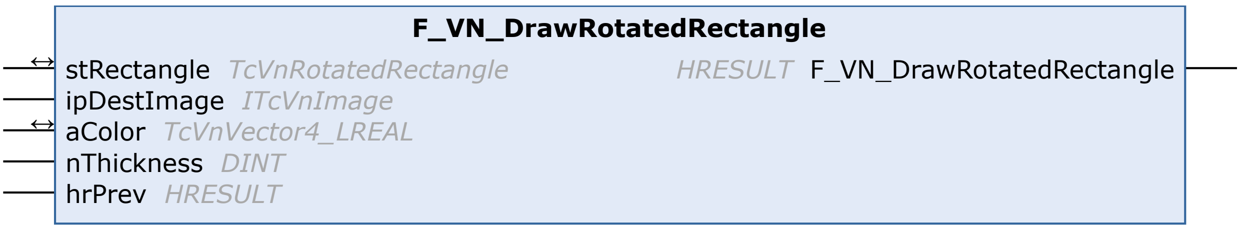
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.40 F_VN_DrawRotatedRectangle



Draw a rectangle.

Syntax

Definition:

```
FUNCTION F_VN_DrawRotatedRectangle : HRESULT
VAR_IN_OUT
    stRectangle : TcVnRotatedRectangle;
END_VAR
VAR_INPUT
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness  : DINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness (if negative, the rectangle is filled)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stRectangle	TcVnRotatedRectangle [▶ 225]	Rectangle to be drawn (rotation angle in degrees)
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

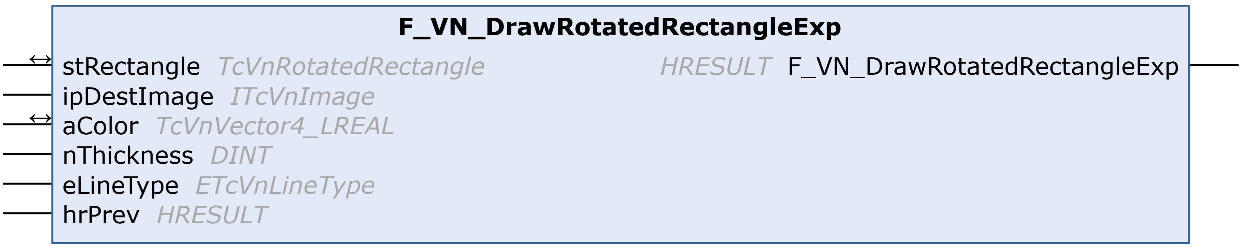
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.41 F_VN_DrawRotatedRectangleExp



Draw a rectangle. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_DrawRotatedRectangleExp : HRESULT
VAR_IN_OUT
    stRectangle : TcVnRotatedRectangle;
END_VAR
VAR_INPUT
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness  : DINT;
    eLineType   : ETcVnLineType;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▶ 390]	Destination image
nThickness	DINT	Line thickness (if negative, the rectangle is filled)
eLineType	ETcVnLineType [▶ 187]	Line type
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stRectangle	TcVnRotatedRectangle [▶ 225]	Rectangle to be drawn (rotation angle in degrees)
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

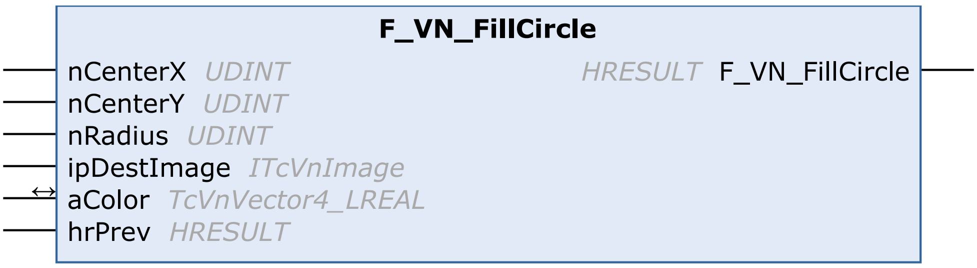
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.42 F_VN_FillCircle



Paint a filled circle.

Syntax

Definition:

```
FUNCTION F_VN_FillCircle : HRESULT
VAR_INPUT
    nCenterX     : UDINT;
    nCenterY     : UDINT;
    nRadius      : UDINT;
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nCenterX	UDINT	x coordinate of the center
nCenterY	UDINT	y coordinate of the center
nRadius	UDINT	Radius
ipDestImage	ITcVnImage [▶ 390]	Destination image
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.43 F_VN_FillContours

Paint a single point set or multiple point sets that are interpreted as contours.

Syntax

Definition:

```
FUNCTION F_VN_FillContours : HRESULT
VAR_INPUT
    ipContours    : ITcVnContainer;
    nContourIndex : DINT;
    ipDestImage   : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor        : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipContours	ITcVnContainer [▸ 349]	Single contour (ContainerType_Vector_TcVnPoint2_DINT) or multiple contours (ContainerType_Vector_Vector_TcVnPoint2_DINT)
nContourIndex	DINT	Index of a specific contour to be drawn (if negative, all contours within the container are drawn)
ipDestImage	ITcVnImage [▸ 390]	Destination image
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▸ 141]	Color

Return value

HRESULT [▸ 122]

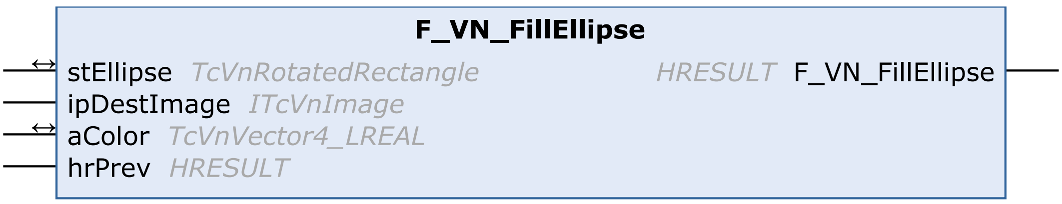
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.44 F_VN_FillEllipse



Paint a filled ellipse.

Syntax

Definition:

```
FUNCTION F_VN_FillEllipse : HRESULT
VAR_IN_OUT
    stEllipse   : TcVnRotatedRectangle;
END_VAR
VAR_INPUT
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▶ 390]	Destination image
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stEllipse	TcVnRotatedRectangle [▶ 225]	Ellipse to be painted (rotation angle in degrees)
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

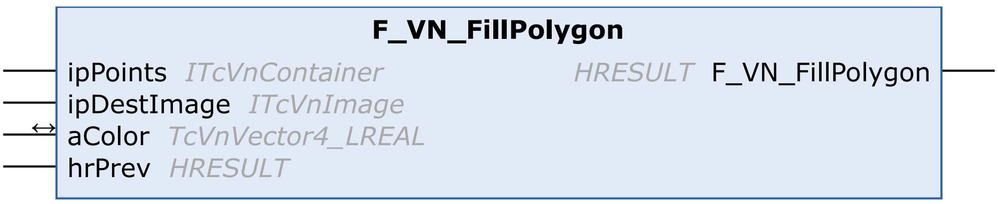
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.45 F_VN_FillPolygon



Paint a filled pologon using a set of points.

Syntax

Definition:

```
FUNCTION F_VN_FillPolygon : HRESULT
VAR_INPUT
    ipPoints     : ITcVnContainer;
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipPoints	ITcVnContainer [▸ 349]	Container with TcVnPoint2_DINT, TcVnPoint2_REAL, or TcVnPoint2_LREAL elements
ipDestImage	ITcVnImage [▸ 390]	Destination image
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▸ 141]	Color

Return value

HRESULT [▸ 122]

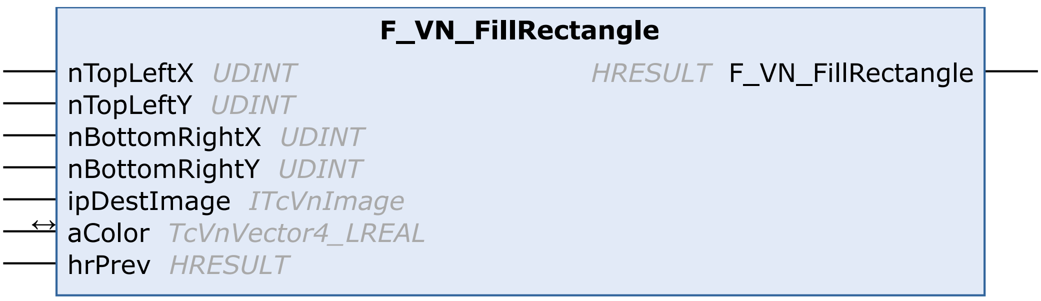
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.46 F_VN_FillRectangle



Paint a filled rectangle.

Syntax

Definition:

```
FUNCTION F_VN_FillRectangle : HRESULT
VAR_INPUT
    nTopLeftX     : UDINT;
    nTopLeftY     : UDINT;
    nBottomRightX : UDINT;
    nBottomRightY : UDINT;
    ipDestImage   : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor        : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nTopLeftX	UDINT	x coordinate of the top left corner
nTopLeftY	UDINT	y coordinate of the top left corner
nBottomRightX	UDINT	x coordinate of the bottom right corner
nBottomRightY	UDINT	y coordinate of the bottom right corner
ipDestImage	ITcVnImage [▸ 390]	Destination image
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▸ 141]	Color

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.47 F_VN_FillRotatedRectangle

Paint a filled rectangle.

Syntax

Definition:

```
FUNCTION F_VN_FillRotatedRectangle : HRESULT
VAR_IN_OUT
    stRectangle : TcVnRotatedRectangle;
END_VAR
VAR_INPUT
    ipDestImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aColor      : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipDestImage	ITcVnImage [▶ 390]	Destination image
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stRectangle	TcVnRotatedRectangle [▶ 225]	Rectangle to be painted (rotation angle in degrees)
aColor	TcVnVector4_LREAL [▶ 141]	Color

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.48 F_VN_PlotIntensityProfile

Plots the pixel intensity profile along a line segment in an image.

Syntax

Definition:

```
FUNCTION F_VN_PlotIntensityProfile : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aStartPoint : TcVnPoint2_REAL;
    aEndPoint   : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (USINT, 1 channel)
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (USINT, 1 channel. x: position, y: intensity, origin: bottom left. An appropriate destination image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aStartPoint	TcVnPoint2_REAL [▸ 139]	Start point of the line segment
aEndPoint	TcVnPoint2_REAL [▸ 139]	End point of the line segment

Return value

HRESULT [▸ 122]

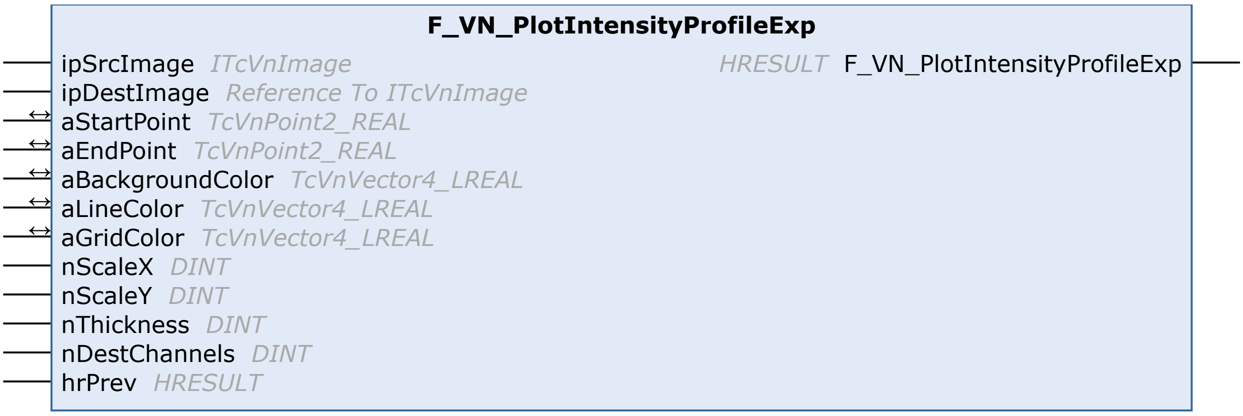
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.49 F_VN_PlotIntensityProfileExp



Plots the pixel intensity profile along a line segment in an image. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_PlotIntensityProfileExp : HRESULT
VAR_INPUT
    ipSrcImage        : ITcVnImage;
    ipDestImage       : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aStartPoint       : TcVnPoint2_REAL;
    aEndPoint         : TcVnPoint2_REAL;
    aBackgroundColor  : TcVnVector4_LREAL;
    aLineColor        : TcVnVector4_LREAL;
    aGridColor        : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nScaleX           : DINT;
    nScaleY           : DINT;
    nThickness        : DINT;
    nDestChannels     : DINT;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT, 1 channel)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (USINT, 1 channel. x: position, y: intensity, origin: bottom left. An appropriate destination image will be created if required.)
nScaleX	DINT	Scale in x direction (ipDestImage will have a width of nScaleX * (ipSrcImage width - 1) + 1)
nScaleY	DINT	Scale in y direction (ipDestImage will have a height of nScaleY * 255 + 1)
nThickness	DINT	Line thickness
nDestChannels	DINT	ipDestImage channels (1 or 3)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aStartPoint	TcVnPoint2_REAL [▶ 139]	Start point of the line segment
aEndPoint	TcVnPoint2_REAL [▶ 139]	End point of the line segment
aBackgroundColor	TcVnVector4_LREAL [▶ 141]	Background color
aLineColor	TcVnVector4_LREAL [▶ 141]	Line color
aGridColor	TcVnVector4_LREAL [▶ 141]	Grid color

Return value

HRESULT [▶ 122]

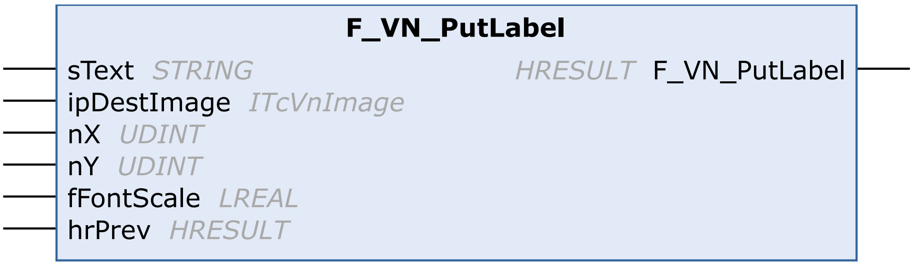
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.50 F_VN_PutLabel



Write a label (black text on white background) into an image.

Syntax

Definition:

```
FUNCTION F_VN_PutLabel : HRESULT
VAR_INPUT
    sText        : STRING;
    ipDestImage : ITcVnImage;
    nX           : UDINT;
    nY           : UDINT;
    fFontScale  : LREAL;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
sText	STRING	Text
ipDestImage	ITcVnImage [▶ 390]	Destination image
nX	UDINT	x coordinate (bottom left)
nY	UDINT	y coordinate (bottom left)
fFontScale	LREAL	Scaling factor
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

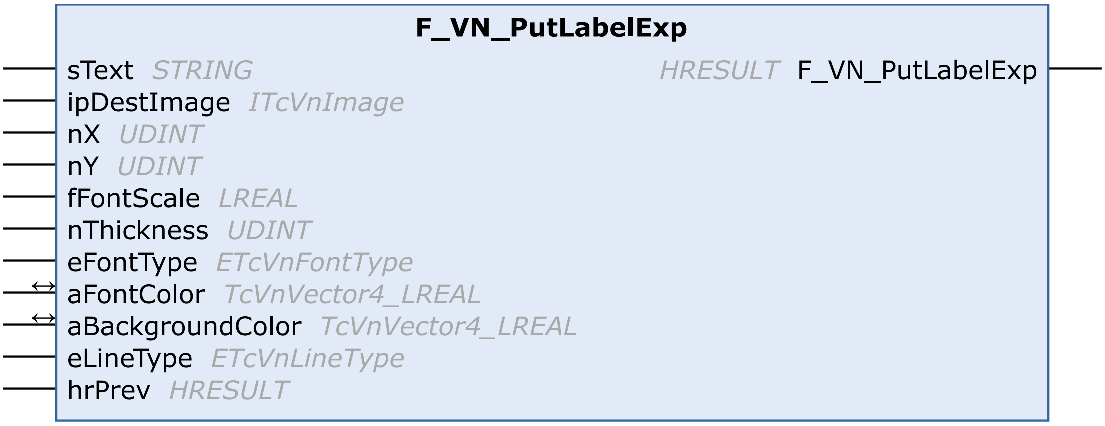
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.51 F_VN_PutLabelExp



Write a label (text on unified background) into an image. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_PutLabelExp : HRESULT
VAR_INPUT
    sText            : STRING;
    ipDestImage      : ITcVnImage;
    nX               : UDINT;
    nY               : UDINT;
    fFontScale       : LREAL;
    nThickness       : UDINT;
    eFontType        : ETcVnFontType;
END_VAR
VAR_IN_OUT
    aFontColor       : TcVnVector4_LREAL;
    aBackgroundColor : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
```

```
    eLineType           : ETcVnLineType;
    hrPrev              : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
sText	STRING	Text
ipDestImage	ITcVnImage [▶ 390]	Destination image
nX	UDINT	x coordinate (bottom left)
nY	UDINT	y coordinate (bottom left)
fFontScale	LREAL	Scaling factor
nThickness	UDINT	Line thickness
eFontType	ETcVnFontType [▶ 183]	Font type
eLineType	ETcVnLineType [▶ 187]	Line type
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aFontColor	TcVnVector4_LREAL [▶ 141]	Font color
aBackgroundColor	TcVnVector4_LREAL [▶ 141]	Background color

Return value

HRESULT [▶ 122]

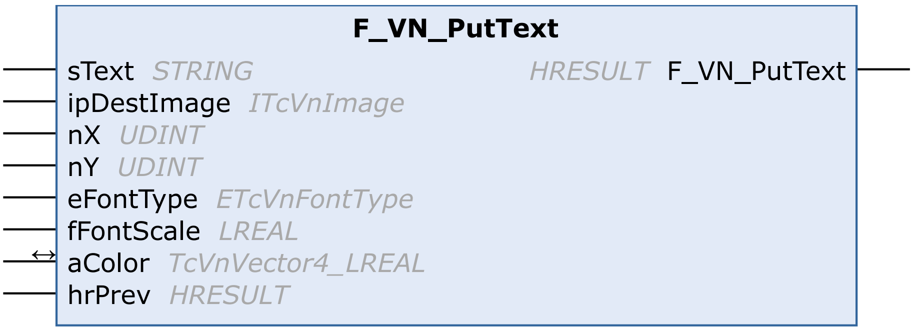
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.52 F_VN_PutText



Write text into an image.

Syntax

Definition:

```
FUNCTION F_VN_PutText : HRESULT
VAR_INPUT
    sText        : STRING;
    ipDestImage  : ITcVnImage;
    nX           : UDINT;
    nY           : UDINT;
    eFontType    : ETcVnFontType;
    fFontScale   : LREAL;
END_VAR
VAR_IN_OUT
    aColor       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
sText	STRING	Text
ipDestImage	ITcVnImage [▸ 390]	Destination image
nX	UDINT	x coordinate (bottom left)
nY	UDINT	y coordinate (bottom left)
eFontType	ETcVnFontType [▸ 183]	Font type
fFontScale	LREAL	Scaling factor
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▸ 141]	Text color

Return value

HRESULT [▸ 122]

更多信息

以下字体可用：

相关函数

- F_VN_PutLabelExp [▶ 1011]
- F_VN_PutText(Exp) [▶ 1012]

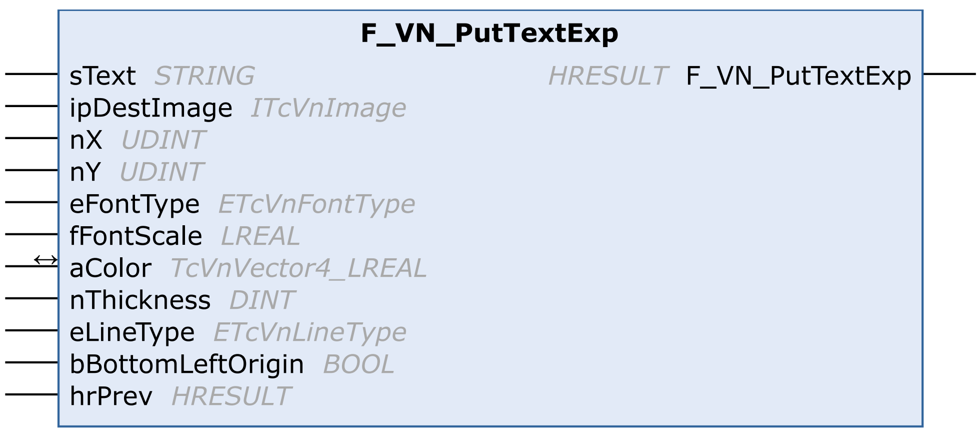
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.11.53 F_VN_PutTextExp



Write text into an image. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_PutTextExp : HRESULT
VAR_INPUT
    sText               : STRING;
    ipDestImage         : ITcVnImage;
    nX                  : UDINT;
    nY                  : UDINT;
    eFontType           : ETcVnFontType;
    fFontScale          : LREAL;
END_VAR
VAR_IN_OUT
    aColor              : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nThickness          : DINT;
    eLineType           : ETcVnLineType;
    bBottomLeftOrigin   : BOOL;
    hrPrev              : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
sText	STRING	Text
ipDestImage	ITcVnImage [▸ 390]	Destination image
nX	UDINT	x coordinate (bottom left)
nY	UDINT	y coordinate (bottom left)
eFontType	ETcVnFontType [▸ 183]	Font type
fFontScale	LREAL	Scaling factor
nThickness	DINT	Line thickness
eLineType	ETcVnLineType [▸ 187]	Line type
bBottomLeftOrigin	BOOL	Sets the image origin to the bottom left corner, if true
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aColor	TcVnVector4_LREAL [▸ 141]	Text color

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.12 Fourier Analysis

该组包含用于图像中的频率分析和频率过滤的函数。

函数

巴特沃斯过滤器

- F_VN_CreateBandpassButterworthFilter [▶ 1016]
- F_VN_CreateBandrejectButterworthFilter [▶ 1018]
- F_VN_CreateHighpassButterworthFilter [▶ 1021]
- F_VN_CreateLowpassButterworthFilter [▶ 1023]

高斯过滤器

- F_VN_CreateBandpassGaussianFilter [▶ 1017]
- F_VN_CreateBandrejectGaussianFilter [▶ 1020]
- F_VN_CreateHighpassGaussianFilter [▶ 1022]
- F_VN_CreateLowpassGaussianFilter [▶ 1024]

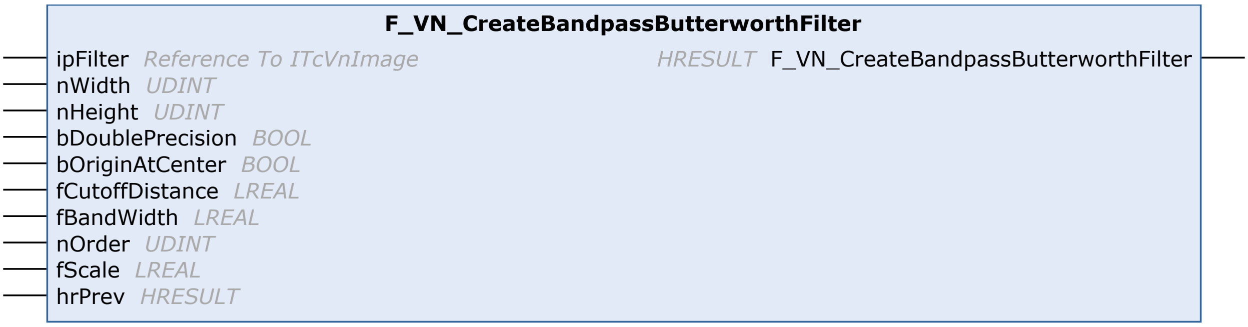
离散傅里叶变换

- F_VN_Dft [▶ 1025]
- F_VN_InverseDft [▶ 1026]
- F_VN_OptimalDftSize [▶ 1027]

其他

- F_VN_PadImageBorder(Exp) [▶ 1028]

6.1.4.12.1 F_VN_CreateBandpassButterworthFilter



Creates a bandpass Butterworth filter, which can be applied to a frequency domain image by element-wise multiplication.

Syntax

Definition:

```
FUNCTION F_VN_CreateBandpassButterworthFilter : HRESULT
VAR_INPUT
    ipFilter         : Reference To ITcVnImage;
    nWidth           : UDINT;
    nHeight          : UDINT;
    bDoublePrecision : BOOL;
    bOriginAtCenter  : BOOL;
    fCutoffDistance  : LREAL;
    fBandWidth       : LREAL;
    nOrder           : UDINT;
    fScale           : LREAL;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipFilter	Reference To ITcVnImage [▶ 390]	Returns the created filter (1 channel of type REAL or LREAL, depending on bDoublePrecision. Non-zero interface pointers are reused.).
nWidth	UDINT	Filter width (even, >= 2, must match the width of the image it is applied to)
nHeight	UDINT	Filter height (even, >= 2, must match the height of the image it is applied to)
bDoublePrecision	BOOL	If true, the filter is generated with double precision (LREAL) instead of single precision (REAL).
bOriginAtCenter	BOOL	If true, the filter origin is shifted to the image center.
fCutoffDistance	LREAL	The cutoff distance of the Butterworth filter (> 0).
fBandWidth	LREAL	The band width of the Butterworth filter (> 0).
nOrder	UDINT	The order of the Butterworth filter (> 0).
fScale	LREAL	The scale factor of the Butterworth filter denominator term (> 0, e.g. 0.414 or 1.0).
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

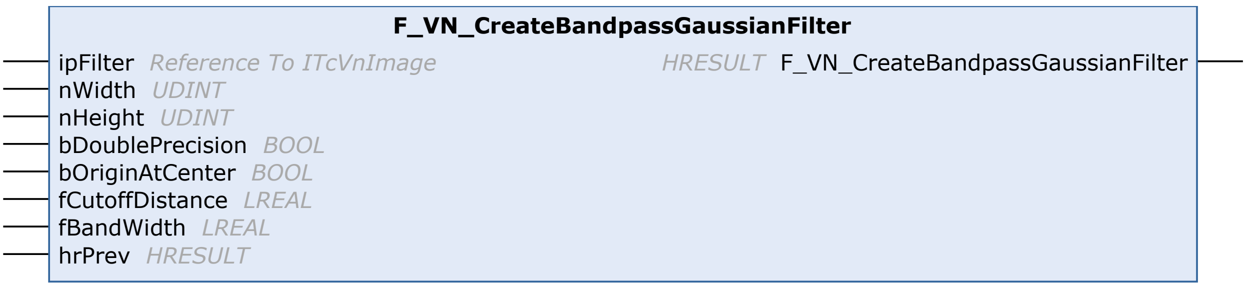
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.12.2 F_VN_CreateBandpassGaussianFilter



Creates a bandpass Gaussian filter, which can be applied to a frequency domain image by element-wise multiplication.

Syntax

Definition:

```
FUNCTION F_VN_CreateBandpassGaussianFilter : HRESULT
VAR_INPUT
    ipFilter          : Reference To ITcVnImage;
```

```
    nWidth            : UDINT;
    nHeight           : UDINT;
    bDoublePrecision  : BOOL;
    bOriginAtCenter   : BOOL;
    fCutoffDistance   : LREAL;
    fBandWidth        : LREAL;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipFilter	Reference To ITcVnImage [▸ 390]	Returns the created filter (1 channel of type REAL or LREAL, depending on bDoublePrecision. Non-zero interface pointers are reused.).
nWidth	UDINT	Filter width (even, >= 2, must match the width of the image it is applied to)
nHeight	UDINT	Filter height (even, >= 2, must match the height of the image it is applied to)
bDoublePrecision	BOOL	If true, the filter is generated with double precision (LREAL) instead of single precision (REAL).
bOriginAtCenter	BOOL	If true, the filter origin is shifted to the image center.
fCutoffDistance	LREAL	The cutoff distance of the Gaussian filter (> 0).
fBandWidth	LREAL	The band width of the Gaussian filter (> 0).
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

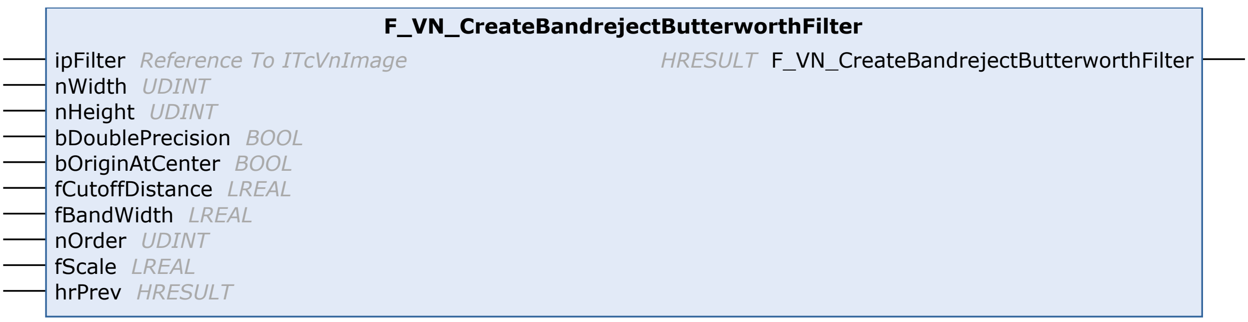
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.12.3 F_VN_CreateBandrejectButterworthFilter



Creates a bandreject Butterworth filter, which can be applied to a frequency domain image by element-wise multiplication.

Syntax

Definition:

```
FUNCTION F_VN_CreateBandrejectButterworthFilter : HRESULT
VAR_INPUT
    ipFilter         : Reference To ITcVnImage;
    nWidth           : UDINT;
    nHeight          : UDINT;
    bDoublePrecision : BOOL;
    bOriginAtCenter  : BOOL;
    fCutoffDistance  : LREAL;
    fBandWidth       : LREAL;
    nOrder           : UDINT;
    fScale           : LREAL;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipFilter	Reference To ITcVnImage [▶ 390]	Returns the created filter (1 channel of type REAL or LREAL, depending on bDoublePrecision. Non-zero interface pointers are reused.).
nWidth	UDINT	Filter width (even, >= 2, must match the width of the image it is applied to)
nHeight	UDINT	Filter height (even, >= 2, must match the height of the image it is applied to)
bDoublePrecision	BOOL	If true, the filter is generated with double precision (LREAL) instead of single precision (REAL).
bOriginAtCenter	BOOL	If true, the filter origin is shifted to the image center.
fCutoffDistance	LREAL	The cutoff distance of the Butterworth filter (> 0).
fBandWidth	LREAL	The band width of the Butterworth filter (> 0).
nOrder	UDINT	The order of the Butterworth filter (> 0).
fScale	LREAL	The scale factor of the Butterworth filter denominator term (> 0, e.g. 0.414 or 1.0).
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.12.4 F_VN_CreateBandrejectGaussianFilter

F_VN_CreateBandrejectGaussianFilter

ipFilter *Reference To ITcVnImage* — *HRESULT* F_VN_CreateBandrejectGaussianFilter
nWidth *UDINT*
nHeight *UDINT*
bDoublePrecision *BOOL*
bOriginAtCenter *BOOL*
fCutoffDistance *LREAL*
fBandWidth *LREAL*
hrPrev *HRESULT*

Creates a bandreject Gaussian filter, which can be applied to a frequency domain image by element-wise multiplication.

Syntax

Definition:

```
FUNCTION F_VN_CreateBandrejectGaussianFilter : HRESULT
VAR_INPUT
    ipFilter         : Reference To ITcVnImage;
    nWidth           : UDINT;
    nHeight          : UDINT;
    bDoublePrecision : BOOL;
    bOriginAtCenter  : BOOL;
    fCutoffDistance  : LREAL;
    fBandWidth       : LREAL;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipFilter	Reference To ITcVnImage [▸ 390]	Returns the created filter (1 channel of type REAL or LREAL, depending on bDoublePrecision. Non-zero interface pointers are reused.).
nWidth	UDINT	Filter width (even, >= 2, must match the width of the image it is applied to)
nHeight	UDINT	Filter height (even, >= 2, must match the height of the image it is applied to)
bDoublePrecision	BOOL	If true, the filter is generated with double precision (LREAL) instead of single precision (REAL).
bOriginAtCenter	BOOL	If true, the filter origin is shifted to the image center.
fCutoffDistance	LREAL	The cutoff distance of the Gaussian filter (> 0).
fBandWidth	LREAL	The band width of the Gaussian filter (> 0).
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

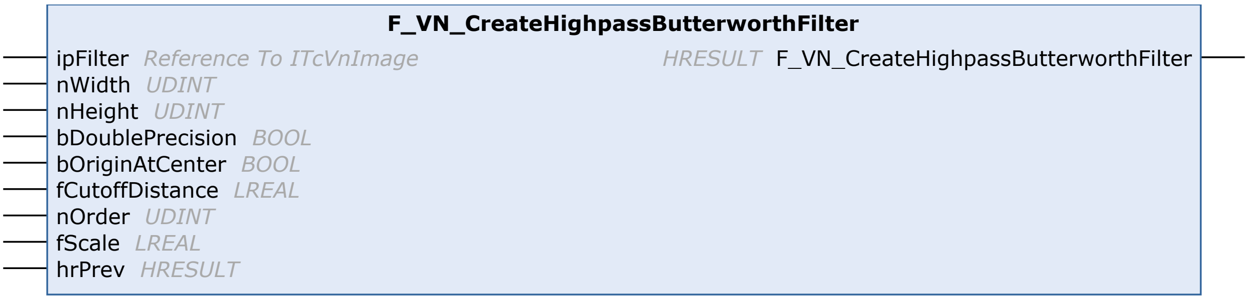
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.12.5 F_VN_CreateHighpassButterworthFilter



Creates a highpass Butterworth filter, which can be applied to a frequency domain image by element-wise multiplication.

Syntax

Definition:

```
FUNCTION F_VN_CreateHighpassButterworthFilter : HRESULT
VAR_INPUT
    ipFilter         : Reference To ITcVnImage;
    nWidth           : UDINT;
    nHeight          : UDINT;
    bDoublePrecision : BOOL;
    bOriginAtCenter  : BOOL;
    fCutoffDistance  : LREAL;
    nOrder           : UDINT;
    fScale           : LREAL;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipFilter	Reference To ITcVnImage [▸ 390]	Returns the created filter (1 channel of type REAL or LREAL, depending on bDoublePrecision. Non-zero interface pointers are reused.).
nWidth	UDINT	Filter width (even, >= 2, must match the width of the image it is applied to)
nHeight	UDINT	Filter height (even, >= 2, must match the height of the image it is applied to)
bDoublePrecision	BOOL	If true, the filter is generated with double precision (LREAL) instead of single precision (REAL).
bOriginAtCenter	BOOL	If true, the filter origin is shifted to the image center.
fCutoffDistance	LREAL	The cutoff distance of the Butterworth filter (> 0).
nOrder	UDINT	The order of the Butterworth filter (> 0).
fScale	LREAL	The scale factor of the Butterworth filter denominator term (> 0, e.g. 0.414 or 1.0).
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

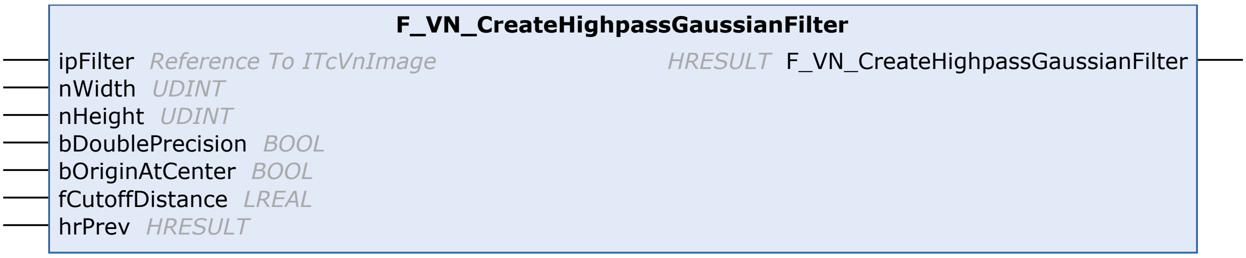
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.12.6 F_VN_CreateHighpassGaussianFilter



Creates a highpass Gaussian filter, which can be applied to a frequency domain image by element-wise multiplication.

Syntax

Definition:

```
FUNCTION F_VN_CreateHighpassGaussianFilter : HRESULT
VAR_INPUT
    ipFilter         : Reference To ITcVnImage;
    nWidth           : UDINT;
    nHeight          : UDINT;
    bDoublePrecision : BOOL;
    bOriginAtCenter  : BOOL;
    fCutoffDistance  : LREAL;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipFilter	Reference To ITcVnImage [▶ 390]	Returns the created filter (1 channel of type REAL or LREAL, depending on bDoublePrecision. Non-zero interface pointers are reused.).
nWidth	UDINT	Filter width (even, >= 2, must match the width of the image it is applied to)
nHeight	UDINT	Filter height (even, >= 2, must match the height of the image it is applied to)
bDoublePrecision	BOOL	If true, the filter is generated with double precision (LREAL) instead of single precision (REAL).
bOriginAtCenter	BOOL	If true, the filter origin is shifted to the image center.
fCutoffDistance	LREAL	The cutoff distance of the Gaussian filter (> 0).
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

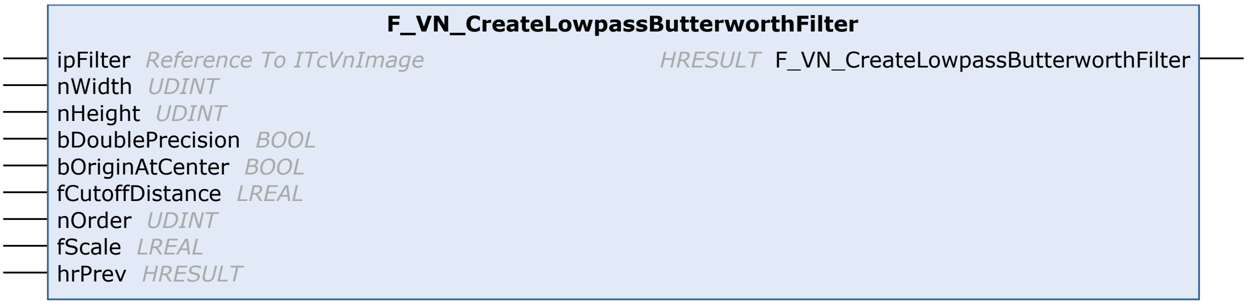
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.12.7 F_VN_CreateLowpassButterworthFilter



Creates a lowpass Butterworth filter, which can be applied to a frequency domain image by element-wise multiplication.

Syntax

Definition:

```
FUNCTION F_VN_CreateLowpassButterworthFilter : HRESULT
VAR_INPUT
    ipFilter          : Reference To ITcVnImage;
    nWidth            : UDINT;
    nHeight           : UDINT;
    bDoublePrecision : BOOL;
    bOriginAtCenter  : BOOL;
    fCutoffDistance  : LREAL;
    nOrder            : UDINT;
    fScale            : LREAL;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipFilter	Reference To ITcVnImage [▶ 390]	Returns the created filter (1 channel of type REAL or LREAL, depending on bDoublePrecision. Non-zero interface pointers are reused.).
nWidth	UDINT	Filter width (even, >= 2, must match the width of the image it is applied to)
nHeight	UDINT	Filter height (even, >= 2, must match the height of the image it is applied to)
bDoublePrecision	BOOL	If true, the filter is generated with double precision (LREAL) instead of single precision (REAL).
bOriginAtCenter	BOOL	If true, the filter origin is shifted to the image center.
fCutoffDistance	LREAL	The cutoff distance of the Butterworth filter (> 0).
nOrder	UDINT	The order of the Butterworth filter (> 0).
fScale	LREAL	The scale factor of the Butterworth filter denominator term (> 0, e.g. 0.414 or 1.0).
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

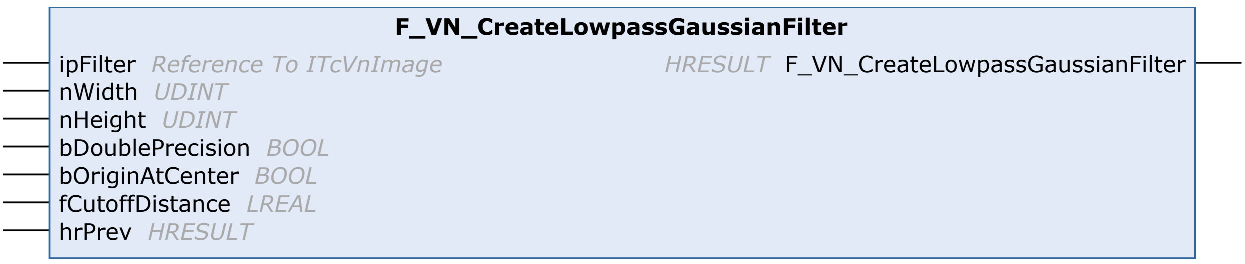
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.12.8 F_VN_CreateLowpassGaussianFilter



Creates a lowpass Gaussian filter, which can be applied to a frequency domain image by element-wise multiplication.

Syntax

Definition:

```
FUNCTION F_VN_CreateLowpassGaussianFilter : HRESULT
VAR_INPUT
    ipFilter          : Reference To ITcVnImage;
    nWidth            : UDINT;
    nHeight           : UDINT;
```

```
    bDoublePrecision : BOOL;
    bOriginAtCenter  : BOOL;
    fCutoffDistance  : LREAL;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipFilter	Reference To ITcVnImage [▶ 390]	Returns the created filter (1 channel of type REAL or LREAL, depending on bDoublePrecision. Non-zero interface pointers are reused.).
nWidth	UDINT	Filter width (even, >= 2, must match the width of the image it is applied to)
nHeight	UDINT	Filter height (even, >= 2, must match the height of the image it is applied to)
bDoublePrecision	BOOL	If true, the filter is generated with double precision (LREAL) instead of single precision (REAL).
bOriginAtCenter	BOOL	If true, the filter origin is shifted to the image center.
fCutoffDistance	LREAL	The cutoff distance of the Gaussian filter (> 0).
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

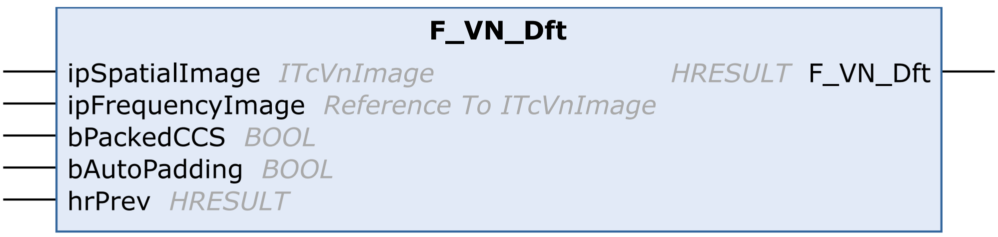
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.12.9 F_VN_Dft



Computes the DFT for a given image.

Syntax

Definition:

```
FUNCTION F_VN_Dft : HRESULT
VAR_INPUT
    ipSpatialImage   : ITcVnImage;
    ipFrequencyImage : Reference To ITcVnImage;
    bPackedCCS       : BOOL;
```

```
    bAutoPadding      : BOOL;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSpatialImag e	ITcVnImage [▶ 390]	Source image (elements of type REAL or LREAL, 1 (Re) or 2 (Re + Im) channels)
ipFrequencyIm age	Reference To ITcVnImage [▶ 390]	Destination image (same type as ipSpatialImage, but number of channels and size can vary depending on bPackedCCS and bAutoPadding.
bPackedCCS	BOOL	If true and ipSpatialImage has only 1 channel, the result image will have 1 channel with packed complex-conjugate-symmetrical format results. Otherwise, the result image will have 2 separate channels (Re + Im), containing the full spectrum.
bAutoPadding	BOOL	If true, the input image is automatically padded (with 0s) to optimal size if required, to speed up dft (Creates a temporary copy so that ipSpatialImage content stays unchanged, which also requires some additional computation power. Therefore, it is recommended to compare execution times with and without padding.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.12.10 F_VN_InverseDft

Compute the inverse DFT for a given frequency image.

Syntax

Definition:

```
FUNCTION F_VN_InverseDft : HRESULT
VAR_INPUT
    ipFrequencyImage : ITcVnImage;
    ipSpatialImage   : Reference To ITcVnImage;
```

```
    bRealOutput       : BOOL;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipFrequencyImage	ITcVnImage [▶ 390]	Source frequency image (elements of type REAL or LREAL, 1 (packed CCS) or 2 (Re + Im) channels)
ipSpatialImage	Reference To ITcVnImage [▶ 390]	Destination image (Same type as ipFrequencyImage, but number of channels can vary depending on bRealOutput. An appropriate image will be created if required.)
bRealOutput	BOOL	Only relevant if ipFrequencyImage has 2 channels. If true, the result image will have only 1 channel. Otherwise, the result image will have 2 separate channels (Re + Im). If ipFrequencyImage has only 1 channel, packed CCS format is assumed and the result image will always have only 1 channel.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

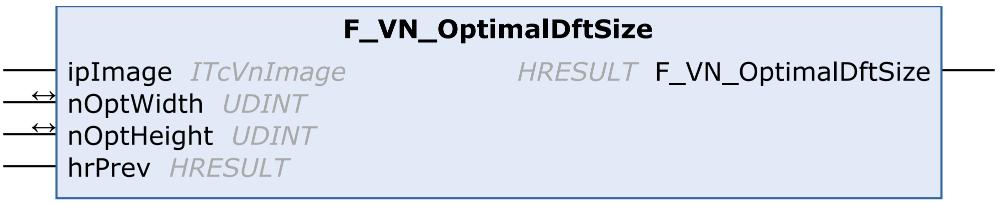
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.12.11 F_VN_OptimalDftSize



Computes the optimal image size for applying a dft (can lead to better performance).

Syntax

Definition:

```
FUNCTION F_VN_OptimalDftSize : HRESULT
VAR_INPUT
    ipImage     : ITcVnImage;
END_VAR
VAR_IN_OUT
    nOptWidth   : UDINT;
    nOptHeight  : UDINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	Source image for which to compute the optimal width and height
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nOptWidth	UDINT	Returns the optimal width of the image
nOptHeight	UDINT	Returns the optimal height of the image

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.12.12 F_VN_PadImageBorder

Add padding (zeros) around the original image borders.

Syntax

Definition:

```
FUNCTION F_VN_PadImageBorder : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipPaddedImage  : Reference To ITcVnImage;
    nTopBorder     : UDINT;
    nBottomBorder  : UDINT;
    nLeftBorder    : UDINT;
    nRightBorder   : UDINT;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipPaddedImage	Reference To ITcVnImage [▶ 390]	Padded destination image (Same type as ipSrcImage, an appropriate destination image will be created if required.)
nTopBorder	UDINT	Padding height in pixels above top border
nBottomBorder	UDINT	Padding height in pixels below bottom border
nLeftBorder	UDINT	Padding width in pixels before left border
nRightBorder	UDINT	Padding width in pixels after right border
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.12.13 F_VN_PadImageBorderExp

Add padding around the original image borders. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_PadImageBorderExp : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipPaddedImage : Reference To ITcVnImage;
    nTopBorder    : UDINT;
    nBottomBorder : UDINT;
    nLeftBorder   : UDINT;
    nRightBorder  : UDINT;
    ePaddingType  : ETcVnBorderInterpolationMethod;
END_VAR
VAR_IN_OUT
    aPaddingValue : TcVnVector4_LREAL;
END_VAR
```

```
VAR_INPUT
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipPaddedImage	Reference To ITcVnImage [▶ 390]	Padded destination image (Same type as ipSrcImage, an appropriate destination image will be created if required.)
nTopBorder	UDINT	Padding height in pixels above top border
nBottomBorder	UDINT	Padding height in pixels below bottom border
nLeftBorder	UDINT	Padding width in pixels before left border
nRightBorder	UDINT	Padding width in pixels after right border
ePaddingType	ETcVnBorderInterpolationMethod [▶ 145]	Specifies how the pixel values of the padding area are determined
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aPaddingValue	TcVnVector4_LREAL [▶ 141]	Specifies the padding value if CONSTANT is used

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13 Geometric and Coordinate Transformations

该组包含用于转换图像和像素的函数。

函数

标定 [▶ 1031]

- 内部和外部标定
- 光学几何畸变补偿
- 图像坐标与世界坐标之间的转换

仿射转换

- F_VN_ApplyRotationToAffineTransformation（Exp） [▶ 1075]
- F_VN_ApplyScalingToAffineTransformation（Exp） [▶ 1077]

- F_VN_ApplyTranslationToAffineTransformation（Exp） [▶ 1079]
- F_VN_ApplyYAxisInversionToAffineTransformation（Exp） [▶ 1080]
- F_VN_DecomposeAffineTransformation(Exp) [▶ 1096]
- F_VN_GenerateAffineTransformationUnitMatrix2D [▶ 1102]
- F_VN_GetAffineTransformation [▶ 1103]
- F_VN_GetAffineTransformation2D(Exp) [▶ 1104]
- F_VN_InvertAffineTransformation [▶ 1110]
- F_VN_WarpAffine（Exp） [▶ 1125]
- F_VN_WarpAffine_Container [▶ 1126]
- F_VN_WarpAffine_Point [▶ 1127]
- F_VN_WarpAffine_Rectangle [▶ 1128]

笛卡尔坐标和极坐标之间的转换

- F_VN_ConvertCartesianToPolarAngleImage（Exp） [▶ 1082]
- F_VN_ConvertCartesianToPolarAngles（Exp） [▶ 1084]
- F_VN_ConvertCartesianToPolarImages（Exp） [▶ 1086]
- F_VN_ConvertCartesianToPolarMagnitudeImage [▶ 1088]
- F_VN_ConvertCartesianToPolarMagnitudes [▶ 1089]
- F_VN_ConvertCartesianToPolarPoints（Exp） [▶ 1090]
- F_VN_ConvertPolarToCartesianImages（Exp） [▶ 1092]
- F_VN_ConvertPolarToCartesianPoints（Exp） [▶ 1094]
- F_VN_RemapImageToLogPolarSpace（Exp） [▶ 1113]
- F_VN_RemapImageToPolarSpace（Exp） [▶ 1116]

简单图像运算

- F_VN_FlipImage [▶ 1101]
- F_VN_PyramidDown [▶ 1111]
- F_VN_PyramidUp [▶ 1112]
- F_VN_ResizeImage [▶ 1120]

图像对齐和旋转

- F_VN_AlignRotatedImageRegion(Exp) [▶ 1073]
- F_VN_RotateImage(Exp) [▶ 1122]

透视转换

- F_VN_DecomposeHomography(Exp) [▶ 1098]
- F_VN_GetPerspectiveTransformation [▶ 1106]
- F_VN_Homography（Exp） [▶ 1107]
- F_VN_PerspectiveTransformation [▶ 1110]
- F_VN_WarpPerspective（Exp） [▶ 1131]F_VN_WarpPerspective(Exp)F_VN_WarpPerspective(Exp)
- F_VN_WarpPerspective_Container [▶ 1132]
- F_VN_WarpPerspective_Point [▶ 1133]
- F_VN_WarpPerspective_Rectangle [▶ 1134]

6.1.4.13.1 Calibration

这个子组包含了像素和现实全局像素之间的相关性函数。它们通常与测量功能 [▶ 1428]一起使用。

函数

标定

- F_VN_CalibrateCamera（Exp） [▶ 1032]
- F_VN_CalibrateCameraManually(Exp) [▶ 1042]
- F_VN_CalibrateCameraPlanar（Exp） [▶ 1045]
- F_VN_CalibrateLinescanCamera(Exp) [▶ 1047]
- F_VN_DetectPatternPoints(Exp) [▶ 1057]
- F_VN_GenerateCalibrationPatternReferencePoints [▶ 1060]
- F_VN_SortDetectedPatternPoints [▶ 1065]
- F_VN_SolvePnP(Exp) [▶ 1063]

镜头几何畸变补偿

- F_VN_CompensateLensDistortion（Exp） [▶ 1051]
- F_VN_CompensateLensDistortionForPoints（Exp） [▶ 1054]

图像坐标与世界坐标之间的转换

- F_VN_TransformCoordinatesImageToWorld [▶ 1067]
- F_VN_TransformCoordinatesWorldToImage [▶ 1072]
- F_VN_TransformCoordinatesPlanar [▶ 1070]
- F_VN_ImagePointsWorldDistance [▶ 1061]

输入图像要求

输入图像不应使用压缩或滤波格式，如Bayer格式。就标定函数而言，这样做的后果是干扰模板识别，从而导致标定结果较差或根本没有。由于在补偿函数过程中进行了插值，改变了像素的原始排列，因此之后无法再正确转换图像。因此，建议在使用该子组中的函数之前一定要先进行转换。

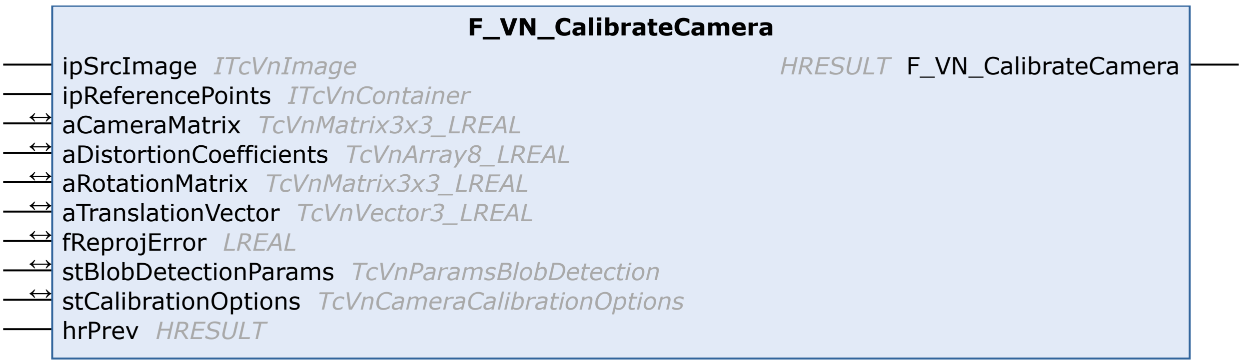
相机矩阵要求

$$CameraMatrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

相机矩阵由焦距 fx 和 fy 以及主点（cx,cy）组成，以像素单位表示。主点是光轴与图像的交点，通常位于图像中心附近。必须满足以下条件：

- 0 < fx 和 fy
- cx >= 0 和 <= 图像宽度 -1
- cy >= 0 和 <= 图像高度 -1
- 如图所示，矩阵中的其他位置必须填 0 或 1。不允许其他值。

6.1.4.13.1.1 F_VN_CalibrateCamera



Compute the camera parameters (intrinsic + extrinsic) by evaluating an image containing a calibration pattern (circles).
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_CalibrateCamera : HRESULT
VAR_INPUT
    ipSrcImage               : ITcVnImage;
    ipReferencePoints        : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aCameraMatrix            : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients  : TcVnArray8_LREAL;
    aRotationMatrix          : TcVnMatrix3x3_LREAL;
    aTranslationVector       : TcVnVector3_LREAL;
    fReprojError             : LREAL;
    stBlobDetectionParams    : TcVnParamsBlobDetection;
    stCalibrationOptions     : TcVnCameraCalibrationOptions;
END_VAR
VAR_INPUT
    hrPrev                   : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image containing a calibration pattern (1 channel, USINT elements)
ipReferencePoints	ITcVnContainer [▶ 349]	Reference calibration pattern point positions (ContainerType_Vector_TcVnPoint3_REAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns the camera matrix
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	Returns the lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns the rotation matrix
aTranslationVector	TcVnVector3_LREAL [▶ 141]	Returns the translation vector
fReprojError	LREAL	Returns the reprojection error
stBlobDetectionParams	TcVnParamsBlobDetection [▶ 213]	Parameters for the internally used F_VN_DetectBlobs function
stCalibrationOptions	TcVnCameraCalibrationOptions [▶ 206]	Calibration options

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_CalibrateCamera根据包含校准图案的输入图像计算相机的内在和外在参数。校准图案的元素必须可检测为斑点。

输入图像

输入图像ipSrcImage必须是位深为 8 位的单通道灰度图像。该图像必须包含一个与参考点ipReferencePoints相匹配的带圆圈的校准图案。该函数期望在白色背景上搭配黑色填充圆圈。为了反转图像，可以使用斑点函数的ETcVnThresholdType = TCVN_TT_BINARY_INV。

专家版F_VN_CalibrateCameraExp [▸ 1035]需要使用多个输入图像进行校准。

参考点

容器ipReferencePoints必须包含输入图像上说明的参考点ipSrcImage，作为校准模式。这些点必须被指定为类型为REAL的 3 维点，以便容器类型为ContainerType_Vector_TcVnPoint3_REAL。

参考点可按以下方式产生：

- 手动创建一个 XML 文件，并用功能块FB_VN_ReadCalibrationPattern [▸ 1523]读取。
- 校准助手 [▸ 94]中的配置，通过点击**写入结果**保存在 TcCOM 对象中，并使用功能块FB_VN_GevCameraControl [▸ 1550]的方法GetCalibPatternRef [▸ 1558]读取。设定的中心点被考虑在内。
- 将参考点单独插入一个容器中。

相机参数的矩阵

这些矩阵返回相机模型的计算参数。然后，它们可用于补偿客观畸变，并用于像素坐标和全局坐标之间的转换。

- 相机矩阵
- 畸变系数
- 旋转矩阵
- 平移矢量

> **舍入误差**
>
> 由于舍入误差，计算出的参数值可能与校准助手中的计算值略有偏差。

再投影误差

斑点检测的参数

> **没有棋盘图案**
>
> 由于参考点被检测为一个斑点结构，因此无法使用棋盘图案。

校准选项

> **默认设置**
>
> 如需从校准向导 [▸ 94]获得默认设置，位bFixAspectRatio、bFixPrincipalPoint、bFixK5、bFixK6和bRationalModel必须设置为 TRUE。
>
> **bFixPrincipalPoint = “修复中心点”**
>
> 该位bFixPrincipalPoint对应于相机校准助手中的“修复中心点”选项。

样本

- 校准助手 [▸ 2720]

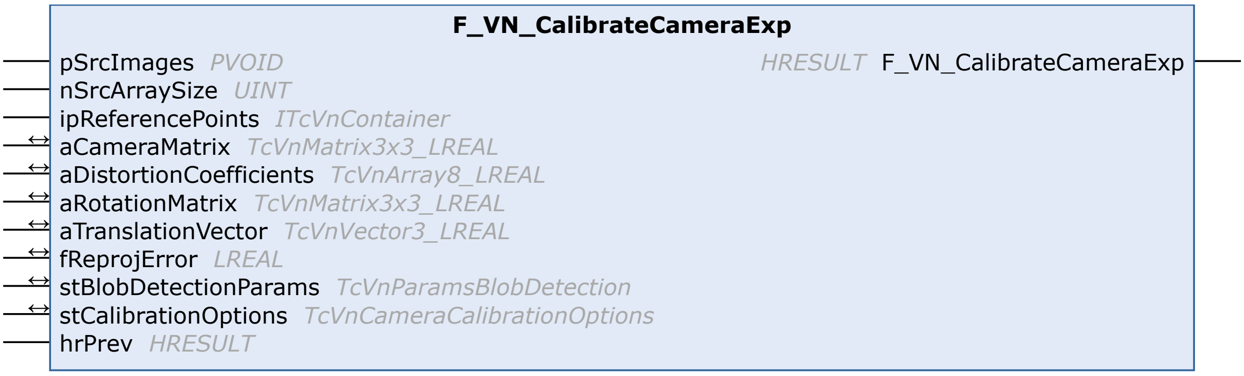
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.2 F_VN_CalibrateCameraExp



Compute the camera parameters (intrinsic + extrinsic) by evaluating images containing a calibration pattern (circles). The extrinsic parameters are computed for the first image in the array. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_CalibrateCameraExp : HRESULT
VAR_INPUT
    pSrcImages                : PVOID;
    nSrcArraySize             : UINT;
    ipReferencePoints         : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aCameraMatrix             : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients   : TcVnArray8_LREAL;
    aRotationMatrix           : TcVnMatrix3x3_LREAL;
    aTranslationVector        : TcVnVector3_LREAL;
    fReprojError              : LREAL;
    stBlobDetectionParams     : TcVnParamsBlobDetection;
    stCalibrationOptions      : TcVnCameraCalibrationOptions;
END_VAR
VAR_INPUT
    hrPrev                    : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
pSrcImages	PVOID	Pointer to an array of source images, each containing the same calibration pattern (1 channel, USINT elements)
nSrcArraySize	UINT	pSrcImages array size
ipReferencePoints	ITcVnContainer [▶ 349]	Reference calibration pattern point positions (ContainerType_Vector_TcVnPoint3_REAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns the camera matrix
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	Returns the lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns the rotation matrix
aTranslationVector	TcVnVector3_LREAL [▶ 141]	Returns the translation vector
fReprojError	LREAL	Returns the reprojection error
stBlobDetectionParams	TcVnParamsBlobDetection [▶ 213]	Parameters for the internally used F_VN_DetectBlobs function
stCalibrationOptions	TcVnCameraCalibrationOptions [▶ 206]	Calibration options

Return value

HRESULT [▶ 122]

更多信息

该函数F_VN_CalibrateCameraExp是F_VN_CalibrateCamera [▶ 1032]的专家版本。它能够使用多个输入图像来校准相机。

参数

输入图像

在指针pSrcImages和图像数量nSrcArraySize的帮助下，相机校准的输入图像被传送到该函数。否则，单个图像必须满足与标准功能相同的要求。

⚠ 警告

指针处理

确保指针的规格和数组的大小与实际数据相符。否则可能发生不允许的内存访问，导致系统崩溃。

参考点

容器ipReferencePoints必须包含输入图像上说明的参考点ipSrcImage，作为校准模式。这些点必须被指定为类型为REAL的 3 维点，以便容器类型为ContainerType_Vector_TcVnPoint3_REAL。

参考点可按以下方式产生：

- 手动创建一个 XML 文件，并用功能块FB_VN_ReadCalibrationPattern [▶ 1523]读取。
- 校准助手 [▶ 94]中的配置，通过点击**写入结果**保存在 TcCOM 对象中，并使用功能块FB_VN_GevCameraControl [▶ 1550]的方法GetCalibPatternRef [▶ 1558]读取。设定的中心点被考虑在内。
- 将参考点单独插入一个容器中。

相机参数的矩阵

这些矩阵返回相机模型的计算参数。然后，它们可用于补偿客观畸变，并用于像素坐标和全局坐标之间的转换。

- 相机矩阵
- 畸变系数
- 旋转矩阵

- 平移矢量

舍入误差

由于舍入误差，计算出的参数值可能与校准助手中的计算值略有偏差。

再投影误差

斑点检测的参数

没有棋盘图案

由于参考点被检测为一个斑点结构，因此无法使用棋盘图案。

校准选项

默认设置

如需从校准向导 [▶ 94]获得默认设置，位bFixAspectRatio、bFixPrincipalPoint、bFixK5、bFixK6和bRationalModel必须设置为 TRUE。

bFixPrincipalPoint = “修复中心点”

该位bFixPrincipalPoint对应于相机校准助手中的“修复中心点”选项。

应用

样本

- 校准助手 [▶ 2720]

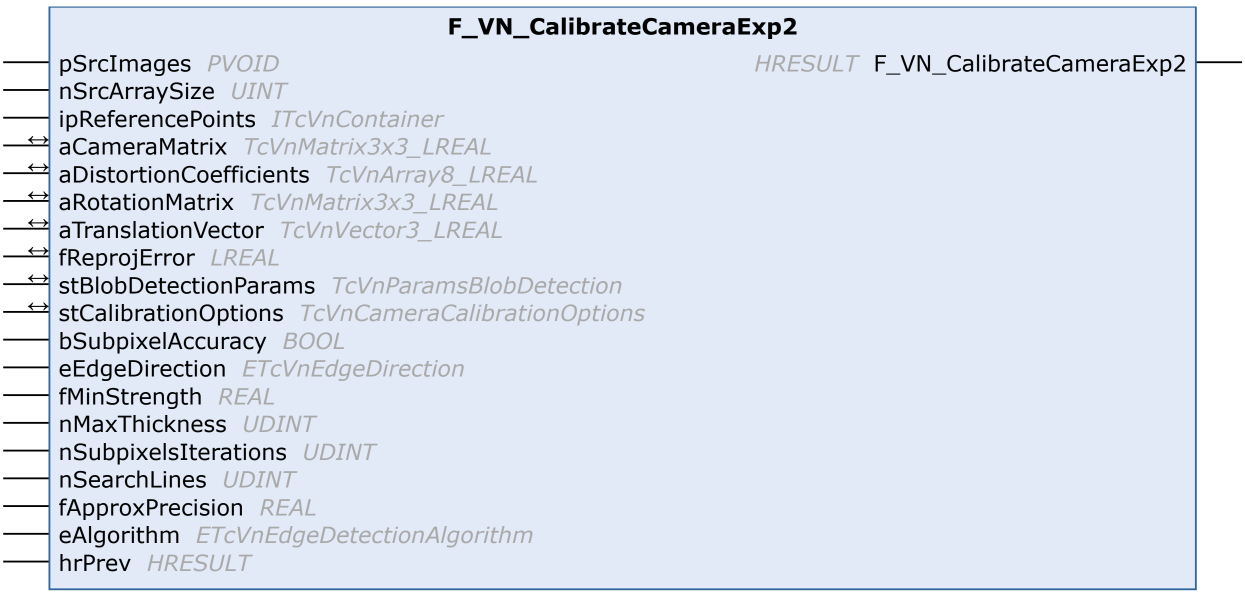
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.3 F_VN_CalibrateCameraExp2



Compute the camera parameters (intrinsic + extrinsic) by evaluating images containing a calibration pattern (circles). The extrinsic parameters are computed for the first image in the array. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_CalibrateCameraExp2 : HRESULT
VAR_INPUT
    pSrcImages               : PVOID;
    nSrcArraySize            : UINT;
    ipReferencePoints        : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aCameraMatrix            : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients : TcVnArray8_LREAL;
    aRotationMatrix          : TcVnMatrix3x3_LREAL;
    aTranslationVector       : TcVnVector3_LREAL;
    fReprojError             : LREAL;
    stBlobDetectionParams    : TcVnParamsBlobDetection;
    stCalibrationOptions     : TcVnCameraCalibrationOptions;
END_VAR
VAR_INPUT
    bSubpixelAccuracy        : BOOL;
    eEdgeDirection           : ETcVnEdgeDirection;
    fMinStrength             : REAL;
    nMaxThickness            : UDINT;
    nSubpixelsIterations     : UDINT;
    nSearchLines             : UDINT;
    fApproxPrecision         : REAL;
    eAlgorithm               : ETcVnEdgeDetectionAlgorithm;
    hrPrev                   : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
pSrcImages	PVOID	Pointer to an array of source images, each containing the same calibration pattern (1 channel, USINT elements)
nSrcArraySize	UINT	pSrcImages array size
ipReferencePoints	ITcVnContainer [▶ 349]	Reference calibration pattern point positions (ContainerType_Vector_TcVnPoint3_REAL)
bSubpixelAccuracy	BOOL	If true, the pattern points are detected with subpixel accuracy
eEdgeDirection	ETcVnEdgeDirection [▶ 178]	Specification of the edge direction to search for (from center to outside ellipse)
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
nMaxThickness	UDINT	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels
nSubpixelsIterations	UDINT	Specifies the number of subpixels (for INTERPOLATION, 10 – 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 – 100 usually is sufficient)
nSearchLines	UDINT	Specifies the amount of search lines, which are equally distributed in all directions (must be >= 8 and a multiple of 4)
fApproxPrecision	REAL	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▶ 177]	Selection of the edge detection algorithm
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns the camera matrix
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	Returns the lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns the rotation matrix
aTranslationVector	TcVnVector3_LREAL [▶ 141]	Returns the translation vector
fReprojError	LREAL	Returns the reprojection error
stBlobDetectionParams	TcVnParamsBlobDetection [▶ 213]	Parameters for the internally used F_VN_DetectBlobs function
stCalibrationOptions	TcVnCameraCalibrationOptions [▶ 206]	Calibration options

Return value

HRESULT [▶ 122]

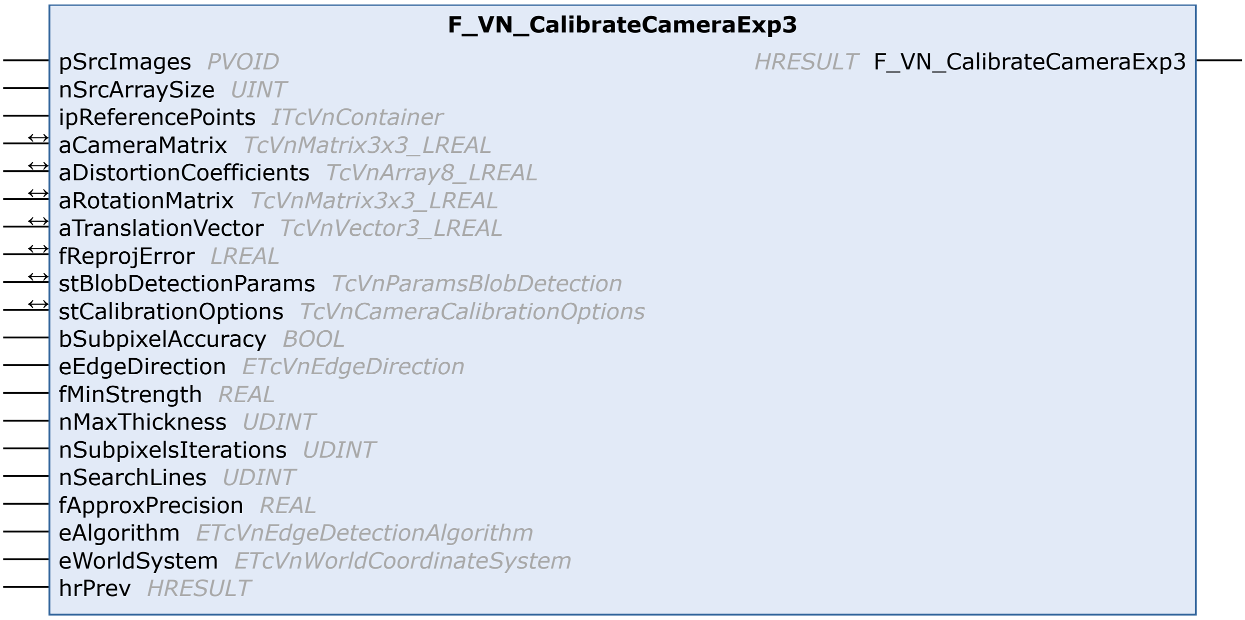
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.4 F_VN_CalibrateCameraExp3



Compute the camera parameters (intrinsic + extrinsic) by evaluating images containing a calibration pattern (circles). The extrinsic parameters are computed for the first image in the array. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_CalibrateCameraExp3 : HRESULT
VAR_INPUT
    pSrcImages                : PVOID;
    nSrcArraySize             : UINT;
    ipReferencePoints         : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aCameraMatrix             : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients   : TcVnArray8_LREAL;
    aRotationMatrix           : TcVnMatrix3x3_LREAL;
    aTranslationVector        : TcVnVector3_LREAL;
    fReprojError              : LREAL;
    stBlobDetectionParams     : TcVnParamsBlobDetection;
    stCalibrationOptions      : TcVnCameraCalibrationOptions;
END_VAR
VAR_INPUT
    bSubpixelAccuracy         : BOOL;
    eEdgeDirection            : ETcVnEdgeDirection;
    fMinStrength              : REAL;
    nMaxThickness             : UDINT;
    nSubpixelsIterations      : UDINT;
    nSearchLines              : UDINT;
    fApproxPrecision          : REAL;
    eAlgorithm                : ETcVnEdgeDetectionAlgorithm;
    eWorldSystem              : ETcVnWorldCoordinateSystem;
    hrPrev                    : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
pSrcImages	PVOID	Pointer to an array of source images, each containing the same calibration pattern (1 channel, USINT elements)
nSrcArraySize	UINT	pSrcImages array size
ipReferencePoints	ITcVnContainer [▸ 349]	Reference calibration pattern point positions (ContainerType_Vector_TcVnPoint3_REAL)
bSubpixelAccuracy	BOOL	If true, the pattern points are detected with subpixel accuracy
eEdgeDirection	ETcVnEdgeDirection [▸ 178]	Specification of the edge direction to search for (from center to outside ellipse)
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
nMaxThickness	UDINT	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels
nSubpixelsIterations	UDINT	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)
nSearchLines	UDINT	Specifies the amount of search lines, which are equally distributed in all directions (must be >= 8 and a multiple of 4)
fApproxPrecision	REAL	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▸ 177]	Selection of the edge detection algorithm
eWorldSystem	ETcVnWorldCoordinateSystem [▸ 202]	World coordinate system orientation
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▸ 141]	Returns the camera matrix
aDistortionCoefficients	TcVnArray8_LREAL [▸ 141]	Returns the lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▸ 141]	Returns the rotation matrix
aTranslationVector	TcVnVector3_LREAL [▸ 141]	Returns the translation vector
fReprojError	LREAL	Returns the reprojection error
stBlobDetectionParams	TcVnParamsBlobDetection [▸ 213]	Parameters for the internally used F_VN_DetectBlobs function
stCalibrationOptions	TcVnCameraCalibrationOptions [▸ 206]	Calibration options

Return value

HRESULT [▶ 122]

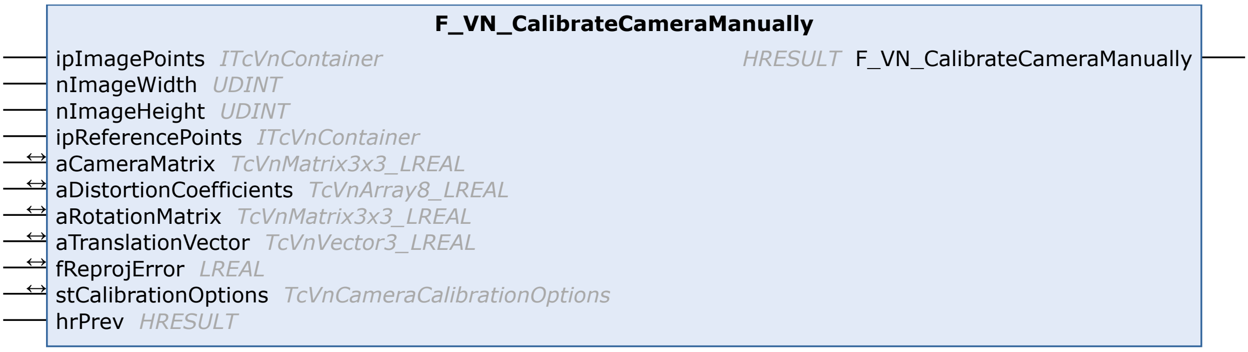
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.5 F_VN_CalibrateCameraManually



Compute the camera parameters (intrinsic + extrinsic). The extrinsic parameters are computed for the first inner container.

Syntax

Definition:

```
FUNCTION F_VN_CalibrateCameraManually : HRESULT
VAR_INPUT
    ipImagePoints            : ITcVnContainer;
    nImageWidth              : UDINT;
    nImageHeight             : UDINT;
    ipReferencePoints        : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aCameraMatrix            : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients : TcVnArray8_LREAL;
    aRotationMatrix          : TcVnMatrix3x3_LREAL;
    aTranslationVector       : TcVnVector3_LREAL;
    fReprojError             : LREAL;
    stCalibrationOptions     : TcVnCameraCalibrationOptions;
END_VAR
VAR_INPUT
    hrPrev                   : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImagePoints	ITcVnContainer [▶ 349]	Provided image points of multiple images (ContainerType_Vector_Vector_TcVnPoint2_REAL, each inner container with at least 6 points). The container and point order must match ipReferencePoints.
nImageWidth	UDINT	Image width
nImageHeight	UDINT	Image height
ipReferencePoints	ITcVnContainer [▶ 349]	Reference world points (ContainerType_Vector_Vector_TcVnPoint3_REAL). The number of inner containers and their amount of points must match ipImagePoints.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns the camera matrix
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	Returns the lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns the rotation matrix (for the first inner container)
aTranslationVector	TcVnVector3_LREAL [▶ 141]	Returns the translation vector (for the first inner container)
fReprojError	LREAL	Returns the reprojection error
stCalibrationOptions	TcVnCameraCalibrationOptions [▶ 206]	Calibration options

Return value

HRESULT [▶ 122]

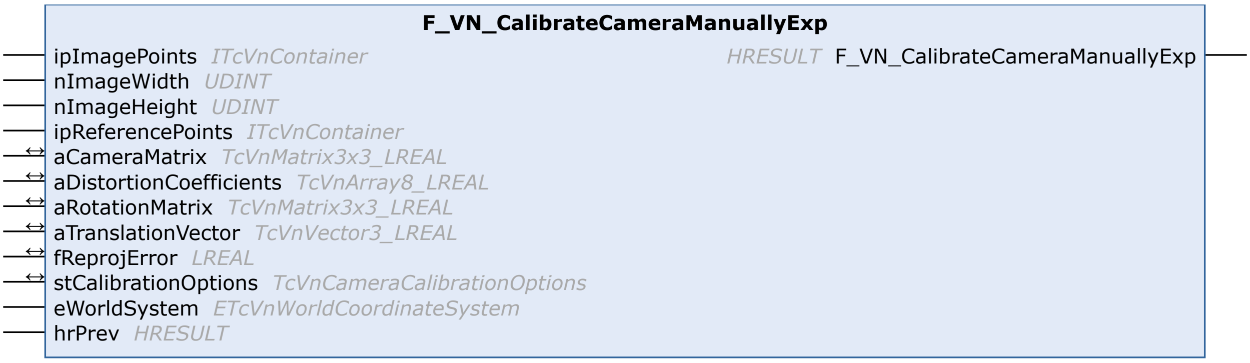
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.6 F_VN_CalibrateCameraManuallyExp



Compute the camera parameters (intrinsic + extrinsic). The extrinsic parameters are computed for the first inner container. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_CalibrateCameraManuallyExp : HRESULT
VAR_INPUT
    ipImagePoints             : ITcVnContainer;
    nImageWidth               : UDINT;
    nImageHeight              : UDINT;
    ipReferencePoints         : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aCameraMatrix             : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients   : TcVnArray8_LREAL;
    aRotationMatrix           : TcVnMatrix3x3_LREAL;
    aTranslationVector        : TcVnVector3_LREAL;
    fReprojError              : LREAL;
    stCalibrationOptions      : TcVnCameraCalibrationOptions;
END_VAR
VAR_INPUT
    eWorldSystem              : ETcVnWorldCoordinateSystem;
    hrPrev                    : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImagePoints	ITcVnContainer [▸ 349]	Provided image points of multiple images (ContainerType_Vector_Vector_TcVnPoint2_REAL, each inner container with at least 6 points). The container and point order must match ipReferencePoints.
nImageWidth	UDINT	Image width
nImageHeight	UDINT	Image height
ipReferencePoints	ITcVnContainer [▸ 349]	Reference world points (ContainerType_Vector_Vector_TcVnPoint3_REAL). The number of inner containers and their amount of points must match ipImagePoints.
eWorldSystem	ETcVnWorldCoordinateSystem [▸ 202]	World coordinate system orientation
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns the camera matrix
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	Returns the lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns the rotation matrix (for the first inner container)
aTranslationVector	TcVnVector3_LREAL [▶ 141]	Returns the translation vector (for the first inner container)
fReprojError	LREAL	Returns the reprojection error
stCalibrationOptions	TcVnCameraCalibrationOptions [▶ 206]	Calibration options

Return value

HRESULT [▶ 122]

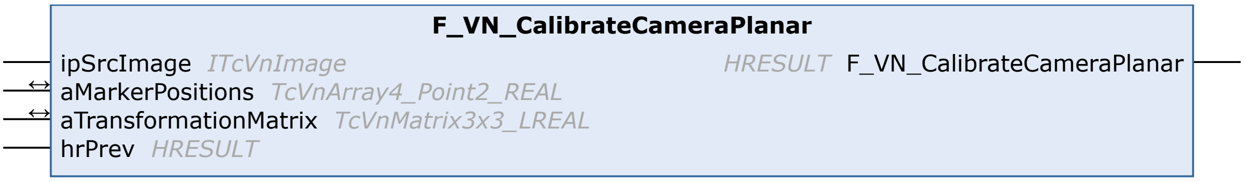
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.7 F_VN_CalibrateCameraPlanar



Calibrate camera using a planar calibration pattern comprised of four circles marking the corners of a rectangle.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_CalibrateCameraPlanar : HRESULT
VAR_INPUT
    ipSrcImage            : ITcVnImage;
END_VAR
VAR_IN_OUT
    aMarkerPositions      : TcVnArray4_Point2_REAL;
    aTransformationMatrix : TcVnMatrix3x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Binary source image (background zero, markers non-zero)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMarkerPositions	TcVnArray4_Point2_REAL [▶ 141]	Marker positions within their plane using world units (right-handed coordinate system)
aTransformationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns a 3-by-3 matrix for the transformation from image coordinates into the reference coordinate system

Return value

HRESULT [▶ 122]

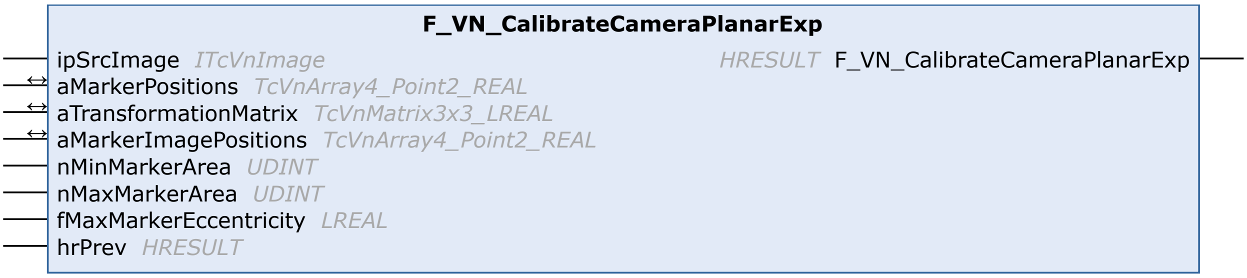
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.8 F_VN_CalibrateCameraPlanarExp



Calibrate camera using a planar calibration pattern comprised of four circles marking the corners of a rectangle. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_CalibrateCameraPlanarExp : HRESULT
VAR_INPUT
    ipSrcImage              : ITcVnImage;
END_VAR
VAR_IN_OUT
    aMarkerPositions        : TcVnArray4_Point2_REAL;
    aTransformationMatrix   : TcVnMatrix3x3_LREAL;
    aMarkerImagePositions   : TcVnArray4_Point2_REAL;
END_VAR
VAR_INPUT
    nMinMarkerArea          : UDINT;
    nMaxMarkerArea          : UDINT;
```

```
    fMaxMarkerEccentricity : LREAL;
    hrPrev                 : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Binary source image (background zero, markers non-zero)
nMinMarkerArea	UDINT	Minimum area for the detection of markers (Smaller markers are ignored.)
nMaxMarkerArea	UDINT	Maximum area for the detection of markers (Larger markers are ignored.)
fMaxMarkerEccentricity	LREAL	Maximum eccentricity for the detection of markers (The eccentricity measures deviations from a round shape (eccentricity = 0: round object; eccentricity = 1.0: linear object).)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMarkerPositions	TcVnArray4_Point2_REAL [▶ 141]	Marker positions within their plane using world units (right-handed coordinate system)
aTransformationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns a 3-by-3 matrix for the transformation from image coordinates into the reference coordinate system
aMarkerImagePositions	TcVnArray4_Point2_REAL [▶ 141]	Returns the marker positions in image coordinates (left-handed coordinate system)

Return value

HRESULT [▶ 122]

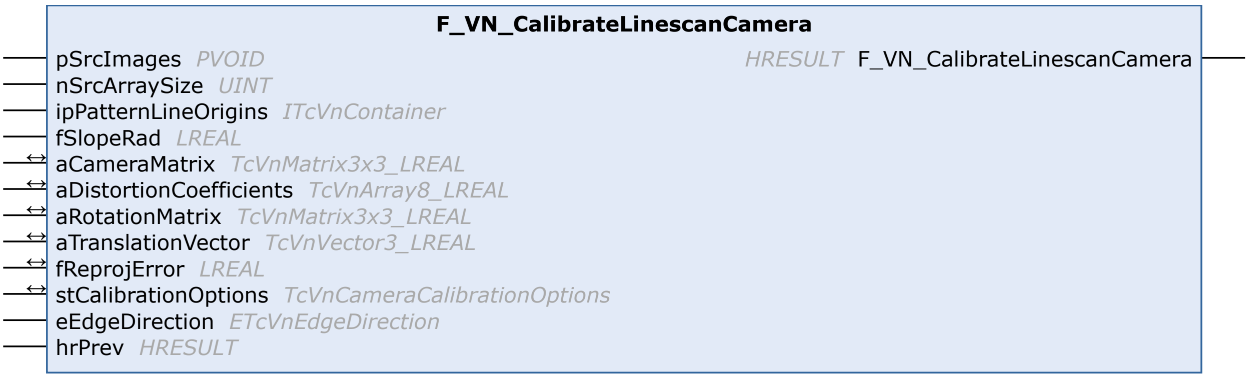
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.9 F_VN_CalibrateLinescanCamera



Compute the camera parameters (intrinsic + extrinsic) by evaluating images containing a line calibration pattern. The pattern must consist of alternating vertical and diagonal lines, with each diagonal having the same constant slope: |\|\|...\|. The results are only valid for the x-direction of a line scan image. To get accurate x-values for a transformation into world coordinates, the y-coordinate must be set to half the sensor width and so equal to the cy-value of the camera matrix. For other y-values, the results are approximated and can be inaccurate. The extrinsic parameters are computed for the first image in the array.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_CalibrateLinescanCamera : HRESULT
VAR_INPUT
    pSrcImages                : PVOID;
    nSrcArraySize             : UINT;
    ipPatternLineOrigins      : ITcVnContainer;
    fSlopeRad                 : LREAL;
END_VAR
VAR_IN_OUT
    aCameraMatrix             : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients : TcVnArray8_LREAL;
    aRotationMatrix           : TcVnMatrix3x3_LREAL;
    aTranslationVector        : TcVnVector3_LREAL;
    fReprojError              : LREAL;
    stCalibrationOptions      : TcVnCameraCalibrationOptions;
END_VAR
VAR_INPUT
    eEdgeDirection            : ETcVnEdgeDirection;
    hrPrev                    : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
pSrcImages	PVOID	Pointer to an array of 1 or more source images (1 channel). If the images have more than 1 row, each row must contain an image of the same pattern position.
nSrcArraySize	UINT	pSrcImages array size
ipPatternLineOrigins	ITcVnContainer [▶ 349]	X position of the pattern line origins (ContainerType_Vector_REAL, usually same origin for vertical and following diagonal line)
fSlopeRad	LREAL	Slope of the diagonal line in rad (> 0, < PI/2), relative to the vertical line (i.e., 0 would be a vertical line, PI/2 a horizontal line)
eEdgeDirection	ETcVnEdgeDirection [▶ 178]	Specification of the edge direction to search for
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns the camera matrix
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	Returns the lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns the rotation matrix
aTranslationVector	TcVnVector3_LREAL [▶ 141]	Returns the translation vector
fReprojError	LREAL	Returns the reprojection error
stCalibrationOptions	TcVnCameraCalibrationOptions [▶ 206]	Calibration options (recommended to set bFixAspectRatio, bFixPrincipalPoint, bZeroTangentDist)

Return value

HRESULT [▶ 122]

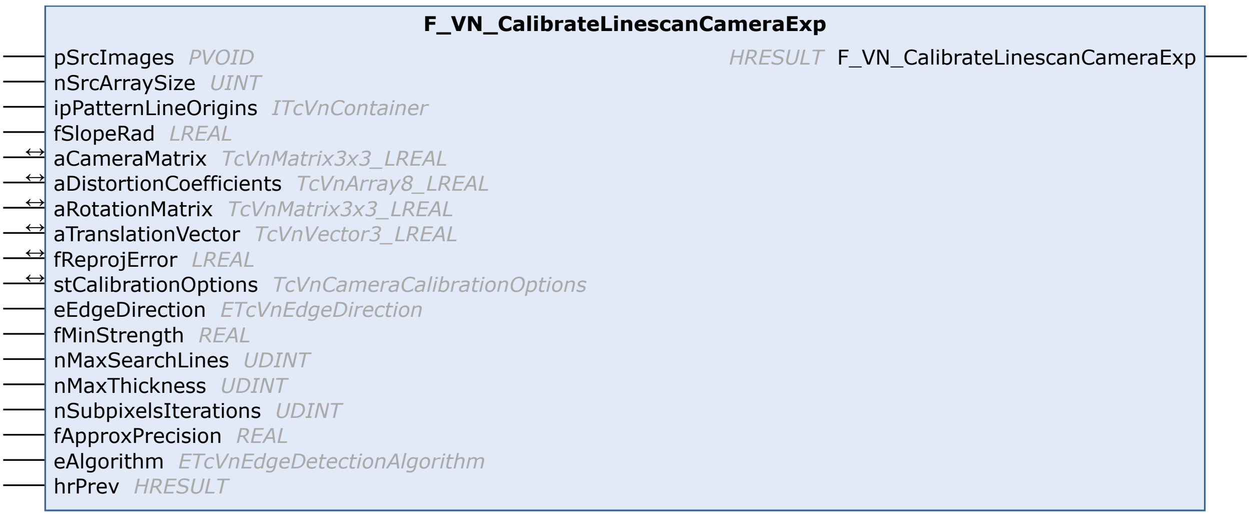
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.10 F_VN_CalibrateLinescanCameraExp



Compute the camera parameters (intrinsic + extrinsic) by evaluating images containing a line calibration pattern. The pattern must consist of alternating vertical and diagonal lines, with each diagonal having the same constant slope: |\|\|...\|. The results are only valid for the x-direction of a line scan image. To get accurate x-values for a transformation into world coordinates, the y-coordinate must be set to half the sensor width and so equal to the cy-value of the camera matrix. For other y-values, the results are approximated and can be inaccurate. The extrinsic parameters are computed for the first image in the array. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_CalibrateLinescanCameraExp : HRESULT
VAR_INPUT
    pSrcImages                 : PVOID;
    nSrcArraySize              : UINT;
    ipPatternLineOrigins       : ITcVnContainer;
    fSlopeRad                  : LREAL;
END_VAR
VAR_IN_OUT
    aCameraMatrix              : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients    : TcVnArray8_LREAL;
    aRotationMatrix            : TcVnMatrix3x3_LREAL;
    aTranslationVector         : TcVnVector3_LREAL;
    fReprojError               : LREAL;
    stCalibrationOptions       : TcVnCameraCalibrationOptions;
END_VAR
VAR_INPUT
    eEdgeDirection             : ETcVnEdgeDirection;
    fMinStrength               : REAL;
    nMaxSearchLines            : UDINT;
    nMaxThickness              : UDINT;
    nSubpixelsIterations       : UDINT;
    fApproxPrecision           : REAL;
    eAlgorithm                 : ETcVnEdgeDetectionAlgorithm;
    hrPrev                     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
pSrcImages	PVOID	Pointer to an array of 1 or more source images (1 channel). If the images have more than 1 row, each row must contain an image of the same pattern position.
nSrcArraySize	UINT	pSrcImages array size
ipPatternLineOrigins	ITcVnContainer [▶ 349]	X position of the pattern line origins (ContainerType_Vector_REAL, usually same origin for vertical and following diagonal line)
fSlopeRad	LREAL	Slope of the diagonal line in rad (> 0, < PI/2), relative to the vertical line (i.e., 0 would be a vertical line, PI/2 a horizontal line)
eEdgeDirection	ETcVnEdgeDirection [▶ 178]	Specification of the edge direction to search for
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
nMaxSearchLines	UDINT	Maximum number of search lines (equally distributed over the image height, at most 1 searchline per image row)
nMaxThickness	UDINT	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels
nSubpixelsIterations	UDINT	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)
fApproxPrecision	REAL	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▶ 177]	Selection of the edge detection algorithm
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns the camera matrix
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	Returns the lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns the rotation matrix
aTranslationVector	TcVnVector3_LREAL [▶ 141]	Returns the translation vector
fReprojError	LREAL	Returns the reprojection error
stCalibrationOptions	TcVnCameraCalibrationOptions [▶ 206]	Calibration options (recommended to set bFixAspectRatio, bFixPrincipalPoint, bZeroTangentDist)

Return value

HRESULT [▶ 122]

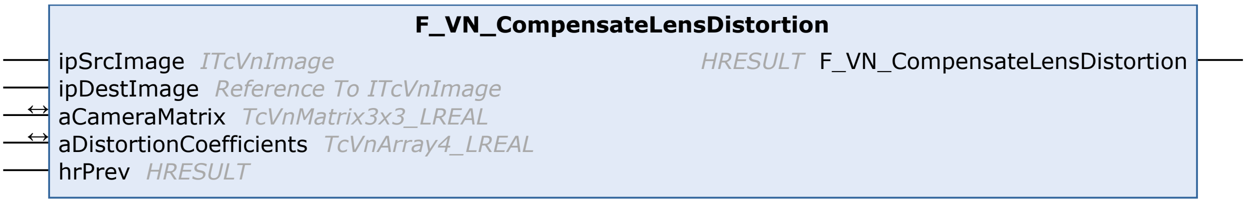
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.11 F_VN_CompensateLensDistortion



Transforms an image to compensate the lens distortion.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_CompensateLensDistortion : HRESULT
VAR_INPUT
    ipSrcImage              : ITcVnImage;
    ipDestImage             : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aCameraMatrix           : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients : TcVnArray4_LREAL;
END_VAR
VAR_INPUT
    hrPrev                  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray4_LREAL [▶ 141]	Lens distortion coefficients [k1, k2, p1, p2]

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.12 F_VN_CompensateLensDistortionExp1

Transforms an image to compensate the lens distortion. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_CompensateLensDistortionExp1 : HRESULT
VAR_INPUT
    ipSrcImage              : ITcVnImage;
    ipDestImage             : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aCameraMatrix           : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients : TcVnArray8_LREAL;
END_VAR
VAR_INPUT
    hrPrev                  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.13 F_VN_CompensateLensDistortionExp2

Transforms an image to compensate the lens distortion. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_CompensateLensDistortionExp2 : HRESULT
VAR_INPUT
    ipSrcImage                : ITcVnImage;
    ipDestImage               : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aCameraMatrix             : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients : TcVnArray8_LREAL;
    aNewCameraMatrix          : TcVnMatrix3x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                    : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aNewCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Allows additional scaling and shifting of the result image

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.14 F_VN_CompensateLensDistortionForPoints

Transforms point coordinates to compensate the lens distortion.

Syntax

Definition:

```
FUNCTION F_VN_CompensateLensDistortionForPoints : HRESULT
VAR_INPUT
    ipSrcPoints             : ITcVnContainer;
    ipDestPoints            : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    aCameraMatrix           : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients : TcVnArray4_LREAL;
END_VAR
VAR_INPUT
    hrPrev                  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPoints	ITcVnContainer [▶ 349]	Container with source point coordinates (ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_DINT)
ipDestPoints	Reference To ITcVnContainer [▶ 349]	Returns a container with the transformed point coordinates (ContainerType_Vector_TcVnPoint2_REAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray4_LREAL [▶ 141]	Lens distortion coefficients [k1, k2, p1, p2]

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.15 F_VN_CompensateLensDistortionForPointsExp1

Transforms point coordinates to compensate the lens distortion. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_CompensateLensDistortionForPointsExp1 : HRESULT
VAR_INPUT
    ipSrcPoints             : ITcVnContainer;
    ipDestPoints            : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    aCameraMatrix           : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients : TcVnArray8_LREAL;
END_VAR
VAR_INPUT
    hrPrev                  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPoints	ITcVnContainer [▶ 349]	Container with source point coordinates (ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_DINT)
ipDestPoints	Reference To ITcVnContainer [▶ 349]	Returns a container with the transformed point coordinates (ContainerType_Vector_TcVnPoint2_REAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]

Return value

HRESULT [▶ 122]

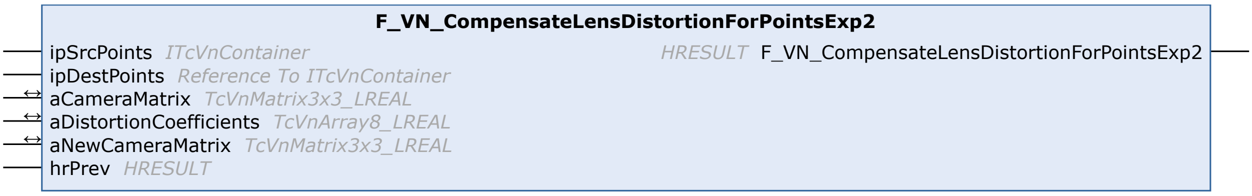
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.16 F_VN_CompensateLensDistortionForPointsExp2



Transforms point coordinates to compensate the lens distortion. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_CompensateLensDistortionForPointsExp2 : HRESULT
VAR_INPUT
    ipSrcPoints              : ITcVnContainer;
    ipDestPoints             : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    aCameraMatrix            : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients  : TcVnArray8_LREAL;
    aNewCameraMatrix         : TcVnMatrix3x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                   : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPoints	ITcVnContainer [▶ 349]	Container with source point coordinates (ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_DINT)
ipDestPoints	Reference To ITcVnContainer [▶ 349]	Returns a container with the transformed point coordinates (ContainerType_Vector_TcVnPoint2_REAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aNewCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Allows additional shifting of the result point coordinates

Return value

HRESULT [▶ 122]

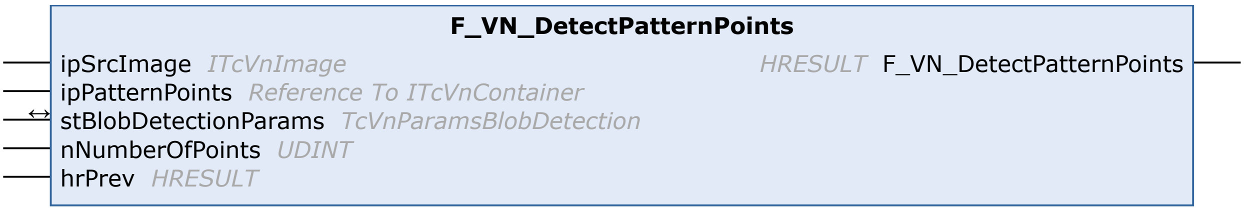
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.17 F_VN_DetectPatternPoints



Detects calibration pattern points (circles) within the provided image.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_DetectPatternPoints : HRESULT
VAR_INPUT
    ipSrcImage              : ITcVnImage;
    ipPatternPoints         : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    stBlobDetectionParams : TcVnParamsBlobDetection;
END_VAR
VAR_INPUT
```

```
    nNumberOfPoints         : UDINT;
    hrPrev                  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT, 1 channel)
ipPatternPoints	Reference To ITcVnContainer [▶ 349]	Returns the pattern point positions (ContainerType_Vector_TcVnPoint2_REAL; Non-zero interface pointers are reused.)
nNumberOfPoints	UDINT	Expected number of pattern points
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stBlobDetectionParams	TcVnParamsBlobDetection [▶ 213]	Parameters to detect and filter contours.

Return value

HRESULT [▶ 122]

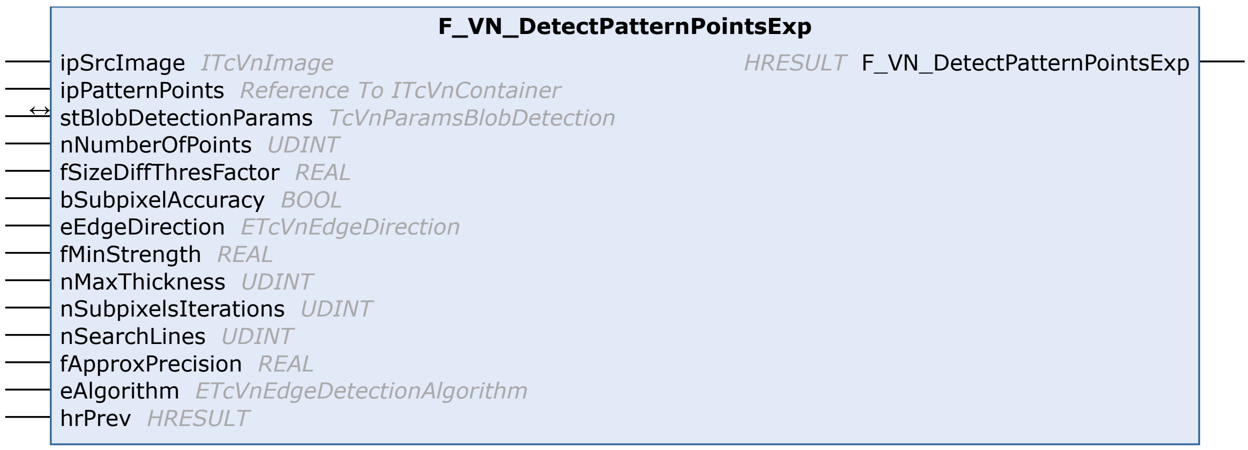
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.18 F_VN_DetectPatternPointsExp



Detects calibration pattern points (circles) within the provided image. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_DetectPatternPointsExp : HRESULT
VAR_INPUT
    ipSrcImage              : ITcVnImage;
    ipPatternPoints         : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    stBlobDetectionParams : TcVnParamsBlobDetection;
END_VAR
VAR_INPUT
    nNumberOfPoints         : UDINT;
    fSizeDiffThresFactor    : REAL;
    bSubpixelAccuracy       : BOOL;
    eEdgeDirection          : ETcVnEdgeDirection;
    fMinStrength            : REAL;
    nMaxThickness           : UDINT;
    nSubpixelsIterations    : UDINT;
    nSearchLines            : UDINT;
    fApproxPrecision        : REAL;
    eAlgorithm              : ETcVnEdgeDetectionAlgorithm;
    hrPrev                  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT, 1 channel)
ipPatternPoints	Reference To ITcVnContainer [▶ 349]	Returns the pattern point positions (ContainerType_Vector_TcVnPoint2_REAL; Non-zero interface pointers are reused.)
nNumberOfPoints	UDINT	Expected number of pattern points
fSizeDiffThresFactor	REAL	Threshold for filtering contours by average size
bSubpixelAccuracy	BOOL	If true, the pattern points are detected with subpixel accuracy
eEdgeDirection	ETcVnEdgeDirection [▶ 178]	Specification of the edge direction to search for (from center to outside circle)
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
nMaxThickness	UDINT	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels
nSubpixelsIterations	UDINT	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)
nSearchLines	UDINT	Specifies the amount of search lines, which are equally distributed in all directions (must be >= 8 and a multiple of 4)
fApproxPrecision	REAL	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▶ 177]	Selection of the edge detection algorithm
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stBlobDetectionParams	TcVnParamsBlobDetection [▶ 213]	Parameters to detect and filter contours.

Return value

HRESULT [▶ 122]

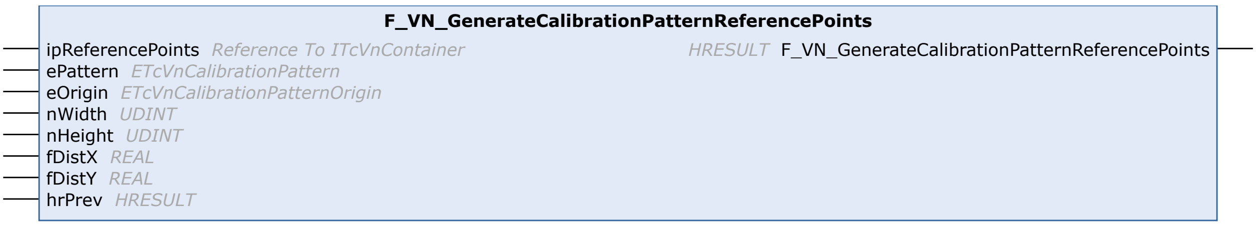
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.19 F_VN_GenerateCalibrationPatternReferencePoints



Generate reference points for common calibration patterns.

Syntax

Definition:

```
FUNCTION F_VN_GenerateCalibrationPatternReferencePoints : HRESULT
VAR_INPUT
    ipReferencePoints : Reference To ITcVnContainer;
    ePattern          : ETcVnCalibrationPattern;
    eOrigin           : ETcVnCalibrationPatternOrigin;
    nWidth            : UDINT;
    nHeight           : UDINT;
    fDistX            : REAL;
    fDistY            : REAL;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipReferencePo ints	Reference To ITcVnContainer [▶ 349]	Returns the generated reference points (ContainerType_Vector_TcVnPoint3_REAL)
ePattern	ETcVnCalibrationPattern [▶ 146]	Calibration pattern type
eOrigin	ETcVnCalibrationPattern Origin [▶ 147]	Calibration pattern origin
nWidth	UDINT	Pattern width, i.e. number of pattern points in x direction
nHeight	UDINT	Pattern height, i.e. number of pattern points in y direction
fDistX	REAL	Distance between pattern points in x direction
fDistY	REAL	Distance between pattern points in y direction
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

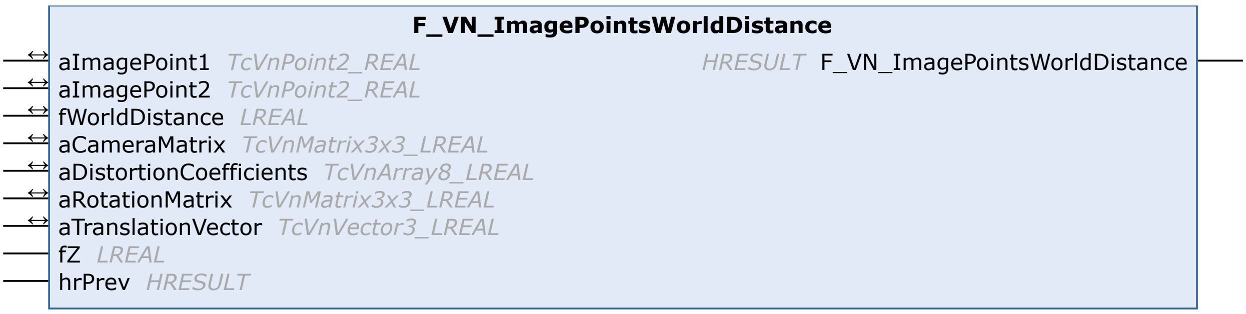
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.20 F_VN_ImagePointsWorldDistance



Computes the distance in the world coordinate system between two points (in the same world coordinate z layer) given in image coordinates.

Syntax

Definition:

```
FUNCTION F_VN_ImagePointsWorldDistance : HRESULT
VAR_IN_OUT
    aImagePoint1            : TcVnPoint2_REAL;
    aImagePoint2            : TcVnPoint2_REAL;
    fWorldDistance          : LREAL;
    aCameraMatrix           : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients : TcVnArray8_LREAL;
```

```
    aRotationMatrix          : TcVnMatrix3x3_LREAL;
    aTranslationVector       : TcVnVector3_LREAL;
END_VAR
VAR_INPUT
    fZ                       : LREAL;
    hrPrev                   : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fZ	LREAL	z coordinate (world coordinate system) of the given points (0 would be at (toplevel) calibration pattern)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aImagePoint1	TcVnPoint2_REAL [▸ 139]	First point in the image coordinate system
aImagePoint2	TcVnPoint2_REAL [▸ 139]	Second point in the image coordinate system
fWorldDistance	LREAL	Returns the distance between aImagePoint1 and aImagePoint2 in the world coordinate system
aCameraMatrix	TcVnMatrix3x3_LREAL [▸ 141]	3x3 camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray8_LREAL [▸ 141]	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▸ 141]	3x3 rotation matrix
aTranslationVector	TcVnVector3_LREAL [▸ 141]	Translation vector

Return value

HRESULT [▸ 122]

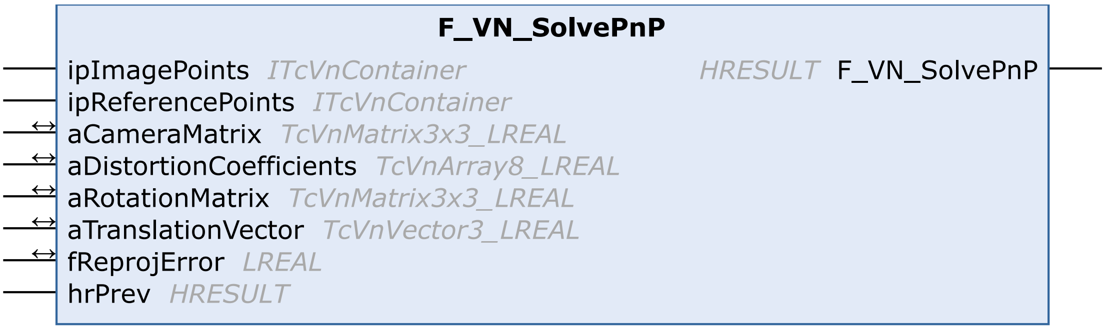
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.21 F_VN_SolvePnP



Computes the rotation matrix and translation vector from 3D-2D point correspondences.

Syntax

Definition:

```
FUNCTION F_VN_SolvePnP : HRESULT
VAR_INPUT
    ipImagePoints           : ITcVnContainer;
    ipReferencePoints       : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aCameraMatrix           : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients : TcVnArray8_LREAL;
    aRotationMatrix         : TcVnMatrix3x3_LREAL;
    aTranslationVector      : TcVnVector3_LREAL;
    fReprojError            : LREAL;
END_VAR
VAR_INPUT
    hrPrev                  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImagePoints	ITcVnContainer [▶ 349]	Image points (ContainerType_Vector_TcVnPoint2_REAL, requires at least 4 points in the same order as in ipReferencePoints)
ipReferencePoints	ITcVnContainer [▶ 349]	Reference world points (ContainerType_Vector_TcVnPoint3_REAL, same number of points as in ipImagePoints)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Camera matrix (e.g. computed with one of the CalibrateCamera functions)
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6] (e.g. computed with one of the CalibrateCamera functions)
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns the rotation matrix
aTranslationVector	TcVnVector3_LREAL [▶ 141]	Returns the translation vector
fReprojError	LREAL	Returns the reprojection error

Return value

HRESULT [▶ 122]

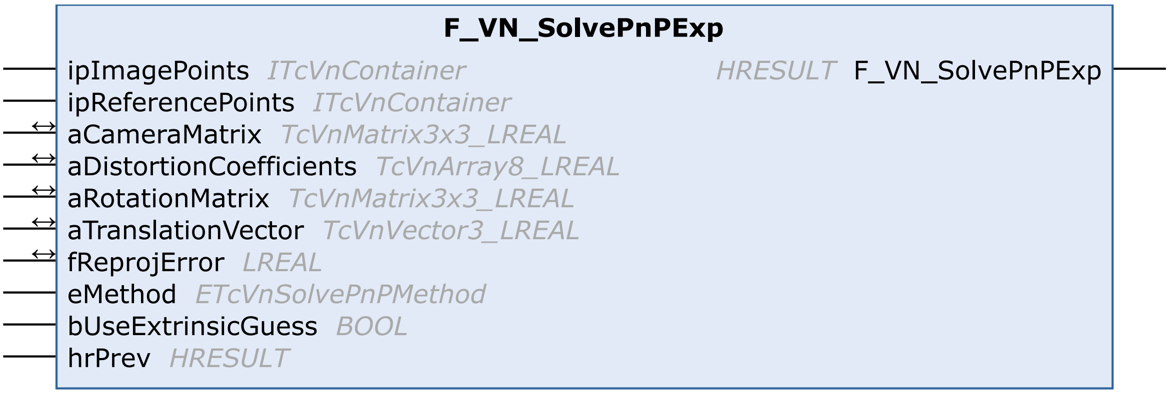
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.22 F_VN_SolvePnPExp



Computes the rotation matrix and translation vector from 3D-2D point correspondences. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_SolvePnPExp : HRESULT
VAR_INPUT
    ipImagePoints            : ITcVnContainer;
    ipReferencePoints        : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aCameraMatrix            : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients : TcVnArray8_LREAL;
    aRotationMatrix          : TcVnMatrix3x3_LREAL;
    aTranslationVector       : TcVnVector3_LREAL;
    fReprojError             : LREAL;
END_VAR
VAR_INPUT
    eMethod                  : ETcVnSolvePnPMethod;
    bUseExtrinsicGuess       : BOOL;
    hrPrev                   : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImagePoints	ITcVnContainer [▶ 349]	Image points (ContainerType_Vector_TcVnPoint2_REAL, requires at least 4 points in the same order as in ipReferencePoints)
ipReferencePoints	ITcVnContainer [▶ 349]	Reference world points (ContainerType_Vector_TcVnPoint3_REAL, same number of points as in ipImagePoints)
eMethod	ETcVnSolvePnPMethod [▶ 196]	Solve PnP method
bUseExtrinsicGuess	BOOL	If true, the provided rotation matrix and translation vector are used as an initial guess for further optimization (only supported for the SPM_ITERATIVE method).
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Camera matrix (e.g. computed with one of the CalibrateCamera functions)
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6] (e.g. computed with one of the CalibrateCamera functions)
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns the rotation matrix
aTranslationVector	TcVnVector3_LREAL [▶ 141]	Returns the translation vector
fReprojError	LREAL	Returns the reprojection error

Return value

HRESULT [▶ 122]

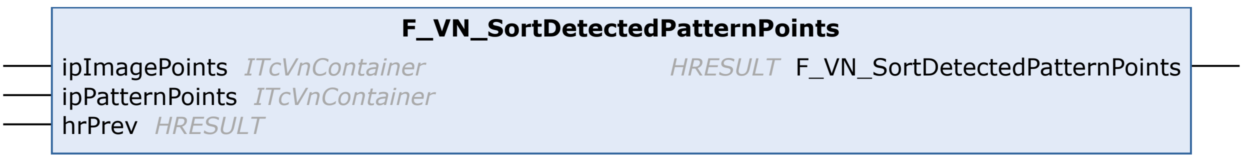
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.23 F_VN_SortDetectedPatternPoints



Sort the detected pattern points according to the relative positions of the calibration pattern points.

Syntax

Definition:

```
FUNCTION F_VN_SortDetectedPatternPoints : HRESULT
VAR_INPUT
    ipImagePoints   : ITcVnContainer;
    ipPatternPoints : ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImagePoints	ITcVnContainer [▶ 349]	Detected pattern point positions in the image coordinate system (will be sorted by this function, ContainerType_Vector_TcVnPoint2_REAL)
ipPatternPoints	ITcVnContainer [▶ 349]	The pattern points in the calibration pattern coordinate system, used as a reference to sort ipImagePoints (ContainerType_Vector_TcVnPoint3_REAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

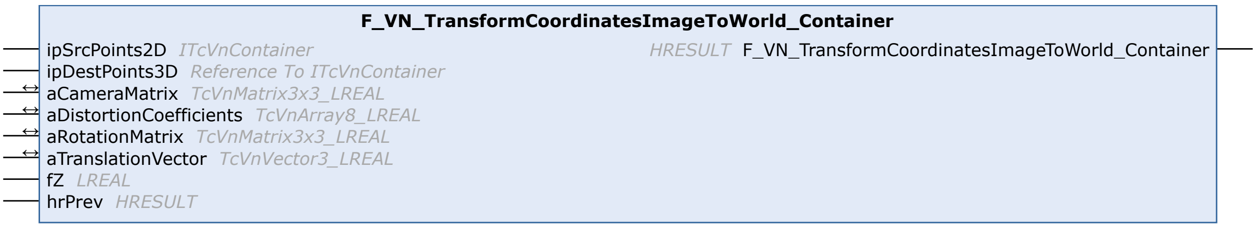
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.24 F_VN_TransformCoordinatesImageToWorld_Container



Transform 2D image point coordinates to 3D world coordinates (using intrinsic + extrinsic calibration results).

Syntax

Definition:

```
FUNCTION F_VN_TransformCoordinatesImageToWorld_Container : HRESULT
VAR_INPUT
    ipSrcPoints2D           : ITcVnContainer;
    ipDestPoints3D          : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    aCameraMatrix           : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients : TcVnArray8_LREAL;
    aRotationMatrix         : TcVnMatrix3x3_LREAL;
    aTranslationVector      : TcVnVector3_LREAL;
```

```
END_VAR
VAR_INPUT
    fZ                      : LREAL;
    hrPrev                  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPoints2D	ITcVnContainer [▶ 349]	Container with 2D source points (TcVnPoint2_DINT or TcVnPoint2_REAL or TcVnPoint2_LREAL)
ipDestPoints3D	Reference To ITcVnContainer [▶ 349]	Returns the transformed 3D points (TcVnPoint3_LREAL)
fZ	LREAL	z coordinate (world coordinate system) of the transformed points (0 would be at (toplevel) calibration pattern)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	3x3 camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	3x3 rotation matrix
aTranslationVector	TcVnVector3_LREAL [▶ 141]	Translation vector

Return value

HRESULT [▶ 122]

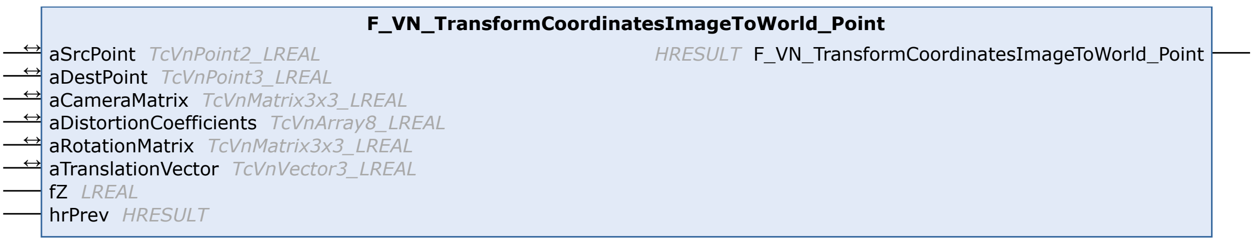
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.25 F_VN_TransformCoordinatesImageToWorld_Point



Transform 2D image point coordinate to 3D world coordinate (using intrinsic + extrinsic calibration results).

Syntax

Definition:

```
FUNCTION F_VN_TransformCoordinatesImageToWorld_Point : HRESULT
VAR_IN_OUT
    aSrcPoint                : TcVnPoint2_LREAL;
    aDestPoint               : TcVnPoint3_LREAL;
    aCameraMatrix            : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients  : TcVnArray8_LREAL;
    aRotationMatrix          : TcVnMatrix3x3_LREAL;
    aTranslationVector       : TcVnVector3_LREAL;
END_VAR
VAR_INPUT
    fZ                       : LREAL;
    hrPrev                   : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fZ	LREAL	z coordinate (world coordinate system) of the transformed points (0 would be at (toplevel) calibration pattern)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aSrcPoint	TcVnPoint2_LREAL [▶ 139]	Point in the image coordinate system
aDestPoint	TcVnPoint3_LREAL [▶ 139]	Returns the point in the world coordinate system
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	3x3 camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	3x3 rotation matrix
aTranslationVector	TcVnVector3_LREAL [▶ 141]	Translation vector

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.26 F_VN_TransformCoordinatesPlanar_Container

F_VN_TransformCoordinatesPlanar_Container

ipSrcPoints *ITcVnContainer* — *HRESULT* F_VN_TransformCoordinatesPlanar_Container
ipDestPoints *Reference To ITcVnContainer*
aTransformationMatrix *TcVnMatrix3x3_LREAL*
hrPrev *HRESULT*

Compute real-world coordinates for given image points or vice versa.

Syntax

Definition:

```
FUNCTION F_VN_TransformCoordinatesPlanar_Container : HRESULT
VAR_INPUT
    ipSrcPoints           : ITcVnContainer;
    ipDestPoints          : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    aTransformationMatrix : TcVnMatrix3x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPoints	ITcVnContainer [▶ 349]	Container with 2D source points (TcVnPoint2_DINT or TcVnPoint2_REAL or TcVnPoint2_LREAL)
ipDestPoints	Reference To ITcVnContainer [▶ 349]	Returns the transformed 2D points (TcVnPoint2_LREAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Transformation matrix obtained with VnCalibrateCameraPlanar() or its inverse matrix obtained with VnInvertMatrix3x3()

Return value

HRESULT [▶ 122]

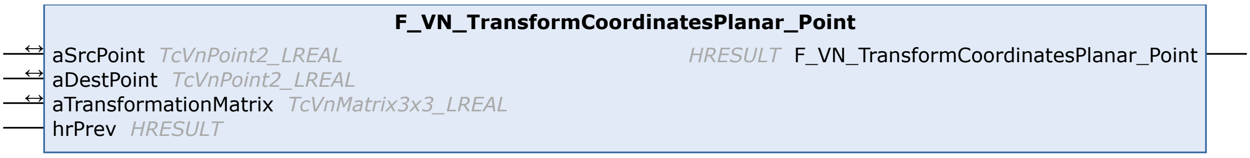
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.27 F_VN_TransformCoordinatesPlanar_Point



Compute real-world coordinates for a given image point or vice versa.

Syntax

Definition:

```
FUNCTION F_VN_TransformCoordinatesPlanar_Point : HRESULT
VAR_IN_OUT
    aSrcPoint             : TcVnPoint2_LREAL;
    aDestPoint            : TcVnPoint2_LREAL;
    aTransformationMatrix : TcVnMatrix3x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aSrcPoint	TcVnPoint2_LREAL [▶ 139]	Point in the source coordinate system
aDestPoint	TcVnPoint2_LREAL [▶ 139]	Returns point in the destination coordinate system
aTransformationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Transformation matrix obtained with VnCalibrateCameraPlanar() or its inverse matrix obtained with VnInvertMatrix3x3()

Return value

HRESULT [▶ 122]

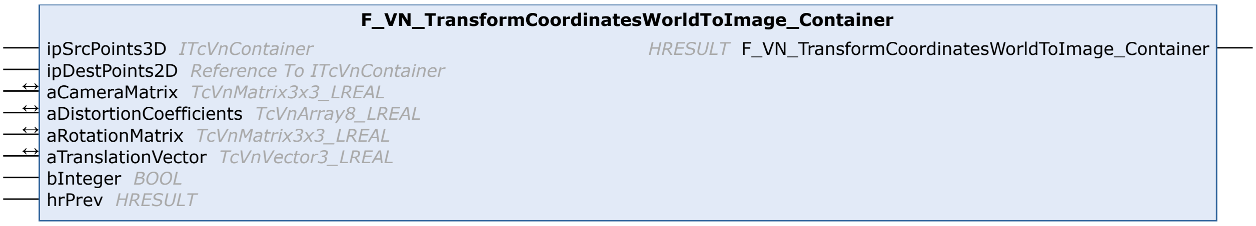
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.28 F_VN_TransformCoordinatesWorldToImage_Container



Transform 3D world coordinates to 2D image coordinates (using intrinsic + extrinsic calibration results).

Syntax

Definition:

```
FUNCTION F_VN_TransformCoordinatesWorldToImage_Container : HRESULT
VAR_INPUT
    ipSrcPoints3D           : ITcVnContainer;
    ipDestPoints2D          : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    aCameraMatrix           : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients : TcVnArray8_LREAL;
    aRotationMatrix         : TcVnMatrix3x3_LREAL;
    aTranslationVector      : TcVnVector3_LREAL;
END_VAR
VAR_INPUT
    bInteger                : BOOL;
    hrPrev                  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPoints3D	ITcVnContainer [▶ 349]	Container with 3D source points (TcVnPoint3_LREAL)
ipDestPoints2D	Reference To ITcVnContainer [▶ 349]	Returns the transformed 2D points (TcVnPoint2_REAL or TcVnPoint2_DINT, depending on bInteger)
bInteger	BOOL	Selects whether the function returns a container of TcVnPoint2_DINT or TcVnPoint2_REAL
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	3x3 camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	3x3 rotation matrix
aTranslationVector	TcVnVector3_LREAL [▶ 141]	Translation vector

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.1.29 F_VN_TransformCoordinatesWorldToImage_Point

Transform 3D world coordinates to 2D image coordinates (using intrinsic + extrinsic calibration results).

Syntax

Definition:

```
FUNCTION F_VN_TransformCoordinatesWorldToImage_Point : HRESULT
VAR_IN_OUT
    aSrcPoint               : TcVnPoint3_LREAL;
    aDestPoint              : TcVnPoint2_REAL;
    aCameraMatrix           : TcVnMatrix3x3_LREAL;
    aDistortionCoefficients : TcVnArray8_LREAL;
    aRotationMatrix         : TcVnMatrix3x3_LREAL;
    aTranslationVector      : TcVnVector3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aSrcPoint	TcVnPoint3_LREAL [▶ 139]	3D source point (world coordinate system)
aDestPoint	TcVnPoint2_REAL [▶ 139]	Returns the transformed 2D point (image coordinate system)
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	3x3 camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	3x3 rotation matrix
aTranslationVector	TcVnVector3_LREAL [▶ 141]	Translation vector

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.2 F_VN_AlignRotatedImageRegion

F_VN_AlignRotatedImageRegion

ipSrcImage *ITcVnImage* — *HRESULT* F_VN_AlignRotatedImageRegion
ipDestImage *Reference To ITcVnImage*
stRegion *TcVnRotatedRectangle*
hrPrev *HRESULT*

Copies the provided image region and aligns it to the image axes.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_AlignRotatedImageRegion : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    stRegion    : TcVnRotatedRectangle;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stRegion	TcVnRotatedRectangle [▸ 225]	Image region to be aligned

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.3 F_VN_AlignRotatedImageRegionExp

Copies the provided image region and aligns it to the image axes. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_AlignRotatedImageRegionExp : HRESULT
VAR_INPUT
    ipSrcImage           : ITcVnImage;
    ipDestImage          : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    stRegion             : TcVnRotatedRectangle;
END_VAR
VAR_INPUT
    eInterpolationType   : ETcVnInterpolationType;
    eBorderInterpolation : ETcVnBorderInterpolationMethod;
END_VAR
VAR_IN_OUT
    aBorderValue         : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev               : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate image will be created if required.)
eInterpolationType	ETcVnInterpolationType [▶ 185]	Interpolation method
eBorderInterpolation	ETcVnBorderInterpolationMethod [▶ 145]	Border interpolation method (ISOLATED not supported)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stRegion	TcVnRotatedRectangle [▶ 225]	Image region to be aligned
aBorderValue	TcVnVector4_LREAL [▶ 141]	Border value, if CONSTANT is used

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.4 F_VN_ApplyRotationToAffineTransformation

Apply a rotation to an existing 2D affine transformation matrix.

Syntax

Definition:

```
FUNCTION F_VN_ApplyRotationToAffineTransformation : HRESULT
VAR_IN_OUT
    aTransformationMatrix : TcVnMatrix2x3_LREAL;
END_VAR
VAR_INPUT
    fAngle                : LREAL;
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fAngle	LREAL	Angle in radians (positive means counter-clockwise)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 141]	Affine transformation matrix to which the rotation is applied

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.5 F_VN_ApplyRotationToAffineTransformationExp

Apply a rotation to an existing 2D affine transformation matrix. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ApplyRotationToAffineTransformationExp : HRESULT
VAR_IN_OUT
    aTransformationMatrix : TcVnMatrix2x3_LREAL;
END_VAR
VAR_INPUT
    fAngle                : LREAL;
    bUsePreMultiplication : BOOL;
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fAngle	LREAL	Angle in radians (positive means counter-clockwise)
bUsePreMultip lication	BOOL	Select if pre- or post-multiplication should be used
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformati onMatrix	TcVnMatrix2x3_LREAL [▸ 141]	Affine transformation matrix to which the rotation is applied

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.6 F_VN_ApplyScalingToAffineTransformation

Apply a scaling to an existing 2D affine transformation matrix.

Syntax

Definition:

```
FUNCTION F_VN_ApplyScalingToAffineTransformation : HRESULT
VAR_IN_OUT
    aTransformationMatrix : TcVnMatrix2x3_LREAL;
END_VAR
VAR_INPUT
    fScaleX               : LREAL;
    fScaleY               : LREAL;
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fScaleX	LREAL	Scaling factor in x direction
fScaleY	LREAL	Scaling factor in y direction
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 141]	Affine transformation matrix to which the scaling is applied

Return value

HRESULT [▶ 122]

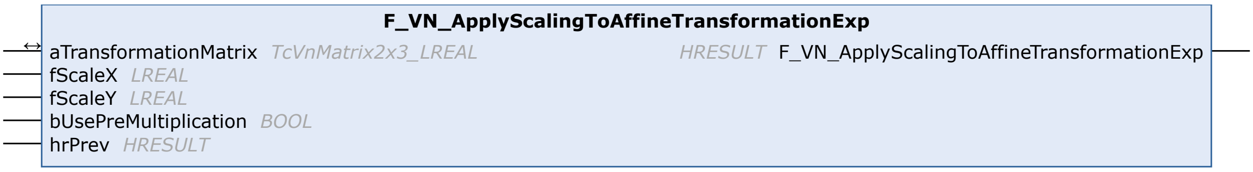
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.7 F_VN_ApplyScalingToAffineTransformationExp



Apply a scaling to an existing 2D affine transformation matrix. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ApplyScalingToAffineTransformationExp : HRESULT
VAR_IN_OUT
    aTransformationMatrix : TcVnMatrix2x3_LREAL;
END_VAR
VAR_INPUT
    fScaleX               : LREAL;
    fScaleY               : LREAL;
    bUsePreMultiplication : BOOL;
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fScaleX	LREAL	Scaling factor in x direction
fScaleY	LREAL	Scaling factor in y direction
bUsePreMultiplication	BOOL	Select if pre- or post-multiplication should be used
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 141]	Affine transformation matrix to which the scaling is applied

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.8 F_VN_ApplyTranslationToAffineTransformation

Apply a translation to an existing 2D affine transformation matrix.

Syntax

Definition:

```
FUNCTION F_VN_ApplyTranslationToAffineTransformation : HRESULT
VAR_IN_OUT
    aTransformationMatrix : TcVnMatrix2x3_LREAL;
END_VAR
VAR_INPUT
    fDeltaX               : LREAL;
    fDeltaY               : LREAL;
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fDeltaX	LREAL	Translation in x direction
fDeltaY	LREAL	Translation in y direction
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformationMatrix	TcVnMatrix2x3_LREAL [▸ 141]	Affine transformation matrix to which the translation is applied

Return value

HRESULT [▸ 122]

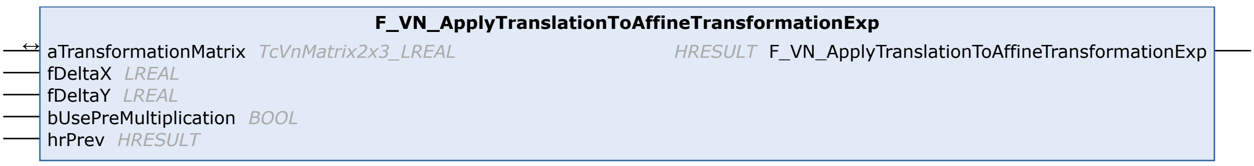
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.9 F_VN_ApplyTranslationToAffineTransformationExp



Apply a translation to an existing 2D affine transformation matrix. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ApplyTranslationToAffineTransformationExp : HRESULT
VAR_IN_OUT
    aTransformationMatrix : TcVnMatrix2x3_LREAL;
END_VAR
VAR_INPUT
    fDeltaX               : LREAL;
    fDeltaY               : LREAL;
    bUsePreMultiplication : BOOL;
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fDeltaX	LREAL	Translation in x direction
fDeltaY	LREAL	Translation in y direction
bUsePreMultiplication	BOOL	Select if pre- or post-multiplication should be used
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 141]	Affine transformation matrix to which the translation is applied

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.10 F_VN_ApplyYAxisInversionToAffineTransformation

Apply an inversion of the y-axis direction to an existing 2D affine transformation matrix.

Syntax

Definition:

```
FUNCTION F_VN_ApplyYAxisInversionToAffineTransformation : HRESULT
VAR_IN_OUT
    aTransformationMatrix : TcVnMatrix2x3_LREAL;
END_VAR
```

```
VAR_INPUT
    hrPrev                  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 141]	Affine transformation matrix to which the inversion of the y-axis direction is applied

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.11 F_VN_ApplyYAxisInversionToAffineTransformationExp

Apply an inversion of the y-axis direction to an existing 2D affine transformation matrix. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ApplyYAxisInversionToAffineTransformationExp : HRESULT
VAR_IN_OUT
    aTransformationMatrix : TcVnMatrix2x3_LREAL;
END_VAR
VAR_INPUT
    bUsePreMultiplication : BOOL;
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
bUsePreMultiplication	BOOL	Select if pre- or post-multiplication should be used
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 141]	Affine transformation matrix to which the inversion of the y-axis direction is applied

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.12 F_VN_ConvertCartesianToPolarAngleImage

Converts cartesian coordinates (x, y) to polar angle.

Syntax

Definition:

```
FUNCTION F_VN_ConvertCartesianToPolarAngleImage : HRESULT
VAR_INPUT
    ipSrcImageX    : ITcVnImage;
    ipSrcImageY    : ITcVnImage;
    ipDestImageAng : Reference To ITcVnImage;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImageX	ITcVnImage [▶ 390]	Source image containing the x values (elements of type REAL or LREAL)
ipSrcImageY	ITcVnImage [▶ 390]	Source image containing the y values (Same element type as ipSrcImageX)
ipDestImageAng	Reference To ITcVnImage [▶ 390]	Destination image containing the angles (Same element type as source images. An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.13 F_VN_ConvertCartesianToPolarAngleImageExp

Converts cartesian coordinates (x, y) to polar angle. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ConvertCartesianToPolarAngleImageExp : HRESULT
VAR_INPUT
    ipSrcImageX      : ITcVnImage;
    ipSrcImageY      : ITcVnImage;
    ipDestImageAng   : Reference To ITcVnImage;
    bAngleInDegrees  : BOOL;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImageX	ITcVnImage [▶ 390]	Source image containing the x values (elements of type REAL or LREAL)
ipSrcImageY	ITcVnImage [▶ 390]	Source image containing the y values (Same element type as ipSrcImageX)
ipDestImageAng	Reference To ITcVnImage [▶ 390]	Destination image containing the angles (Same element type as source images. An appropriate destination image will be created if required.)
bAngleInDegrees	BOOL	Specifys, if the angles should be in degrees or radians
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.14 F_VN_ConvertCartesianToPolarAngles

Converts cartesian coordinates (x, y) to polar angle.

Syntax

Definition:

```
FUNCTION F_VN_ConvertCartesianToPolarAngles : HRESULT
VAR_INPUT
    ipSrcPointsCart : ITcVnContainer;
    ipDestAngles    : Reference To ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPointsCa rt	ITcVnContainer [▶ 349]	Container with cartesian points (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_LREAL)
ipDestAngles	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the angles (ContainerType_Vector_REAL, if source points are of type TcVnPoint2_DINT or TcVnPoint2_REAL; else ContainerType_Vector_LREAL. Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

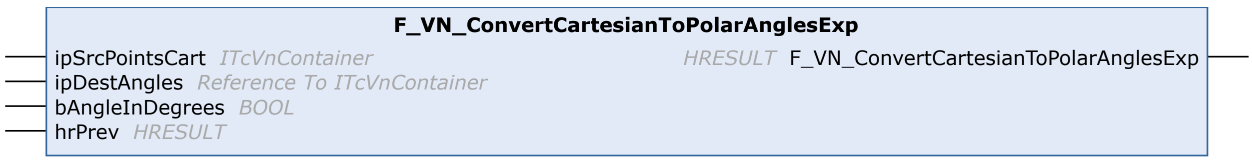
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.15 F_VN_ConvertCartesianToPolarAnglesExp



Converts cartesian coordinates (x, y) to polar angle. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ConvertCartesianToPolarAnglesExp : HRESULT
VAR_INPUT
    ipSrcPointsCart : ITcVnContainer;
    ipDestAngles    : Reference To ITcVnContainer;
    bAngleInDegrees : BOOL;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPointsCart	ITcVnContainer [▶ 349]	Container with cartesian points (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_LREAL)
ipDestAngles	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the angles (ContainerType_Vector_REAL, if source points are of type TcVnPoint2_DINT or TcVnPoint2_REAL; else ContainerType_Vector_LREAL. Non-zero interface pointers are reused.)
bAngleInDegrees	BOOL	Specify, if the angles should be in degrees or radians
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

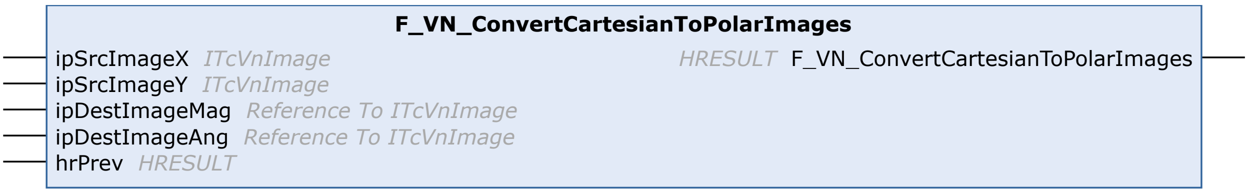
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.16 F_VN_ConvertCartesianToPolarImages



Converts cartesian coordinates (x, y) to polar coordinates (magnitude, angle).

Syntax

Definition:

```
FUNCTION F_VN_ConvertCartesianToPolarImages : HRESULT
VAR_INPUT
    ipSrcImageX     : ITcVnImage;
    ipSrcImageY     : ITcVnImage;
    ipDestImageMag : Reference To ITcVnImage;
    ipDestImageAng : Reference To ITcVnImage;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImageX	ITcVnImage [▶ 390]	Source image containing the x values (elements of type REAL or LREAL)
ipSrcImageY	ITcVnImage [▶ 390]	Source image containing the y values (Same element type as ipSrcImageX)
ipDestImageMag	Reference To ITcVnImage [▶ 390]	Destination image containing the magnitudes (Same element type as source images. An appropriate destination image will be created if required.)
ipDestImageAng	Reference To ITcVnImage [▶ 390]	Destination image containing the angles (Same element type as source images. An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.17 F_VN_ConvertCartesianToPolarImagesExp

Converts cartesian coordinates (x, y) to polar coordinates (magnitude, angle). (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ConvertCartesianToPolarImagesExp : HRESULT
VAR_INPUT
    ipSrcImageX      : ITcVnImage;
    ipSrcImageY      : ITcVnImage;
    ipDestImageMag   : Reference To ITcVnImage;
    ipDestImageAng   : Reference To ITcVnImage;
    bAngleInDegrees : BOOL;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImageX	ITcVnImage [▶ 390]	Source image containing the x values (elements of type REAL or LREAL)
ipSrcImageY	ITcVnImage [▶ 390]	Source image containing the y values (Same element type as ipSrcImageX)
ipDestImageMag	Reference To ITcVnImage [▶ 390]	Destination image containing the magnitudes (Same element type as source images. An appropriate destination image will be created if required.)
ipDestImageAng	Reference To ITcVnImage [▶ 390]	Destination image containing the angles (Same element type as source images. An appropriate destination image will be created if required.)
bAngleInDegrees	BOOL	Specifys, if the angles should be in degrees or radians
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.18 F_VN_ConvertCartesianToPolarMagnitudeImage

Converts cartesian coordinates (x, y) to polar magnitude.

Syntax

Definition:

```
FUNCTION F_VN_ConvertCartesianToPolarMagnitudeImage : HRESULT
VAR_INPUT
    ipSrcImageX     : ITcVnImage;
    ipSrcImageY     : ITcVnImage;
    ipDestImageMag : Reference To ITcVnImage;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImageX	ITcVnImage [▶ 390]	Source image containing the x values (elements of type REAL or LREAL)
ipSrcImageY	ITcVnImage [▶ 390]	Source image containing the y values (Same element type as ipSrcImageX)
ipDestImageMag	Reference To ITcVnImage [▶ 390]	Destination image containing the magnitudes (Same element type as source images. An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

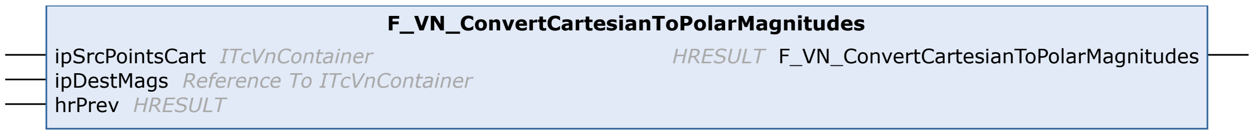
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.19 F_VN_ConvertCartesianToPolarMagnitudes



Converts cartesian coordinates (x, y) to polar magnitude.

Syntax

Definition:

```
FUNCTION F_VN_ConvertCartesianToPolarMagnitudes : HRESULT
VAR_INPUT
    ipSrcPointsCart : ITcVnContainer;
    ipDestMags      : Reference To ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPointsCart	ITcVnContainer [▶ 349]	Container with cartesian points (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_LREAL)
ipDestMags	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the magnitudes (ContainerType_Vector_REAL, if source points are of type TcVnPoint2_DINT or TcVnPoint2_REAL; else ContainerType_Vector_LREAL. Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.20 F_VN_ConvertCartesianToPolarPoints

Converts cartesian coordinates (x, y) to polar coordinates (magnitude, angle).

Syntax

Definition:

```
FUNCTION F_VN_ConvertCartesianToPolarPoints : HRESULT
VAR_INPUT
    ipSrcPointsCart   : ITcVnContainer;
    ipDestPointsPolar : Reference To ITcVnContainer;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPointsCa rt	ITcVnContainer [▶ 349]	Container with cartesian points (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_LREAL)
ipDestPointsP olar	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the converted points in polar coordinates (ContainerType_Vector_TcVnPoint2_REAL, if source points are of type TcVnPoint2_DINT or TcVnPoint2_REAL; else ContainerType_Vector_TcVnPoint2_LREAL. Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.21 F_VN_ConvertCartesianToPolarPointsExp

Converts cartesian coordinates (x, y) to polar coordinates (magnitude, angle). (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ConvertCartesianToPolarPointsExp : HRESULT
VAR_INPUT
    ipSrcPointsCart   : ITcVnContainer;
    ipDestPointsPolar : Reference To ITcVnContainer;
    bAngleInDegrees   : BOOL;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPointsCart	ITcVnContainer [▶ 349]	Container with cartesian points (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_LREAL)
ipDestPointsPolar	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the converted points in polar coordinates (ContainerType_Vector_TcVnPoint2_REAL, if source points are of type TcVnPoint2_DINT or TcVnPoint2_REAL; else ContainerType_Vector_TcVnPoint2_LREAL. Non-zero interface pointers are reused.)
bAngleInDegrees	BOOL	Specifys, if the angles should be in degrees or radians
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

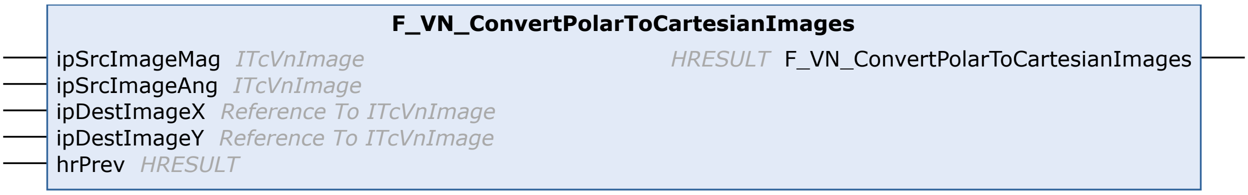
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.22 F_VN_ConvertPolarToCartesianImages



Converts polar coordinates (magnitude, angle) to cartesian coordinates (x, y).

Syntax

Definition:

```
FUNCTION F_VN_ConvertPolarToCartesianImages : HRESULT
VAR_INPUT
    ipSrcImageMag  : ITcVnImage;
    ipSrcImageAng  : ITcVnImage;
    ipDestImageX   : Reference To ITcVnImage;
    ipDestImageY   : Reference To ITcVnImage;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImageMag	ITcVnImage [▸ 390]	Source image containing the magnitudes (elements of type REAL or LREAL)
ipSrcImageAng	ITcVnImage [▸ 390]	Source image containing the angles (Same element type as ipSrcImageMag)
ipDestImageX	Reference To ITcVnImage [▸ 390]	Destination image containing the x values (Same element type as source images. An appropriate destination image will be created if required.)
ipDestImageY	Reference To ITcVnImage [▸ 390]	Destination image containing the y values (Same element type as source images. An appropriate destination image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

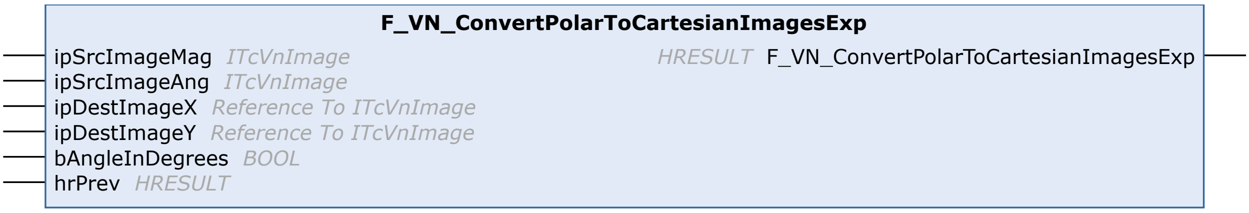
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.23 F_VN_ConvertPolarToCartesianImagesExp



Converts polar coordinates (magnitude, angle) to cartesian coordinates (x, y). (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ConvertPolarToCartesianImagesExp : HRESULT
VAR_INPUT
    ipSrcImageMag   : ITcVnImage;
    ipSrcImageAng   : ITcVnImage;
    ipDestImageX    : Reference To ITcVnImage;
    ipDestImageY    : Reference To ITcVnImage;
    bAngleInDegrees : BOOL;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImageMag	ITcVnImage [▶ 390]	Source image containing the magnitudes (elements of type REAL or LREAL)
ipSrcImageAng	ITcVnImage [▶ 390]	Source image containing the angles (Same element type as ipSrcImageMag)
ipDestImageX	Reference To ITcVnImage [▶ 390]	Destination image containing the x values (Same element type as source images. An appropriate destination image will be created if required.)
ipDestImageY	Reference To ITcVnImage [▶ 390]	Destination image containing the y values (Same element type as source images. An appropriate destination image will be created if required.)
bAngleInDegrees	BOOL	Specifys, if the angles are in degrees or radians
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

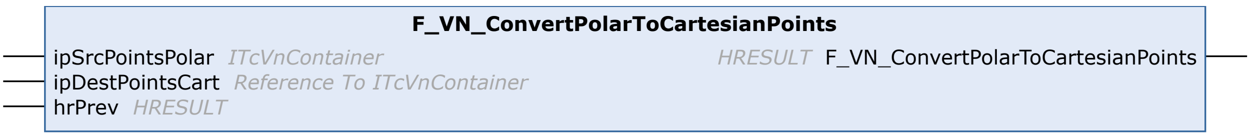
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.24 F_VN_ConvertPolarToCartesianPoints



Converts polar coordinates (magnitude, angle) to cartesian coordinates (x, y).

Syntax

Definition:

```
FUNCTION F_VN_ConvertPolarToCartesianPoints : HRESULT
VAR_INPUT
    ipSrcPointsPolar : ITcVnContainer;
    ipDestPointsCart : Reference To ITcVnContainer;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPointsPolar	ITcVnContainer [▶ 349]	Container with polar coordinates (ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_LREAL)
ipDestPointsCart	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the converted points in cartesian coordinates (same type ID as ipSrcPointsPolar. Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.25 F_VN_ConvertPolarToCartesianPointsExp

Converts polar coordinates (magnitude, angle) to cartesian coordinates (x, y). (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ConvertPolarToCartesianPointsExp : HRESULT
VAR_INPUT
    ipSrcPointsPolar : ITcVnContainer;
    ipDestPointsCart : Reference To ITcVnContainer;
    bAngleInDegrees  : BOOL;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPointsPolar	ITcVnContainer [▶ 349]	Container with polar coordinates (ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_LREAL)
ipDestPointsCart	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the converted points in cartesian coordinates (same type ID as ipSrcPointsPolar. Non-zero interface pointers are reused.)
bAngleInDegrees	BOOL	Specifys, if the angles are in degrees or radians
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

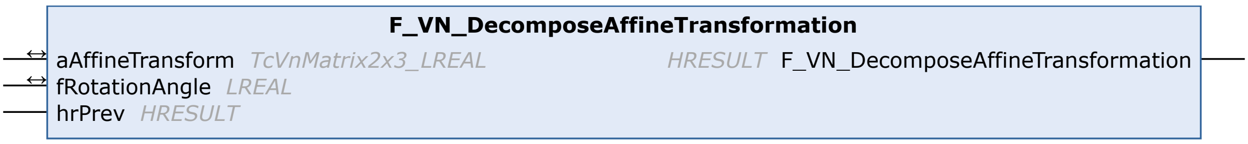
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.26 F_VN_DecomposeAffineTransformation



Decompose an affine transformation matrix and return the rotation angle (clockwise) in radian. The decomposition method assumes the following order of transformation: Translation, Rotation, Scale, and Skew.

Syntax

Definition:

```
FUNCTION F_VN_DecomposeAffineTransformation : HRESULT
VAR_IN_OUT
    aAffineTransform : TcVnMatrix2x3_LREAL;
    fRotationAngle   : LREAL;
END_VAR
VAR_INPUT
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aAffineTransform	TcVnMatrix2x3_LREAL [▶ 141]	Affine transformation matrix
fRotationAngle	LREAL	Rotation angle (radian) in the clockwise direction. The rotation center is the origin point (0,0)

Return value

HRESULT [▶ 122]

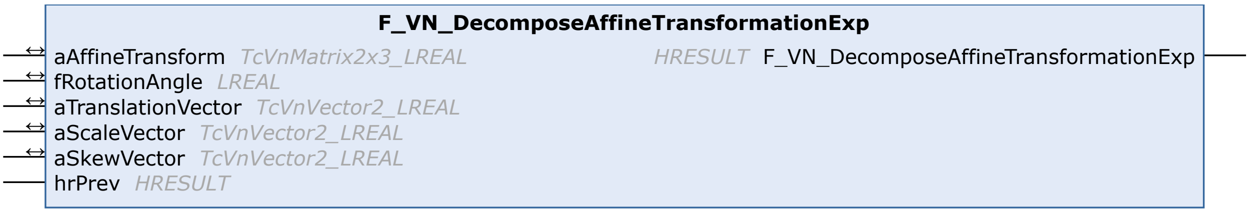
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.27 F_VN_DecomposeAffineTransformationExp



Decompose an affine transformation matrix and return the rotation angle (clockwise) in radian, translation vector (pixels), scale, and skew vectors (angle in radian). The decomposition method assumes the following order of transformation: Translation, Rotation, Scale, and Skew.

Syntax

Definition:

```
FUNCTION F_VN_DecomposeAffineTransformationExp : HRESULT
VAR_IN_OUT
    aAffineTransform   : TcVnMatrix2x3_LREAL;
    fRotationAngle     : LREAL;
    aTranslationVector : TcVnVector2_LREAL;
    aScaleVector       : TcVnVector2_LREAL;
    aSkewVector        : TcVnVector2_LREAL;
END_VAR
VAR_INPUT
    hrPrev             : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aAffineTransform	TcVnMatrix2x3_LREAL [▶ 141]	Affine transformation matrix
fRotationAngle	LREAL	Rotation angle (radian) in the clockwise direction. The rotation center is the origin point (0,0)
aTranslationVector	TcVnVector2_LREAL [▶ 141]	2D translation vector. The values are in pixels
aScaleVector	TcVnVector2_LREAL [▶ 141]	2D scale vector
aSkewVector	TcVnVector2_LREAL [▶ 141]	2D skew vector. The values are the skew angles in x and y directions in radian

Return value

HRESULT [▶ 122]

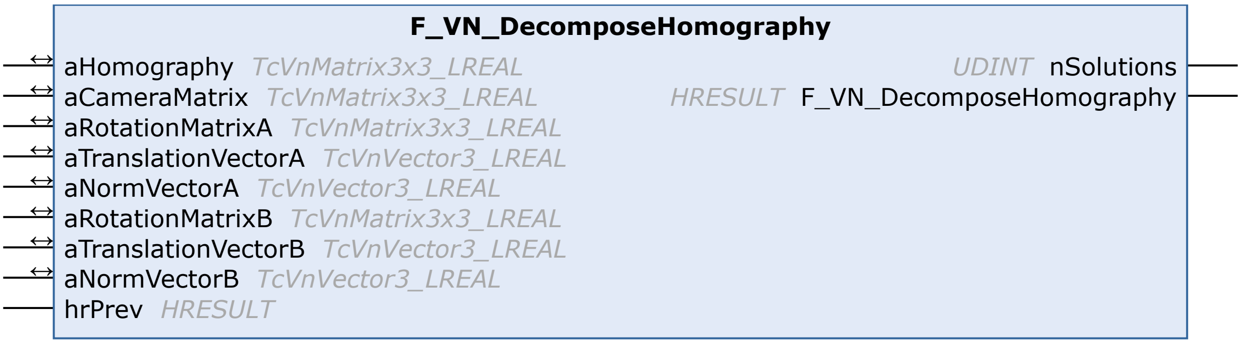
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.28 F_VN_DecomposeHomography



Decompose a homography matrix into four solutions of rotation, translation, and plane normal. It returns two solutions (Ra, Ta, Nb) and (Rb, Tb, Nb). The third and the fourth solutions can be calculated as follows: (Ra, -Ta, -Na) and (Rb, -Tb, -Nb).

Syntax

Definition:

```
FUNCTION F_VN_DecomposeHomography : HRESULT
VAR_IN_OUT
    aHomography         : TcVnMatrix3x3_LREAL;
    aCameraMatrix       : TcVnMatrix3x3_LREAL;
    aRotationMatrixA    : TcVnMatrix3x3_LREAL;
    aTranslationVectorA : TcVnVector3_LREAL;
    aNormVectorA        : TcVnVector3_LREAL;
    aRotationMatrixB    : TcVnMatrix3x3_LREAL;
    aTranslationVectorB : TcVnVector3_LREAL;
    aNormVectorB        : TcVnVector3_LREAL;
END_VAR
VAR_INPUT
    hrPrev              : HRESULT;
```

```
END_VAR
VAR_OUTPUT
    nSolutions          : UDINT;
END_VAR
```

Inputs

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aHomography	TcVnMatrix3x3_LREAL [▶ 141]	Homography matrix (a perspective transformation between two planes)
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Camera matrix
aRotationMatrixA	TcVnMatrix3x3_LREAL [▶ 141]	Rotation matrix of the first solution (Ra)
aTranslationVectorA	TcVnVector3_LREAL [▶ 141]	Translation vector of the first solution (Ta)
aNormVectorA	TcVnVector3_LREAL [▶ 141]	Norm vector of the first solution (Na)
aRotationMatrixB	TcVnMatrix3x3_LREAL [▶ 141]	Rotation matrix of the second solution (Rb)
aTranslationVectorB	TcVnVector3_LREAL [▶ 141]	Translation vector of the second solution (Tb)
aNormVectorB	TcVnVector3_LREAL [▶ 141]	Norm vector of the second solution Nb

Outputs

Name	Type	Description
nSolutions	UDINT	Returns the number of solutions. In normal cases, it is equal four.

Return value

HRESULT [▶ 122]

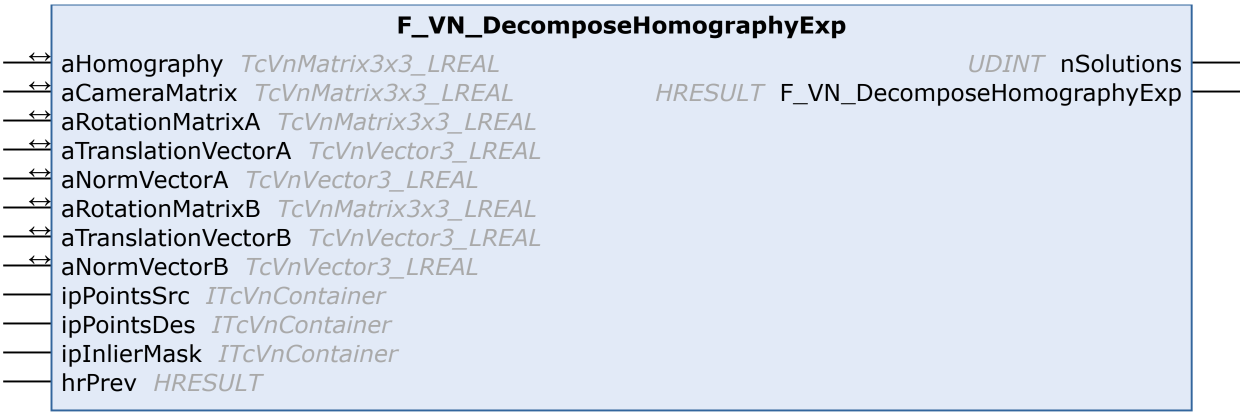
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.29 F_VN_DecomposeHomographyExp



Decompose a homography matrix and verify four solutions of rotation, translation, and plane normal. It uses visible reference points being in front of the camera to confirm which solution(s) of the four solutions (maximum two solutions) are consistent with all reference points.

Syntax

Definition:

```
FUNCTION F_VN_DecomposeHomographyExp : HRESULT
VAR_IN_OUT
    aHomography         : TcVnMatrix3x3_LREAL;
    aCameraMatrix       : TcVnMatrix3x3_LREAL;
    aRotationMatrixA    : TcVnMatrix3x3_LREAL;
    aTranslationVectorA : TcVnVector3_LREAL;
    aNormVectorA        : TcVnVector3_LREAL;
    aRotationMatrixB    : TcVnMatrix3x3_LREAL;
    aTranslationVectorB : TcVnVector3_LREAL;
    aNormVectorB        : TcVnVector3_LREAL;
END_VAR
VAR_INPUT
    ipPointsSrc         : ITcVnContainer;
    ipPointsDes         : ITcVnContainer;
    ipInlierMask        : ITcVnContainer;
    hrPrev              : HRESULT;
END_VAR
VAR_OUTPUT
    nSolutions          : UDINT;
END_VAR
```

Inputs

Name	Type	Description
ipPointsSrc	ITcVnContainer [▶ 349]	Container with source points (ContainerType_Vector_TcVnPoint2_REAL)
ipPointsDes	ITcVnContainer [▶ 349]	Container with destination points (same number as ipPoints1, ContainerType_Vector_TcVnPoint2_REAL)
ipInlierMask	ITcVnContainer [▶ 349]	A mask marking the inliers (optional, set to 0 if not available; ContainerType_Vector_SINT or ContainerType_Vector_USINT)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aHomography	TcVnMatrix3x3_LREAL [▶ 141]	Homography matrix (a perspective transformation between two planes)
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Camera matrix
aRotationMatrixA	TcVnMatrix3x3_LREAL [▶ 141]	Rotation matrix of the first solution (Ra)
aTranslationVectorA	TcVnVector3_LREAL [▶ 141]	Translation vector of the first solution (Ta)
aNormVectorA	TcVnVector3_LREAL [▶ 141]	Norm vector of the first solution (Na)
aRotationMatrixB	TcVnMatrix3x3_LREAL [▶ 141]	Rotation matrix of the second solution (Rb)
aTranslationVectorB	TcVnVector3_LREAL [▶ 141]	Translation vector of the second solution (Tb)
aNormVectorB	TcVnVector3_LREAL [▶ 141]	Norm vector of the second solution (Nb)

Outputs

Name	Type	Description
nSolutions	UDINT	Return the number of true potential solutions [0,1, or 2]. 0: no true potential solutions for the given corresponding points are confirmed. 1: the solution (Ra, Ta, Na) is confirmed. 2: both solution (Ra, Ta, Na) and (Rb, Tb, Nb) are confirmed. In all cases the function returns two solutions (Ra, Ta, Nb) and (Rb, Tb, Nb). The third and the fourth solutions can be calculated as follows: (Ra, -Ta, -Na) and (Rb, -Tb, -Nb).

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.30 F_VN_FlipImage

Flip an image (mirror and shift back to prior coordinates).

Syntax

Definition:

```
FUNCTION F_VN_FlipImage : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    eFlipAxis   : ETcVnFlipAxis;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
eFlipAxis	ETcVnFlipAxis [▶ 182]	Selects the axis around which to flip the image
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.31 F_VN_GenerateAffineTransformationUnitMatrix2D

Generate an affine transformation 2D unit matrix.

Syntax

Definition:

```
FUNCTION F_VN_GenerateAffineTransformationUnitMatrix2D : HRESULT
VAR_IN_OUT
    aTransformationMatrix : TcVnMatrix2x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 141]	Resulting affine transformation unit matrix

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.32 F_VN_GetAffineTransformation

Calculate the affine transformation between three corresponding point pairs. The points mark the corners of the corresponding triangles.

Syntax

Definition:

```
FUNCTION F_VN_GetAffineTransformation : HRESULT
VAR_IN_OUT
    aSrcPoints            : TcVnArray3_Point2_REAL;
    aDestPoints           : TcVnArray3_Point2_REAL;
    aTransformationMatrix : TcVnMatrix2x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aSrcPoints	TcVnArray3_Point2_REAL [▶ 141]	Source points
aDestPoints	TcVnArray3_Point2_REAL [▶ 141]	Destination points
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 141]	Resulting transformation matrix

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.33 F_VN_GetAffineTransformation2D

Estimate the 2D affine transformation between two planar point sets using RANSAC algorithm.

Syntax

Definition:

```
FUNCTION F_VN_GetAffineTransformation2D : HRESULT
VAR_INPUT
    ipSrcPoints      : ITcVnContainer;
    ipDestPoints     : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aAffineTransform : TcVnMatrix2x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPoints	ITcVnContainer [▶ 349]	Container with at least 3 source points (ContainerType_Vector_TcVnPoint2_REAL)
ipDestPoints	ITcVnContainer [▶ 349]	Container with destination points (same number as ipSrcPoints, ContainerType_Vector_TcVnPoint2_REAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aAffineTransform	TcVnMatrix2x3_LREAL [▶ 141]	Returns the affine transformation matrix, which transforms the source points to the destination points

Return value

HRESULT [▶ 122]

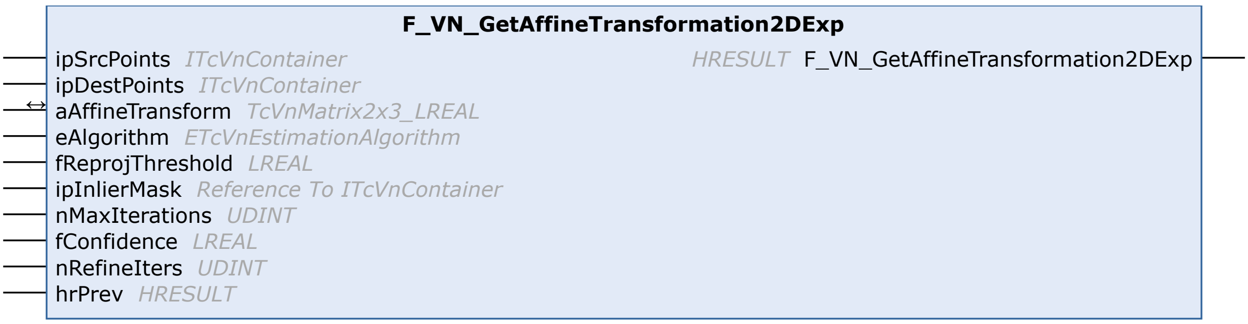
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.34 F_VN_GetAffineTransformation2DExp



Estimate the affine transformation between two planar point sets. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_GetAffineTransformation2DExp : HRESULT
VAR_INPUT
    ipSrcPoints       : ITcVnContainer;
    ipDestPoints      : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aAffineTransform : TcVnMatrix2x3_LREAL;
END_VAR
VAR_INPUT
    eAlgorithm        : ETcVnEstimationAlgorithm;
    fReprojThreshold : LREAL;
    ipInlierMask      : Reference To ITcVnContainer;
    nMaxIterations    : UDINT;
    fConfidence       : LREAL;
    nRefineIters      : UDINT;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPoints	ITcVnContainer [▸ 349]	Container with at least 3 source points (ContainerType_Vector_TcVnPoint2_REAL)
ipDestPoints	ITcVnContainer [▸ 349]	Container with destination points (same number as ipSrcPoints, ContainerType_Vector_TcVnPoint2_REAL)
eAlgorithm	ETcVnEstimationAlgorithm [▸ 179]	Estimation algorithm (only RANSAC and LMEDS are supported)
fReprojThreshold	LREAL	Maximum allowed reprojection error to treat a point pair as an inlier
ipInlierMask	Reference To ITcVnContainer [▸ 349]	Returns a mask marking the inliers (optional, set to 0 if not required; ContainerType_Vector_USINT)
nMaxIterations	UDINT	Maximum number of iterations
fConfidence	LREAL	Confidence (0..1)
nRefineIters	UDINT	Maximum number of iterations of Levenberg-Marquardt algorithm to refine further the affine transform (using only inliers). Set to 0 to disable refining.
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aAffineTransform	TcVnMatrix2x3_LREAL [▸ 141]	Returns the affine transformation matrix, which transforms the source points to the destination points

Return value

HRESULT [▸ 122]

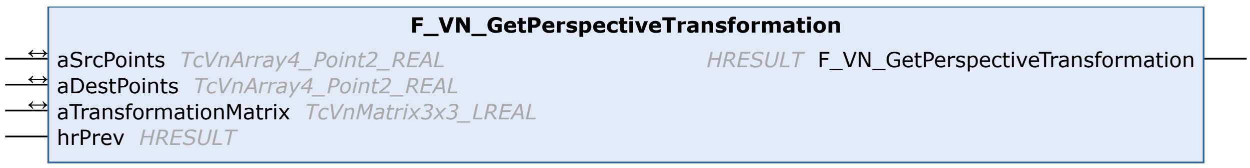
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.35 F_VN_GetPerspectiveTransformation



Calculate the perspective transformation between four corresponding point pairs. The points mark the corners of the corresponding rectangles.

Syntax

Definition:

```
FUNCTION F_VN_GetPerspectiveTransformation : HRESULT
VAR_IN_OUT
    aSrcPoints            : TcVnArray4_Point2_REAL;
    aDestPoints           : TcVnArray4_Point2_REAL;
    aTransformationMatrix : TcVnMatrix3x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aSrcPoints	TcVnArray4_Point2_REAL [▶ 141]	Source points
aDestPoints	TcVnArray4_Point2_REAL [▶ 141]	Destination points
aTransformationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Resulting transformation matrix

Return value

HRESULT [▶ 122]

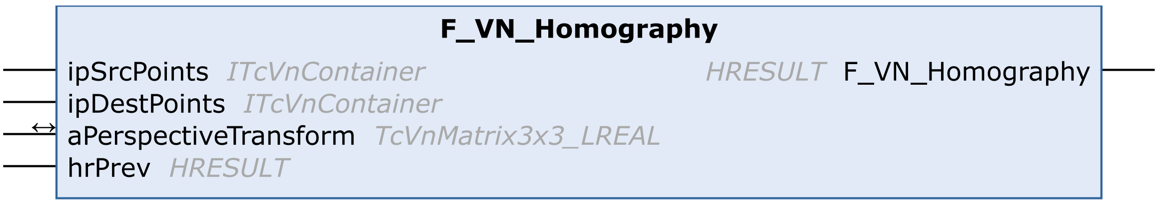
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.36 F_VN_Homography



Find the homography (perspective transformation) between two planar point sets using RANSAC algorithm.

Syntax

Definition:

```
FUNCTION F_VN_Homography : HRESULT
VAR_INPUT
    ipSrcPoints           : ITcVnContainer;
    ipDestPoints          : ITcVnContainer;
END_VAR
```

```
VAR_IN_OUT
    aPerspectiveTransform : TcVnMatrix3x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPoints	ITcVnContainer [▶ 349]	Container with at least 4 source points (ContainerType_Vector_TcVnPoint2_REAL)
ipDestPoints	ITcVnContainer [▶ 349]	Container with destination points (same number as ipSrcPoints, ContainerType_Vector_TcVnPoint2_REAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aPerspectiveTransform	TcVnMatrix3x3_LREAL [▶ 141]	Returns the perspective transformation matrix, which transforms the source points to the destination points

Return value

HRESULT [▶ 122]

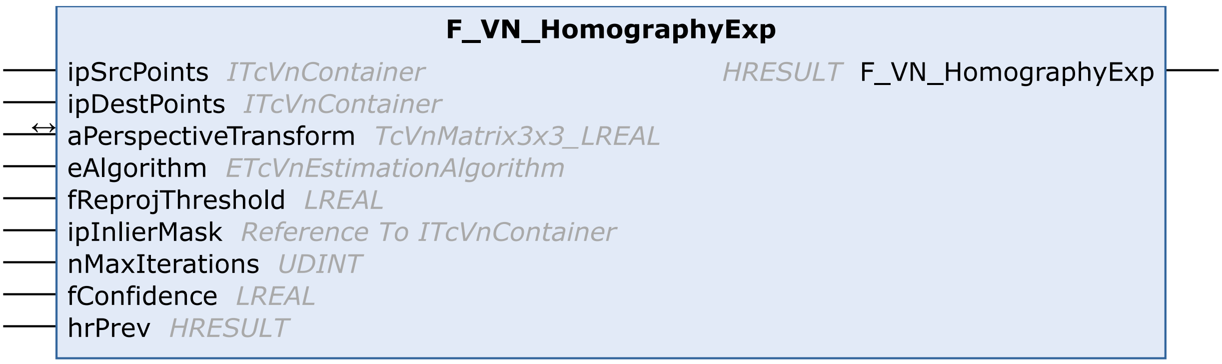
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.37 F_VN_HomographyExp



Find the homography (perspective transformation) between two planar point sets. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_HomographyExp : HRESULT
VAR_INPUT
    ipSrcPoints            : ITcVnContainer;
    ipDestPoints           : ITcVnContainer;
END_VAR
VAR_IN_OUT
    aPerspectiveTransform : TcVnMatrix3x3_LREAL;
END_VAR
VAR_INPUT
    eAlgorithm             : ETcVnEstimationAlgorithm;
    fReprojThreshold       : LREAL;
    ipInlierMask           : Reference To ITcVnContainer;
    nMaxIterations         : UDINT;
    fConfidence            : LREAL;
    hrPrev                 : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPoints	ITcVnContainer [▶ 349]	Container with at least 4 source points (ContainerType_Vector_TcVnPoint2_REAL)
ipDestPoints	ITcVnContainer [▶ 349]	Container with destination points (same number as ipSrcPoints, ContainerType_Vector_TcVnPoint2_REAL)
eAlgorithm	ETcVnEstimationAlgorithm [▶ 179]	Estimation algorithm
fReprojThreshold	LREAL	Maximum allowed reprojection error to treat a point pair as an inlier (only for RANSAC, RHO)
ipInlierMask	Reference To ITcVnContainer [▶ 349]	Returns a mask marking the inliers (optional, set to 0 if not required; ContainerType_Vector_SINT; only for RANSAC, LMEDS)
nMaxIterations	UDINT	Maximum number of RANSAC iterations
fConfidence	LREAL	Confidence (0..1)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aPerspectiveTransform	TcVnMatrix3x3_LREAL [▶ 141]	Returns the perspective transformation matrix, which transforms the source points to the destination points

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.38 F_VN_InvertAffineTransformation

Invert a 2D affine transformation matrix. The inverted transformation compensates the affine transformation on an image if it is used in the F_VN_WarpAffine function.

Syntax

Definition:

```
FUNCTION F_VN_InvertAffineTransformation : HRESULT
VAR_IN_OUT
    aTransformationMatrix         : TcVnMatrix2x3_LREAL;
    aInvertedTransformationMatrix : TcVnMatrix2x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 141]	Source affine transformation matrix
aInvertedTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 141]	Resulting inverted affine transformation matrix

Return value

HRESULT [▶ 122]

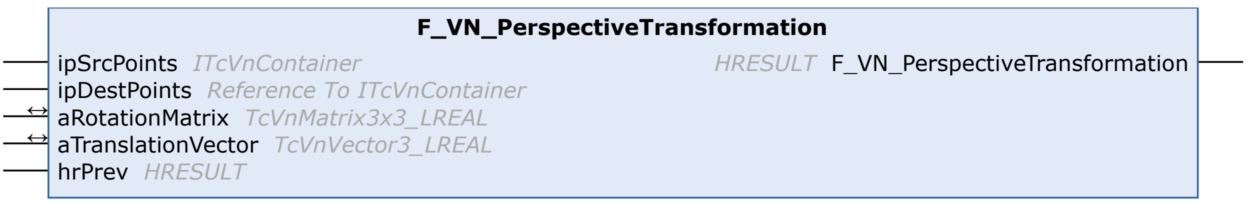
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.39 F_VN_PerspectiveTransformation



Apply a perspective transformation (rotation + translation, e.g. from extrinsic calibration) to 3D point coordinates.

Syntax

Definition:

```
FUNCTION F_VN_PerspectiveTransformation : HRESULT
VAR_INPUT
    ipSrcPoints         : ITcVnContainer;
    ipDestPoints        : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    aRotationMatrix     : TcVnMatrix3x3_LREAL;
    aTranslationVector  : TcVnVector3_LREAL;
END_VAR
VAR_INPUT
    hrPrev              : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPoints	ITcVnContainer [▸ 349]	Container with 3D source points (TcVnPoint3_REAL or TcVnPoint3_LREAL)
ipDestPoints	Reference To ITcVnContainer [▸ 349]	Returns the transformed points (same type as ipSrcPoints)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aRotationMatrix	TcVnMatrix3x3_LREAL [▸ 141]	3x3 rotation matrix
aTranslationVector	TcVnVector3_LREAL [▸ 141]	Translation vector

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.40 F_VN_PyramidDown

Downsamples an image to half width and height.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_PyramidDown : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.41 F_VN_PyramidUp

Upsamples an image to double width and height.

Syntax

Definition:

```
FUNCTION F_VN_PyramidUp : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.42 F_VN_RemapImageToLogPolarSpace

Remap an image to log-polar space.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_RemapImageToLogPolarSpace : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aCenter     : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (Must not be the same as ipSrcImage! An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCenter	TcVnPoint2_REAL [▸ 139]	Center point for the transformation

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.43 F_VN_RemapImageToLogPolarSpaceExp

Remap an image to log-polar space. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_RemapImageToLogPolarSpaceExp : HRESULT
VAR_INPUT
    ipSrcImage          : ITcVnImage;
    ipDestImage         : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aCenter             : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    fScale              : LREAL;
    eInterpolationType : ETcVnInterpolationType;
    hrPrev              : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (Must not be the same as ipSrcImage! An appropriate destination image will be created if required.)
fScale	LREAL	Magnitude scale parameter, i.e. image width / ln(radius) (set to 0 for auto select)
eInterpolationType	ETcVnInterpolationType [▶ 185]	Interpolation type
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCenter	TcVnPoint2_REAL [▶ 139]	Center point for the transformation

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.44 F_VN_RemapImageToLogPolarSpaceExp2

Remap an image to log-polar space. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_RemapImageToLogPolarSpaceExp2 : HRESULT
VAR_INPUT
    ipSrcImage          : ITcVnImage;
    ipDestImage         : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aCenter             : TcVnPoint2_REAL;
END_VAR
```

```
VAR_INPUT
    fScale              : LREAL;
    eInterpolationType : ETcVnInterpolationType;
    nDestWidth          : DINT;
    nDestHeight         : DINT;
    hrPrev              : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (Must not be the same as ipSrcImage! An appropriate destination image will be created if required.)
fScale	LREAL	Magnitude scale parameter, i.e. image width / ln(radius) (set to 0 for auto select)
eInterpolationType	ETcVnInterpolationType [▶ 185]	Interpolation type
nDestWidth	DINT	Destination image width (set to -1 for source image width, 0 to auto scale to transformed radius, > 0 for a user defined width)
nDestHeight	DINT	destination image height (set to -1 for source image height, 0 to auto scale to transformed radius * PI, > 0 for a user defined height)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCenter	TcVnPoint2_REAL [▶ 139]	Center point for the transformation

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.45 F_VN_RemapImageToPolarSpace

Remap an image to polar space.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_RemapImageToPolarSpace : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aCenter     : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (Must not be the same as ipSrcImage! An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCenter	TcVnPoint2_REAL [▶ 139]	Center point for the transformation

Return value

HRESULT [▶ 122]

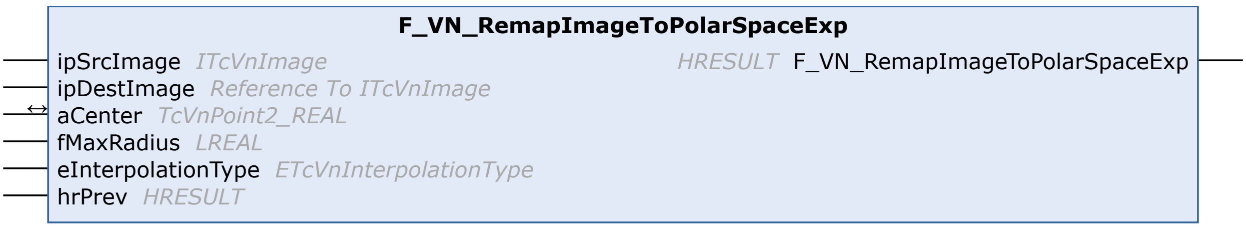
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.46 F_VN_RemapImageToPolarSpaceExp



Remap an image to polar space. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_RemapImageToPolarSpaceExp : HRESULT
VAR_INPUT
    ipSrcImage         : ITcVnImage;
    ipDestImage        : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aCenter            : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    fMaxRadius         : LREAL;
    eInterpolationType : ETcVnInterpolationType;
    hrPrev             : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (Must not be the same as ipSrcImage! An appropriate destination image will be created if required.)
fMaxRadius	LREAL	Maximum radius for the transformation (set to 0 for auto select)
eInterpolationType	ETcVnInterpolationType [▸ 185]	Interpolation type
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCenter	TcVnPoint2_REAL [▸ 139]	Center point for the transformation

Return value

HRESULT [▸ 122]

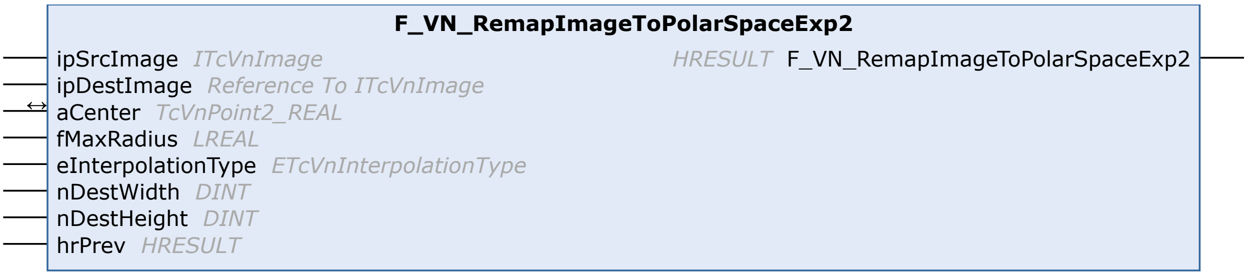
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.47 F_VN_RemapImageToPolarSpaceExp2



Remap an image to polar space. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_RemapImageToPolarSpaceExp2 : HRESULT
VAR_INPUT
    ipSrcImage         : ITcVnImage;
    ipDestImage        : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aCenter            : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    fMaxRadius         : LREAL;
    eInterpolationType : ETcVnInterpolationType;
    nDestWidth         : DINT;
    nDestHeight        : DINT;
    hrPrev             : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (Must not be the same as ipSrcImage! An appropriate destination image will be created if required.)
fMaxRadius	LREAL	Maximum radius for the transformation (set to 0 for auto select)
eInterpolationType	ETcVnInterpolationType [▶ 185]	Interpolation type
nDestWidth	DINT	Destination image width (set to -1 for source image width, 0 to auto scale to transformed radius, > 0 for a user defined width)
nDestHeight	DINT	destination image height (set to -1 for source image height, 0 to auto scale to transformed radius * PI, > 0 for a user defined height)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCenter	TcVnPoint2_REAL [▶ 139]	Center point for the transformation

Return value

HRESULT [▶ 122]

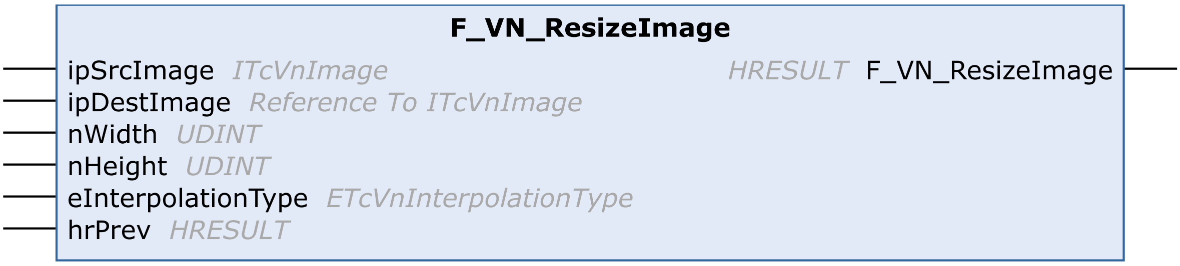
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.48 F_VN_ResizeImage



Resize an image using a specific interpolation type.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_ResizeImage : HRESULT
VAR_INPUT
    ipSrcImage          : ITcVnImage;
    ipDestImage         : Reference To ITcVnImage;
    nWidth              : UDINT;
    nHeight             : UDINT;
    eInterpolationType  : ETcVnInterpolationType;
    hrPrev              : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate image will be created if required.)
nWidth	UDINT	New width
nHeight	UDINT	New height
eInterpolationType	ETcVnInterpolationType [▶ 185]	Interpolation type
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

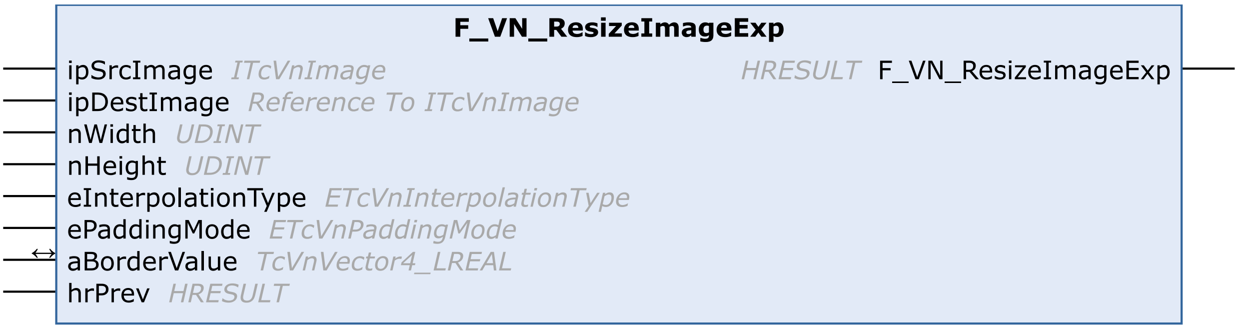
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.49 F_VN_ResizeImageExp



Resize an image using a specific interpolation type while maintaining its aspect ratio. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_ResizeImageExp : HRESULT
VAR_INPUT
    ipSrcImage         : ITcVnImage;
    ipDestImage        : Reference To ITcVnImage;
    nWidth             : UDINT;
    nHeight            : UDINT;
    eInterpolationType : ETcVnInterpolationType;
    ePaddingMode       : ETcVnPaddingMode;
END_VAR
VAR_IN_OUT
    aBorderValue       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev             : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate image will be created if required.)
nWidth	UDINT	New width
nHeight	UDINT	New height
eInterpolationType	ETcVnInterpolationType [▸ 185]	Interpolation type
ePaddingMode	ETcVnPaddingMode [▸ 192]	Image padding mode
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aBorderValue	TcVnVector4_LREAL [▶ 141]	Border value, if TCVN_PM_LETTERBOX is used

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.50 F_VN_RotateImage

Rotate an image by 90, 180, or 270 degrees in clockwise direction.

Syntax

Definition:

```
FUNCTION F_VN_RotateImage : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    eAngle      : ETcVnRotationAngle;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate image will be created if required.)
eAngle	ETcVnRotationAngle [▶ 195]	Rotation angle
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.51 F_VN_RotateImageExp

F_VN_RotateImageExp

ipSrcImage *ITcVnImage* — *HRESULT* F_VN_RotateImageExp
ipDestImage *Reference To ITcVnImage*
fAngleDeg *LREAL*
bAdjustDestSize *BOOL*
hrPrev *HRESULT*

Rotate an image by an individual rotation angle in degrees.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_RotateImageExp : HRESULT
VAR_INPUT
    ipSrcImage      : ITcVnImage;
    ipDestImage     : Reference To ITcVnImage;
    fAngleDeg       : LREAL;
    bAdjustDestSize : BOOL;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate image will be created if required.)
fAngleDeg	LREAL	Rotation angle in degree (positive: counter-clockwise, negative: clockwise)
bAdjustDestSize	BOOL	If true, ipDestImage size is adjusted so that the whole rotated ipSrcImage is contained. Otherwise, ipDestImage size is set to ipSrcImage size.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

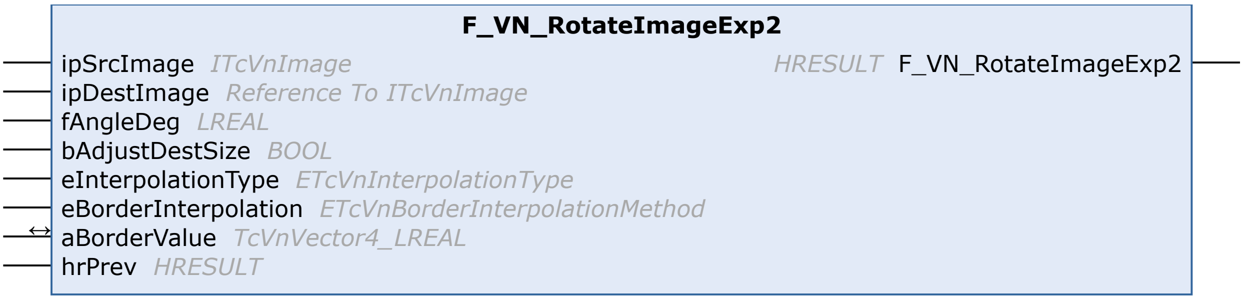
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.52 F_VN_RotateImageExp2



Rotate an image by an individual rotation angle in degrees. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_RotateImageExp2 : HRESULT
VAR_INPUT
    ipSrcImage           : ITcVnImage;
    ipDestImage          : Reference To ITcVnImage;
    fAngleDeg            : LREAL;
    bAdjustDestSize      : BOOL;
    eInterpolationType   : ETcVnInterpolationType;
    eBorderInterpolation : ETcVnBorderInterpolationMethod;
END_VAR
VAR_IN_OUT
    aBorderValue         : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev               : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate image will be created if required.)
fAngleDeg	LREAL	Rotation angle in degree (positive: counter-clockwise, negative: clockwise)
bAdjustDestSize	BOOL	If true, ipDestImage size is adjusted so that the whole rotated ipSrcImage is contained. Otherwise, ipDestImage size is set to ipSrcImage size.
eInterpolationType	ETcVnInterpolationType [▸ 185]	Interpolation method
eBorderInterpolation	ETcVnBorderInterpolationMethod [▸ 145]	Border interpolation method (ISOLATED not supported)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aBorderValue	TcVnVector4_LREAL [▶ 141]	Border value, if CONSTANT is used

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.53 F_VN_WarpAffine

Apply an affine transformation to an image.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_WarpAffine : HRESULT
VAR_INPUT
    ipSrcImage            : ITcVnImage;
    ipDestImage           : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aTransformationMatrix : TcVnMatrix2x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 141]	Affine transformation matrix

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.54 F_VN_WarpAffine_Container

F_VN_WarpAffine_Container
ipSrcPoints *ITcVnContainer* — *HRESULT* F_VN_WarpAffine_Container
ipDestPoints *Reference To ITcVnContainer*
aTransformationMatrix *TcVnMatrix2x3_LREAL*
hrPrev *HRESULT*

Apply an affine transformation to a container of 2D points.

Syntax

Definition:

```
FUNCTION F_VN_WarpAffine_Container : HRESULT
VAR_INPUT
    ipSrcPoints           : ITcVnContainer;
    ipDestPoints          : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    aTransformationMatrix : TcVnMatrix2x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPoints	ITcVnContainer [▶ 349]	Source points (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_LREAL)
ipDestPoints	Reference To ITcVnContainer [▶ 349]	Returns the transformed points (same type ID as ipSrcPoints; Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 141]	Affine transformation matrix

Return value

HRESULT [▶ 122]

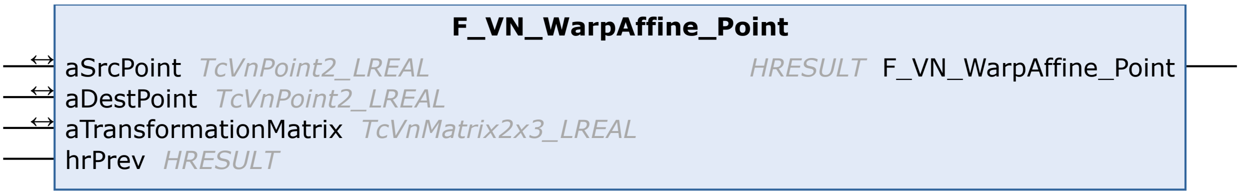
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.55 F_VN_WarpAffine_Point



Apply an affine transformation to a 2D point.

Syntax

Definition:

```
FUNCTION F_VN_WarpAffine_Point : HRESULT
VAR_IN_OUT
    aSrcPoint             : TcVnPoint2_LREAL;
    aDestPoint            : TcVnPoint2_LREAL;
    aTransformationMatrix : TcVnMatrix2x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aSrcPoint	TcVnPoint2_LREAL [▶ 139]	Source point
aDestPoint	TcVnPoint2_LREAL [▶ 139]	Destination point
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 141]	Affine transformation matrix

Return value

HRESULT [▶ 122]

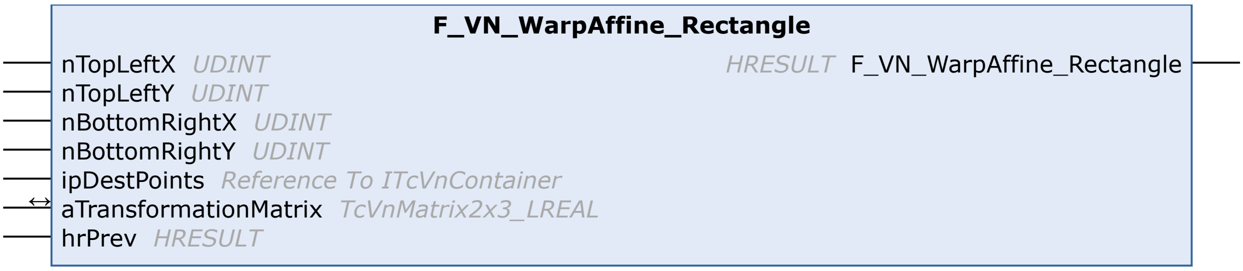
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.56 F_VN_WarpAffine_Rectangle



Use an affine transform to warp a rectangle.

Syntax

Definition:

```
FUNCTION F_VN_WarpAffine_Rectangle : HRESULT
VAR_INPUT
    nTopLeftX             : UDINT;
    nTopLeftY             : UDINT;
    nBottomRightX         : UDINT;
    nBottomRightY         : UDINT;
    ipDestPoints          : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    aTransformationMatrix : TcVnMatrix2x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nTopLeftX	UDINT	x coordinate of the top left corner
nTopLeftY	UDINT	y coordinate of the top left corner
nBottomRightX	UDINT	x coordinate of the bottom right corner
nBottomRightY	UDINT	y coordinate of the bottom right corner
ipDestPoints	Reference To ITcVnContainer [▶ 349]	Returns the 4 transformed points (ContainerType_Vector_TcVnPoint2_REAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 141]	Affine transformation matrix

Return value

HRESULT [▶ 122]

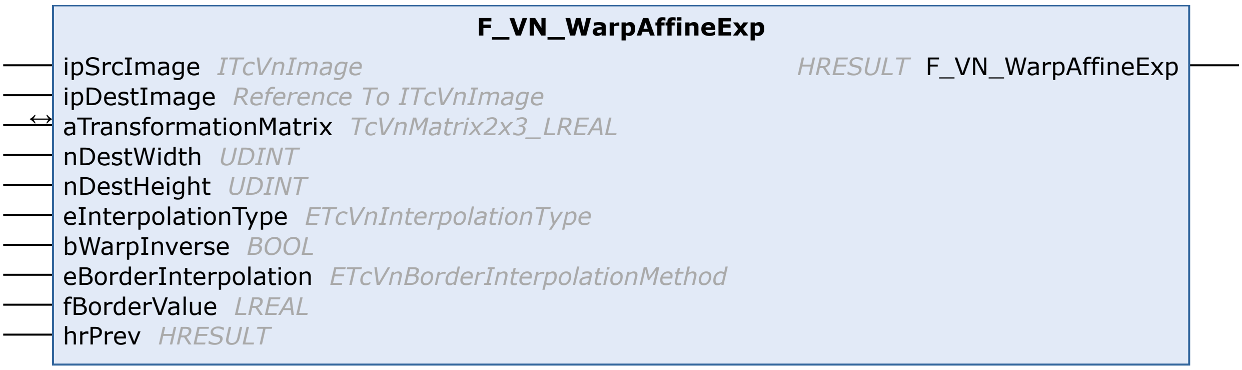
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.57 F_VN_WarpAffineExp



Apply an affine transformation to an image. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_WarpAffineExp : HRESULT
VAR_INPUT
    ipSrcImage              : ITcVnImage;
    ipDestImage             : Reference To ITcVnImage;
```

```
END_VAR
VAR_IN_OUT
    aTransformationMatrix : TcVnMatrix2x3_LREAL;
END_VAR
VAR_INPUT
    nDestWidth             : UDINT;
    nDestHeight            : UDINT;
    eInterpolationType     : ETcVnInterpolationType;
    bWarpInverse           : BOOL;
    eBorderInterpolation   : ETcVnBorderInterpolationMethod;
    fBorderValue           : LREAL;
    hrPrev                 : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate image will be created if required.)
nDestWidth	UDINT	Width of the destination image
nDestHeight	UDINT	Height of the destination image
eInterpolationType	ETcVnInterpolationType [▸ 185]	Interpolation method
bWarpInverse	BOOL	Handle aTransformationMatrix as the inverse transformation
eBorderInterpolation	ETcVnBorderInterpolationMethod [▸ 145]	Border interpolation method (ISOLATED not supported)
fBorderValue	LREAL	Border value, if CONSTANT is used
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformationMatrix	TcVnMatrix2x3_LREAL [▸ 141]	Affine transformation matrix

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.58 F_VN_WarpPerspective

F_VN_WarpPerspective

ipSrcImage *ITcVnImage* — *HRESULT* F_VN_WarpPerspective
ipDestImage *Reference To ITcVnImage*
aTransformationMatrix *TcVnMatrix3x3_LREAL*
hrPrev *HRESULT*

Apply a perspective transformation to an image.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_WarpPerspective : HRESULT
VAR_INPUT
    ipSrcImage            : ITcVnImage;
    ipDestImage           : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aTransformationMatrix : TcVnMatrix3x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Perspective transformation matrix

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.59 F_VN_WarpPerspective_Container

Apply a perspective transformation to a container of 2D points.

Syntax

Definition:

```
FUNCTION F_VN_WarpPerspective_Container : HRESULT
VAR_INPUT
    ipSrcPoints           : ITcVnContainer;
    ipDestPoints          : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    aTransformationMatrix : TcVnMatrix3x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcPoints	ITcVnContainer [▶ 349]	Source points (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_LREAL)
ipDestPoints	Reference To ITcVnContainer [▶ 349]	Returns the transformed points (same type ID as ipSrcPoints; Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Perspective transformation matrix

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.60 F_VN_WarpPerspective_Point

Apply a perspective transformation to a 2D point.

Syntax

Definition:

```
FUNCTION F_VN_WarpPerspective_Point : HRESULT
VAR_IN_OUT
    aSrcPoint             : TcVnPoint2_LREAL;
    aDestPoint            : TcVnPoint2_LREAL;
    aTransformationMatrix : TcVnMatrix3x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aSrcPoint	TcVnPoint2_LREAL [▸ 139]	Source point
aDestPoint	TcVnPoint2_LREAL [▸ 139]	Destination point
aTransformationMatrix	TcVnMatrix3x3_LREAL [▸ 141]	Perspective transformation matrix

Return value

HRESULT [▸ 122]

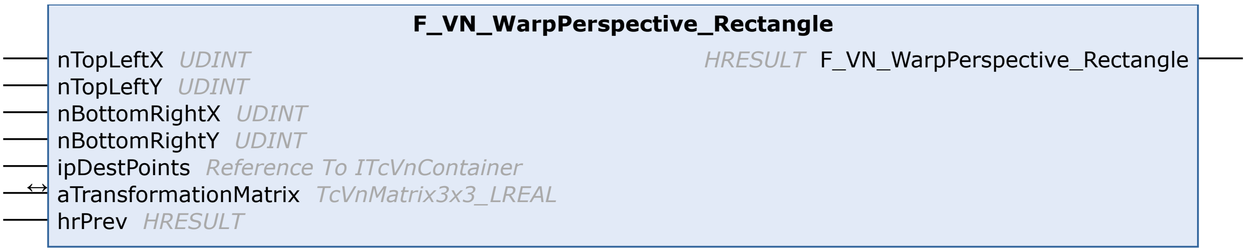
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.61 F_VN_WarpPerspective_Rectangle



Use a perspective transform to warp a rectangle.

Syntax

Definition:

```
FUNCTION F_VN_WarpPerspective_Rectangle : HRESULT
VAR_INPUT
    nTopLeftX             : UDINT;
    nTopLeftY             : UDINT;
    nBottomRightX         : UDINT;
    nBottomRightY         : UDINT;
    ipDestPoints          : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    aTransformationMatrix : TcVnMatrix3x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nTopLeftX	UDINT	x coordinate of the top left corner
nTopLeftY	UDINT	y coordinate of the top left corner
nBottomRightX	UDINT	x coordinate of the bottom right corner
nBottomRightY	UDINT	y coordinate of the bottom right corner
ipDestPoints	Reference To ITcVnContainer [▸ 349]	Returns the 4 transformed points (ContainerType_Vector_TcVnPoint2_REAL)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformationMatrix	TcVnMatrix3x3_LREAL [▸ 141]	Perspective transformation matrix

Return value

HRESULT [▸ 122]

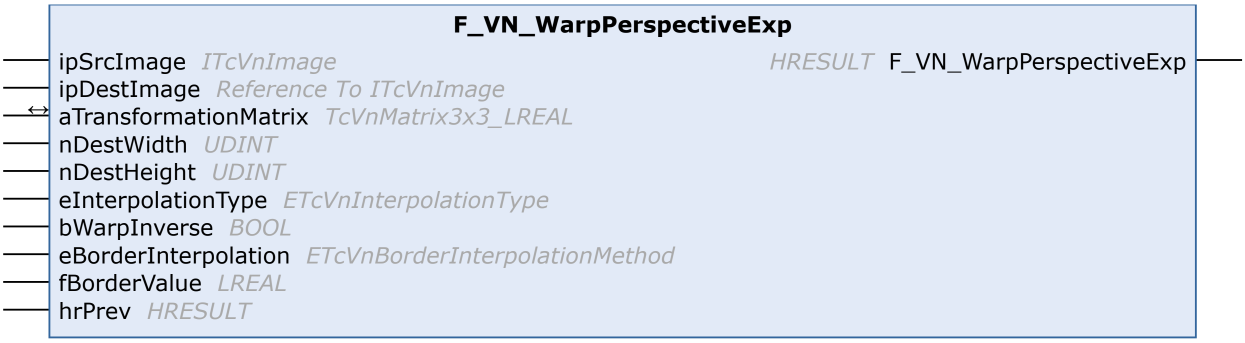
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.13.62 F_VN_WarpPerspectiveExp



Apply a perspective transformation to an image. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_WarpPerspectiveExp : HRESULT
VAR_INPUT
    ipSrcImage            : ITcVnImage;
    ipDestImage           : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aTransformationMatrix : TcVnMatrix3x3_LREAL;
END_VAR
VAR_INPUT
    nDestWidth            : UDINT;
    nDestHeight           : UDINT;
    eInterpolationType    : ETcVnInterpolationType;
    bWarpInverse          : BOOL;
    eBorderInterpolation  : ETcVnBorderInterpolationMethod;
    fBorderValue          : LREAL;
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate image will be created if required.)
nDestWidth	UDINT	Width of the destination image
nDestHeight	UDINT	Height of the destination image
eInterpolationType	ETcVnInterpolationType [▶ 185]	Interpolation method
bWarpInverse	BOOL	Handle aTransformationMatrix as the inverse transformation
eBorderInterpolation	ETcVnBorderInterpolationMethod [▶ 145]	Border interpolation method (ISOLATED not supported)
fBorderValue	LREAL	Border value, if CONSTANT is used
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aTransformationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Perspective transformation matrix

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14 Image Analysis

该组包含从图像中提取信息的函数。

函数

对象检测 [▶ 1137]

- 轮廓搜索
- 通过 Hough 变换查找对象
- 通过图像对比进行匹配

统计图像特征

- F_VN_ImageAverage(Exp) [▶ 1178]
- F_VN_ImageAverageStdDev(Exp) [▶ 1179]

- F_VN_ImageMedian（Exp） [▶ 1183]
- F_VN_MaxPixelValue（Exp） [▶ 1186]
- F_VN_MinPixelValue（Exp） [▶ 1188]

边缘检测

- F_VN_CannyEdgeDetection（Exp） [▶ 1164]

查找连续图像区域

- F_VN_ConnectedComponents(Exp) [▶ 1167]
- F_VN_ConnectedComponentsWithStats (Exp) [▶ 1170]
- F_VN_GetConnectedComponent(Exp) [▶ 1176]

区域对齐

- F_VN_RegionOrientation(Exp) [▶ 1190]

其它

- F_VN_CountNonZeroPixels [▶ 1172]
- F_VN_DistanceTransformation（Exp） [▶ 1173]
- F_VN_ImageCenterOfMass（Exp） [▶ 1181]
- F_VN_ImageMoments [▶ 1185]

样本

图像分析 [▶ 2664]

6.1.4.14.1 Object Detection

该子组包含用于物体识别的函数。

函数

对象检测

- F_VN_DetectBlobs(Exp) [▶ 1139]
- F_VN_FindContours(Exp) [▶ 1149]
- F_VN_FindContourHierarchyExp [▶ 1143]
- F_VN_HoughCircles(Exp) [▶ 1152]
- F_VN_HoughLines(Exp) [▶ 1155]
- F_VN_HoughLinesP(Exp) [▶ 1157]

匹配

- F_VN_MatchImageHuMoments [▶ 1158]
- F_VN_MatchTemplate(Exp) [▶ 1163]
- F_VN_MatchTemplateAndEvaluate(Exp) [▶ 1161]

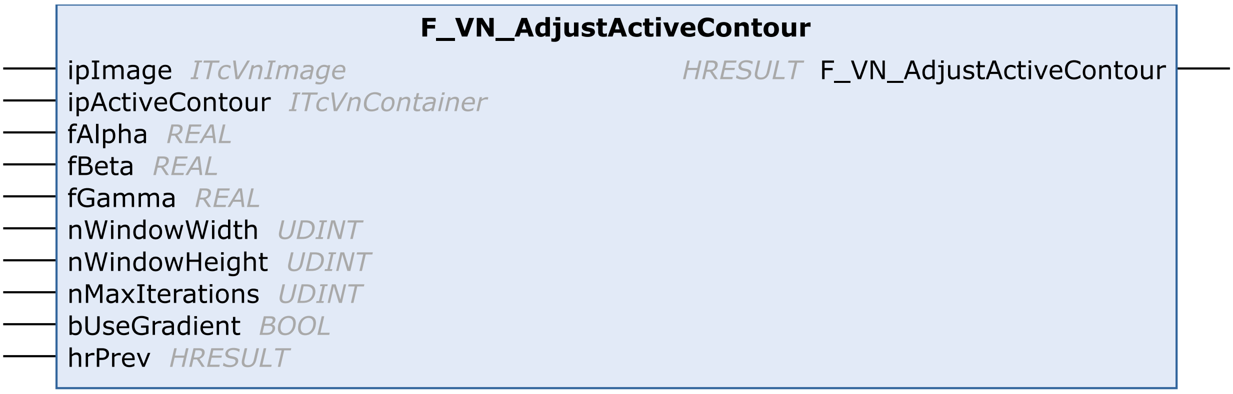
其他

- F_VN_AdjustActiveContour [▶ 1138]

样本

对象检测 [▶ 2664]

6.1.4.14.1.1 F_VN_AdjustActiveContour



Adjust active contour (snake) in order to minimize its cummulative (internal and external) energy.

Syntax

Definition:

```
FUNCTION F_VN_AdjustActiveContour : HRESULT
VAR_INPUT
    ipImage          : ITcVnImage;
    ipActiveContour  : ITcVnContainer;
    fAlpha           : REAL;
    fBeta            : REAL;
    fGamma           : REAL;
    nWindowWidth     : UDINT;
    nWindowHeight    : UDINT;
    nMaxIterations   : UDINT;
    bUseGradient     : BOOL;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	Source image (1 channel, USINT elements)
ipActiveContour	ITcVnContainer [▶ 349]	Initial contour, which will be adjusted
fAlpha	REAL	Continuity energy coefficient
fBeta	REAL	Curvature energy coefficient
fGamma	REAL	Image energy coefficient
nWindowWidth	UDINT	Window width (3, 5, 7, ...)
nWindowHeight	UDINT	Window height (3, 5, 7, ...)
nMaxIterations	UDINT	Maximum iterations
bUseGradient	BOOL	If true, the gradient magnitude is used as image energy (otherwise: pixel intensity)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

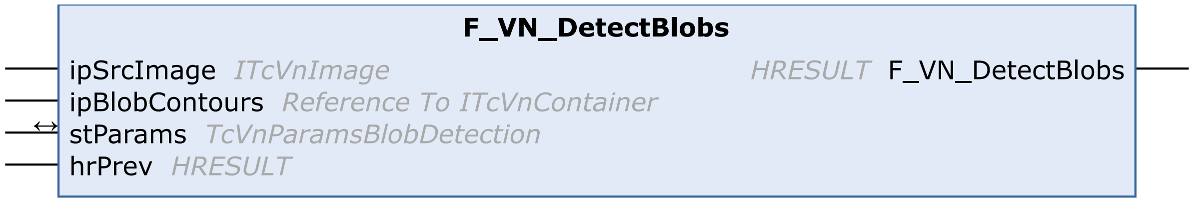
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.1.2 F_VN_DetectBlobs



Detects blob-contours. Applies a threshold, a contour search and offers several options for filtering the found contours. Provides easy setup for multiple thresholds and combination of results.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_DetectBlobs : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipBlobContours : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    stParams       : TcVnParamsBlobDetection;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (USINT or REAL, 1 channel or 3 channel (3 channel input is expected to be RGB and internally converted to Gray))
ipBlobContours	Reference To ITcVnContainer [▸ 349]	Returns a container which is filled with the found contours (ContainerType_Vector_Vector_TcVnPoint2_DINT; The elements of this container are single contours. Non-zero interface pointers are reused.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stParams	TcVnParamsBlobDetection [▸ 213]	Parameters to filter the detected contours.

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_DetectBlobs根据输入图像中可定义的特征检测斑点轮廓。在内部，这是通过类似于F_VN_FindContours [▶ 1147]的轮廓查找和随后根据轮廓特征对找到的轮廓进行过滤来进行的。

参数

输入图像

输入图像ipSrcImage必须是类型为USINT或REAL的单通道或三通道图像。三通道图像被解释为 RGB 并在内部转换为灰度图像。

找到的轮廓列表（返回值）

在输入图像中发现的所有轮廓都会在容器ipBlobContours中返回。

斑点检测的参数

TcVnParamsBlobDetection [▶ 213]结构的参数stParams定义了输入图像中的轮廓必须具备哪些特征才能被找到并返回。

专家参数

专家版F_VN_DetectBlobsExp [▶ 1141]包含额外的参数。

斑点参数

阈值

在第一步，在内部使用一个或多个二进制图像进行轮廓搜索。为此，使用一个或多个阈值和阈值类型将输入图像转换为二进制图像。

设置一个单一的阈值对应于调用F_VN_Threshold [▶ 1287]。为此，*fThresholdStep*必须设置为 0。仅使用*fMinThreshold*的值。

另外，也可以考虑几个阈值。第一个阈值是*fMinThreshold*。从这个值开始，对于下一个阈值增加*fThresholdStep*，直到阈值大于*fMaxThreshold*。例如，如果需要考虑三个阈值 100、150 和 200，则参数化如下所示：

```
stBlobParams.fMinThreshold     := 100;
stBlobParams.fMaxThreshold     := 200;
stBlobParams.fThresholdStep    := 50;
```

相同轮廓，不同的阈值

可能发现相同的轮廓而阈值不同。相同轮廓通过中心点比较进行识别。在这种情况下，*fMinBlobDistance*指定不同轮廓的最小中心距离。如果没有达到这个距离，则根据*eBlobCombination*（ETcVnBlobCombination [▶ 144]）选择轮廓。

此外，在不同的阈值下多次找到一个轮廓可能为一个过滤标准。一个轮廓必须被找到多少次才能被考虑，这一点通过*nMinRepeatability*指定。

过滤器

在第二步，根据定义的参数对找到的轮廓进行过滤。各个过滤标准可以通过布尔变量激活。

面积（单位：像素）

- 参数：bFilterByArea、fMinArea、fMaxArea
- 或者，可以通过F_VN_ContourArea [▶ 916]确定轮廓的面积。

圆度

- 参数：bFilterByCircularity、fMinCicularity、fMaxCircularity
- 值范围从 0 到 1；一个完美的圆的圆度为 1。
- 另外，轮廓的圆度可以通过F_VN_ContourCircularity [▶ 918]确定。

凸性

- 参数：bFilterByConvexity、fMinConvexity、fMaxConvexity
- 其数值范围为 0 至 1。完美凸型的凸度为 1。

偏心度

- 参数：bFilterByEccentricity、fMinEccentricity、fMaxEccentricity
- 值范围从 0 到 1；完美圆形的偏心度为 0。
- 另外，轮廓的偏心度可以通过F_VN_ContourEccentricity [▸ 920]确定。

惯性比

- 参数：bFilterByInertiaRatio、fMinInertiaRatio、fMaxInertiaRatio
- 数值范围从 0 到 1；具有相同高度和宽度的形状的惯性比率为 1。

应用

```
VAR
    stBlobParams     :   TcVnParamsBlobDetection;
END_VAR

stBlobParams.bFilterByArea  :=  TRUE;
stBlobParams.fMinArea       :=  100;
stBlobParams.fMaxArea       :=  100_000;

hr := F_VN_DetectBlobs(
    ipSrcImage       :=  ipImageIn,
    ipBlobContours   :=  ipContours,
    stParams         :=  stBlobsParams,
    hrPrev           :=  hr
);
```

样本

- 带看门狗监控的斑点检测 [▸ 2664]

相关函数

- F_VN_FindContours(Exp) [▸ 1147]用于寻找一般的轮廓
- F_VN_FindContourHierarchyExp [▸ 1143]返回层次结构
- F_VN_DetectBlobs [▸ 1139]对轮廓进行综合过滤

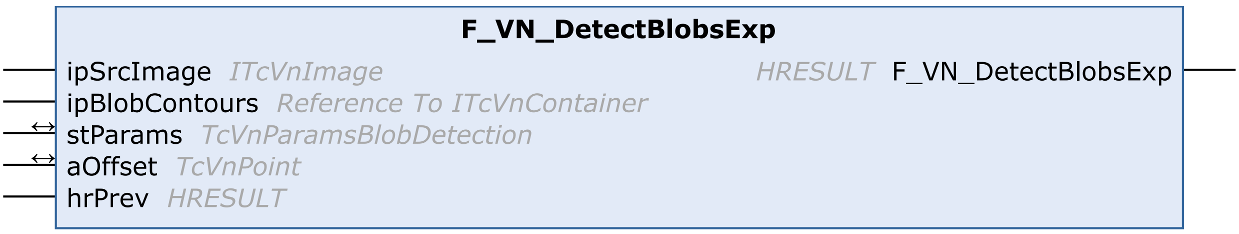
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.1.3 F_VN_DetectBlobsExp



Detects blob-contours. Applies a threshold, a contour search and offers several options for filtering the found contours. Provides easy setup for multiple thresholds and combination of results. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_DetectBlobsExp : HRESULT
VAR_INPUT
    ipSrcImage      : ITcVnImage;
    ipBlobContours : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    stParams        : TcVnParamsBlobDetection;
    aOffset         : TcVnPoint;
END_VAR
VAR_INPUT
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT or REAL, 1 channel or 3 channel (3 channel input is expected to be RGB and internally converted to Gray))
ipBlobContours	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the found contours (ContainerType_Vector_Vector_TcVnPoint2_DINT; The elements of this container are single contours. Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stParams	TcVnParamsBlobDetection [▶ 213]	Parameters to filter the detected contours.
aOffset	TcVnPoint [▶ 139]	Offset by which every contour point is shifted

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_DetectBlobsExp是F_VN_DetectBlobs [▶ 1139]的专家版本。它包含额外的参数。

参数

输入图像

输入图像ipSrcImage必须是类型为USINT或REAL的单通道或三通道图像。三通道图像被解释为 RGB 并在内部转换为灰度图像。

找到的轮廓列表（返回值）

在输入图像中发现的所有轮廓都会在容器ipBlobContours中返回。

斑点检测的参数

TcVnParamsBlobDetection [▶ 213]结构的参数stParams定义了输入图像中的轮廓必须具备哪些特征才能被找到并返回。

偏移量（专家）

偏移量aOffset定义了常量 X/Y 值，通过数值所有的轮廓点都被移位。当在一个 ROI 中寻找轮廓且它们的位置应该为原始图像提供参考时，这可能有所帮助。

相关函数

- F_VN_FindContours(Exp) [▶ 1147]用于寻找一般的轮廓
- F_VN_FindContourHierarchyExp [▶ 1143]返回层次结构
- F_VN_DetectBlobs [▶ 1139]对轮廓进行综合过滤

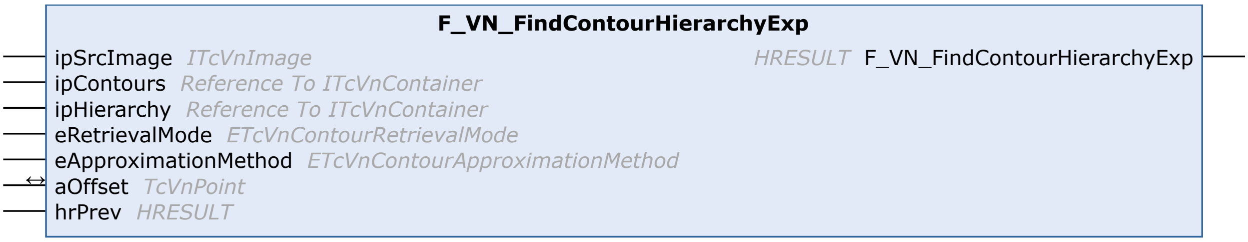
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.1.4 F_VN_FindContourHierarchyExp



Search for object contours in a binary image and determine their hierarchical relationship. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_FindContourHierarchyExp : HRESULT
VAR_INPUT
    ipSrcImage           : ITcVnImage;
    ipContours           : Reference To ITcVnContainer;
    ipHierarchy          : Reference To ITcVnContainer;
    eRetrievalMode       : ETcVnContourRetrievalMode;
    eApproximationMethod : ETcVnContourApproximationMethod;
END_VAR
VAR_IN_OUT
    aOffset              : TcVnPoint;
END_VAR
VAR_INPUT
    hrPrev               : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (1 channel, binary)
ipContours	Reference To ITcVnContainer [▸ 349]	Returns a container which is filled with the found contours (ContainerType_Vector_Vector_TcVnPoint2_DINT; The elements of this container are single contours. Non-zero interface pointers are reused.)
ipHierarchy	Reference To ITcVnContainer [▸ 349]	Returns a container which is filled with information on the found contours' hierarchy (ContainerType_Vector_TcVnVector4_DINT; The elements of this container are four-dimensional vectors containing the 0-based indices of the next [0] and the previous contour [1] at the same level, the first child [2], and the parent [3]. Non-zero interface pointers are reused.)
eRetrievalMode	ETcVnContourRetrievalMode [▸ 170]	Specifies which contours are retrieved and how their relationship is encoded
eApproximationMethod	ETcVnContourApproximationMethod [▸ 167]	Contour encoding
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aOffset	TcVnPoint [▸ 139]	Offset by which every contour point is shifted

Return value

HRESULT [▸ 122]

更多信息

通过返回轮廓层次结构，函数F_VN_FindContourHierarchyExp对函数F_VN_FindContoursExp [▸ 1149]进行了扩展。

参数

输入图像

输入图像ipSrcImage必须正好有 1 个通道，而且应该是二进制图像。

如果输入图像不是二进制图像，它将通过 1 的阈值在内部转换成二进制图像。因此，输入的图像应该通过阈值或二进制分割进行手动预处理。

找到的轮廓列表（返回值）

在输入图像中发现的所有轮廓都会在容器ipContours中返回。容器是一个二维的点阵列，因此，其类型为ContainerType_Vector_Vector_TcVnPoint2_DINT。

找到的轮廓会以ipContours返回。每个轮廓由一个点的总和组成，这些点被组合成一个容器 [▸ 132]。然后，另一个更高层次的容器将所有的轮廓合并。

层次结构（专家，返回值）

参数ipHierarchy返回在 TcVnVector4_DINT 类型的容器 [▸ 132]中发现的轮廓的层次结构。容器的向量有以下元素：

[0] - NEXT - 同层下一个轮廓的索引
[1] - PREVIOUS - 同层上一个轮廓的索引
[2] - CHILD - 第一个子轮廓的索引
[3] - PARENT - 父轮廓的索引

搜索模式（专家）

ETcVnContourRetrievalMode [▶ 170]类型的搜索模式eRetrievalMode定义了要寻找哪些轮廓：

- TCVN_CRM_LIST：所有轮廓
- TCVN_CRM_EXTERNAL：仅外部轮廓（如同F_VN_FindContours [▶ 1147]）
- TCVN_CRM_CONNECTED_COMPONENTS：具有两层层次结构的轮廓
- TCVN_CRM_TREE：所有轮廓；与函数F_VN_FindContourHierarchyExp [▶ 1143]交互使用，以确定轮廓的层次结构
- TCVN_CRM_FLOODFILL：返回在泛洪填充算法帮助下发现的轮廓（仅适用于 DINT 图像）

近似法（专家）

ETcVnContourApproximationMethod [▶ 167]类型的近似法eApproximationMethod指定是否在所有周围点的基础上描述轮廓（TCVN_CAM_NONE），或者是否在近似法的帮助下减少描述点的数量。

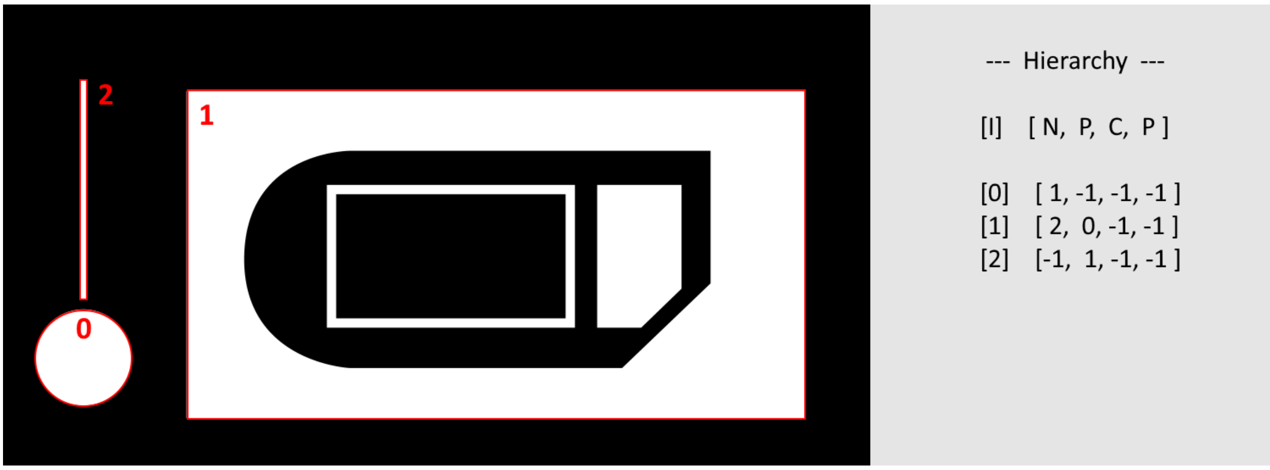
偏移量（专家）

偏移量aOffset定义了常量 X/Y 值，通过数值所有的轮廓点都被移位。当在一个 ROI 中寻找轮廓且它们的位置应该为原始图像提供参考时，这可能有所帮助。

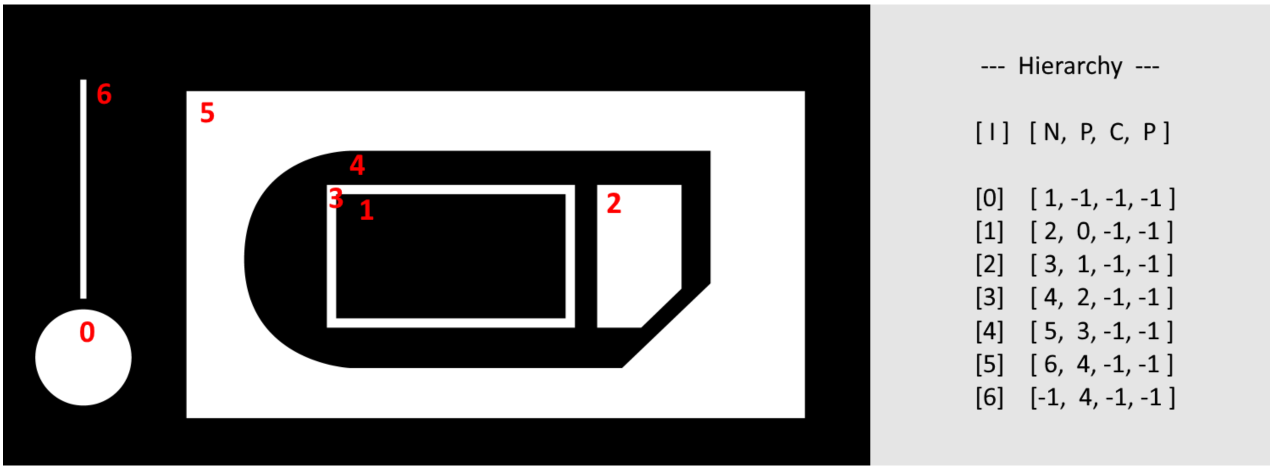
轮廓层次的示例

通过 eRetrievalMode，不仅可以指定是寻找所有的轮廓还是只寻找外轮廓，还可以指定如何对轮廓进行排序以及它们之间的层次关系。这可以从参数ipHierarchy读取：

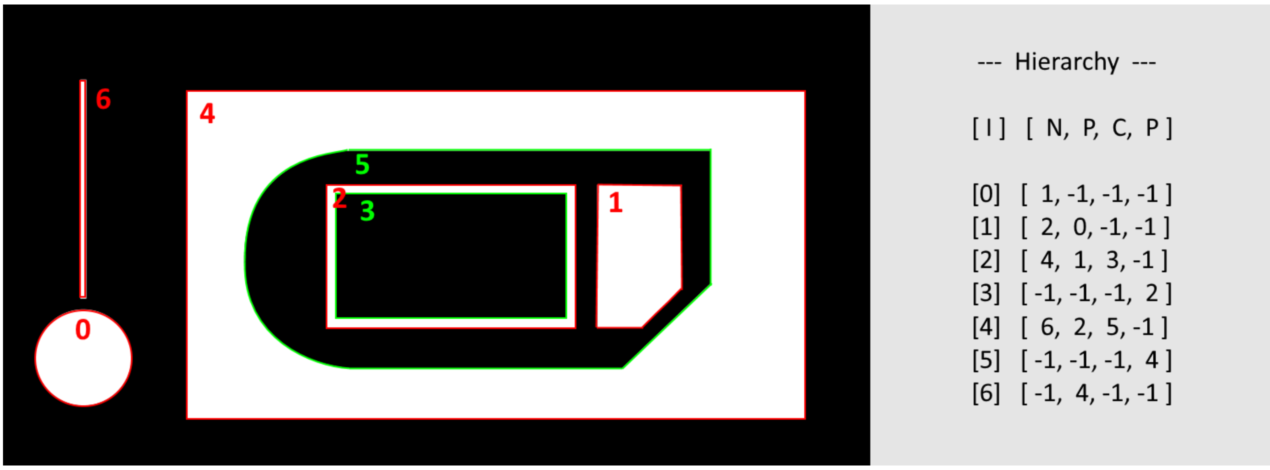
- TCVN_CRM_EXTERNAL
  仅外部轮廓以排序列表形式返回。子[2]和父[3]关系不确定，且处于无效状态-1。



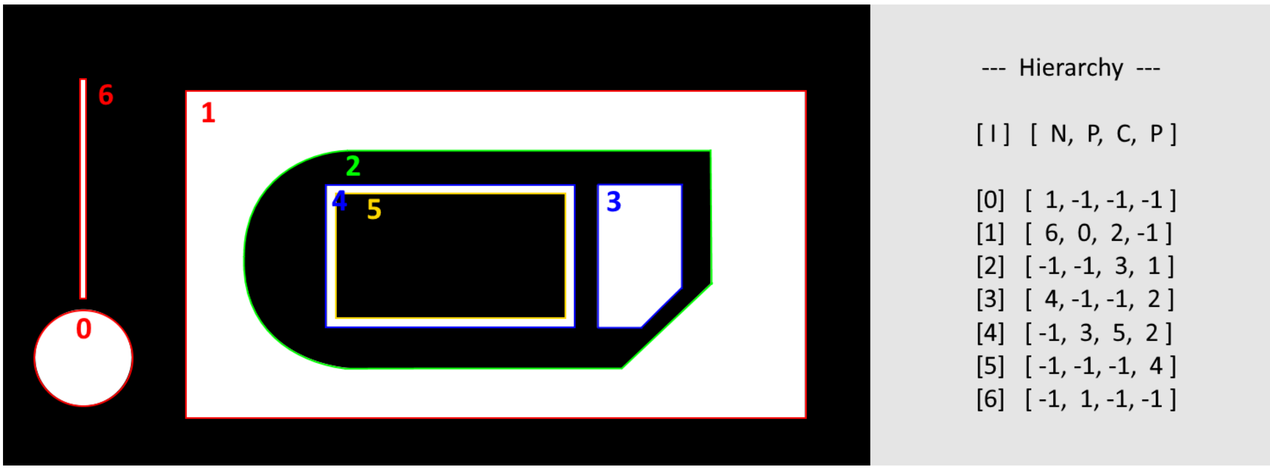
- TCVN_CRM_LIST
  所有轮廓都以排序列表形式返回。子[2]和父[3]关系不确定，且处于无效状态-1。



- TCVN_CRM_CONNECTED_COMPONENTS
  所有轮廓都是按照 2 个步骤的层次结构进行排序。首先考虑外部轮廓，然后在第二步考虑直接位于其内的轮廓。如果在第二步内存在其他轮廓，它们会像外轮廓一样被设置在第一步中
  在这个样本中，这意味着内部的子[2]轮廓（图像中的绿色）不能同时是父[3]轮廓（图像中的红色）。因此，将位于第 3 级树状结构中的轮廓再次被视为外部轮廓，并设置为第 1 级（父[3] = -1），在这种情况下，轮廓 1 和 2。



- TCVN_CRM_TREE
  所有轮廓均按照树状结构设置。轮廓有可能既是子[2]，又是父[3]，具体取决于它们如何相互嵌套。



样本

- 查找轮廓 - 层次结构和检索模式 [▶ 2674]用于参数ETcVnContourRetrievalMode [▶ 170]

- 查找轮廓 - 近似法 [▶ 2672]用于参数ETcVnContourApproximationMethod [▶ 167]
- 查找轮廓而不是斑点检测 [▶ 2670]用于与函数F_VN_DetectBlobs [▶ 1139]进行比较

相关函数

- F_VN_FindContours(Exp) [▶ 1147]用于寻找一般的轮廓
- F_VN_FindContourHierarchyExp [▶ 1143]返回层次结构
- F_VN_DetectBlobs [▶ 1139]对轮廓进行综合过滤

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.1.5 F_VN_FindContours

Search for object contours in a binary image. Returns only external contours.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_FindContours : HRESULT
VAR_INPUT
    ipSrcImage : ITcVnImage;
    ipContours : Reference To ITcVnContainer;
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel, binary)
ipContours	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the found contours (ContainerType_Vector_Vector_TcVnPoint2_DINT; The elements of this container are single contours. Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_FindContours检测输入图像中的外轮廓。

参数

输入图像

输入图像ipSrcImage必须正好有 1 个通道，而且应该是二进制图像。

如果输入图像不是二进制图像，它将通过 1 的阈值在内部转换成二进制图像。因此，输入的图像应该通过阈值或二进制分割进行手动预处理。

找到的轮廓列表（返回值）

在输入图像中发现的所有轮廓都会在容器ipContours中返回。容器是一个二维的点阵列，因此，其类型为ContainerType_Vector_Vector_TcVnPoint2_DINT。

找到的轮廓会以ipContours返回。每个轮廓由一个点的总和组成，这些点被组合成一个容器 [▶ 132]。然后，另一个更高层次的容器将所有的轮廓合并。

专家参数

专家版F_VN_FindContoursExp [▶ 1149]包含额外的参数。例如，这也使得内部轮廓能够被发现。

应用

例如，寻找和处理轮廓如下所示：

预处理

由于要使用二进制图像作为输入图像ipSrcImage，因此在调用函数之前应将阈值或二进制分割应用于图像，例如使用函数F_VN_Threshold [▶ 1287]：

```
hr := F_VN_Threshold(ipImageIn, ipImageWork, 128, 255, TCVN_TT_BINARY, hr);
```

函数调用

关于函数调用，没有什么需要注意。参数设置仅在专家版F_VN_FindContoursExp [▶ 1149]下需要。

```
hr := F_VN_FindContours(
    ipSrcImage  :=  ipImageIn,
    ipContours  :=  ipContours,
    hrPrev      :=  hr
);
```

进一步处理

F_VN_GetForwardIterator [▶ 738]可用于将迭代器应用于上层容器，从而可以通过F_VN_GetContainer [▶ 736]一个接一个地检索各个轮廓的容器，以便使用轮廓分析函数来分析它们。记住通过F_VN_IncrementIterator [▶ 740]递增迭代器。

```
hr := F_VN_GetForwardIterator(ipContours, ipIterator, hr);
WHILE SUCCEEDED(hr) AND_THEN ipIterator.CheckIfEnd() <> S_OK DO
    hr := F_VN_GetContainer( ipIterator, ipContour, hr);
    hr := F_VN_IncrementIterator(ipIterator, hr);

    // Analyse the single contour in ipContours with the contour analysis functions

END_WHILE
```

另外，也可以用F_VN_GetAt_ITcVnContainer [▶ 617]从所有轮廓的容器中检索出单个轮廓。

找到的轮廓数量可以通过F_VN_GetNumberOfElements [▶ 738]查询。

可视化

使用函数F_VN_DrawContours(Exp) [▶ 967]绘制所发现的轮廓。

```
hr := F_VN_DrawContours(
    ipContours      :=  ipContours,
    nContourIndex   :=  -1,
    ipDestImage     :=  ipImageRes,
    aColor          :=  aColorRed,
```

```
    nThickness      :=  5,
    hrPrev          :=  hr
);
```

轮廓列表ipContours中的所有轮廓都是通过设置nContoursIndex到-1进行绘制。

样本

- 查找轮廓而不是斑点检测 [▶ 2670]

相关函数

- F_VN_FindContours(Exp) [▶ 1147]用于寻找一般的轮廓
- F_VN_FindContourHierarchyExp [▶ 1143]返回层次结构
- F_VN_DetectBlobs [▶ 1139]对轮廓进行综合过滤

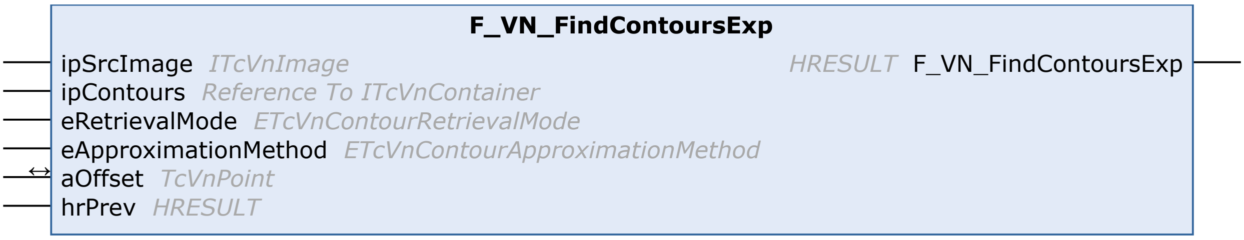
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.1.6 F_VN_FindContoursExp



Search for object contours in a binary image. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_FindContoursExp : HRESULT
VAR_INPUT
    ipSrcImage            : ITcVnImage;
    ipContours            : Reference To ITcVnContainer;
    eRetrievalMode        : ETcVnContourRetrievalMode;
    eApproximationMethod  : ETcVnContourApproximationMethod;
END_VAR
VAR_IN_OUT
    aOffset               : TcVnPoint;
END_VAR
VAR_INPUT
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (1 channel, binary)
ipContours	Reference To ITcVnContainer [▸ 349]	Returns a container which is filled with the found contours (ContainerType_Vector_Vector_TcVnPoint2_DINT; The elements of this container are single contours. Non-zero interface pointers are reused.)
eRetrievalMode	ETcVnContourRetrievalMode [▸ 170]	Specifies which contours are retrieved and how their relationship is encoded
eApproximationMethod	ETcVnContourApproximationMethod [▸ 167]	Contour encoding
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aOffset	TcVnPoint [▸ 139]	Offset by which every contour point is shifted

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_FindContoursExp是F_VN_FindContours [▸ 1147]的专家版本。它包含额外的参数。

参数

输入图像

输入图像ipSrcImage必须正好有 1 个通道，而且应该是二进制图像。

如果输入图像不是二进制图像，它将通过 1 的阈值在内部转换成二进制图像。因此，输入的图像应该通过阈值或二进制分割进行手动预处理。

找到的轮廓列表（返回值）

在输入图像中发现的所有轮廓都会在容器ipContours中返回。容器是一个二维的点阵列，因此，其类型为ContainerType_Vector_Vector_TcVnPoint2_DINT。

找到的轮廓会以ipContours返回。每个轮廓由一个点的总和组成，这些点被组合成一个容器 [▸ 132]。然后，另一个更高层次的容器将所有的轮廓合并。

搜索模式（专家）

ETcVnContourRetrievalMode [▸ 170]类型的搜索模式eRetrievalMode定义了要寻找哪些轮廓：

- TCVN_CRM_LIST：所有轮廓
- TCVN_CRM_EXTERNAL：仅外部轮廓（如同F_VN_FindContours [▸ 1147]）
- TCVN_CRM_CONNECTED_COMPONENTS：具有两层层次结构的轮廓
- TCVN_CRM_TREE：所有轮廓；与函数F_VN_FindContourHierarchyExp [▸ 1143]交互使用，以确定轮廓的层次结构
- TCVN_CRM_FLOODFILL：返回在泛洪填充算法帮助下发现的轮廓（仅适用于 DINT 图像）

近似法（专家）

ETcVnContourApproximationMethod [▶ 167]类型的近似法eApproximationMethod指定是否在所有周围点的基础上描述轮廓（TCVN_CAM_NONE），或者是否在近似法的帮助下减少描述点的数量。

偏移量（专家）

偏移量aOffset定义了常量 X/Y 值，通过数值所有的轮廓点都被移位。当在一个 ROI 中寻找轮廓且它们的位置应该为原始图像提供参考时，这可能有所帮助。

应用

例如，查找输入图像中的所有轮廓如下所示，其中的轮廓点不是近似的，而是在 x 方向上移动了 100px，在 y 方向上移动了 -50px：

```
VAR
    aOffset :    TcVnPoint    :=  [100, -50];
END_VAR

hr := F_VN_FindContours(
    ipSrcImage                  :=  ipImageIn,
    ipContours                  :=  ipContours,
    eRetrievalMode              :=  TCVN_CRM_LIST,
    eApproximationMethod        :=  TCVN_CAM_NONE,
    aOffset                     :=  aOffset,
    hrPrev                      :=  hr
);
```

样本

- 查找轮廓 – 层次结构和检索模式 [▶ 2674]用于参数ETcVnContourRetrievalMode [▶ 170]
- 查找轮廓 – 近似法 [▶ 2672]用于参数ETcVnContourApproximationMethod [▶ 167]
- 查找轮廓而不是斑点检测 [▶ 2670]用于与函数F_VN_DetectBlobs [▶ 1139]进行比较

相关函数

- F_VN_FindContours(Exp) [▶ 1147]用于寻找一般的轮廓
- F_VN_FindContourHierarchyExp [▶ 1143]返回层次结构
- F_VN_DetectBlobs [▶ 1139]对轮廓进行综合过滤

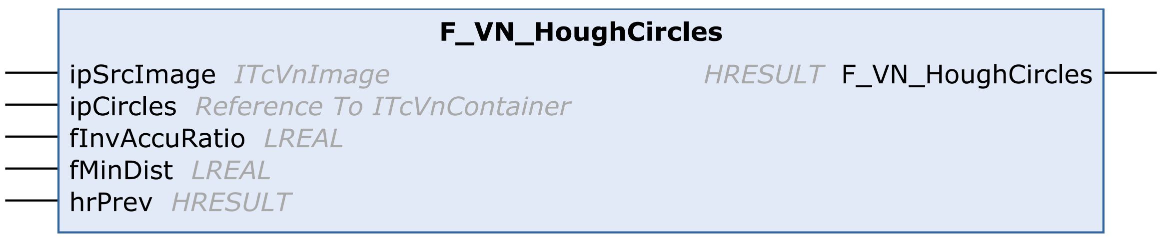
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.1.7 F_VN_HoughCircles



Search for circles using the Hough transform.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_HoughCircles : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipCircles      : Reference To ITcVnContainer;
    fInvAccuRatio : LREAL;
    fMinDist       : LREAL;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT, 1 channel, gray-level)
ipCircles	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the found circles (ContainerType_Vector_TcVnVector3_REAL; Each container element contains the x coordinate of the circle center [0], the y coordinate of the circle center [1], and the radius [2], respectively. Non-zero interface pointers are reused.)
fInvAccuRatio	LREAL	Inverted ratio of the accumulator size in relation to the source image's size (must be > 0. A value of 2 means that the size is halved in both directions.)
fMinDist	LREAL	Smallest allowed distance of two circles (must be > 0)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

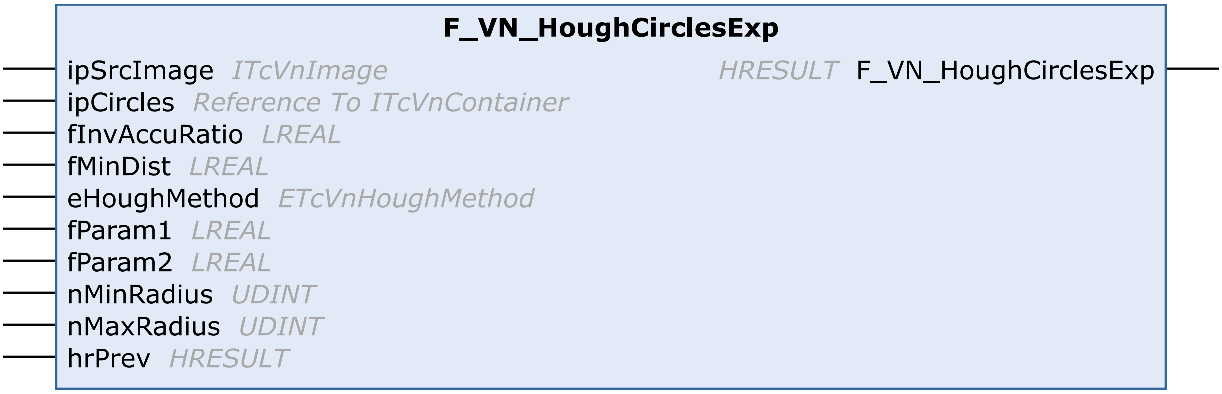
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.1.8 F_VN_HoughCirclesExp



Search for circles using the Hough transform. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_HoughCirclesExp : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipCircles     : Reference To ITcVnContainer;
    fInvAccuRatio : LREAL;
    fMinDist      : LREAL;
    eHoughMethod  : ETcVnHoughMethod;
    fParam1       : LREAL;
    fParam2       : LREAL;
    nMinRadius    : UDINT;
    nMaxRadius    : UDINT;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT, 1 channel, gray-level)
ipCircles	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the found circles (ContainerType_Vector_TcVnVector3_REAL; Each container element contains the x coordinate of the circle center [0], the y coordinate of the circle center [1], and the radius [2], respectively. Non-zero interface pointers are reused.)
fInvAccuRatio	LREAL	Inverted ratio of the accumulator size in relation to the source image's size (must be > 0. A value of 2 means that the size is halved in both directions.)
fMinDist	LREAL	Smallest allowed distance of two circles (must be > 0)
eHoughMethod	ETcVnHoughMethod [▶ 184]	Hough method to use (GRADIENT or GRADIENT_ALT)
fParam1	LREAL	First method specific parameter (GRADIENT, GRADIENT_ALT: upper threshold for canny edge detection, which must be > 0)
fParam2	LREAL	Second method specific parameter (GRADIENT: accumulator threshold for detecting circle centers, which must be > 0. GRADIENT_ALT: required circle perfectness, which must be > 0 and < 1, 1 would be a perfect circle)
nMinRadius	UDINT	Minimum circle radius allowed
nMaxRadius	UDINT	Maximum circle radius allowed (if 0, the value is internally set to max(image rows, image columns))
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

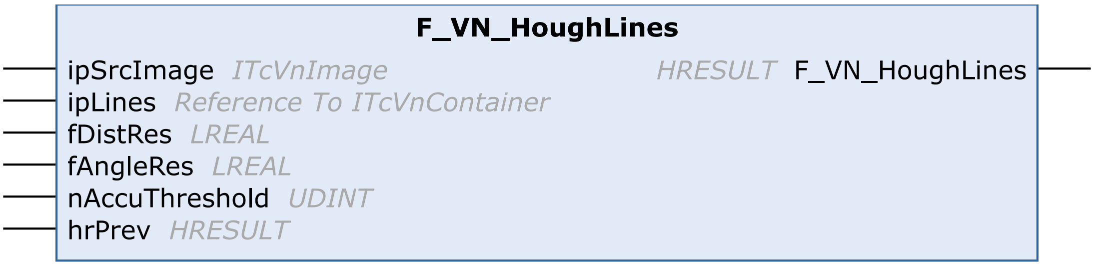
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.1.9 F_VN_HoughLines



Search for lines using the standard Hough transform.

Syntax

Definition:

```
FUNCTION F_VN_HoughLines : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipLines        : Reference To ITcVnContainer;
    fDistRes       : LREAL;
    fAngleRes      : LREAL;
    nAccuThreshold : UDINT;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (USINT, 1 channel, binary, may be modified by the function)
ipLines	Reference To ITcVnContainer [▸ 349]	Returns a container which is filled with the found lines (ContainerType_Vector_TcVnVector2_REAL; Each container element contains the distance from the origin [0] in pixels and the rotation angle [1] in radians, respectively. Non-zero interface pointers are reused.)
fDistRes	LREAL	Distance resolution of the accumulator (in pixels, must be > 0)
fAngleRes	LREAL	Angle resolution of the accumulator (in radians, must be > 0)
nAccuThreshold	UDINT	Accumulator threshold
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

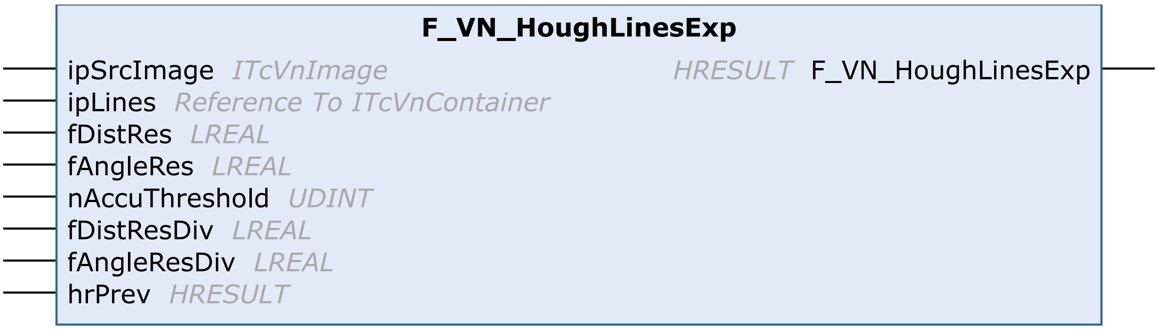
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.1.10 F_VN_HoughLinesExp



Search for lines using the standard Hough transform. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_HoughLinesExp : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipLines        : Reference To ITcVnContainer;
    fDistRes       : LREAL;
    fAngleRes      : LREAL;
    nAccuThreshold : UDINT;
    fDistResDiv    : LREAL;
    fAngleResDiv   : LREAL;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (USINT, 1 channel, binary, may be modified by the function)
ipLines	Reference To ITcVnContainer [▸ 349]	Returns a container which is filled with the found lines (ContainerType_Vector_TcVnVector2_REAL; Each container element contains the distance from the origin [0] in pixels and the rotation angle [1] in radians, respectively. Non-zero interface pointers are reused.)
fDistRes	LREAL	Distance resolution of the accumulator (in pixels, must be > 0)
fAngleRes	LREAL	Angle resolution of the accumulator (in radians, must be > 0)
nAccuThreshold	UDINT	Accumulator threshold
fDistResDiv	LREAL	Divisor of the distance resolution for the multi-scale Hough transform (>= 0, must be > 0 if fAngleResDiv is > 0)
fAngleResDiv	LREAL	Divisor of the angle resolution for the multi-scale Hough transform (>= 0, must be > 0 if fDistResDiv is > 0)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

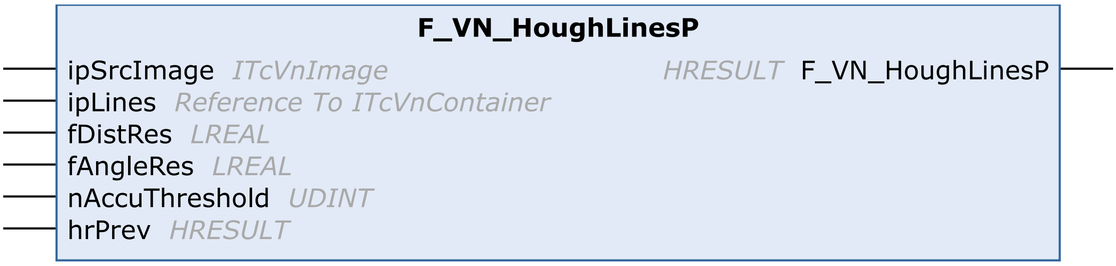
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.1.11 F_VN_HoughLinesP



Search for line segments using the probabilistic Hough transform.

Syntax

Definition:

```
FUNCTION F_VN_HoughLinesP : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipLines        : Reference To ITcVnContainer;
    fDistRes       : LREAL;
    fAngleRes      : LREAL;
    nAccuThreshold : UDINT;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT, 1 channel, binary, may be modified by the function)
ipLines	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the found line segments (ContainerType_Vector_TcVnVector4_DINT; Each container element contains the two ending points. Non-zero interface pointers are reused.)
fDistRes	LREAL	Distance resolution of the accumulator (in pixels, must be > 0)
fAngleRes	LREAL	Angle resolution of the accumulator (in radians, must be > 0)
nAccuThreshold	UDINT	Accumulator threshold
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

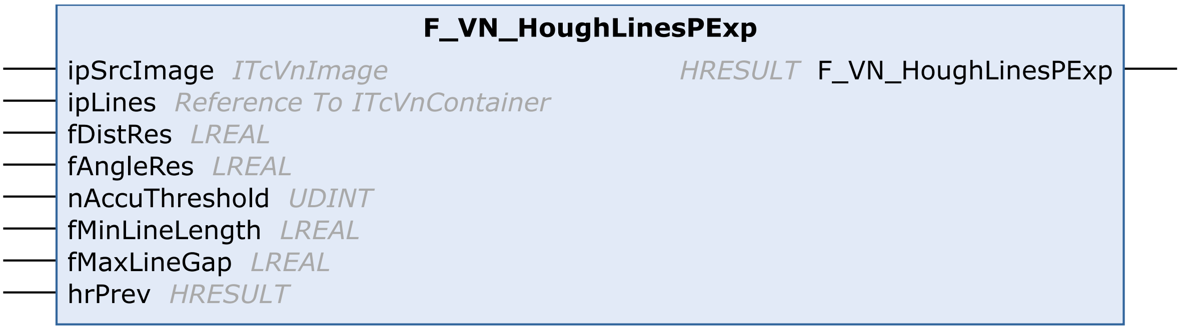
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.1.12 F_VN_HoughLinesPExp



Search for line segments using the probabilistic Hough transform. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_HoughLinesPExp : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipLines        : Reference To ITcVnContainer;
    fDistRes       : LREAL;
    fAngleRes      : LREAL;
    nAccuThreshold : UDINT;
    fMinLineLength : LREAL;
    fMaxLineGap    : LREAL;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT, 1 channel, binary, may be modified by the function)
ipLines	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the found line segments (ContainerType_Vector_TcVnVector4_DINT; Each container element contains the two ending points. Non-zero interface pointers are reused.)
fDistRes	LREAL	Distance resolution of the accumulator (in pixels, must be > 0)
fAngleRes	LREAL	Angle resolution of the accumulator (in radians, must be > 0)
nAccuThreshold	UDINT	Accumulator threshold
fMinLineLength	LREAL	Minimum line length to search for
fMaxLineGap	LREAL	Maximum gap between points on the same line
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

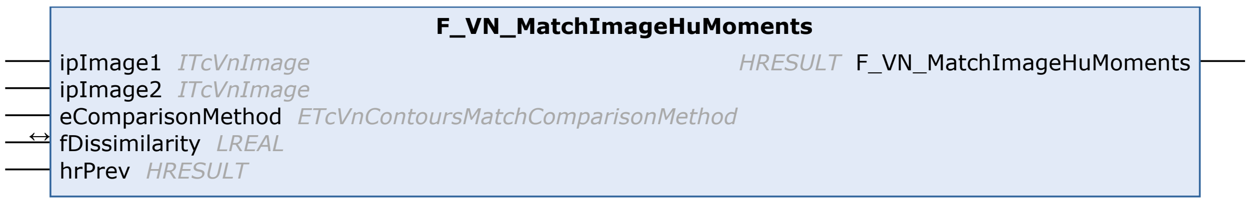
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.1.13 F_VN_MatchImageHuMoments



Compare two images using the Hu moment invariants.

Syntax

Definition:

```
FUNCTION F_VN_MatchImageHuMoments : HRESULT
VAR_INPUT
    ipImage1          : ITcVnImage;
    ipImage2          : ITcVnImage;
    eComparisonMethod : ETcVnContoursMatchComparisonMethod;
END_VAR
VAR_IN_OUT
    fDissimilarity    : LREAL;
END_VAR
```

```
VAR_INPUT
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage1	ITcVnImage [▶ 390]	First image (1 channel)
ipImage2	ITcVnImage [▶ 390]	Second image (1 channel)
eComparisonMethod	ETcVnContoursMatchComparisonMethod [▶ 172]	Method used for comparing the Hu moment invariants of the images
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fDissimilarity	LREAL	Returns the dissimilarity of the image Hu moment invariants depending on the chosen comparison method

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.1.14 F_VN_MatchTemplate

Match a template image with every location in the source image (using the TCVN_TMM_CCORR_NORMED method) and save the comparison results.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_MatchTemplate : HRESULT
VAR_INPUT
    ipSrcImage      : ITcVnImage;
    ipTemplateImage : ITcVnImage;
    ipResultImage   : Reference To ITcVnImage;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT or REAL, 1 or 3 channels)
ipTemplateImage	ITcVnImage [▶ 390]	Template image (same type as ipSrcImage, smaller width and height)
ipResultImage	Reference To ITcVnImage [▶ 390]	Returns the result image (REAL, normalized to [0..1], 1 channel, dimensions: (ipSrcImage.width - ipTemplateImage.width + 1) x (ipSrcImage.height - ipTemplateImage.height + 1). The best match is the global maximum. The position in ipResultImage is the top-left corner of ipTemplateImage position in ipSrcImage)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

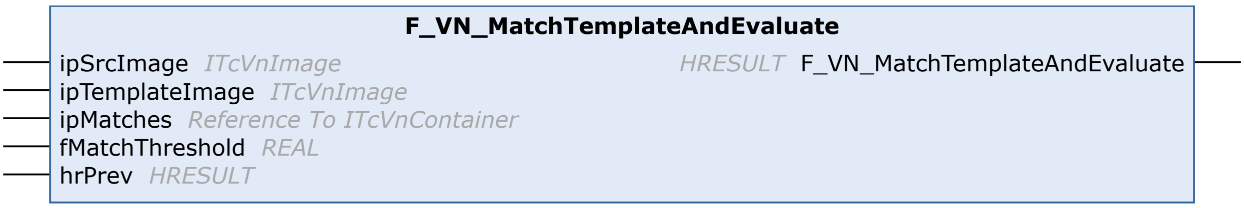
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.1.15 F_VN_MatchTemplateAndEvaluate



Match a template image with every location in the source image (using the TCVN_TMM_CCORR_NORMED method) and evaluate the comparison results. Returns a sorted list of possible matches (best match first).
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_MatchTemplateAndEvaluate : HRESULT
VAR_INPUT
    ipSrcImage       : ITcVnImage;
    ipTemplateImage  : ITcVnImage;
    ipMatches        : Reference To ITcVnContainer;
    fMatchThreshold  : REAL;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT or REAL, 1 or 3 channels)
ipTemplateImage	ITcVnImage [▶ 390]	Template image (same type as ipSrcImage, smaller width and height)
ipMatches	Reference To ITcVnContainer [▶ 349]	Returns the matching positions (ContainerType_Vector_TcVnPoint2_DINT, where each element represents the top-left corner of ipTemplateImage) in ipSrcImage, sorted by relevance (best match first)
fMatchThreshold	REAL	Threshold to separate relevant from irrelevant matches (0..1, 1.0 would be a perfect match. To find a suitable value, you could evaluate some sample result images of F_VN_MatchTemplate.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

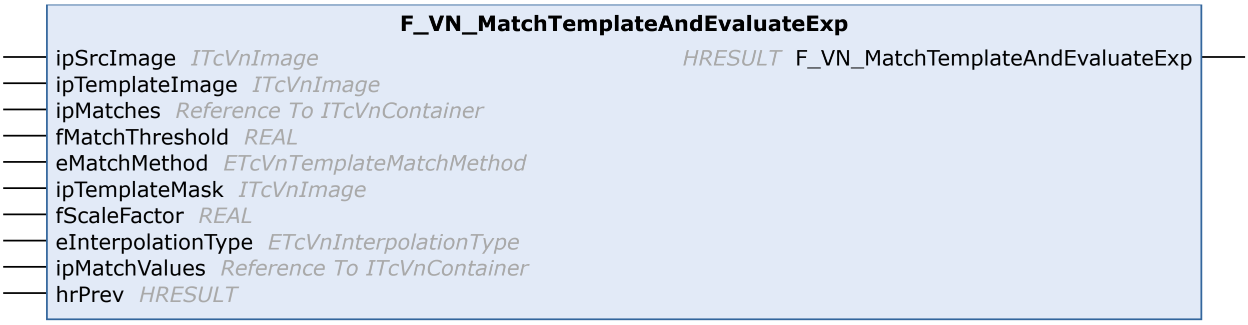
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.1.16 F_VN_MatchTemplateAndEvaluateExp



Match a template image with every location in the source image and evaluate the comparison results. Returns a sorted list of possible matches (best match first). (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_MatchTemplateAndEvaluateExp : HRESULT
VAR_INPUT
    ipSrcImage          : ITcVnImage;
    ipTemplateImage     : ITcVnImage;
    ipMatches           : Reference To ITcVnContainer;
    fMatchThreshold     : REAL;
    eMatchMethod        : ETcVnTemplateMatchMethod;
    ipTemplateMask      : ITcVnImage;
```

```
    fScaleFactor        : REAL;
    eInterpolationType : ETcVnInterpolationType;
    ipMatchValues       : Reference To ITcVnContainer;
    hrPrev              : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT or REAL, 1 or 3 channels)
ipTemplateImage	ITcVnImage [▶ 390]	Template image (same type as ipSrcImage, smaller width and height)
ipMatches	Reference To ITcVnContainer [▶ 349]	Returns the matching positions (ContainerType_Vector_TcVnPoint2_DINT, where each element represents the top-left corner of ipTemplateImage) in ipSrcImage, sorted by relevance (best match first)
fMatchThreshold	REAL	Threshold to separate relevant from irrelevant matches (0..1 for NORMED methods, otherwise dependent on template size and content. To find a suitable value, you could evaluate some sample result images of F_VN_MatchTemplateExp.)
eMatchMethod	ETcVnTemplateMatchMethod [▶ 199]	Specifies the template match method
ipTemplateMask	ITcVnImage [▶ 390]	Optional mask for ipTemplateImage (same type and size as ipTemplateImage)
fScaleFactor	REAL	Factor (0..1] to reduce source and template image width and height for better performance (but less accuracy!)
eInterpolationType	ETcVnInterpolationType [▶ 185]	Image resize interpolation type (only used if fScaleFactor != 1, TCVN_IT_BILINEAR recommended for most cases)
ipMatchValues	Reference To ITcVnContainer [▶ 349]	Optionally returns the matching values (ContainerType_Vector_REAL, same size and sort order as ipMatches. Set to 0 if not required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

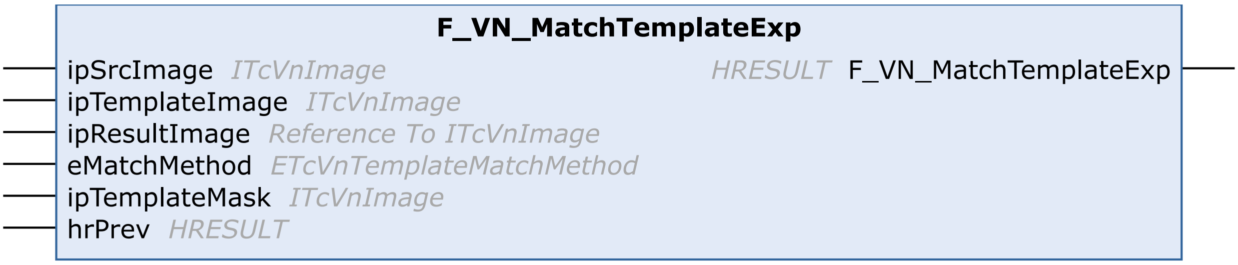
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.1.17 F_VN_MatchTemplateExp



Match a template image with every location in the source image and save the comparison results. (expert function)
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_MatchTemplateExp : HRESULT
VAR_INPUT
    ipSrcImage      : ITcVnImage;
    ipTemplateImage : ITcVnImage;
    ipResultImage   : Reference To ITcVnImage;
    eMatchMethod    : ETcVnTemplateMatchMethod;
    ipTemplateMask  : ITcVnImage;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT or REAL, 1 or 3 channels)
ipTemplateImage	ITcVnImage [▶ 390]	Template image (same type as ipSrcImage, smaller width and height)
ipResultImage	Reference To ITcVnImage [▶ 390]	Returns the result image (REAL, 1 channel, dimensions: (ipSrcImage.width - ipTemplateImage.width + 1) x (ipSrcImage.height - ipTemplateImage.height + 1). The best match is the global minimum (SQDIFF(_NORMED)) or maximum (CCORR(_NORMED), CCOEFF(_NORMED)). The position in ipResultImage is the top-left corner of ipTemplateImage position in ipSrcImage)
eMatchMethod	ETcVnTemplateMatchMethod [▶ 199]	Specifies the template match method
ipTemplateMask	ITcVnImage [▶ 390]	Optional mask for ipTemplateImage (same type and size as ipTemplateImage)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

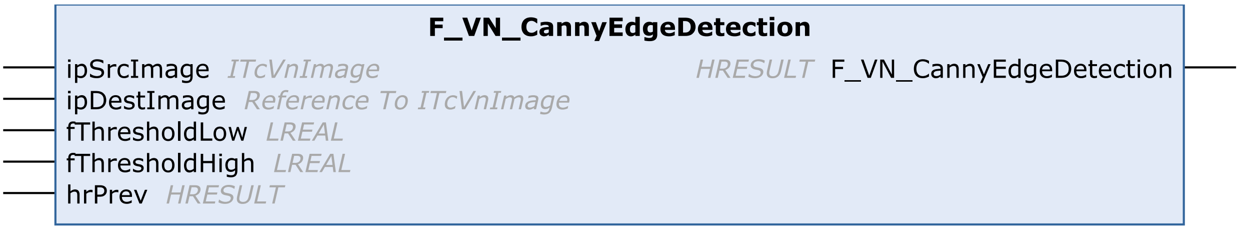
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.2 F_VN_CannyEdgeDetection



Find edges using the Canny edge detection algorithm.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_CannyEdgeDetection : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipDestImage    : Reference To ITcVnImage;
    fThresholdLow  : LREAL;
    fThresholdHigh : LREAL;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (elements of type USINT)
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (elements of type USINT. An appropriate destination image will be created if required.)
fThresholdLow	LREAL	Low threshold
fThresholdHigh	LREAL	High threshold
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

这个函数在图像中寻找边缘，并在黑色背景上通过白线进行展示说明。此外，这个专家版 F_VN_CannyEdgeDetectionExp [▸ 1166]存在高级参数化选项。

算法

Canny Edge 边缘检测是一种多阶段的算法，用于实现最稳健的边缘检测。它包括下列步骤：

1. 通过高斯过滤器减少噪音。
2. 通过索贝尔算子寻找边缘。
3. 通过非最大限度的抑制来进行线条瘦身，即所产生的线条为 1px 厚。

4. 滞后，所以只留下粗线（见下面的阈值参数）。

参数

原始图像

预计原始图像ipSrcImage是一个单通道 8 位强度图像（像素格式TCVN_ET_USINT）。因此，可以使用灰度图像或彩色图像的各个通道。

结果图像

结果图像ipDestImage格式相同。

阈值

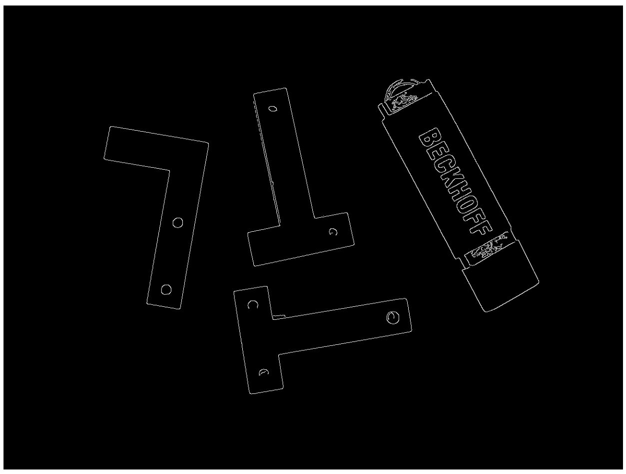
两个阈值fThresholdLow和fThresholdHigh决定接受哪些发现的边缘：接受梯度高于fThresholdHigh的像素。丢弃颜色梯度低于fThresholdLow的像素。中间梯度的像素仅在与已经接受的像素相连接时才会被接受。

通过这两个阈值，可实现灵活设置。通过设置两个相同数值，也可以实现硬阈值。

如果阈值设置过高，可能会丢失重要的边缘信息。如果阈值过低，噪声和其他不相关的信息可能会解释为边缘。

应用

```
hr := F_VN_CannyEdgeDetection(
    ipSrcImage      :=  ipImageIn,
    ipDestImage     :=  ipImageWork,
    fThresholdLow   :=  0,
    fThresholdHigh  :=  200,
    hr
);
```



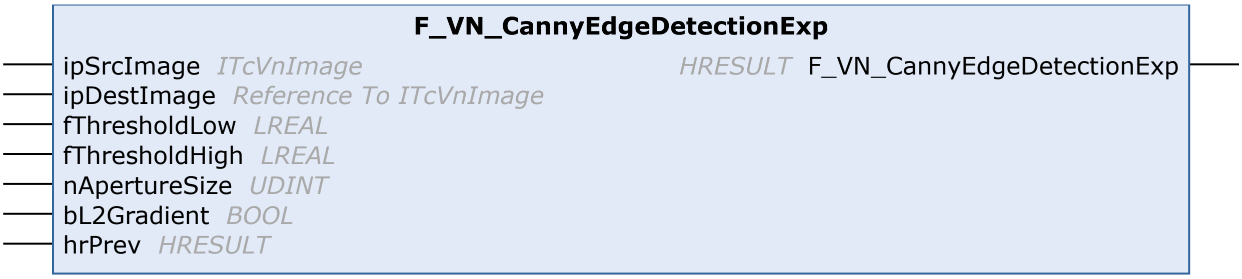
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.3 F_VN_CannyEdgeDetectionExp



Find edges using the Canny edge detection algorithm. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_CannyEdgeDetectionExp : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipDestImage    : Reference To ITcVnImage;
    fThresholdLow  : LREAL;
    fThresholdHigh : LREAL;
    nApertureSize  : UDINT;
    bL2Gradient    : BOOL;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (elements of type USINT)
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (elements of type USINT. An appropriate destination image will be created if required.)
fThresholdLow	LREAL	Low threshold
fThresholdHigh	LREAL	High threshold
nApertureSize	UDINT	Aperture size for the Sobel operator (3, 5, 7)
bL2Gradient	BOOL	If true, the more accurate (and slower) L2 norm is used instead of the L1 norm
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

该函数是F_VN_CannyEdgeDetection [▸ 1164]的专家版。它包含高级参数化选项。

参数

在该专家版中额外提供了以下参数：

光圈大小

仅奇数值3或更大对于nApertureSize有效。

L1/L2 标准

两个标准可用于计算梯度。L1 标准更快。反之，L2 标准则更准确。

$$L_1 = \left|\frac{dl}{dx}\right| + \left|\frac{dl}{dy}\right| \qquad L_2 = \sqrt{\left(\frac{dl}{dx}\right)^2 + \left(\frac{dl}{dy}\right)^2}$$

将参数bL2Gradient设为FALSE，以使用 L1 标准，设为TRUE，以使用 L2 标准。

应用

例如，这个函数的应用可以如下所示：

```
hr := F_VN_CannyEdgeDetectionExp(
    ipSrcImage      :=  ipImageIn,
    ipDestImage     :=  ipImageWork,
    fThresholdLow   :=  0,
    fThresholdHigh  :=  200,
    nApertureSize   :=  5,
    bL2Gradient     :=  TRUE,
    hr
);
```

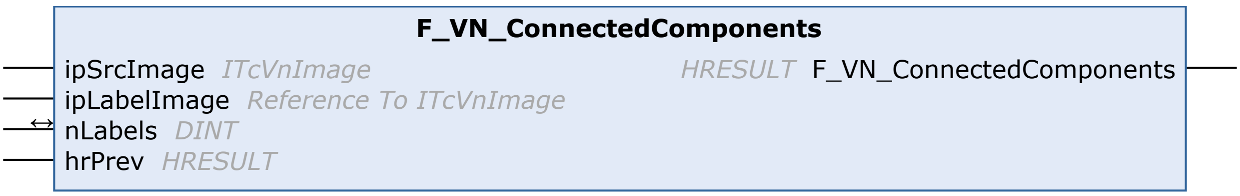
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.4 F_VN_ConnectedComponents



Computes the connected components of an image.

Syntax

Definition:

```
FUNCTION F_VN_ConnectedComponents : HRESULT
VAR_INPUT
    ipSrcImage   : ITcVnImage;
    ipLabelImage : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    nLabels      : DINT;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT, 1 channel, binary)
ipLabelImage	Reference To ITcVnImage [▶ 390]	Returns the labels for each source image pixel (1 channel, DINT. An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nLabels	DINT	Returns the number of labels

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.5 F_VN_ConnectedComponentsExp

Computes the connected components of an image. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ConnectedComponentsExp : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipLabelImage  : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    nLabels       : DINT;
END_VAR
VAR_INPUT
    eConnectivity : ETcVnPixelConnectivity;
    eLabelType    : ETcVnElementType;
    eAlgorithm    : ETcVnConnectedComponentsAlgorithm;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT, 1 channel, binary)
ipLabelImage	Reference To ITcVnImage [▶ 390]	Returns the labels for each source image pixel (1 channel, type depends on eLabelType. An appropriate destination image will be created if required.)
eConnectivity	ETcVnPixelConnectivity [▶ 192]	Selects if 4- or 8-way pixel connectivity should be used
eLabelType	ETcVnElementType [▶ 178]	Selects the type of ipLabelImage (only UINT or DINT supported)
eAlgorithm	ETcVnConnectedComponentsAlgorithm [▶ 166]	Selects the applied algorithm
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nLabels	DINT	Returns the number of labels

Return value

HRESULT [▶ 122]

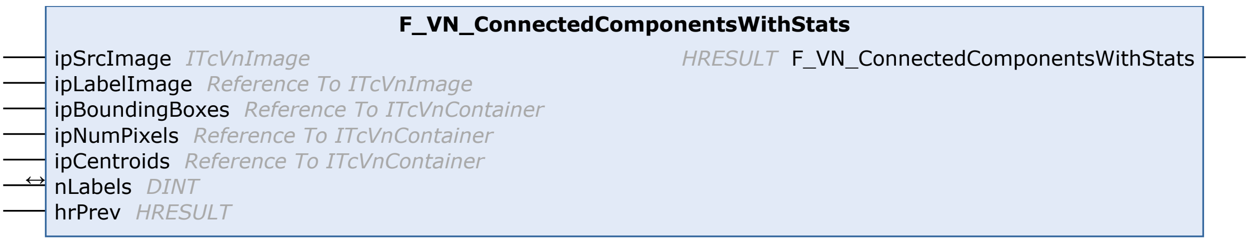
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.6 F_VN_ConnectedComponentsWithStats



Computes the connected components and corresponding statistics of an image.

Syntax

Definition:

```
FUNCTION F_VN_ConnectedComponentsWithStats : HRESULT
VAR_INPUT
    ipSrcImage      : ITcVnImage;
    ipLabelImage    : Reference To ITcVnImage;
    ipBoundingBoxes : Reference To ITcVnContainer;
    ipNumPixels     : Reference To ITcVnContainer;
    ipCentroids     : Reference To ITcVnContainer;
```

```
END_VAR
VAR_IN_OUT
    nLabels          : DINT;
END_VAR
VAR_INPUT
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT, 1 channel, binary)
ipLabelImage	Reference To ITcVnImage [▶ 390]	Returns the labels for each source image pixel (1 channel, DINT. An appropriate destination image will be created if required.)
ipBoundingBoxes	Reference To ITcVnContainer [▶ 349]	Returns the bounding boxes for each labeled region. (ContainerType_Vector_TcVnRectangle_DINT; Non-zero interface pointers are reused.)
ipNumPixels	Reference To ITcVnContainer [▶ 349]	Returns the number of pixels for each labeled region. (ContainerType_Vector_DINT; Non-zero interface pointers are reused.)
ipCentroids	Reference To ITcVnContainer [▶ 349]	Returns the centroids for each labeled region. (ContainerType_Vector_TcVnPoint2_LREAL; Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nLabels	DINT	Returns the number of labels

Return value

HRESULT [▶ 122]

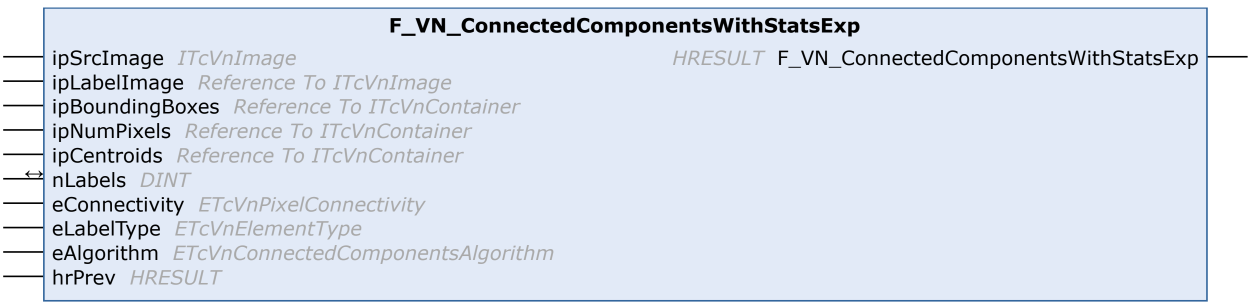
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.7 F_VN_ConnectedComponentsWithStatsExp



Computes the connected components and corresponding statistics of an image. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ConnectedComponentsWithStatsExp : HRESULT
VAR_INPUT
    ipSrcImage       : ITcVnImage;
    ipLabelImage     : Reference To ITcVnImage;
    ipBoundingBoxes  : Reference To ITcVnContainer;
    ipNumPixels      : Reference To ITcVnContainer;
    ipCentroids      : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    nLabels          : DINT;
END_VAR
VAR_INPUT
    eConnectivity    : ETcVnPixelConnectivity;
    eLabelType       : ETcVnElementType;
    eAlgorithm       : ETcVnConnectedComponentsAlgorithm;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (USINT, 1 channel, binary)
ipLabelImage	Reference To ITcVnImage [▸ 390]	Returns the labels for each source image pixel (1 channel, DINT. An appropriate destination image will be created if required.)
ipBoundingBoxes	Reference To ITcVnContainer [▸ 349]	Returns the bounding boxes for each labeled region. (ContainerType_Vector_TcVnRectangle_DINT; Non-zero interface pointers are reused.)
ipNumPixels	Reference To ITcVnContainer [▸ 349]	Returns the number of pixels for each labeled region. (ContainerType_Vector_DINT; Non-zero interface pointers are reused.)
ipCentroids	Reference To ITcVnContainer [▸ 349]	Returns the centroids for each labeled region. (ContainerType_Vector_TcVnPoint2_LREAL; Non-zero interface pointers are reused.)
eConnectivity	ETcVnPixelConnectivity [▸ 192]	Selects if 4- or 8-way pixel connectivity should be used
eLabelType	ETcVnElementType [▸ 178]	Selects the type of ipLabelImage (only UINT or DINT supported)
eAlgorithm	ETcVnConnectedComponentsAlgorithm [▸ 166]	Selects the applied algorithm
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nLabels	DINT	Returns the number of labels

Return value

HRESULT [▸ 122]

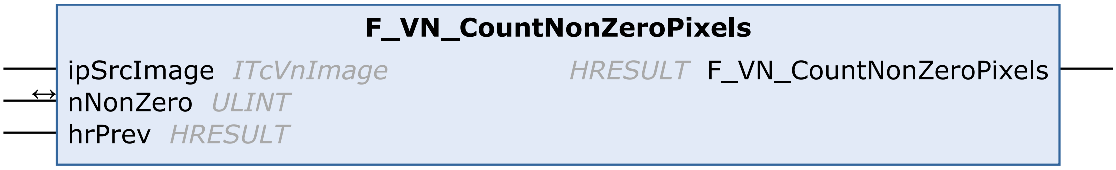
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.8 F_VN_CountNonZeroPixels



Counts the non-zero pixels in a single-channel image (e.g. useful to analyze threshold results).

Syntax

Definition:

```
FUNCTION F_VN_CountNonZeroPixels : HRESULT
VAR_INPUT
    ipSrcImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    nNonZero    : ULINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nNonZero	ULINT	Returns the number of non-zero pixels in ipSrcImage

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.9 F_VN_DistanceTransformation

F_VN_DistanceTransformation

ipSrcImage *ITcVnImage* — *HRESULT* F_VN_DistanceTransformation
ipDestImage *Reference To ITcVnImage*
eDistanceType *ETcVnDistanceType*
eMaskSize *ETcVnDistanceTransformationMask*
hrPrev *HRESULT*

Calculates the distance transformation, which is the distance to the closest zero pixel in a binary image.

Syntax

Definition:

```
FUNCTION F_VN_DistanceTransformation : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipDestImage   : Reference To ITcVnImage;
    eDistanceType : ETcVnDistanceType;
    eMaskSize     : ETcVnDistanceTransformationMask;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT, 1 channel)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (REAL, 1 channel. An appropriate destination image will be created if required.)
eDistanceType	ETcVnDistanceType [▶ 175]	Distance computation method (supported: L1, L2, C)
eMaskSize	ETcVnDistanceTransformationMask [▶ 174]	Size of the distance transformation mask
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

这个函数执行图像的距离变换。此外，还有一个专家版F_VN_DistanceTransformationExp [▶ 1175]，配有高级参数化选项。

参数

下列参数在基本版中可用：

原始图像

预计原始图像ipSrcImage为 1 通道的 8 位图像（元素类型TCVN_ET_USINT）。其他图像格式必须首先进行转换。

结果图像

结果图像ipDestImage以元素类型TCVN_ET_REAL返回。每个像素的值描述了各自像素与原始图像中最近的 0 值像素之间的距离。

> **在 ADS 图像查看中显示**
> 为了能够在 ADS 图像查看中显示结果图像，必须首先将其转换为兼容的图像格式。根据使用情况，对像素值进行缩放可能有用。

距离类型

距离类型eDistanceType确定计算像素和最近 0 值像素之间距离的方程式。支持枚举ETcVnDistanceType [▶ 175]的下列距离类型：

- TCVN_DT_L1
- TCVN_DT_L2
- TCVN_DT_C

掩码尺寸

掩码大小eMaskSize描述了，在选择 L2 标准作为距离类型的情况下，欧几里得距离的近似值。

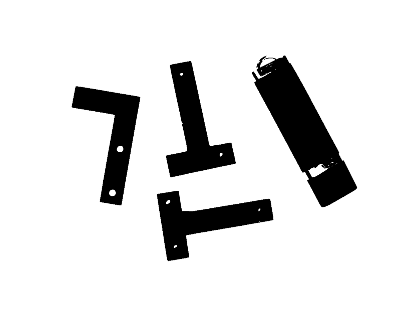
- TCVN_DTM_PRECISE：精确计算欧几里得距离。
- TCVN_DTM_3：欧几里得距离被近似为 3x3 掩码（水平、垂直和对角线）内距离元素的总和。
- TCVN_DTM_5：欧几里得距离被近似为 5x5 掩码（水平、垂直、对角线和骑士的移动（如国际象棋））内的距离元素之和。

掩码大小对距离类型 L1 和 C 没有影响。相反，内部总是使用 3x3 掩码。

应用

例如，这个函数的应用可以如下所示：

```
// convert to binary image
hr := F_VN_Threshold(ipImageIn, ipImageWork, 128, 255, TCVN_TT_BINARY, hr);
// apply distance transformation
hr := F_VN_DistanceTransformation(
    ipSrcImage      :=  ipImageWork,
    ipDestImage     :=  ipImageWork,
    eDistanceType   :=  TCVN_DT_C,
    eMaskSize       :=  TCVN_DTM_3,
    hr
);
// convert image to display it in ADS Image Watch
hr := F_VN_ConvertElementType(ipImageWork, ipImageWork, TCVN_ET_USINT, hr);
```

原始图像	二进制图像	结果图像
		

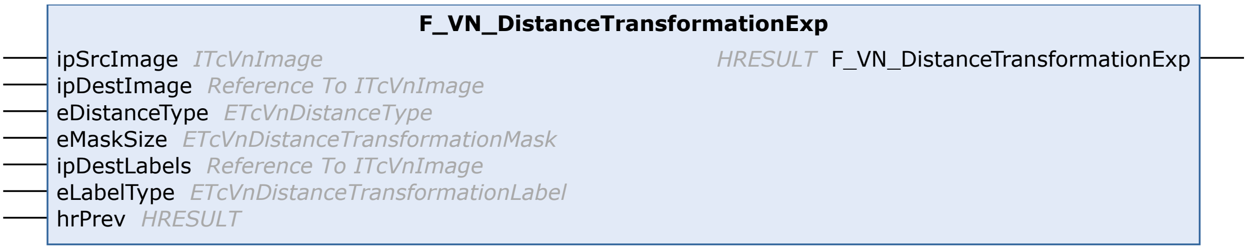
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.10 F_VN_DistanceTransformationExp



Calculates the distance transformation, which is the distance to the closest zero pixel in a binary image. Additionally, an image with component labels is created (discrete Voronoi diagram). (expert function)

Syntax

Definition:

```
FUNCTION F_VN_DistanceTransformationExp : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipDestImage   : Reference To ITcVnImage;
    eDistanceType : ETcVnDistanceType;
    eMaskSize     : ETcVnDistanceTransformationMask;
    ipDestLabels  : Reference To ITcVnImage;
    eLabelType    : ETcVnDistanceTransformationLabel;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (USINT, 1 channel)
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (REAL, 1 channel. An appropriate destination image will be created if required.)
eDistanceType	ETcVnDistanceType [▸ 175]	Distance computation method (supported: L1, L2, C)
eMaskSize	ETcVnDistanceTransformationMask [▸ 174]	Size of the distance transformation mask (PRECISE not supported for label computation)
ipDestLabels	Reference To ITcVnImage [▸ 390]	Returns the component labels (Discrete Voronoi diagram; DINT, 1 channel. An appropriate image will be created if required.)
eLabelType	ETcVnDistanceTransformationLabel [▸ 174]	Type of the labels
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

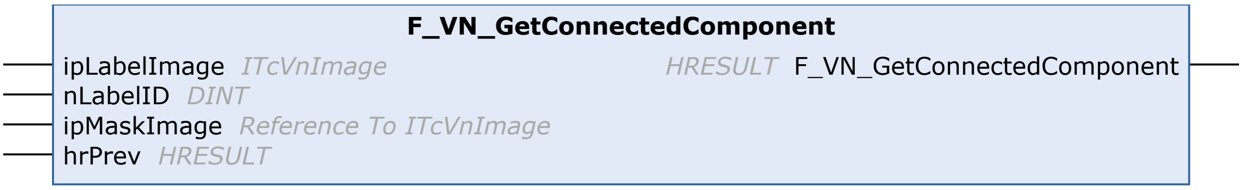
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.11 F_VN_GetConnectedComponent



Get the mask image of a connected component from a label image based on the label ID.

Syntax

Definition:

```
FUNCTION F_VN_GetConnectedComponent : HRESULT
VAR_INPUT
    ipLabelImage : ITcVnImage;
    nLabelID     : DINT;
    ipMaskImage  : Reference To ITcVnImage;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipLabelImage	ITcVnImage [▸ 390]	Image containing labels for each pixel (1 channel DINT or UINT).
nLabelID	DINT	Value of the requested label
ipMaskImage	Reference To ITcVnImage [▸ 390]	Returns a mask image marking the pixels that belong to the component (1 channel USINT.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.12 F_VN_GetConnectedComponentExp

F_VN_GetConnectedComponentExp

ipLabelImage *ITcVnImage* — *HRESULT* F_VN_GetConnectedComponentExp
nLabelID *DINT*
ipMaskImage *Reference To ITcVnImage*
ipContourPoints *Reference To ITcVnContainer*
hrPrev *HRESULT*

Get the mask image (and optionally the outer contour) of a connected component from a label image based on the label ID. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_GetConnectedComponentExp : HRESULT
VAR_INPUT
    ipLabelImage     : ITcVnImage;
    nLabelID         : DINT;
    ipMaskImage      : Reference To ITcVnImage;
    ipContourPoints  : Reference To ITcVnContainer;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipLabelImage	ITcVnImage [▶ 390]	Image containing labels for each pixel (1 channel DINT or UINT).
nLabelID	DINT	Value of the requested label
ipMaskImage	Reference To ITcVnImage [▶ 390]	Returns a mask image marking the pixels that belong to the component (1 channel USINT. Mask is optional, set to 0 if not required.)
ipContourPoints	Reference To ITcVnContainer [▶ 349]	Returns the outer contour of the component(ContainerType_Vector_TcVnPoint2_DINT. Contour is optional, set to 0 if not required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.13 F_VN_ImageAverage

Computes the (channel-wise) average pixel value of an image.

Syntax

Definition:

```
FUNCTION F_VN_ImageAverage : HRESULT
VAR_INPUT
    ipSrcImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aAverage   : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (1 - 4 channels)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aAverage	TcVnVector4_LREAL [▸ 141]	Returns the (channel-wise) average pixel value of ipSrcImage

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.14 F_VN_ImageAverageExp

Computes the (channel-wise) average pixel value of an image. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ImageAverageExp : HRESULT
VAR_INPUT
    ipSrcImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aAverage   : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipMask     : ITcVnImage;
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (1 - 4 channels)
ipMask	ITcVnImage [▸ 390]	Mask of type USINT (1 channel)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aAverage	TcVnVector4_LREAL [▸ 141]	Returns the (channel-wise) average pixel value of ipSrcImage

Return value

HRESULT [▸ 122]

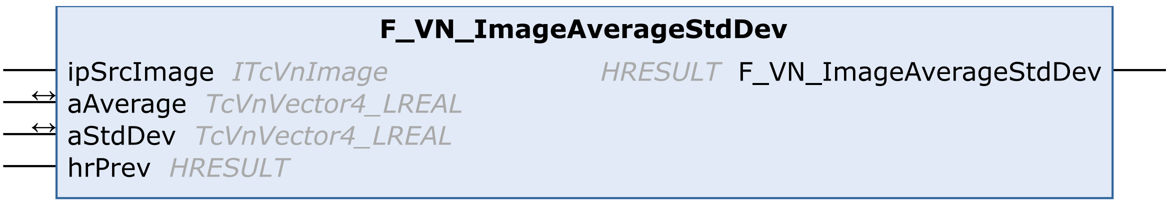
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.15 F_VN_ImageAverageStdDev



Computes the (channel-wise) average pixel value and the corresponding standard deviation of an image.

Syntax

Definition:

```
FUNCTION F_VN_ImageAverageStdDev : HRESULT
VAR_INPUT
    ipSrcImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aAverage   : TcVnVector4_LREAL;
    aStdDev    : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 - 4 channels)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aAverage	TcVnVector4_LREAL [▶ 141]	Returns the (channel-wise) average pixel value of ipSrcImage
aStdDev	TcVnVector4_LREAL [▶ 141]	Returns the (channel-wise) pixel value standard deviation of ipSrcImage

Return value

HRESULT [▶ 122]

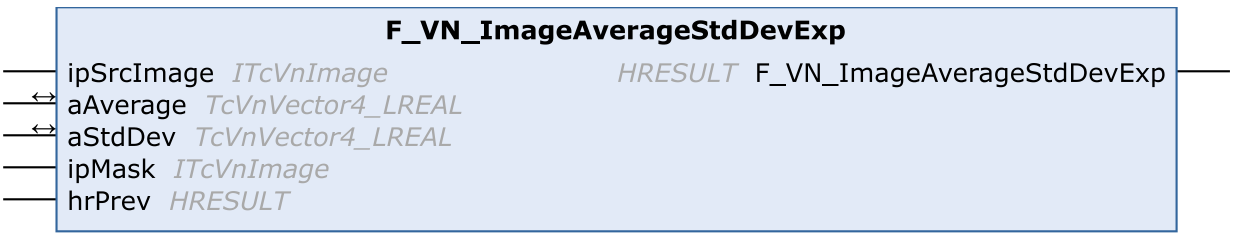
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.16 F_VN_ImageAverageStdDevExp



Computes the (channel-wise) average pixel value and the corresponding standard deviation of an image. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ImageAverageStdDevExp : HRESULT
VAR_INPUT
    ipSrcImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aAverage    : TcVnVector4_LREAL;
    aStdDev     : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipMask      : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (1 - 4 channels)
ipMask	ITcVnImage [▸ 390]	Mask of type USINT (1 channel)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aAverage	TcVnVector4_LREAL [▸ 141]	Returns the (channel-wise) average pixel value of ipSrcImage
aStdDev	TcVnVector4_LREAL [▸ 141]	Returns the (channel-wise) pixel value standard deviation of ipSrcImage

Return value

HRESULT [▸ 122]

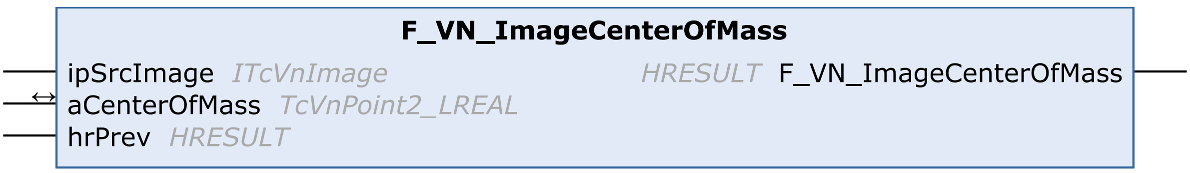
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.17 F_VN_ImageCenterOfMass



Computes the center of mass of an image.

Syntax

Definition:

```
FUNCTION F_VN_ImageCenterOfMass : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
END_VAR
```

```
VAR_IN_OUT
    aCenterOfMass : TcVnPoint2_LREAL;
END_VAR
VAR_INPUT
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (1 channel)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCenterOfMass	TcVnPoint2_LREAL [▸ 139]	Returns the center of mass of the image

Return value

HRESULT [▸ 122]

更多信息

这个函数计算输入图像的质量中心。专家版F_VN_ImageCenterOfMassExp [▸ 1182]存在高级参数化选项。

参数

输入图像

输入图像ipSrcImage必须为单通道图像。如果想要计算多通道图像的质量中心，你有以下选项：

- 可以使用F_VN_SplitImageChannels [▸ 783]将多通道图像分割成各个通道，并为每个通道计算出一个质心。
- 可以使用F_VN_ConvertColorSpace [▸ 1197]将多通道图像转换成单通道图像，并为其计算出质心。

质量中心（返回值）

计算的质心通过参考aCenterOfMass返回。

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.18 F_VN_ImageCenterOfMassExp

Computes the center of mass of an image.

Syntax

Definition:

```
FUNCTION F_VN_ImageCenterOfMassExp : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
END_VAR
VAR_IN_OUT
    aCenterOfMass : TcVnPoint2_LREAL;
END_VAR
VAR_INPUT
    ipMask         : ITcVnImage;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel)
ipMask	ITcVnImage [▶ 390]	Optional mask (1 channel of type USINT, same width and height as ipSrcImage)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aCenterOfMass	TcVnPoint2_LREAL [▶ 139]	Returns the center of mass of the image

Return value

HRESULT [▶ 122]

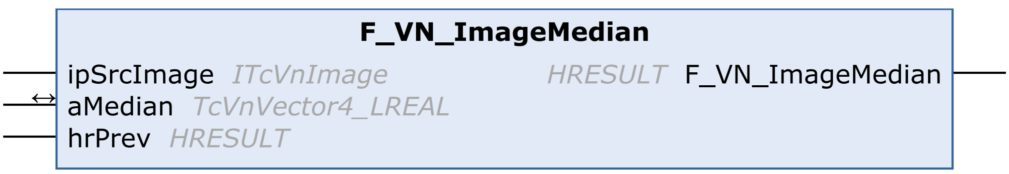
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.19 F_VN_ImageMedian



Computes the (approximated, channel-wise) median pixel value of an image.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_ImageMedian : HRESULT
VAR_INPUT
    ipSrcImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aMedian    : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (1 - 4 channels)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector4_LREAL [▸ 141]	Returns the (approximated, channel-wise) median pixel value of ipSrcImage

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.20 F_VN_ImageMedianExp

Computes the (approximated, channel-wise) median pixel value of an image. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_ImageMedianExp : HRESULT
VAR_INPUT
    ipSrcImage : ITcVnImage;
END_VAR
VAR_IN_OUT
    aMedian    : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
```

```
    ipMask      : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 – 4 channels)
ipMask	ITcVnImage [▶ 390]	Mask of type USINT (1 channel)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMedian	TcVnVector4_LREAL [▶ 141]	Returns the (approximated, channel-wise) median pixel value of ipSrcImage

Return value

HRESULT [▶ 122]

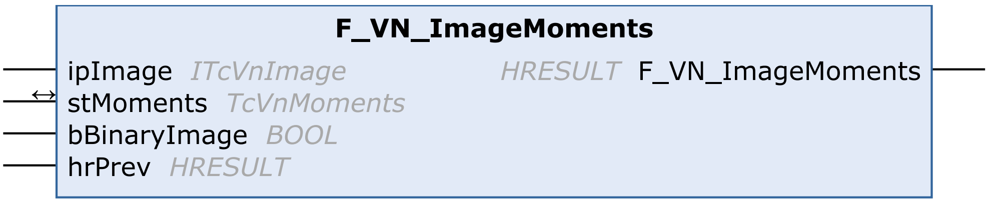
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.21 F_VN_ImageMoments



Computes the spatial moments, the central moments, and the central normalized moments of an image up to the third order.

Syntax

Definition:

```
FUNCTION F_VN_ImageMoments : HRESULT
VAR_INPUT
    ipImage       : ITcVnImage;
END_VAR
VAR_IN_OUT
    stMoments     : TcVnMoments;
END_VAR
VAR_INPUT
    bBinaryImage : BOOL;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	Source image (1 channel, elements of type USINT or REAL)
bBinaryImage	BOOL	If true, all non-zero pixels are treated as ones
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stMoments	TcVnMoments [▶ 211]	Returns a struct containing the moments

Return value

HRESULT [▶ 122]

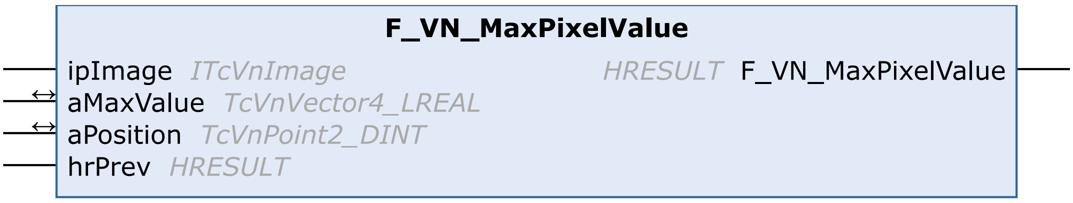
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.22 F_VN_MaxPixelValue



Finds the maximum pixel value in an image (1 - 4 channels supported).

Syntax

Definition:

```
FUNCTION F_VN_MaxPixelValue : HRESULT
VAR_INPUT
    ipImage    : ITcVnImage;
END_VAR
VAR_IN_OUT
    aMaxValue : TcVnVector4_LREAL;
    aPosition : TcVnPoint2_DINT;
END_VAR
VAR_INPUT
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	Source image
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMaxValue	TcVnVector4_LREAL [▶ 141]	Returns the maximum pixel value
aPosition	TcVnPoint2_DINT [▶ 139]	Returns the first found position of aMaxValue

Return value

HRESULT [▶ 122]

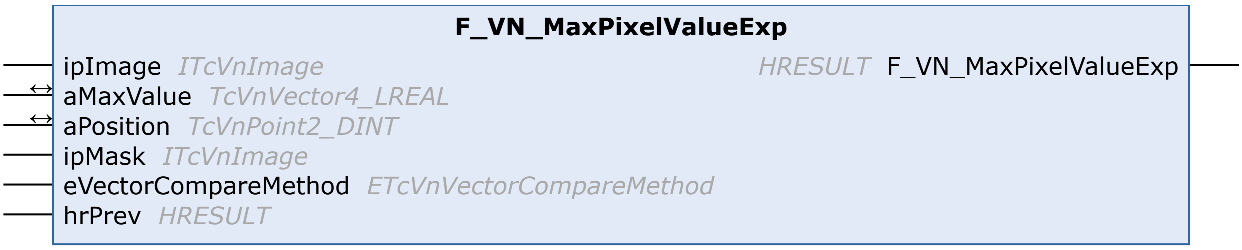
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.23 F_VN_MaxPixelValueExp



Finds the maximum pixel value in an image (1 - 4 channels supported). (expert function)

Syntax

Definition:

```
FUNCTION F_VN_MaxPixelValueExp : HRESULT
VAR_INPUT
    ipImage              : ITcVnImage;
END_VAR
VAR_IN_OUT
    aMaxValue            : TcVnVector4_LREAL;
    aPosition            : TcVnPoint2_DINT;
END_VAR
VAR_INPUT
    ipMask               : ITcVnImage;
    eVectorCompareMethod : ETcVnVectorCompareMethod;
    hrPrev               : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	Source image
ipMask	ITcVnImage [▶ 390]	Optional mask to specify which pixel positions are considered (USINT, set parameter to 0 if not required)
eVectorCompareMethod	ETcVnVectorCompareMethod [▶ 202]	Select a vector compare method for multi-channel images
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMaxValue	TcVnVector4_LREAL [▶ 141]	Returns the maximum pixel value
aPosition	TcVnPoint2_DINT [▶ 139]	Returns the first found position of aMaxValue (not supported for multi-channel images with ELEMENTWISE)

Return value

HRESULT [▶ 122]

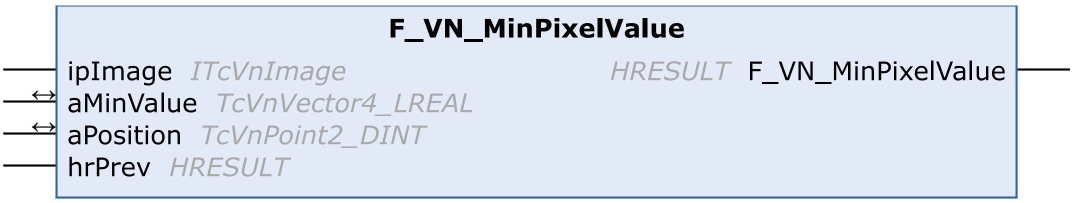
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.24 F_VN_MinPixelValue



Finds the minimum pixel value in an image (1 - 4 channels supported).

Syntax

Definition:

```
FUNCTION F_VN_MinPixelValue : HRESULT
VAR_INPUT
    ipImage    : ITcVnImage;
END_VAR
VAR_IN_OUT
    aMinValue  : TcVnVector4_LREAL;
    aPosition  : TcVnPoint2_DINT;
END_VAR
VAR_INPUT
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	Source image
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMinValue	TcVnVector4_LREAL [▶ 141]	Returns the minimum pixel value
aPosition	TcVnPoint2_DINT [▶ 139]	Returns the first found position of aMinValue

Return value

HRESULT [▶ 122]

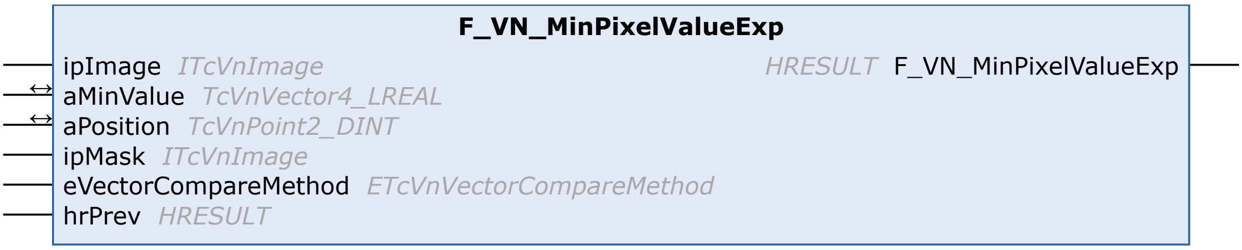
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.25 F_VN_MinPixelValueExp



Finds the minimum pixel value in an image (1 - 4 channels supported). (expert function)

Syntax

Definition:

```
FUNCTION F_VN_MinPixelValueExp : HRESULT
VAR_INPUT
    ipImage              : ITcVnImage;
END_VAR
VAR_IN_OUT
    aMinValue            : TcVnVector4_LREAL;
    aPosition            : TcVnPoint2_DINT;
END_VAR
VAR_INPUT
    ipMask               : ITcVnImage;
    eVectorCompareMethod : ETcVnVectorCompareMethod;
    hrPrev               : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▸ 390]	Source image
ipMask	ITcVnImage [▸ 390]	Optional mask to specify which pixel positions are considered (USINT, set parameter to 0 if not required)
eVectorCompareMethod	ETcVnVectorCompareMethod [▸ 202]	Select a vector compare method for multi-channel images
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aMinValue	TcVnVector4_LREAL [▸ 141]	Returns the minimum pixel value
aPosition	TcVnPoint2_DINT [▸ 139]	Returns the first found position of aMinValue (not supported for multi-channel images with ELEMENTWISE)

Return value

HRESULT [▸ 122]

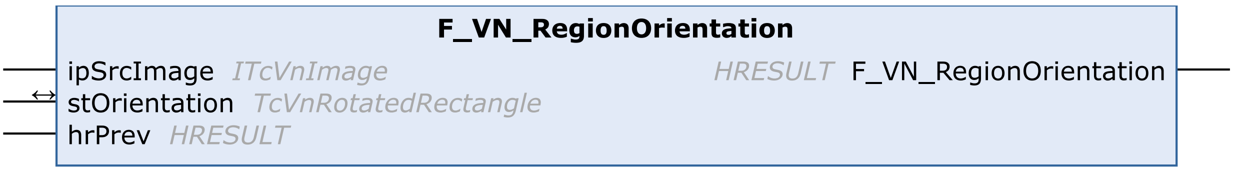
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.26 F_VN_RegionOrientation



Calculate the orientation of a region based on a binary image that contains a non zero value for every pixel in that region.

Syntax

Definition:

```
FUNCTION F_VN_RegionOrientation : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
END_VAR
VAR_IN_OUT
    stOrientation : TcVnRotatedRectangle;
END_VAR
VAR_INPUT
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel, binary).
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stOrientation	TcVnRotatedRectangle [▶ 225]	Resulting rotated rectangle, containig center of mass, lenghts of axes, and rotation angle of the region in clockwise direction.

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.14.27 F_VN_RegionOrientationExp

Calculate the orientation of a region based on a binary image that contains a non zero value for every pixel in that region. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_RegionOrientationExp : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
END_VAR
VAR_IN_OUT
    stOrientation : TcVnRotatedRectangle;
END_VAR
VAR_INPUT
    eMethod        : ETcVnOrientationMethod;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel, binary)
eMethod	ETcVnOrientationMethod [▶ 191]	Method for calculating the orientation.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stOrientation	TcVnRotatedRectangle [▶ 225]	Resulting rotated rectangle, containig center of mass, lenghts of axes, and rotation angle of the region in clockwise direction.

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15 Image Color and Contrast Processing

该组包含色彩和对比度处理的函数。

函数

图像的频率分布、强度和对比度调整

- F_VN_Clahe(Exp) [▶ 1195]
- F_VN_Histogram(Exp) [▶ 1201]
- F_VN_HistogramEqualization(Exp) [▶ 1202]
- F_VN_NormalizeImage(Exp) [▶ 1208]
- F_VN_NormalizeImageForDisplay [▶ 1209]

查询表和色图

- F_VN_ApplyColorMap [▶ 1193]
- F_VN_ApplyLut [▶ 1194]
- F_VN_GenerateColorMap [▶ 1198]
- F_VN_GenerateCustomColorMap [▶ 1200]

颜色匹配

- F_VN_ReferenceColorSimilarity(Exp) [▶ 1210]
- F_VN_TrainImageColor(Exp) [▶ 1218]
- F_VN_TrainImageColor(Exp)_ITcVnMlModel [▶ 1219]

其他

- F_VN_ConvertColorSpace [▶ 1197]
- F_VN_InvertImageColor(Exp) [▶ 1206]

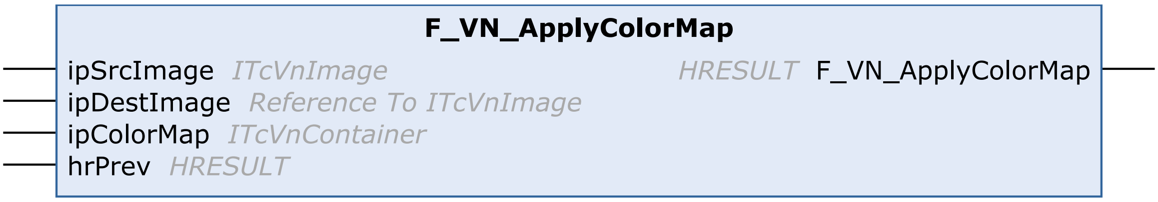
样本

图像颜色和对比度处理 [▶ 2684]

还请参阅有关此

F_VN_WhiteBalance [▶ 1230]

6.1.4.15.1 F_VN_ApplyColorMap



Apply a color map to a gray-level image.

Syntax

Definition:

```
FUNCTION F_VN_ApplyColorMap : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    ipColorMap  : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (TCVN_ET_USINT or TCVN_ET_UINT, 1 channel)
ipDestImage	Reference To ITcVnImage [▶ 390]	Returns the colored image (TCVN_ET_USINT, 3 channel. An appropriate image will be created if required.)
ipColorMap	ITcVnContainer [▶ 349]	Color map to be applied to ipSrcImage (ContainerType_Vector_TcVnVector3_REAL with 256 or 65536 elements, dependent on ipSrcImage type). Can be either custom or created with F_VN_GenerateColorMap.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_ApplyColorMap将一个色表应用于输入图像。

参数

输入图像

输入图像ipSrcImage必须为单通道灰度图像，且其元素类型为USINT（8 位）或UINT（16 位）。

结果图像

结果图像ipDestImage返回通过色表转换的 3 通道 RGB 彩色图像，其元素类型为USINT（8 位）。

色表

色表ipColorMap定义了输入图像的转换方式。色表是一个容器，可以手动创建，也可以通过以下两种函数之一创建：

- F_VN_GenerateColorMap [▶ 1198]
- F_VN_GenerateCustomColorMap [▶ 1200]

容器必须包含 256 个元素（对于 8 位）或 65,536 个元素（对于 16 位）（取决于输入图像的元素类型），并且必须是类型ContainerType_Vector_TcVnVector3_REAL。

应用

例如，一个典型的热图在 8 位图像上的应用如下所示：

```
hr := F_VN_GenerateColorMap(ipColorMap, TCVN_CMM_HOT, TCVN_CMS_256, hr);
hr := F_VN_ApplyColorMap(
    ipSrcImage       :=  ipImageIn,
    ipDestImage      :=  ipImageRes,
    ipColorMap       :=  ipColorMap,
    hrPrev           :=  hr
);
```

样本

- 通过颜色表显示 [▶ 2687]

相关函数

- F_VN_GenerateColorMap [▶ 1198]
- F_VN_GenerateCustomColorMap [▶ 1200]
- F_VN_ApplyColorMap [▶ 1193]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.2 F_VN_ApplyLut

Apply a lookup table to an image to manipulate its colors.

Syntax

Definition:

```
FUNCTION F_VN_ApplyLut : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
```

```
    ipLut        : ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1-4 channels of 8 or 16 bit types SINT, USINT, INT, UINT)
ipDestImage	Reference To ITcVnImage [▶ 390]	Returns the result image (same size, type and channels as ipSrcImage. An appropriate image will be created if required.)
ipLut	ITcVnContainer [▶ 349]	Lookup table with elements matching ipSrcImage type and either 256 (8 bit images) or 65536 (16 bit images) elements. E.g. ContainerType_Vector_USINT for 1-4 channel USINT image (in this case, the same lookup values are used for each channel) or ContainerType_Vector_TcVnVector3_USINT for a 3 channel USINT image (in this case, each channel uses an individual lookup value).
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.3 F_VN_Clahe

Apply Constrast Limited Adaptive Histogram Equalization. This applies local histogram equalization to individual tiles.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_Clahe : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (TCVN_ET_USINT or TCVN_ET_UINT, 1 channel)
ipDestImage	Reference To ITcVnImage [▶ 390]	Returns the resulting image (An appropriate image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.4 F_VN_ClaheExp

Apply Constrast Limited Adaptive Histogram Equalization. This applies local histogram equalization to individual tiles. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_ClaheExp : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    fClipLimit  : LREAL;
    nTilesX     : UDINT;
    nTilesY     : UDINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (TCVN_ET_USINT or TCVN_ET_UINT, 1 channel)
ipDestImage	Reference To ITcVnImage [▶ 390]	Returns the resulting image (An appropriate image will be created if required.)
fClipLimit	LREAL	Threshold for contrast limit (set to <= 0 to disable clipping)
nTilesX	UDINT	Number of tiles in x direction
nTilesY	UDINT	Number of tiles in y direction
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

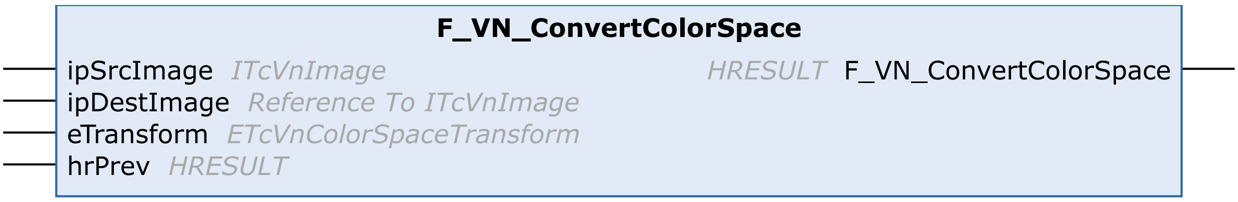
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.5 F_VN_ConvertColorSpace



Convert image from one color space to another.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_ConvertColorSpace : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    eTransform  : ETcVnColorSpaceTransform;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
eTransform	ETcVnColorSpaceTransform [▶ 154]	Transforms to be applied
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

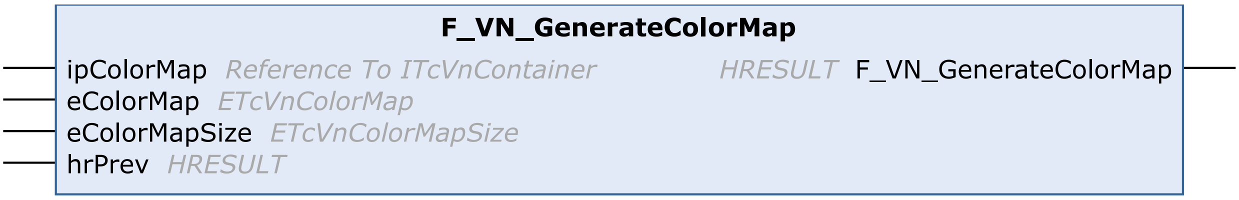
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.6 F_VN_GenerateColorMap



Generates a pre-defined color map.

Syntax

Definition:

```
FUNCTION F_VN_GenerateColorMap : HRESULT
VAR_INPUT
    ipColorMap    : Reference To ITcVnContainer;
    eColorMap     : ETcVnColorMap;
    eColorMapSize : ETcVnColorMapSize;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipColorMap	Reference To ITcVnContainer [▶ 349]	Returns the color map (ContainerType_Vector_TcVnVector3_REAL with 256 or 65536 elements)
eColorMap	ETcVnColorMap [▶ 149]	Selects a color map (similar to GNU Octave/MATLAB types)
eColorMapSize	ETcVnColorMapSize [▶ 154]	Defines how many elements the generated color map should have
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_GenerateColorMap创建一个可以通过函数F_VN_ApplyColorMap [▶ 1193]应用到图像上色表。

参数

色图

参数ipColorMap返回创建的色表。

颜色梯度

色表的颜色梯度eColorMap通过枚举ETcVnColorMap [▶ 149]定义。如果想要手动定义颜色梯度，请使用函数F_VN_GenerateCustomColorMap [▶ 1200]。

色表尺寸

大小eColorMapSize通过枚举ETcVnColorMapSize [▶ 154]指定，并定义是否为 8 位图像（TCVN_CMS_256）或 16 位图像（TCVN_CMS_65536）创建色表。

应用

```
VAR
    ipColorMap      :   ITcVnColorMap;
END_VAR

hr := F_VN_GenerateColorMap(
    ipColorMap      :=  ipColorMap,
    eColorMap       :=  TCVN_CM_HOT,
    eColorMapSize   :=  TCVN_CMS_256,
    hrPrev          :=  hr
);
```

样本

- 通过颜色表显示 [▶ 2687]

相关函数

- F_VN_GenerateColorMap [▶ 1198]
- F_VN_GenerateCustomColorMap [▶ 1200]
- F_VN_ApplyColorMap [▶ 1193]

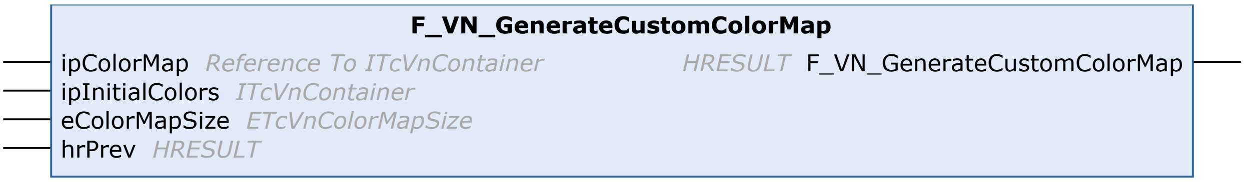
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.7 F_VN_GenerateCustomColorMap



Generate a custom color map by interpolating between user defined colors (equally distributed, linear for each channel).

Syntax

Definition:

```
FUNCTION F_VN_GenerateCustomColorMap : HRESULT
VAR_INPUT
    ipColorMap      : Reference To ITcVnContainer;
    ipInitialColors : ITcVnContainer;
    eColorMapSize   : ETcVnColorMapSize;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipColorMap	Reference To ITcVnContainer [▸ 349]	Returns the color map (ContainerType_Vector_TcVnVector3_REAL with 256 or 65536 elements)
ipInitialColors	ITcVnContainer [▸ 349]	User defined colors (ContainerType_Vector_TcVnVector3_REAL, at least 2 elements)
eColorMapSize	ETcVnColorMapSize [▸ 154]	Defines how many elements the generated color map should have
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

样本

- 通过颜色表显示 [▸ 2687]

相关函数

- F_VN_GenerateColorMap [▸ 1198]
- F_VN_GenerateCustomColorMap [▸ 1200]
- F_VN_ApplyColorMap [▸ 1193]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.8 F_VN_Histogram

Calculate the (multi-channel) histogram of an image.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_Histogram : HRESULT
VAR_INPUT
    ipSrcImage      : ITcVnImage;
    ipDestHistogram : Reference To ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestHistogram	Reference To ITcVnContainer [▸ 349]	Returns a container with a multi-channel histogram, where every channel is represented as a vector of UDINT (ContainerType_Vector_Vector_UDINT. Non-zero interface pointers are reused.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_Histogram逐个通道计算输入图像的直方图。

参数

输入图像

输入图像ipSrcImage可以有任何格式。

直方图

计算出的多通道直方图将作为类型ContainerType_Vector_Vector_UDINT的容器ipDestHistogram返回。每个子元素对应于一个通道的直方图。直方图的每个容器均以UDINT表示，其中指定了属于该容器的像素数量。

专家参数

专家版F_VN_HistogramExp [▸ 1204]包含额外的参数。

应用

例如，确定灰度图像中最频繁出现的像素值如下所示：

```
VAR
    ipDestHistograms    :   ITcVnContainer;
    ipSingleHistogram   :   ITcVnContainer;
    nIndex              :   ULINT;
    ipIterator          :   ITcVnBidirectionalIterator;
END_VAR

hr := F_VN_Histogram(
    ipSrcImage        :=  ipImageIn,
    ipDestHistogram   :=  ipHistograms,
    hrPrev            :=  hr
);
hr := F_VN_GetAt_ITcVnContainer(ipHistograms, ipSingleHistogram, 0, hr);
hr := F_VN_MaxElement(ipSingleHistogram, ipIterator, nIndex, hr);
// nIndex now contains the most frequent pixel value in ipImageIn.
```

样本

- 处理直方图

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.9 F_VN_HistogramEqualization

F_VN_HistogramEqualization	
ipSrcImage *ITcVnImage*	*HRESULT* F_VN_HistogramEqualization
ipDestImage *Reference To ITcVnImage*	
hrPrev *HRESULT*	

Equalize the histogram of a grayscale or rgb image, which normalizes the brightness and improves the contrast.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_HistogramEqualization : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

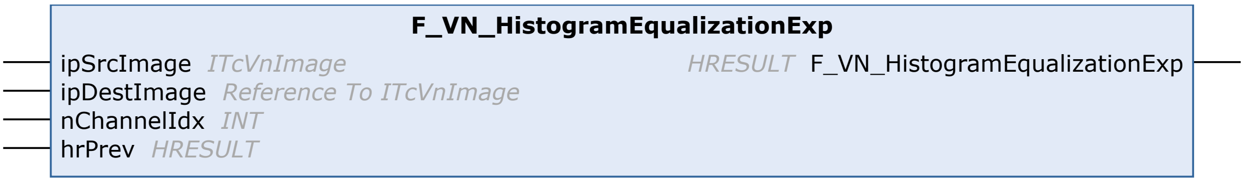
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.10 F_VN_HistogramEqualizationExp



Equalize the histogram of a grayscale or color image, which normalizes the brightness and improves the contrast. (expert function) The channel index that should be equalized has to be specified (-1 expects a RGB image, converts it to YCbCr, equalizes the Y channel and converts the image back to RGB).
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_HistogramEqualizationExp : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    nChannelIdx : INT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
nChannelIdx	INT	Index of the image channel that should be equalized (-1 expects a RGB image, converts it to YCbCr, equalizes the Y channel and converts the image back to RGB)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.11 F_VN_HistogramExp

Calculate the (multi-channel) histogram of an image. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_HistogramExp : HRESULT
VAR_INPUT
    ipSrcImage       : ITcVnImage;
    ipDestHistogram : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    nBins            : UDINT;
    fLowerBound      : LREAL;
    fUpperBound      : LREAL;
END_VAR
VAR_INPUT
    ipMask           : ITcVnImage;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestHistogram	Reference To ITcVnContainer [▸ 349]	Returns a container with a multi-channel histogram, where every channel is represented as a vector of UDINT (ContainerType_Vector_Vector_UDINT. Non-zero interface pointers are reused.)
ipMask	ITcVnImage [▸ 390]	Source image mask (TCVN_ET_USINT, 1 channel. Mask is optional, set to 0 if not required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nBins	UDINT	Desired number of bins or 0 to keep the default for the corresponding image format (in) and default number of bins (out)
fLowerBound	LREAL	Lower (inclusive) boundary of the 0-th histogram bin (in), or receive the default if fLowerBound AND fUpperBound are set to 0 (out)
fUpperBound	LREAL	Upper (exclusive) boundary of the last histogram bin nBins-1 (in), or receive the default if fLowerBound AND fUpperBound are set to 0 (out)

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_HistogramExp是F_VN_Histogram [▸ 1201]的专家版本。它包含额外的参数。

参数

输入图像

输入图像ipSrcImage可以有任何格式。

直方图

计算出的多通道直方图将作为类型ContainerType_Vector_Vector_UDINT的容器ipDestHistogram返回。每个子元素对应于一个通道的直方图。直方图的每个容器均以UDINT表示，其中指定了属于该容器的像素数量。

容器

数字nBins定义了观察到的像素光谱要被分成多少个容器。

限制

下限fLowerBound和上限fUpperBound定义了在创建直方图时哪些像素值被考虑在内。

掩码

掩码ipMask定义了在计算直方图时输入图像ipSrcImage的哪些像素被考虑在内。它由一个元素类型为USINT的单通道图像描述，其大小与输入图像ipSrcImage相同。在计算直方图时，掩码图像中所有数值为<> 0的像素都被考虑在内。

应用

例如，一个有 10 个容器的直方图，数值范围为 100-200，有一个圆形掩码，其计算如下：

```
VAR
    ipImageMask :   ITcVnImage;
END_VAR

// Create mask image with a circle at image center.
hr := F_VN_CopyImage(ipImageIn, ipImageMask, hr);
hr := F_VN_SetPixels(ipImageMask, aBlack, hr);
hr := F_VN_DrawCircle(400, 300, 300, ipImageMask, aWhite, -1, hr);

hr := F_VN_HistogramExp(
    ipSrcImage       :=  ipImageIn,
    ipDestHistogram  :=  ipHistograms,
    nBins            :=  10,
    fLowerBound      :=  100,
    fUpperBound      :=  201,
```

```
    ipMask           :=  ipImageMask,
    hrPrev           :=  hr
);
```

样本

- 处理直方图

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.12 F_VN_InvertImageColor

Invert the color of an image. For signed integer or negative floating point values, only the signs will be switched. If the image contains only positive floating point values, each pixel value is subtracted from the maximum available pixel value (or 1.0, whatever is higher).

Syntax

Definition:

```
FUNCTION F_VN_InvertImageColor : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.13 F_VN_InvertImageColorExp

F_VN_InvertImageColorExp	
ipSrcImage *ITcVnImage*	*HRESULT* F_VN_InvertImageColorExp
ipDestImage *Reference To ITcVnImage*	
fMaxValue *LREAL*	
hrPrev *HRESULT*	

Invert the color of an image. (expert function) For signed integer or negative floating point values, only the signs will be switched. If the image contains only positive floating point values, each pixel value is subtracted from the maximum available pixel value (or 1.0, whatever is higher).

Syntax

Definition:

```
FUNCTION F_VN_InvertImageColorExp : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    fMaxValue   : LREAL;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate image will be created if required.)
fMaxValue	LREAL	Maximum pixel value (e.g. if a 16 bit image contains 12 bit values). -1 means the default values are used.
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.14 F_VN_NormalizeImage

F_VN_NormalizeImage
ipSrcImage *ITcVnImage* — *HRESULT* F_VN_NormalizeImage
ipDestImage *Reference To ITcVnImage*
hrPrev *HRESULT*

Normalize an image regarding its value range (e.g. stretch pixel values [50..150] to full range [0..255]).

Syntax

Definition:

```
FUNCTION F_VN_NormalizeImage : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.15 F_VN_NormalizeImageExp

F_VN_NormalizeImageExp
ipSrcImage *ITcVnImage* — *HRESULT* F_VN_NormalizeImageExp
ipDestImage *Reference To ITcVnImage*
fAlpha *LREAL*
fBeta *LREAL*
eNormType *ETcVnNormalizationType*
eDestType *ETcVnElementType*
ipMask *ITcVnImage*
hrPrev *HRESULT*

Normalize an image regarding its value range (e.g. stretch pixel values [50..150] to full range [0..255]) or scale the values regarding a specific normalization (e.g. L2-norm). (expert function)

Syntax

Definition:

```
FUNCTION F_VN_NormalizeImageExp : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    fAlpha      : LREAL;
    fBeta       : LREAL;
    eNormType   : ETcVnNormalizationType;
    eDestType   : ETcVnElementType;
    ipMask      : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
fAlpha	LREAL	Lower range boundary (in case of normalizing the value range) or value to normalize to (||ipDestImage|| = fAlpha)
fBeta	LREAL	Upper range boundary (in case of normalizing the value range)
eNormType	ETcVnNormalizationType [▶ 189]	Normalization type (only INF, L1, L2 or MINMAX)
eDestType	ETcVnElementType [▶ 178]	Destination image depth (usually SAME_AS_SOURCE)
ipMask	ITcVnImage [▶ 390]	Mask to restrict the normalization to specific pixel positions (set 0 to normalize the whole image)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.16 F_VN_NormalizeImageForDisplay

Normalize an image for display, i.e. scale it to the full value range of the underlying data type (-1 to 1 for floating point).
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_NormalizeImageForDisplay : HRESULT
VAR_INPUT
    ipSrcImage              : ITcVnImage;
    ipDestImage             : Reference To ITcVnImage;
    eSignedNormalization : ETcVnSignedNormalization;
    hrPrev                  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
eSignedNormalization	ETcVnSignedNormalization [▸ 195]	Option for normalizing signed values
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.17 F_VN_ReferenceColorSimilarity_ITcVnColorModel

F_VN_ReferenceColorSimilarity_ITcVnColorModel
ipSrcImage *ITcVnImage* — *HRESULT* F_VN_ReferenceColorSimilarity_ITcVnColorModel
ipDestImage *Reference To ITcVnImage*
ipColorModel *ITcVnColorModel*
hrPrev *HRESULT*

Computes the similarity to a reference color model for each pixel in the source image.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_ReferenceColorSimilarity_ITcVnColorModel : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipDestImage   : Reference To ITcVnImage;
```

```
    ipColorModel : ITcVnColorModel;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	RGB source image (USINT, 3 channels)
ipDestImage	Reference To ITcVnImage [▶ 390]	Returns the similarity to ipColorModel for each pixel in ipSrcImage (USINT, 1 channel. An appropriate destination image will be created if required.)
ipColorModel	ITcVnColorModel [▶ 397]	Color model
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

有三种方法可以将参考颜色传递给相应的函数：

- 作为一个经过训练的颜色模型，类型ITcVnColorModel
- 作为一个经过训练的颜色模型，类型ITcVnMlModel
- 直接通过TcVnVector3_LREAL中的数值， 见样本

如需训练一个颜色模型，必须使用相应的F_VN_TrainImageColor [▶ 1218]函数。这两种颜色模型仅在数据类型上有所区别，而在功能上没有区别。ITcVnMlModel是 ML 模型的一般数据类型，对其也有加载和保存的功能块。因此，如果需要将训练好的颜色模型保存到磁盘上，应该使用这种数据类型和相应的函数。

样本

- 与 RGB 参考颜色的颜色相似度 [▶ 2684]

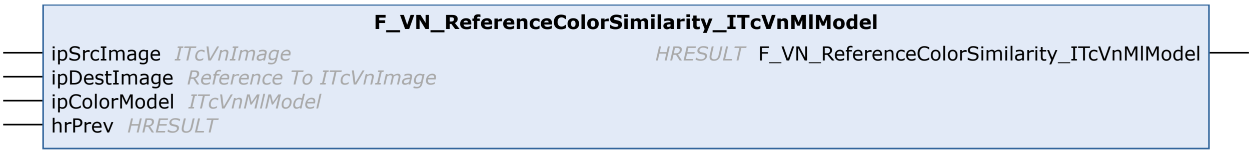
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.18 F_VN_ReferenceColorSimilarity_ITcVnMlModel



Computes the similarity to a reference color model for each pixel in the source image.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_ReferenceColorSimilarity_ITcVnMlModel : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipDestImage   : Reference To ITcVnImage;
    ipColorModel  : ITcVnMlModel;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	RGB source image (USINT, 3 channels)
ipDestImage	Reference To ITcVnImage [▶ 390]	Returns the similarity to ipColorModel for each pixel in ipSrcImage (USINT, 1 channel. An appropriate destination image will be created if required.)
ipColorModel	ITcVnMlModel [▶ 399]	Color model
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

有三种方法可以将参考颜色传递给相应的函数：

- 作为一个经过训练的颜色模型，类型ITcVnColorModel
- 作为一个经过训练的颜色模型，类型ITcVnMlModel
- 直接通过TcVnVector3_LREAL中的数值， 见样本

如需训练一个颜色模型，必须使用相应的F_VN_TrainImageColor [▶ 1218]函数。这两种颜色模型仅在数据类型上有所区别，而在功能上没有区别。ITcVnMlModel是 ML 模型的一般数据类型，对其也有加载和保存的功能块。因此，如果需要将训练好的颜色模型保存到磁盘上，应该使用这种数据类型和相应的函数。

样本

- 与 RGB 参考颜色的颜色相似度 [▶ 2684]

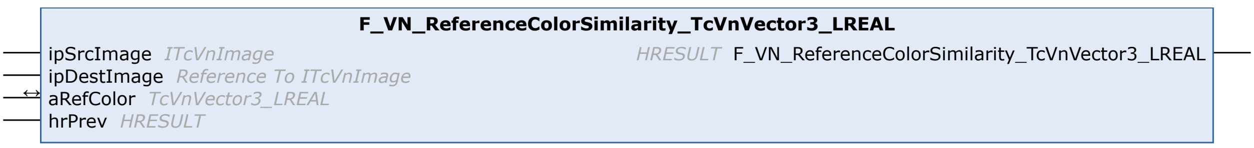
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.19 F_VN_ReferenceColorSimilarity_TcVnVector3_LREAL



Computes the similarity to a reference color for each pixel in the source image.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_ReferenceColorSimilarity_TcVnVector3_LREAL : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aRefColor   : TcVnVector3_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	RGB source image (USINT, 3 channels)
ipDestImage	Reference To ITcVnImage [▸ 390]	Returns the similarity to aRefColor for each pixel in ipSrcImage (USINT, 1 channel. An appropriate destination image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aRefColor	TcVnVector3_LREAL [▸ 141]	Reference color (RGB, [0..255])

Return value

HRESULT [▸ 122]

更多信息

有三种方法可以将参考颜色传递给相应的函数：

- 作为一个经过训练的颜色模型，类型ITcVnColorModel
- 作为一个经过训练的颜色模型，类型ITcVnMlModel
- 直接通过TcVnVector3_LREAL中的数值， 见样本

如需训练一个颜色模型，必须使用相应的F_VN_TrainImageColor [▸ 1218]函数。这两种颜色模型仅在数据类型上有所区别，而在功能上没有区别。ITcVnMlModel是 ML 模型的一般数据类型，对其也有加载和保存的功能块。因此，如果需要将训练好的颜色模型保存到磁盘上，应该使用这种数据类型和相应的函数。

样本

- 与 RGB 参考颜色的颜色相似度 [▸ 2684]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.20 F_VN_ReferenceColorSimilarityExp_ITcVnColorModel

Computes the similarity to a reference color model for each pixel in the source image. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_ReferenceColorSimilarityExp_ITcVnColorModel : HRESULT
VAR_INPUT
    ipSrcImage        : ITcVnImage;
    ipDestImage       : Reference To ITcVnImage;
    ipColorModel      : ITcVnColorModel;
    fVariance         : REAL;
    fLuminanceWeight  : REAL;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	RGB source image (USINT, 3 channels)
ipDestImage	Reference To ITcVnImage [▸ 390]	Returns the similarity to ipColorModel for each pixel in ipSrcImage (USINT, 1 channel. An appropriate destination image will be created if required.)
ipColorModel	ITcVnColorModel [▸ 397]	Color model
fVariance	REAL	Allowed color variance (0.1 - 0.3 might be a good start to try)
fLuminanceWeight	REAL	Weight the impact of the luminance ([0..1], e.g. set to 0 to be more resistant to unequal illumination, but might be required to differentiate between some colors. Ignored if ipColorModel is of type RGB.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

有三种方法可以将参考颜色传递给相应的函数：

- 作为一个经过训练的颜色模型，类型ITcVnColorModel

- 作为一个经过训练的颜色模型，类型ITcVnMlModel
- 直接通过TcVnVector3_LREAL中的数值， 见样本

如需训练一个颜色模型，必须使用相应的F_VN_TrainImageColor [▶ 1218]函数。这两种颜色模型仅在数据类型上有所区别，而在功能上没有区别。ITcVnMlModel是 ML 模型的一般数据类型，对其也有加载和保存的功能块。因此，如果需要将训练好的颜色模型保存到磁盘上，应该使用这种数据类型和相应的函数。

样本

- 与 RGB 参考颜色的颜色相似度 [▶ 2684]

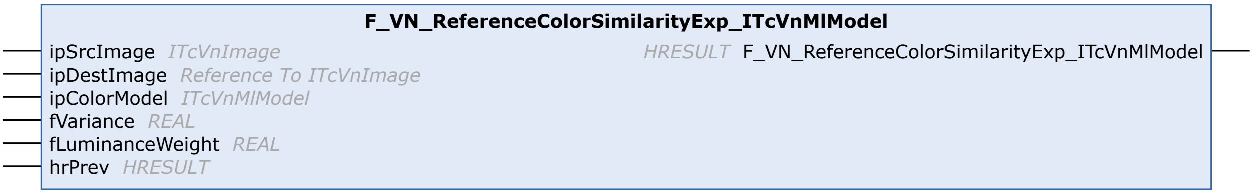
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.21 F_VN_ReferenceColorSimilarityExp_ITcVnMlModel



Computes the similarity to a reference color model for each pixel in the source image. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_ReferenceColorSimilarityExp_ITcVnMlModel : HRESULT
VAR_INPUT
    ipSrcImage        : ITcVnImage;
    ipDestImage       : Reference To ITcVnImage;
    ipColorModel      : ITcVnMlModel;
    fVariance         : REAL;
    fLuminanceWeight  : REAL;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	RGB source image (USINT, 3 channels)
ipDestImage	Reference To ITcVnImage [▶ 390]	Returns the similarity to ipColorModel for each pixel in ipSrcImage (USINT, 1 channel. An appropriate destination image will be created if required.)
ipColorModel	ITcVnMlModel [▶ 399]	Color model
fVariance	REAL	Allowed color variance (0.1 - 0.3 might be a good start to try)
fLuminanceWeight	REAL	Weight the impact of the luminance ([0..1], e.g. set to 0 to be more resistant to unequal illumination, but might be required to differentiate between some colors. Ignored if ipColorModel is of type RGB.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

有三种方法可以将参考颜色传递给相应的函数：

- 作为一个经过训练的颜色模型，类型ITcVnColorModel
- 作为一个经过训练的颜色模型，类型ITcVnMlModel
- 直接通过TcVnVector3_LREAL中的数值， 见样本

如需训练一个颜色模型，必须使用相应的F_VN_TrainImageColor [▶ 1218]函数。这两种颜色模型仅在数据类型上有所区别，而在功能上没有区别。ITcVnMlModel是 ML 模型的一般数据类型，对其也有加载和保存的功能块。因此，如果需要将训练好的颜色模型保存到磁盘上，应该使用这种数据类型和相应的函数。

样本

- 与 RGB 参考颜色的颜色相似度 [▶ 2684]

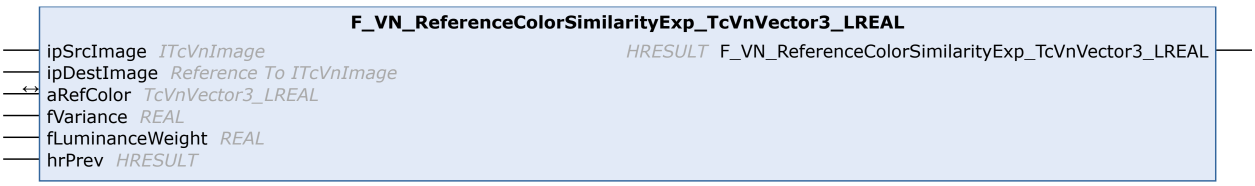
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.22 F_VN_ReferenceColorSimilarityExp_TcVnVector3_LREAL



Computes the similarity to a reference color for each pixel in the source image. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_ReferenceColorSimilarityExp_TcVnVector3_LREAL : HRESULT
VAR_INPUT
    ipSrcImage        : ITcVnImage;
    ipDestImage       : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aRefColor         : TcVnVector3_LREAL;
END_VAR
VAR_INPUT
    fVariance         : REAL;
    fLuminanceWeight  : REAL;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	RGB source image (USINT, 3 channels)
ipDestImage	Reference To ITcVnImage [▶ 390]	Returns the similarity to aRefColor for each pixel in ipSrcImage (USINT, 1 channel. An appropriate destination image will be created if required.)
fVariance	REAL	Allowed color variance (0.1 - 0.3 might be a good start to try)
fLuminanceWeight	REAL	Weight the impact of the luminance ([0..1], e.g. set to 0 to be more resistant to unequal illumination, but might be required to differentiate between some colors)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aRefColor	TcVnVector3_LREAL [▶ 141]	Reference color (RGB, [0..255])

Return value

HRESULT [▶ 122]

更多信息

有三种方法可以将参考颜色传递给相应的函数：

- 作为一个经过训练的颜色模型，类型ITcVnColorModel
- 作为一个经过训练的颜色模型，类型ITcVnMlModel
- 直接通过TcVnVector3_LREAL中的数值， 见样本

如需训练一个颜色模型，必须使用相应的F_VN_TrainImageColor [▶ 1218]函数。这两种颜色模型仅在数据类型上有所区别，而在功能上没有区别。ITcVnMlModel是 ML 模型的一般数据类型，对其也有加载和保存的功能块。因此，如果需要将训练好的颜色模型保存到磁盘上，应该使用这种数据类型和相应的函数。

样本

- 与 RGB 参考颜色的颜色相似度 [▶ 2684]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.23 F_VN_TrainImageColor

Create a new color model, describing the image color.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_TrainImageColor : HRESULT
VAR_INPUT
    ipSrcImage       : ITcVnImage;
    ipColorModel     : Reference To ITcVnColorModel;
    nDifferentColors : UDINT;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (3 channels (RGB) of type USINT)
ipColorModel	Reference To ITcVnColorModel [▶ 397]	Returns the color model
nDifferentColors	UDINT	Number of different colors to distinguish
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_TrainImageColor根据参考图像训练颜色模型。如果要将训练好的颜色模型保存到磁盘，必须使用类型为ITcVnMlModel的F_VN_TrainImageColor_ITcVnMlModel [▶ 1219]。

参数

参考图像

参考图像ipSrcImage必须有 3 个 RGB 格式的通道，且元素类型为USINT（8 位）。

颜色模型

参数ipColorModel返回训练的颜色模型。这是一个类型ITcVnColorModel [▶ 397]的接口指针。

需要区分的颜色数量

数字nDifferentColors定义了需要训练的颜色数量。请注意，图像背景也可以有自己的颜色。

专家参数

关于更多参数，可查看专家版F_VN_TrainImageColorExp [▶ 1221]。

应用

例如，对于包括 4 个不同颜色物体以及深色背景的参考图像，进行颜色模型的训练如下所示：

```
VAR
    ipColorModel     :    ITcVnColorModel;
END_VAR

hr := F_VN_TrainImageColor(
            ipSrcImage       := ipImageRef,
            ipColorModel     := ipColorModel,
            nDifferentColors:= 5, // 4 colors + 1 background = 5
            hrPrev           := hr);
```

相关函数

有三种方法可以将参考颜色传递给相应的F_VN_ReferenceColorSimilarity函数：

- 作为一个经过训练的颜色模型，类型ITcVnColorModel
- 作为一个经过训练的颜色模型，类型ITcVnMlModel
- 直接使用TcVnVector3_LREAL中的数值

对于颜色模型，有相应的函数用于训练，而对于所有三种数据类型，各有一个函数用于执行：

- F_VN_TrainImageColor [▶ 1218]用于训练一个颜色模型
- F_VN_ReferenceColorSimilarity [▶ 1210]通过颜色模型进行分割

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.24 F_VN_TrainImageColor_ITcVnMlModel

Create a new color model, describing the image color.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_TrainImageColor_ITcVnMlModel : HRESULT
VAR_INPUT
    ipSrcImage        : ITcVnImage;
    ipColorModel      : Reference To ITcVnMlModel;
    nDifferentColors : UDINT;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (3 channels (RGB) of type USINT)
ipColorModel	Reference To ITcVnMlModel [▸ 399]	Returns the color model
nDifferentColors	UDINT	Number of different colors to distinguish
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_TrainImageColor_ITcVnMlModel根据参考图像训练颜色模型。通过相应的功能块，颜色模型可以在硬盘上保存和加载。

参数

参考图像

参考图像ipSrcImage必须有 3 个 RGB 格式的通道，且元素类型为USINT（8 位）。

颜色模型

参数ipColorModel返回训练的颜色模型。这是一个接口指针，类型为ITcVnMlModel [▸ 399]。

需要区分的颜色数量

数字nDifferentColors定义了需要训练的颜色数量。请注意，图像背景也可以有自己的颜色。

专家参数

关于更多参数，可查看专家版F_VN_TrainImageColorExp_ITcVnMlModel [▸ 1223]。

应用

例如，对于包括 4 个不同颜色物体以及深色背景的参考图像，进行颜色模型的训练如下所示：

```
VAR
    ipColorModel    :   ITcVnMlModel;
END_VAR

hr := F_VN_TrainImageColor_ITcVnMlModel(
```

```
            ipSrcImage       := ipImageRef,
            ipColorModel     := ipColorModel,
            nDifferentColors:= 5, // 4 colors + 1 background = 5
            hrPrev           := hr);
```

相关函数

有三种方法可以将参考颜色传递给相应的F_VN_ReferenceColorSimilarity函数：

- 作为一个经过训练的颜色模型，类型ITcVnColorModel
- 作为一个经过训练的颜色模型，类型ITcVnMlModel
- 直接使用TcVnVector3_LREAL中的数值

对于颜色模型，有相应的函数用于训练，而对于所有三种数据类型，各有一个函数用于执行：

- F_VN_TrainImageColor [▶ 1218]用于训练一个颜色模型
- F_VN_ReferenceColorSimilarity [▶ 1210]通过颜色模型进行分割

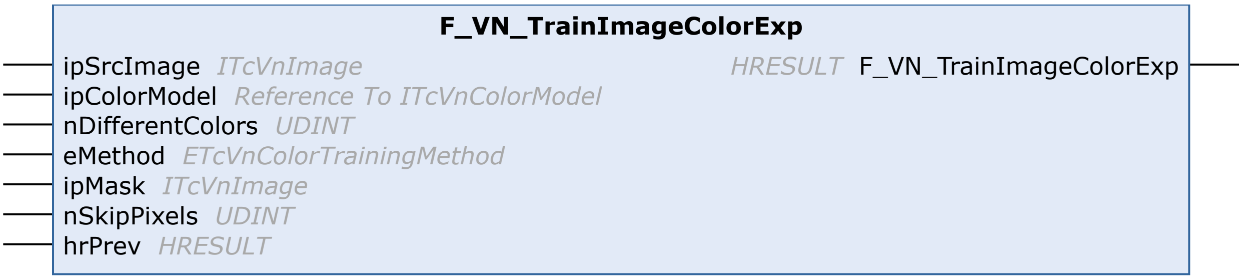
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.25 F_VN_TrainImageColorExp



Create a new color model, describing the image color. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_TrainImageColorExp : HRESULT
VAR_INPUT
    ipSrcImage       : ITcVnImage;
    ipColorModel     : Reference To ITcVnColorModel;
    nDifferentColors : UDINT;
    eMethod          : ETcVnColorTrainingMethod;
    ipMask           : ITcVnImage;
    nSkipPixels      : UDINT;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (3 channels (RGB) of type USINT)
ipColorModel	Reference To ITcVnColorModel [▶ 397]	Returns the color model
nDifferentColors	UDINT	Number of different colors to distinguish
eMethod	ETcVnColorTrainingMethod [▶ 166]	Color training method
ipMask	ITcVnImage [▶ 390]	Optional image mask (1 channel of type USINT, set to 0 if not required)
nSkipPixels	UDINT	Number of pixels to skip between each evaluated color sample (to achieve a better performance). 0 takes every pixel into account and tends to be more accurate.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

该函数F_VN_TrainImageColorExp是F_VN_TrainImageColor [▶ 1218]的专家版本。它包含进一步的参数。如果需要将训练好的颜色模型保存到磁盘，必须使用类型ITcVnMlModel的F_VN_TrainImageColorExp_ITcVnMlModel [▶ 1223]。

参数

参考图像

参考图像ipSrcImage必须有 3 个 RGB 格式的通道，且元素类型为USINT（8 位）。

颜色模型

参数ipColorModel返回训练的颜色模型。这是一个类型ITcVnColorModel [▶ 397]的接口指针。

需要区分的颜色数量

数字nDifferentColors定义了需要训练的颜色数量。请注意，图像背景也可以有自己的颜色。

专家参数

关于其他参数，可查看另一专家版本F_VN_TrainImageColorExp2 [▶ 1225]。

培训方法

在选择训练方法时，可以选择 LAB 和 RGB 颜色空间。

掩码

掩码ipMask定义了输入图像ipSrcImage的哪些像素在计算中被考虑进去。掩码由一个元素类型为 USINT 的单通道图像描述，其大小与输入图像ipSrcImage相同。在计算集群时，掩码图像中所有数值 <>0 的像素都被考虑在内。

像素差距

为了优化性能，可以跳过一些像素，这样在划分各个集群时需要考虑的像素就会减少。

应用

```
VAR
    ipColorModel     :   ITcVnColorModel;
END_VAR

hr := F_VN_TrainImageColor(
            ipSrcImage       :=  ipImageRef,
            ipColorModel     :=  ipColorModel,
            nDifferentColors:=  5, // 4 colors + 1 background = 5
            eMethod          :=  TCVN_CTM_LAB,
            ipMask           :=  0,
            nSkipPixels      :=  3,
            hrPrev           :=  hr);
```

相关函数

有三种方法可以将参考颜色传递给相应的F_VN_ReferenceColorSimilarity函数：

- 作为一个经过训练的颜色模型，类型ITcVnColorModel
- 作为一个经过训练的颜色模型，类型ITcVnMlModel
- 直接使用TcVnVector3_LREAL中的数值

对于颜色模型，有相应的函数用于训练，而对于所有三种数据类型，各有一个函数用于执行：

- F_VN_TrainImageColor [▶ 1218]用于训练一个颜色模型
- F_VN_ReferenceColorSimilarity [▶ 1210]通过颜色模型进行分割

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.26 F_VN_TrainImageColorExp_ITcVnMlModel

Create a new color model, describing the image color. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_TrainImageColorExp_ITcVnMlModel : HRESULT
VAR_INPUT
    ipSrcImage        : ITcVnImage;
    ipColorModel      : Reference To ITcVnMlModel;
    nDifferentColors  : UDINT;
    eMethod           : ETcVnColorTrainingMethod;
    ipMask            : ITcVnImage;
```

```
    nSkipPixels       : UDINT;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (3 channels (RGB) of type USINT)
ipColorModel	Reference To ITcVnMlModel [▶ 399]	Returns the color model
nDifferentColors	UDINT	Number of different colors to distinguish
eMethod	ETcVnColorTrainingMethod [▶ 166]	Color training method
ipMask	ITcVnImage [▶ 390]	Optional image mask (1 channel of type USINT, set to 0 if not required)
nSkipPixels	UDINT	Number of pixels to skip between each evaluated color sample (to achieve a better performance). 0 takes every pixel into account and tends to be more accurate.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_TrainImageColorExp_ITcVnMlModel是F_VN_TrainImageColor_ITcVnMlModel [▶ 1219]的专家版。它包含进一步的参数。通过相应的功能块，颜色模型可以在硬盘上保存和加载。

参数

参考图像

参考图像ipSrcImage必须有 3 个 RGB 格式的通道，且元素类型为USINT（8 位）。

颜色模型

参数ipColorModel返回训练的颜色模型。这是一个接口指针，类型为ITcVnMlModel [▶ 399]。

需要区分的颜色数量

数字nDifferentColors定义了需要训练的颜色数量。请注意，图像背景也可以有自己的颜色。

专家参数

关于其他参数，可查看另一专家版本F_VN_TrainImageColorExp2_ITcVnMlModel [▶ 1228]。

培训方法

在选择训练方法时，可以选择 LAB 和 RGB 颜色空间。

掩码

掩码ipMask定义了输入图像ipSrcImage的哪些像素在计算中被考虑进去。掩码由一个元素类型为 USINT 的单通道图像描述，其大小与输入图像ipSrcImage相同。在计算集群时，掩码图像中所有数值 <>0 的像素都被考虑在内。

像素差距

为了优化性能，可以跳过一些像素，这样在划分各个集群时需要考虑的像素就会减少。

应用

```
VAR
    ipColorModel    :   ITcVnMlModel;
END_VAR

hr := F_VN_TrainImageColorExp_ITcVnMlModel (
            ipSrcImage      := ipImageRef,
            ipColorModel    := ipColorModel,
            nDifferentColors:= 5, // 4 colors + 1 background = 5
            eMethod         := TCVN_CTM_LAB,
            ipMask          := 0,
            nSkipPixels     := 3,
            hrPrev          := hr);
```

相关函数

有三种方法可以将参考颜色传递给相应的F_VN_ReferenceColorSimilarity函数：

- 作为一个经过训练的颜色模型，类型ITcVnColorModel
- 作为一个经过训练的颜色模型，类型ITcVnMlModel
- 直接使用TcVnVector3_LREAL中的数值

对于颜色模型，有相应的函数用于训练，而对于所有三种数据类型，各有一个函数用于执行：

- F_VN_TrainImageColor [▶ 1218]用于训练一个颜色模型
- F_VN_ReferenceColorSimilarity [▶ 1210]通过颜色模型进行分割

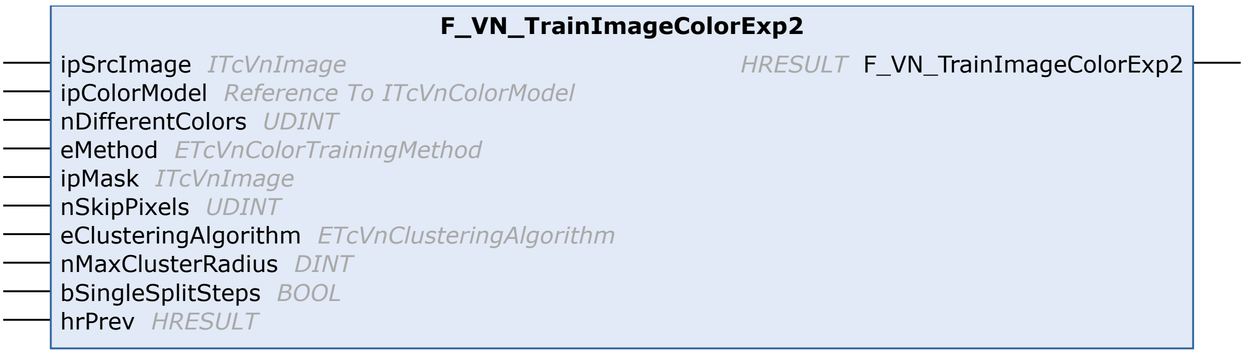
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.27 F_VN_TrainImageColorExp2



Create a new color model, describing the image color. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_TrainImageColorExp2 : HRESULT
VAR_INPUT
    ipSrcImage           : ITcVnImage;
```

```
    ipColorModel          : Reference To ITcVnColorModel;
    nDifferentColors      : UDINT;
    eMethod               : ETcVnColorTrainingMethod;
    ipMask                : ITcVnImage;
    nSkipPixels           : UDINT;
    eClusteringAlgorithm  : ETcVnClusteringAlgorithm;
    nMaxClusterRadius     : DINT;
    bSingleSplitSteps     : BOOL;
    hrPrev                : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (3 channels (RGB) of type USINT)
ipColorModel	Reference To ITcVnColorModel [▶ 397]	Returns the color model
nDifferentColors	UDINT	Maximum number of different colors to distinguish (if LBG is used as a clustering algorithm, the result might have less different colors, depending on nMaxClusterRadius)
eMethod	ETcVnColorTrainingMethod [▶ 166]	Color training method
ipMask	ITcVnImage [▶ 390]	Optional image mask (1 channel of type USINT, set to 0 if not required)
nSkipPixels	UDINT	Number of pixels to skip between each evaluated color sample (to achieve a better performance). 0 takes every pixel into account and tends to be more accurate.
eClusteringAlgorithm	ETcVnClusteringAlgorithm [▶ 148]	Clustering algorithm
nMaxClusterRadius	DINT	Only used for the LBG clustering algorithm. Maximum allowed radius (> 0) of a single cluster, i.e. clusters with a higher radius will be split into smaller ones, until a global number of nDifferentColors is reached.
bSingleSplitSteps	BOOL	Only used for the LBG clustering algorithm. If true, the global optimization is always run after a single cluster has been split. If false, several clusters are split within the same step before applying the global optimization. Applying the global optimization less often is faster, but can lead to less optimal results, especially having 2 nearby clusters that could be represented by 1)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

该函数F_VN_TrainImageColorExp2是F_VN_TrainImageColorExp [▶ 1221]的扩展专家版本。它包含进一步的参数。如果需要将训练好的颜色模型保存到磁盘，必须使用类型ITcVnMlModel的F_VN_TrainImageColorExp2_ITcVnMlModel [▶ 1228]。

参数

参考图像

参考图像ipSrcImage必须有 3 个 RGB 格式的通道，且元素类型为USINT（8 位）。

颜色模型

参数ipColorModel返回训练的颜色模型。这是一个类型ITcVnColorModel [▶ 397]的接口指针。

需要区分的颜色数量

数字nDifferentColors定义了需要训练的颜色数量。请注意，图像背景也可以有自己的颜色。

集群算法

在选择集群算法时，有 KMeans++（固定集群数量）和 LBG（动态集群数量）可供选择。

最大集群半径

仅与 LBG 集群算法一起使用。这可以用来影响各个集群的分布。

全球优化

仅与 LBG 集群算法一起使用。这允许在更短的执行时间（bSingleSplitSteps = FALSE）和更好的结果（bSingleSplitSteps = TRUE）之间做出选择。

更少使用全局优化会更快，但可能导致不太理想的结果。

应用

```
VAR
    ipColorModel    :   ITcVnColorModel;
END_VAR

hr := F_VN_TrainImageColorExp2(
            ipSrcImage          := ipImageRef,
            ipColorModel        := ipColorModel,
            nDifferentColors    := 5, // 4 colors + 1 background = 5
            eMethod             := TCVN_CTM_LAB,
            ipMask              := 0,
            nSkipPixels         := 3,
            eClusteringAlgorithm:= TCVN_CA_KMEANSPP,
            nMaxClusterRadius   := 0,
            bSingleSplitSteps   := FALSE,
            hrPrev              := hr);
```

相关函数

有三种方法可以将参考颜色传递给相应的F_VN_ReferenceColorSimilarity函数：

- 作为一个经过训练的颜色模型，类型ITcVnColorModel
- 作为一个经过训练的颜色模型，类型ITcVnMlModel
- 直接使用TcVnVector3_LREAL中的数值

对于颜色模型，有相应的函数用于训练，而对于所有三种数据类型，各有一个函数用于执行：

- F_VN_TrainImageColor [▶ 1218]用于训练一个颜色模型
- F_VN_ReferenceColorSimilarity [▶ 1210]通过颜色模型进行分割

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.28 F_VN_TrainImageColorExp2_ITcVnMlModel

Create a new color model, describing the image color. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_TrainImageColorExp2_ITcVnMlModel : HRESULT
VAR_INPUT
    ipSrcImage           : ITcVnImage;
    ipColorModel         : Reference To ITcVnMlModel;
    nDifferentColors     : UDINT;
    eMethod              : ETcVnColorTrainingMethod;
    ipMask               : ITcVnImage;
    nSkipPixels          : UDINT;
    eClusteringAlgorithm : ETcVnClusteringAlgorithm;
    nMaxClusterRadius    : DINT;
    bSingleSplitSteps    : BOOL;
    hrPrev               : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (3 channels (RGB) of type USINT)
ipColorModel	Reference To ITcVnMlModel [▶ 399]	Returns the color model
nDifferentColors	UDINT	Maximum number of different colors to distinguish (if LBG is used as a clustering algorithm, the result might have less different colors, depending on nMaxClusterRadius)
eMethod	ETcVnColorTrainingMethod [▶ 166]	Color training method
ipMask	ITcVnImage [▶ 390]	Optional image mask (1 channel of type USINT, set to 0 if not required)
nSkipPixels	UDINT	Number of pixels to skip between each evaluated color sample (to achieve a better performance). 0 takes every pixel into account and tends to be more accurate.
eClusteringAlgorithm	ETcVnClusteringAlgorithm [▶ 148]	Clustering algorithm
nMaxClusterRadius	DINT	Only used for the LBG clustering algorithm. Maximum allowed radius (> 0) of a single cluster, i.e. clusters with a higher radius will be split into smaller ones, until a global number of nDifferentColors is reached.
bSingleSplitSteps	BOOL	Only used for the LBG clustering algorithm. If true, the global optimization is always run after a single cluster has been split. If false, several clusters are split within the same step before applying the global optimization. Applying the global optimization less often is faster, but can lead to less optimal results, especially having 2 nearby clusters that could be represented by 1)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

该函数F_VN_TrainImageColorExp2_ITcVnMlModel 是F_VN_TrainImageColorExp_ITcVnMlModel [▶ 1223]的扩展专家版。它包含进一步的参数。通过相应的功能块，颜色模型可以在硬盘上保存和加载。

参数

参考图像

参考图像ipSrcImage必须有 3 个 RGB 格式的通道，且元素类型为USINT（8 位）。

颜色模型

参数ipColorModel返回训练的颜色模型。这是一个接口指针，类型为ITcVnMlModel [▶ 399]。

需要区分的颜色数量

数字nDifferentColors定义了需要训练的颜色数量。请注意，图像背景也可以有自己的颜色。

集群算法

在选择集群算法时，有 KMeans++（固定集群数量）和 LBG（动态集群数量）可供选择。

最大集群半径

仅与 LBG 集群算法一起使用。这可以用来影响各个集群的分布。

全球优化

仅与 LBG 集群算法一起使用。这允许在更短的执行时间（bSingleSplitSteps = FALSE）和更好的结果（bSingleSplitSteps = TRUE）之间做出选择。

更少使用全局优化会更快，但可能导致不太理想的结果。

应用

```
VAR
    ipColorModel     :    ITcVnColorModel;
END_VAR

hr := F_VN_TrainImageColorExp2_ITcVnMlModel(
            ipSrcImage           := ipImageRef,
            ipColorModel         := ipColorModel,
            nDifferentColors     := 5, // 4 colors + 1 background = 5
            eMethod              := TCVN_CTM_LAB,
            ipMask               := 0,
            nSkipPixels          := 3,
            eClusteringAlgorithm:= TCVN_CA_KMEANSPP,
            nMaxClusterRadius    := 0,
            bSingleSplitSteps    := FALSE,
            hrPrev               := hr);
```

相关函数

有三种方法可以将参考颜色传递给相应的F_VN_ReferenceColorSimilarity函数：

- 作为一个经过训练的颜色模型，类型ITcVnColorModel
- 作为一个经过训练的颜色模型，类型ITcVnMlModel
- 直接使用TcVnVector3_LREAL中的数值

对于颜色模型，有相应的函数用于训练，而对于所有三种数据类型，各有一个函数用于执行：

- F_VN_TrainImageColor [▶ 1218]用于训练一个颜色模型
- F_VN_ReferenceColorSimilarity [▶ 1210]通过颜色模型进行分割

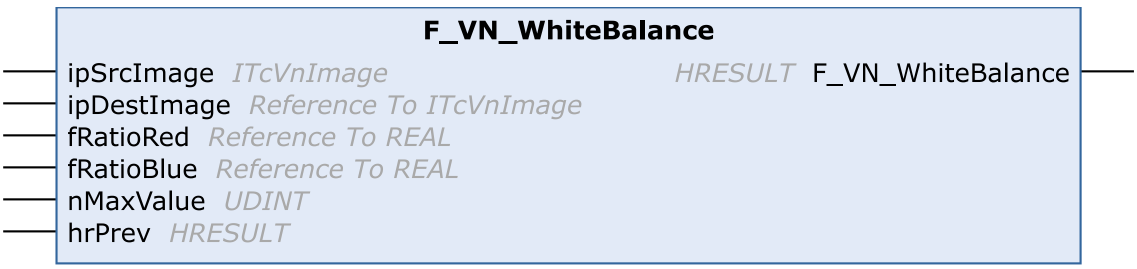
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.15.29 F_VN_WhiteBalance



Calculates the white balance ratios for a RGB image. Uses the green channel as a reference and computes the relative ratios for the red and blue channels. Requires a non-reflecting, not overexposed white colored object (e.g. a sheet of paper) in at least 100 pixels of the image to provide reasonable results. The resulting ratios can then be set on the camera (multiply with existing ratios), or applied in the PLC via MultiplyImageWithVector.

Syntax

Definition:

```
FUNCTION F_VN_WhiteBalance : HRESULT
VAR_INPUT
    ipSrcImage   : ITcVnImage;
    ipDestImage  : Reference To ITcVnImage;
    fRatioRed    : Reference To REAL;
    fRatioBlue   : Reference To REAL;
    nMaxValue    : UDINT;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (3 channels (RGB), USINT or UINT)
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image with white balance applied (optional, set to 0 if not required. An appropriate image will be created if required.)
fRatioRed	Reference To REAL	Returns the ratio for the red channel
fRatioBlue	Reference To REAL	Returns the ratio for the blue channel
nMaxValue	UDINT	Specify the maximum pixel value that would be achieved by overexposure (usually 255 for 8 bit or 4095 for 12 bit data)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16 Image Filtering

该组包含过滤图像的函数。

函数

模糊化

- F_VN_BilateralFilter(Exp) [▸ 1232]
- F_VN_BoxFilter(Exp) [▸ 1237]
- F_VN_GaussianFilter(Exp) [▸ 1250]

- F_VN_MedianFilter [▶ 1261]

边缘检测

- F_VN_LaplacianFilter(Exp) [▶ 1254]
- F_VN_ScharrFilter(Exp) [▶ 1268]
- F_VN_SobelFilter(Exp) [▶ 1276]

用户定义的过滤器

- F_VN_CustomFilter(Exp) [▶ 1242]
- F_VN_SeparableCustomFilter(Exp) [▶ 1272]

填充区域

- F_VN_BrightBorderObjects [▶ 1239]
- F_VN_DarkBorderObjects [▶ 1247]
- F_VN_FillHoles [▶ 1248]

形态学算子

- F_VN_CreateStructuringElement [▶ 1240]
- F_VN_MorphologicalOperator [▶ 1264]

本地极端情况

- F_VN_LocalMaxima [▶ 1258]
- F_VN_LocalMinima [▶ 1259]
- F_VN_RemoveLocalMaxima [▶ 1265]
- F_VN_RemoveLocalMinima [▶ 1267]

本地差异

- F_VN_VarianceFilter(Exp) [▶ 1280]

样本

图像过滤 [▶ 2688]

6.1.4.16.1 F_VN_BilateralFilter

Apply a Bilateral filter to smooth the image but preserve edges.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_BilateralFilter : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    nDiameter   : DINT;
    fSigmaColor : LREAL;
    fSigmaSpace : LREAL;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (Must be different from ipSrcImage! An appropriate destination image will be created if required.)
nDiameter	DINT	Diameter of the pixel neighborhood used for filtering (e.g. 5, 7, 9). If <= 0, it is automatically chosen dependend on fSigmaSpace.
fSigmaColor	LREAL	Sigma used for color space filtering (> 0). A larger value means that farther colors are mixed together.
fSigmaSpace	LREAL	Sigma used for coordinate space filtering (> 0). A larger value means that farther pixels can influence each other.
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_BilateralFilter对原始图像进行双边过滤。在保留图像中的边缘的同时，该过滤器使图像变得平滑并减少噪音。

参数

原始图像

原始图像ipSrcImage必须是元素类型TCVN_ET_USINT或浮点类型，且有 1 或 3 个通道。

结果图像

结果图像ipDestImage必须是与原始图像ipSrcImage不同的接口指针！

过滤器直径

过滤器直径nDiameter决定了一个过滤区域的大小。预计为奇数。如果nDiameter = 0，那么过滤器直径将根据fSigmaSpace自动计算。

颜色区域

这个值可能不是0。

房间面积

这个值可能不是0。

专家参数

关于更多参数，可查看专家版F_VN_BilateralFilterExp [▸ 1235]。

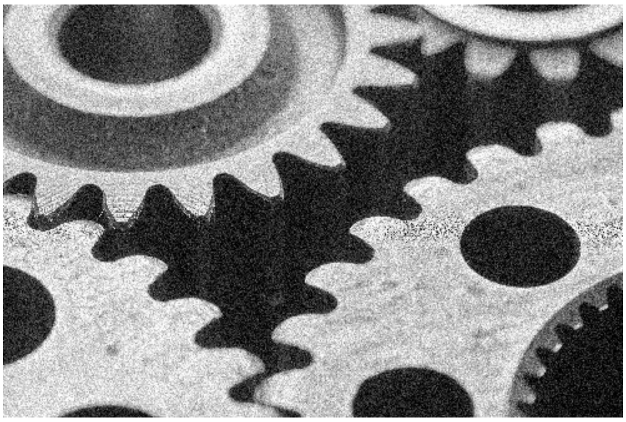
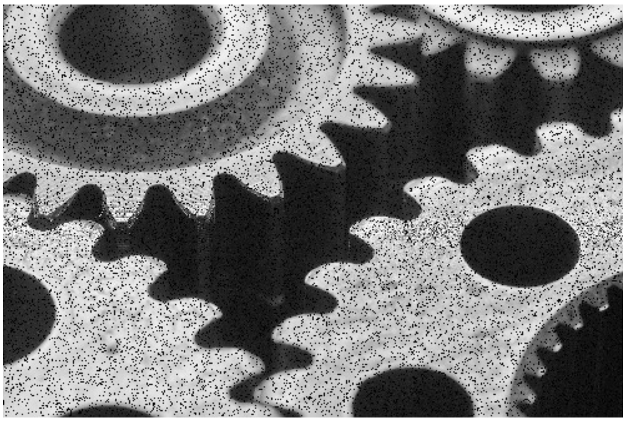
应用

例如，这个函数的应用可以如下所示：

```
hr := F_VN_BilateralFilter(
    ipSrcImage      :=  ipImage,
    ipDestImage     :=  ipImageRes,
    nDiameter       :=  nDiameter,
    fSigmaColor     :=  fSigmaColor,
```

```
    fSigmaSpace     :=  fSigmaSpace,
    hrPrev          :=  hr
);
```

未经处理的原始图像（第[1]行）在细节上已经表现出齿轮表面的精细结构。为了说明过滤器的效果，在原始图像中增加了高斯噪声（第[2] 行）和椒盐噪声（第[3]行）的额外干扰。

原始图像	结果图像： 直径 = 7 Sigma = 100
	
	
	

样本

- 模糊过滤器 [▸ 2688]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.2 F_VN_BilateralFilterExp

Apply a Bilateral filter to smooth the image but preserve edges. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_BilateralFilterExp : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    nDiameter   : DINT;
    fSigmaColor : LREAL;
    fSigmaSpace : LREAL;
    eBorderType : ETcVnBorderInterpolationMethod;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (Must be different from ipSrcImage! An appropriate destination image will be created if required.)
nDiameter	DINT	Diameter of the pixel neighborhood used for filtering (e.g. 5, 7, 9). If <= 0, it is automatically chosen dependend on fSigmaSpace.
fSigmaColor	LREAL	Sigma used for color space filtering (> 0). A larger value means that farther colors are mixed together.
fSigmaSpace	LREAL	Sigma used for coordinate space filtering (> 0). A larger value means that farther pixels can influence each other.
eBorderType	ETcVnBorderInterpolationMethod [▸ 145]	Image border handling
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

该函数F_VN_BilateralFilterExp是F_VN_BilateralFilter [▶ 1232]的专家版本。它包含额外的参数。

参数

原始图像

原始图像ipSrcImage必须是元素类型TCVN_ET_USINT或浮点类型，且有 1 或 3 个通道。

结果图像

结果图像ipDestImage必须是与原始图像ipSrcImage不同的接口指针！

过滤器直径

过滤器直径nDiameter决定了一个过滤区域的大小。预计为奇数。如果nDiameter = 0，那么过滤器直径将根据fSigmaSpace自动计算。

颜色区域

这个值可能不是0。

房间面积

这个值可能不是0。

边界推断

边界推断eBorderType的方法定义了如何将不存在的像素外推到图像边界之外，以计算图像边界的过滤值。更多细节，请参见ETcVnBorderInterpolationMethod [▶ 145]。

应用

例如，这个函数的应用可以如下所示：

```
hr := F_VN_BilateralFilterExp(
    ipSrcImage      :=  ipImage,
    ipDestImage     :=  ipImageRes,
    nDiameter       :=  nDiameter,
    fSigmaColor     :=  fSigmaColor,
    fSigmaSpace     :=  fSigmaSpace,
    eBorderType     :=  eBorderType,
    hrPrev          :=  hr
);
```

样本

- 模糊过滤器 [▶ 2688]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.3 F_VN_BoxFilter

Apply a box filter to an image.

Syntax

Definition:

```
FUNCTION F_VN_BoxFilter : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipDestImage   : Reference To ITcVnImage;
    nFilterWidth  : UDINT;
    nFilterHeight : UDINT;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
nFilterWidth	UDINT	Filter width in pixels
nFilterHeight	UDINT	Filter height in pixels
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

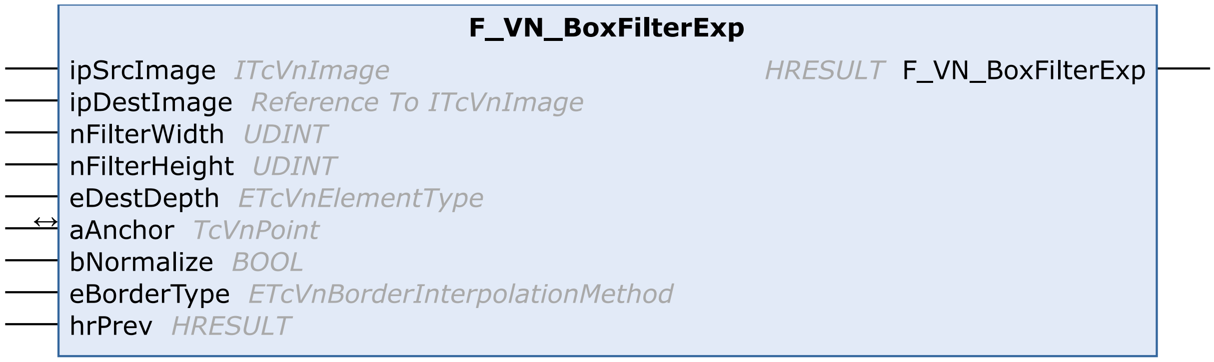
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.4 F_VN_BoxFilterExp



Apply a box filter to an image. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_BoxFilterExp : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipDestImage   : Reference To ITcVnImage;
    nFilterWidth  : UDINT;
    nFilterHeight : UDINT;
    eDestDepth    : ETcVnElementType;
END_VAR
VAR_IN_OUT
    aAnchor       : TcVnPoint;
END_VAR
VAR_INPUT
    bNormalize    : BOOL;
    eBorderType   : ETcVnBorderInterpolationMethod;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
nFilterWidth	UDINT	Filter width in pixels
nFilterHeight	UDINT	Filter height in pixels
eDestDepth	ETcVnElementType [▶ 178]	Destination image depth
bNormalize	BOOL	If true, the kernel is normalized by nFilterWidth * nFilterHeight
eBorderType	ETcVnBorderInterpolationMethod [▶ 145]	Image border handling
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aAnchor	TcVnPoint [▶ 139]	Anchor point of the kernel ([-1, -1] for center)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.5 F_VN_BrightBorderObjects

Find bright objects connected to the image border within a gray-scale single-channel image.

Syntax

Definition:

```
FUNCTION F_VN_BrightBorderObjects : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT, UINT, INT, REAL, or LREAL, 1 channel)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

这个函数将输入图像中的所有明亮的边框对象转移到结果图像中。

算法

图像中所有比周围像素更亮的物体都被移除。

参数

输入图像

输入图像ipSrcImage只能有一个通道，并且必须为以下元素类型之一：USINT、UINT、INT、REAL或LREAL。

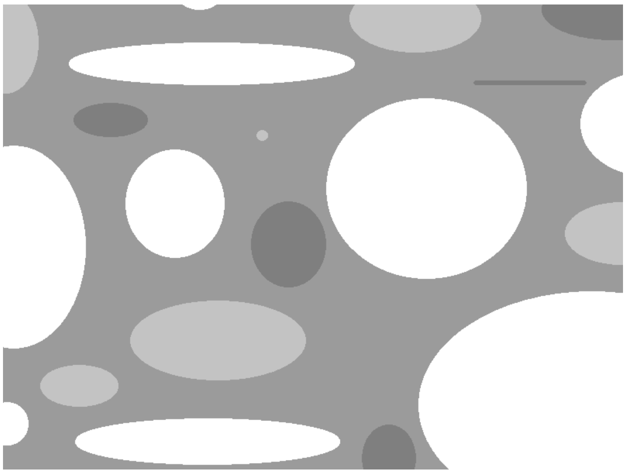
结果图像

结果图像ipDestImage返回明亮的边界对象。

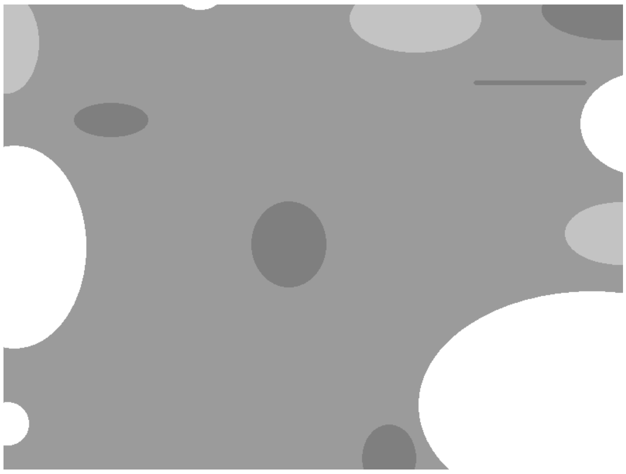
应用

```
hr := F_VN_BrightBorderObjects(
    ipSrcImage      :=  ipImageIn,
    ipDestImage     :=  ipImageRes,
    hrPrev          :=  hr
);
```

输入图像



结果图像



相关函数

- F_VN_DarkBorderObjects [▶ 1247]
- F_VN_FillHoles [▶ 1248]

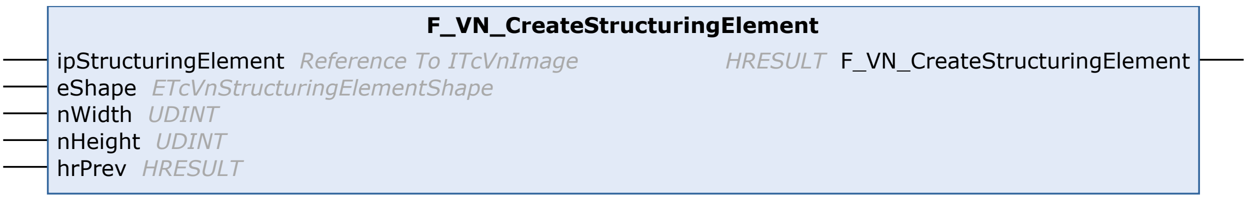
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.6 F_VN_CreateStructuringElement



Creates a structuring element for the usage with morphological operators and allocate an appropriate data buffer. The initial reference count is set to one if a new image interface is created and kept, otherwise.

Syntax

Definition:

```
FUNCTION F_VN_CreateStructuringElement : HRESULT
VAR_INPUT
    ipStructuringElement : Reference To ITcVnImage;
    eShape               : ETcVnStructuringElementShape;
    nWidth               : UDINT;
    nHeight              : UDINT;
    hrPrev               : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipStructuringElement	Reference To ITcVnImage [▸ 390]	Returns the created structuring element (Non-zero interface pointers are reused.)
eShape	ETcVnStructuringElementShape [▸ 197]	Shape of the structuring element (rectangle, cross, or ellipse)
nWidth	UDINT	Width
nHeight	UDINT	Height
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

该函数创建了一个结构元素，可通过F_VN_MorphologicalOperator [▸ 1264]用于形态学操作。

参数

结构元素

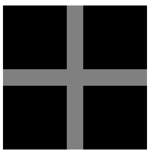
创建的结构元素ipStructuringElement通过使用数值1以像素表示结构元素返回为ITcVnImage [▸ 390]。

形状

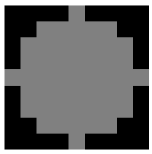
结构元素的形状eShape由枚举ETcVnStructuringElementShape [▸ 197]定义。以下形状可用：

TCVN_SES_RECTANGLE

TCVN_SES_CROSS



TCVN_SES_ELLIPSE



由于结构元素的像素值为 1，为了更好地显示，这里将它们按比例放大。

大小

结构元素的大小通过待创建图像ipStructuringElement的宽度nWidth和高度nHeight定义。

应用

例如，创建大小为9的方形结构元素如下所示：

```
hr := F_VN_CreateStructuringElement(
    ipStructuringElement    :=  ipElement,
    eShape                  :=  TCVN_SES_RECTANGLE,
```

```
    nWidth                  :=  9,
    nHeight                 :=  9,
    hrPrev                  :=  hr
);
```

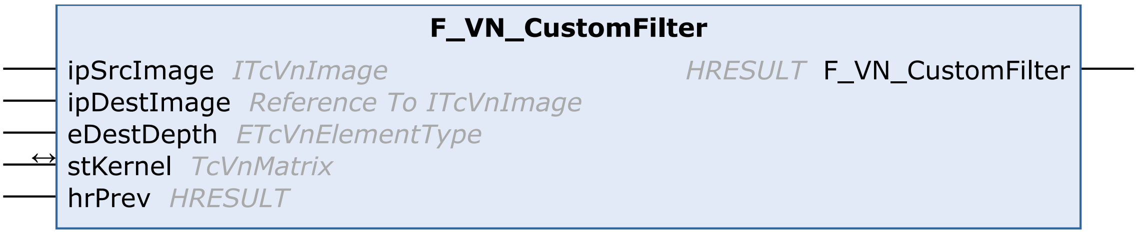
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.7 F_VN_CustomFilter



Apply a custom filter to the image.

Syntax

Definition:

```
FUNCTION F_VN_CustomFilter : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    eDestDepth  : ETcVnElementType;
END_VAR
VAR_IN_OUT
    stKernel    : TcVnMatrix;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
eDestDepth	ETcVnElementType [▸ 178]	Destination image depth
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stKernel	TcVnMatrix [▸ 211]	Custom filter kernel with values of type REAL or LREAL

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_CustomFilter对输入图像应用用户定义的过滤器。过滤器内核由一个矩阵定义。这个矩阵与图像一起被折叠起来。

参数

输入图像

输入图像ipSrcImage可以有任何可用的格式。

结果图像

结果图像ipDestImage的格式与输入图像ipSrcImage相同。

结果深度

结果深度eDestDepth定义了结果图像ipDestImage的元素类型。一个更大的元素类型可以显示更多的信息。

过滤器内核

用户定义的过滤器内核stKernel以矩阵形式传输。允许的矩阵元素类型为TCVN_ET_REAL和TCVN_ET_LREAL。通过函数F_VN_InitMatrixStruct [▶ 1484]，可以创建类型为TcVnMatrix [▶ 211]的相应矩阵。

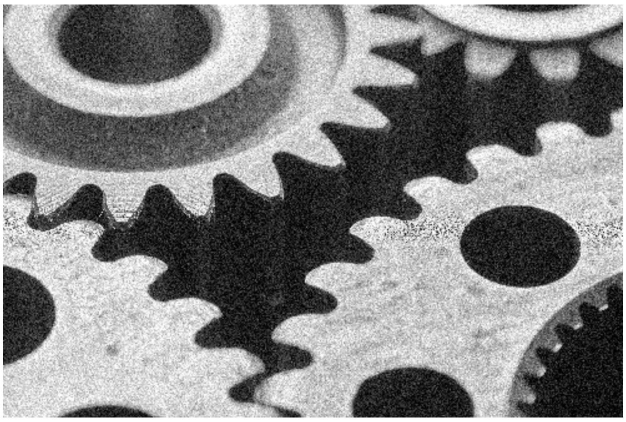
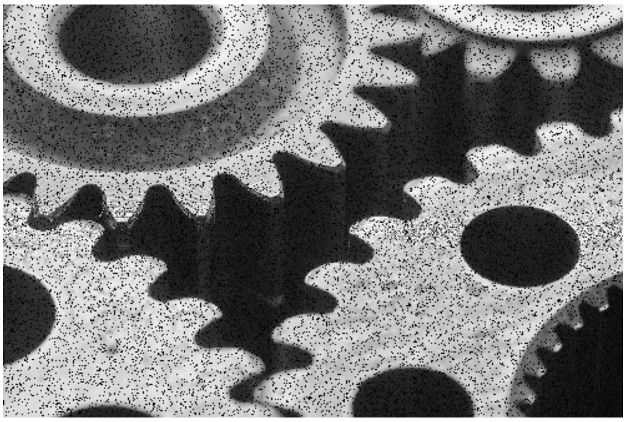
专家参数

专家版F_VN_CustomFilterExp [▶ 1245]包含额外的参数。

应用

```
hr := F_VN_InitMatrixStruct(
    pSrcBuffer      := ADR(aMatrixArray7x7),
    stDestMatrix    := stKernelMatrix,
    nRows           := 7,
    nCols           := 7,
    eElementType    := TCVN_ET_REAL,
    hrPrev          := hr
);

hr := F_VN_CustomFilter(
    ipSrcImage  :=  ipImageIn,
    ipDestImage :=  ipImageRes,
    eDestDepth  :=  TCVN_ET_USINT,
    stKernel    :=  stKernelMatrix,
    hrPrev      :=  hr
);
```

- 未经处理的原始图像（第1行）在细节上已经表现出齿轮表面的精细结构。为了说明过滤器的效果，在原始图像中增加了高斯噪声（第2 行）和椒盐噪声（第3行）的额外干扰。作为一个例子，实现了一个平均过滤器；关于更详细的描述，可查看模糊过滤器 [▶ 2688]的样本：

原始图像	结果图像： 平均数过滤器
	
	
	

样本

- 模糊过滤器 [▶ 2688]
-

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.8 F_VN_CustomFilterExp

F_VN_CustomFilterExp

ipSrcImage *ITcVnImage* — *HRESULT* F_VN_CustomFilterExp
ipDestImage *Reference To ITcVnImage*
eDestDepth *ETcVnElementType*
stKernel *TcVnMatrix*
aAnchor *TcVnPoint*
fDelta *LREAL*
eBorderType *ETcVnBorderInterpolationMethod*
hrPrev *HRESULT*

Apply a custom filter to the image. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_CustomFilterExp : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    eDestDepth  : ETcVnElementType;
END_VAR
VAR_IN_OUT
    stKernel    : TcVnMatrix;
    aAnchor     : TcVnPoint;
END_VAR
VAR_INPUT
    fDelta      : LREAL;
    eBorderType : ETcVnBorderInterpolationMethod;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
eDestDepth	ETcVnElementType [▸ 178]	Destination image depth
fDelta	LREAL	Value added to each pixel after filtering
eBorderType	ETcVnBorderInterpolationMethod [▸ 145]	Image border handling
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stKernel	TcVnMatrix [▸ 211]	Custom filter kernel with values of type REAL or LREAL
aAnchor	TcVnPoint [▸ 139]	Anchor point of the kernel

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_CustomFilterExp是F_VN_CustomFilter [▶ 1242]的专家版本。它包含额外的参数。

参数

输入图像

输入图像ipSrcImage可以有任何可用的格式。

结果图像

结果图像ipDestImage的格式与输入图像ipSrcImage相同。

结果深度

结果深度eDestDepth定义了结果图像ipDestImage的元素类型。一个更大的元素类型可以显示更多的信息。

过滤器内核

用户定义的过滤器内核stKernel以矩阵形式传输。允许的矩阵元素类型为TCVN_ET_REAL和TCVN_ET_LREAL。通过函数F_VN_InitMatrixStruct [▶ 1484]，可以创建类型为TcVnMatrix [▶ 211]的相应矩阵。

锚点

锚点aAnchor定义了在输入图像ipSrcImage的每个像素上移动过滤器内核的哪个点。

Delta

常量 deltafDelta被添加到过滤操作的结果中。

边界推断

边界推断eBorderType的方法定义了如何将不存在的像素外推到图像边界之外，以计算图像边界的过滤值。更多细节，请参见ETcVnBorderInterpolationMethod [▶ 145]。

应用

```
hr := F_VN_CustomFilterExp(
    ipSrcImage  :=  ipImageIn,
    ipDestImage :=  ipImageRes,
    eDestDepth  :=  TCVN_ET_USINT,
    stKernel    :=  stKernelMatrix,
    aAnchor     :=  aAnchorPoint,
    fDelta      :=  0,
    eBorderType :=  TCVN_BIM_DEFAULT,
    hrPrev      :=  hr
);
```

样本

-

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.9 F_VN_DarkBorderObjects

F_VN_DarkBorderObjects
ipSrcImage *ITcVnImage* — *HRESULT* F_VN_DarkBorderObjects
ipDestImage *Reference To ITcVnImage*
hrPrev *HRESULT*

Find dark objects connected to the image border within a gray-scale single-channel image. (equivalent to filling holes)

Syntax

Definition:

```
FUNCTION F_VN_DarkBorderObjects : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (USINT, UINT, INT, REAL, or LREAL, 1 channel)
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

这个函数将输入图像中的所有暗色边界对象转移到结果图像中。

算法

图像中所有比周围像素更暗的对象都被删除。

参数

输入图像

输入图像ipSrcImage只能有一个通道，并且必须为以下元素类型之一：USINT、UINT、INT、REAL或LREAL。

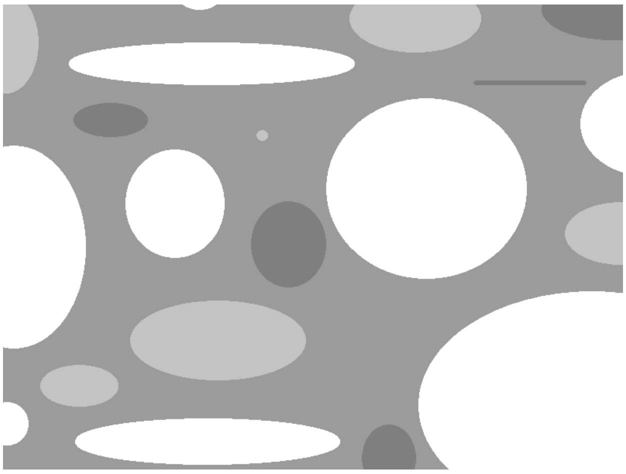
结果图像

结果图像ipDestImage返回深色边界对象。

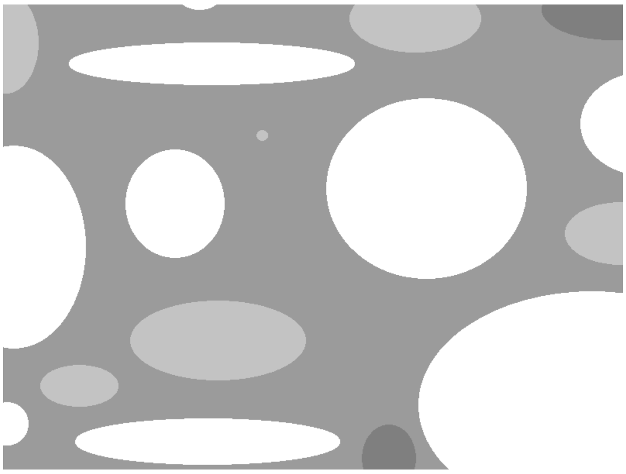
应用

```
hr := F_VN_DarkBorderObjects(
    ipSrcImage      :=  ipImageIn,
    ipDestImage     :=  ipImageRes,
    hrPrev          :=  hr
);
```

输入图像



结果图像



相关函数

- F_VN_FillHoles [▶ 1248] als Alias
- F_VN_BrightBorderObjects [▶ 1239]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.10 F_VN_FillHoles

Fill holes within a gray-scale single-channel image. (equivalent to finding dark border objects)

Syntax

Definition:

```
FUNCTION F_VN_FillHoles : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT, UINT, INT, REAL, or LREAL, 1 channel)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

这个函数填补了输入图像中的孔。

算法

图像中所有比周围像素更暗的对象都被删除。

参数

输入图像

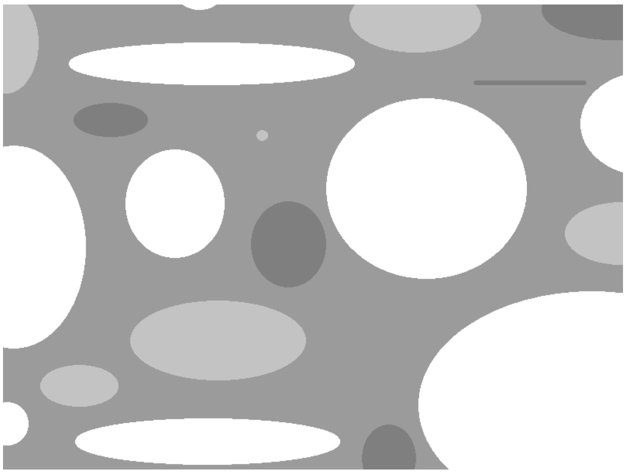
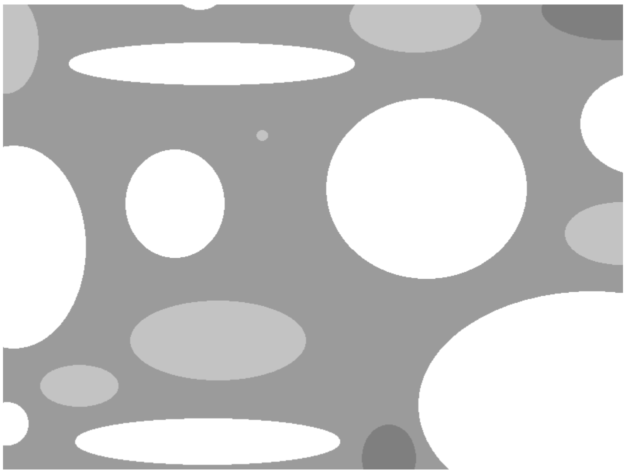
输入图像ipSrcImage只能有一个通道，并且必须为以下元素类型之一：USINT、UINT、INT、REAL或LREAL。

结果图像

结果图像ipDestImage返回填充了孔的图像。

应用

```
hr := F_VN_FillHoles(
    ipSrcImage      :=  ipImageIn,
    ipDestImage     :=  ipImageRes,
    hrPrev          :=  hr
);
```

输入图像	结果图像
	

相关函数

- F_VN_DarkBorderObjects [▶ 1247] as alias
- F_VN_BrightBorderObjects [▶ 1239]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.11 F_VN_GaussianFilter

Apply a Gaussian filter to smooth the image.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_GaussianFilter : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipDestImage   : Reference To ITcVnImage;
    nFilterWidth  : UDINT;
    nFilterHeight : UDINT;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
nFilterWidth	UDINT	Filter width in pixels (1, 3, 5, 7, ...)
nFilterHeight	UDINT	Filter height in pixels (1, 3, 5, 7, ...)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_GaussianFilter对输入图像应用高斯过滤器。

算法

高斯过滤器在图像处理中经常用于减少噪音或平滑图像。每个像素被其环境的加权平均数所取代，去除较小的细节但保留较大的细节。过滤器越大，图像就越平滑（因此显得更模糊）。

参数

输入图像

输入图像ipSrcImage可以有任何可用的格式。

结果图像

结果图像ipDestImage格式与输入图像ipSrcImage相同。

过滤器尺寸

高斯过滤器的大小在 X 方向由nFilterWidth且在 Y 方向由nFilterHeight描述。尺寸规格必须是奇数，因为必须始终有一个中心像素。

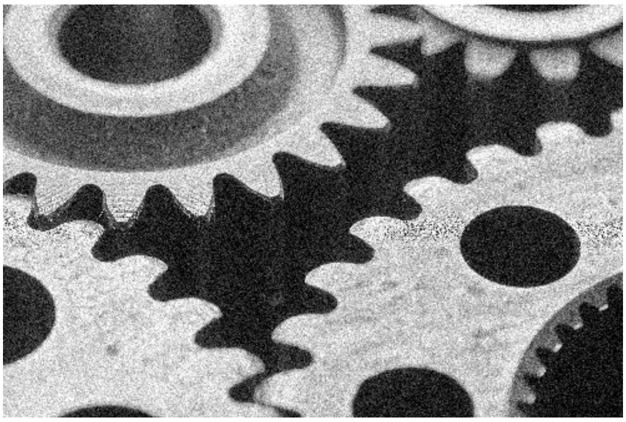
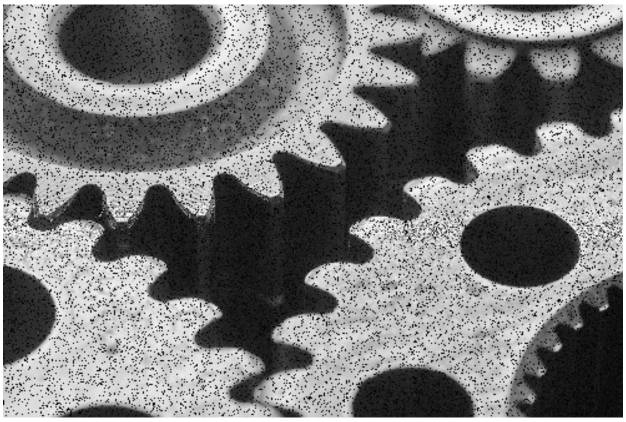
专家参数

专家版F_VN_GaussianFilterExp [▸ 1253]包含额外的参数。

应用

```
hr := F_VN_GaussianFilter(
    ipSrcImage      :=  ipImageIn,
    ipDestImage     :=  ipImageRes,
    nFilterWidth    :=  7,
    nFilterHeight   :=  7,
    hrPrev          :=  hr
);
```

未经处理的原始图像（第1行）在细节上已经表现出齿轮表面的精细结构。为了说明过滤器的效果，在原始图像中增加了高斯噪声（第2 行）和椒盐噪声（第3行）的额外干扰。

原始图像	结果图片
	
	
	

样本

- 模糊过滤器 [▶ 2688]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.12 F_VN_GaussianFilterExp

Apply a Gaussian filter to smooth the image. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_GaussianFilterExp : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipDestImage   : Reference To ITcVnImage;
    nFilterWidth  : UDINT;
    nFilterHeight : UDINT;
    fSigmaX       : LREAL;
    fSigmaY       : LREAL;
    eBorderType   : ETcVnBorderInterpolationMethod;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
nFilterWidth	UDINT	Filter width in pixels (1, 3, 5, 7, ...)
nFilterHeight	UDINT	Filter height in pixels (1, 3, 5, 7, ...)
fSigmaX	LREAL	Gaussian kernel standard deviation in X direction (>= 0, automatically chosen if 0)
fSigmaY	LREAL	Gaussian kernel standard deviation in Y direction (>= 0, automatically chosen if 0)
eBorderType	ETcVnBorderInterpolationMethod [▶ 145]	Image border handling
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_GaussianFilterExp是F_VN_GaussianFilter [▶ 1250]的专家版本。它包含额外的参数。

参数

输入图像

输入图像ipSrcImage可以有任何可用的格式。

结果图像

结果图像ipDestImage格式与输入图像ipSrcImage相同。

过滤器尺寸

高斯过滤器的大小在 X 方向由nFilterWidth且在 Y 方向由nFilterHeight描述。尺寸规格必须是奇数，因为必须始终有一个中心像素。

标准偏差

X 方向的标准偏差 fSigmaX 和 Y 方向的标准偏差 fSigmaY 不仅定义了过滤器的大小，而且还定义了过滤器中使用的高斯曲线的形状。

边界推断

边界推断eBorderType的方法定义了如何将不存在的像素外推到图像边界之外，以计算图像边界的过滤值。更多细节，请参见ETcVnBorderInterpolationMethod [▶ 145]。

应用

```
hr := F_VN_GaussianFilterExp(
    ipSrcImage       :=  ipImageIn,
    ipDestImage      :=  ipImageRes,
    nFilterWidth     :=  3,
    nFilterHeight    :=  3,
    fSigmaX          :=  5,
    fSigmaY          :=  5,
    eBorderType      :=  TCVN_BIM_DEFAULT,
    hrPrev           :=  hr
);
```

样本

- 模糊过滤器 [▶ 2688]

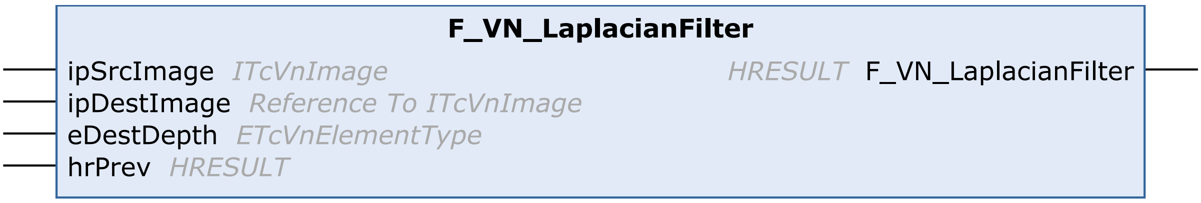
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.13 F_VN_LaplacianFilter



Apply a Laplacian filter to an image.

Syntax

Definition:

```
FUNCTION F_VN_LaplacianFilter : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
```

```
    eDestDepth   : ETcVnElementType;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
eDestDepth	ETcVnElementType [▶ 178]	Destination image depth
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_LaplacianFilter对输入图像应用了拉普拉斯过滤器。

算法

拉普拉斯过滤器可用于通过近似相邻像素强度转换的二阶导数来检测边缘。由于导数也有负值，建议对结果图像使用有符号的元素类型（例如TCVN_ET_INT）。

对于 nKernelSize = 3, 5, ...，拉普拉斯过滤器的计算方法是在指定的窗口大小上加上 x 和 y 方向的二次导数。对于 nKernelSize = 1，使用以下特殊的 3 x 3 内核：

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

参数

输入图像

输入图像 ipSrcImage 可以是任何可用的格式，除了一般不支持的数据类型 SINT 和 DINT。

结果图像

结果图像 ipDestImage 的大小与输入图像 ipSrcImage 相同，但像素格式由结果深度eDestDepth决定。

结果深度

结果深度eDestDepth定义了结果图像ipDestImage的元素类型。一个更大的元素类型可以显示更多的信息。

注意

输入图像和结果深度的组合

只有与输入图像相同或更大的数据类型才能作为结果深度，否则数据就会丢失。例如，一个 UINT 不能被转换为 USINT。此外，还有两个限制，所以 UINT 不能被转换成 INT，且 REAL 不能被转换成 LREAL。

专家参数

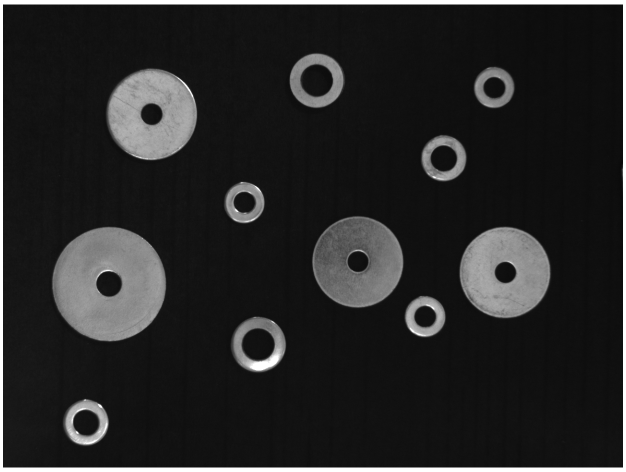
专家版F_VN_LaplacianFilterExp [▶ 1256]包含额外的参数。

应用

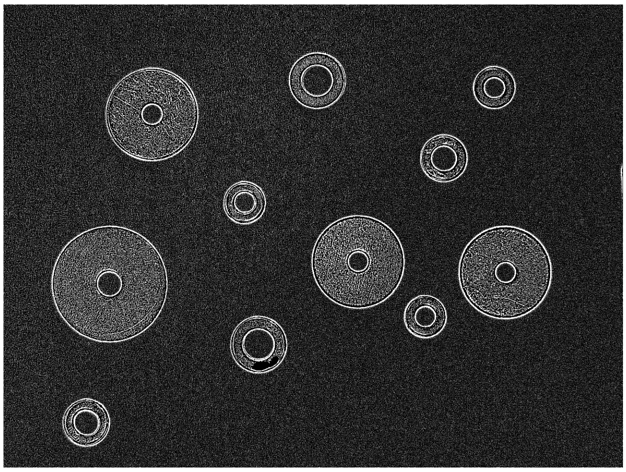
例如，结果深度为 8 位的拉普拉斯过滤器的应用如下所示：

```
hr := F_VN_LaplacianFilter(
    ipSrcImage  :=  ipImageIn,
    ipDestImage :=  ipImageRes,
    eDestDepth  :=  TCVN_ET_USINT,
    hrPrev      :=  hr
);
```

原始图像



结果图像



样本

- 边缘检测过滤器 [▶ 2690]

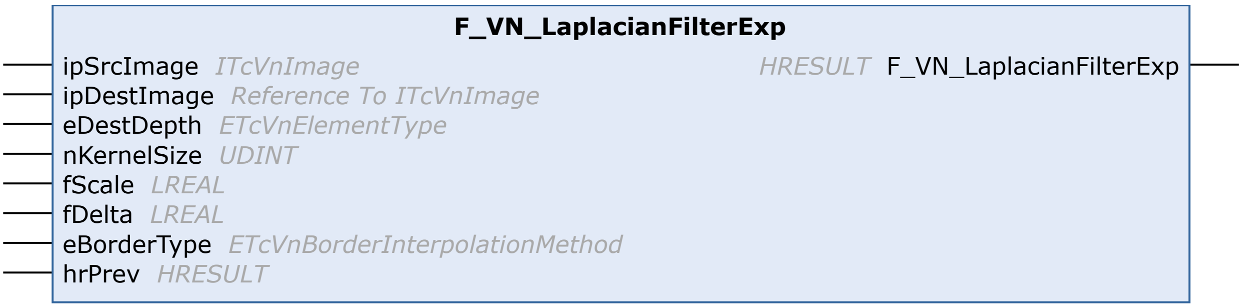
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.14 F_VN_LaplacianFilterExp



Apply a Laplacian filter to an image. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_LaplacianFilterExp : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    eDestDepth  : ETcVnElementType;
```

```
    nKernelSize : UDINT;
    fScale      : LREAL;
    fDelta      : LREAL;
    eBorderType : ETcVnBorderInterpolationMethod;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
eDestDepth	ETcVnElementType [▶ 178]	Destination image depth
nKernelSize	UDINT	Aperture size used to compute the second-derivative filters (1, 3, 5, ..., 31)
fScale	LREAL	Scale factor for the computed derivative values
fDelta	LREAL	Delta value that is added to the results before storing them
eBorderType	ETcVnBorderInterpolationMethod [▶ 145]	Image border handling
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_LaplacianFilterExp是F_VN_LaplacianFilter [▶ 1254]的专家版本。它包含额外的参数。

参数

输入图像

输入图像 ipSrcImage 可以是任何可用的格式，除了一般不支持的数据类型 SINT 和 DINT。

结果图像

结果图像 ipDestImage 的大小与输入图像 ipSrcImage 相同，但像素格式由结果深度eDestDepth决定。

结果深度

结果深度eDestDepth定义了结果图像ipDestImage的元素类型。一个更大的元素类型可以显示更多的信息。

> *注意*
>
> **输入图像和结果深度的组合**
>
> 只有与输入图像相同或更大的数据类型才能作为结果深度，否则数据就会丢失。例如，一个 UINT 不能被转换为 USINT。此外，还有两个限制，所以 UINT 不能被转换成 INT，且 REAL 不能被转换成 LREAL。

内核大小

参数nKernelSize定义了用于近似二阶导数的过滤器核的大小。只允许使用1和31之间的奇数。

缩放

拉普拉斯运算的结果乘以缩放系数fScale。

Delta

在通过fScale缩放后，常量 deltafDelta被添加到夏尔运算的结果中。

边界推断

边界推断eBorderType的方法定义了如何将不存在的像素外推到图像边界之外，以计算图像边界的过滤值。更多细节，请参见ETcVnBorderInterpolationMethod [▶ 145]。

应用

例如，内核大小为3×3的拉普拉斯过滤器的应用如下所示：

```
hr := F_VN_LaplacianFilterExp(
    ipSrcImage   :=  ipImageIn,
    ipDestImage  :=  ipImageRes,
    eDestDepth   :=  TCVN_ET_USINT,
    nKernelSize  :=  3,
    fScale       :=  1,
    fDelta       :=  0,
    eBorderType  :=  TCVN_BIM_DEFAULT,
    hrPrev       :=  hr
);
```

样本

- 边缘检测过滤器 [▶ 2690]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.15 F_VN_LocalMaxima

Find local maxima in a gray-scale single-channel image. The found maxima are marked by a value of 1 in the destination image.

Syntax

Definition:

```
FUNCTION F_VN_LocalMaxima : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT or UINT, 1 channel)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (Must be different from ipSrcImage! An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_LocalMaxima找到输入图像中的局部极大值。

参数

输入图像

输入图像ipSrcImage必须是位深为 8 或 16 位的单通道灰度图像（USINT 或 UINT）。

结果图像

结果图像ipDestImage与输入图像ipSrcImage具有相同的类型。在这里，输入图像的每个局部最大值都以像素值 1 标记。

应用

寻找局部最大值如下所示：

```
hr := F_VN_LocalMaxima(
    ipSrcImage  :=  ipImageIn,
    ipDestImage :=  ipImageRes,
    hrPrev      :=  hr
);
```

样本

- 消除局部极值

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.16 F_VN_LocalMinima

Find local minima in a gray-scale single-channel image. The found minima are marked by a value of 1 in the destination image.

Syntax

Definition:

```
FUNCTION F_VN_LocalMinima : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (USINT or UINT, 1 channel)
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (Must be different from ipSrcImage! An appropriate destination image will be created if required.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_LocalMinima在输入图像中寻找局部极小值。

参数

输入图像

输入图像ipSrcImage必须是位深为 8 或 16 位的单通道灰度图像（USINT 或 UINT）。

结果图像

结果图像ipDestImage与输入图像ipSrcImage具有相同的类型。在这里，输入图像的每个局部最小值都以像素值 1 标记。

应用

寻找局部最小值如下所示：

```
hr := F_VN_LocalMinima(
    ipSrcImage  :=  ipImageIn,
    ipDestImage :=  ipImageRes,
    hrPrev      :=  hr
);
```

样本

- 消除局部极值

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.17 F_VN_MedianFilter

Apply a Median filter to an image.

Syntax

Definition:

```
FUNCTION F_VN_MedianFilter : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    nFilterSize : UDINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (for nFilterSize 3 or 5: USINT, UINT, REAL. For bigger filters, only USINT is supported.
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
nFilterSize	UDINT	Size (width and height) of the filter (3, 5, 7, ...)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

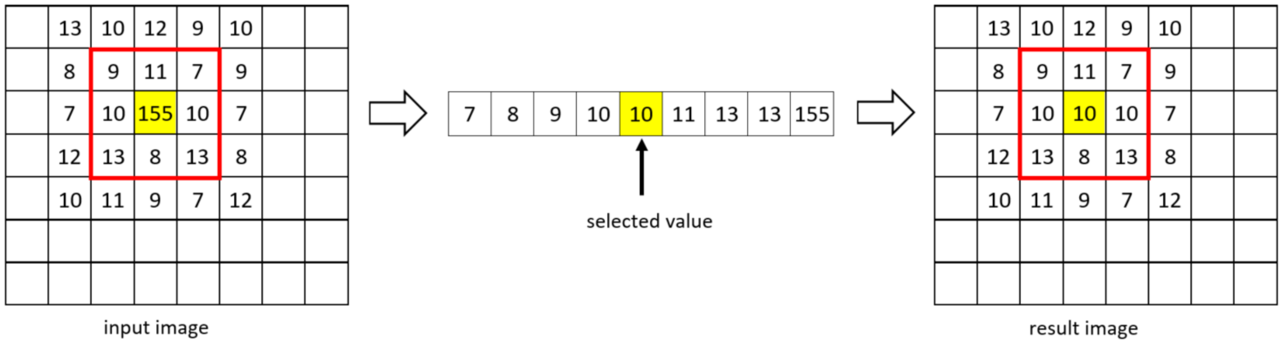
更多信息

函数F_VN_MedianFilter对输入图像应用了中位数过滤器。

算法

中位数过滤器是一种排序过滤器，经常用来抑制噪音。过滤器矩阵捕捉到的所有像素值按大小排序，然后用排序后的列表中心位置的值替换当前值。与具有加权平均值的过滤器（如高斯过滤器）相比，其优势在于个别像素的强离群值（如以下样本图像中的 155）被完全忽略，因此不会对结果产生负面影响。

过滤器矩阵的大小由nFilterSize指定，且只接受奇数整数>=3。



参数

输入图像

输入图像ipSrcImage可以有任何数量的通道。元素类型USINT、UINT和REAL允许使用过滤器尺寸nFilterSize的3或5。对于较大的过滤器，只支持元素类型USINT。

结果图像

结果图像ipDestImage返回过滤后的图像。

过滤器尺寸

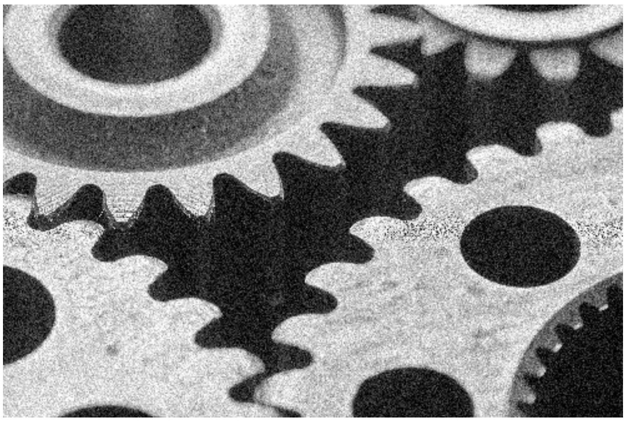
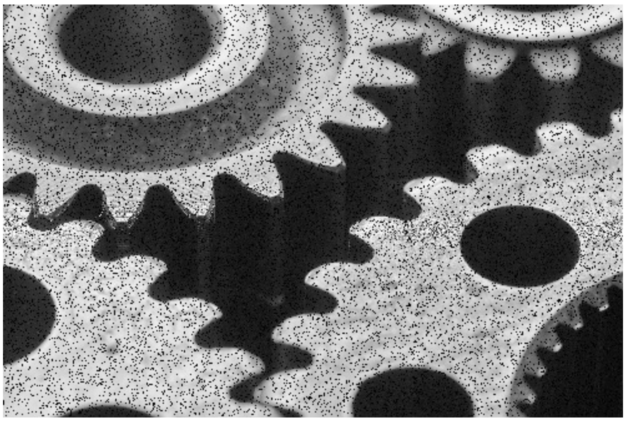
过滤器尺寸nFilterSize定义了过滤器区域的高度和宽度。它必须是一个奇数且>= 3。

应用

例如，尺寸7×7的中位数过滤器的应用如下所示：

```
hr := F_VN_MedianFilter(
    ipSrcImage      :=  ipImageIn,
    ipDestImage     :=  ipImageRes,
    nFilterSize     :=  7,
    hrPrev          :=  hr
);
```

未经处理的原始图像（第1行）在细节上已经表现出齿轮表面的精细结构。为了说明过滤器的效果，在原始图像中增加了高斯噪声（第2 行）和椒盐噪声（第3行）的额外干扰。

原始图像	结果图片
	
	
	

样本

- 模糊过滤器 [▶ 2688]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.18 F_VN_MorphologicalOperator

F_VN_MorphologicalOperator	
ipSrcImage *ITcVnImage*	*HRESULT* F_VN_MorphologicalOperator
ipDestImage *Reference To ITcVnImage*	
eOperator *ETcVnMorphologicalOperator*	
ipStructuringElement *ITcVnImage*	
hrPrev *HRESULT*	

Apply a morphological operator.

Syntax

Definition:

```
FUNCTION F_VN_MorphologicalOperator : HRESULT
VAR_INPUT
    ipSrcImage           : ITcVnImage;
    ipDestImage          : Reference To ITcVnImage;
    eOperator            : ETcVnMorphologicalOperator;
    ipStructuringElement : ITcVnImage;
    hrPrev               : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (only 1 channel supported for reconstruction operators)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
eOperator	ETcVnMorphologicalOperator [▶ 188]	Operator type
ipStructuringElement	ITcVnImage [▶ 390]	Structuring element to be applied (Typically created via F_VN_CreateStructuringElement.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_MorphologicalOperator对输入图像进行形态学运算。

参数

原始图像

重构算子只能应用于单通道图像，其他算子对原始图像没有特殊要求ipSrcImage。

结果图像

修改后的图像会在结果图像ipDestImage中返回。

形态学算子

参数eOperator定义了哪种形态学算子被应用于原始图像ipSrcImage。ETcVnMorphologicalOperator [▶ 188] 枚举中的所有算子都可用。

结构元素

结构元素ipStructuringElement是指定形态学算子的过滤区域的元素。它可以在函数 F_VN_CreateStructuringElement [▶ 1240]的帮助下创建。

应用

例如，通过尺寸3×3的矩形过滤区域应用开放性算子如下所示：

```
VAR
    ipElement : ITcVnImage;
END_VAR

hr := F_VN_CreateStructuringElement(ipElement, TCVN_SES_RECTANGLE, 3, 3, hr);

hr := F_VN_MorphologicalOperator(
    ipSrcImage   :=  ipImageIn,
    ipDestImage  :=  ipImageRes,
    eOperator    :=  TCVN_MO_OPENING,
    ipStructuringElement  :=  ipElement,
    hrPrev       :=  hr
);
```

样本

- 形态学算子 [▶ 2692]

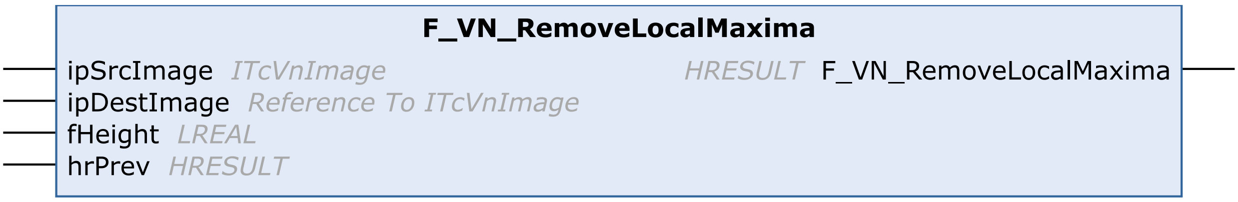
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.19 F_VN_RemoveLocalMaxima



Remove local maxima up to a given height from a gray-scale single-channel image.

Syntax

Definition:

```
FUNCTION F_VN_RemoveLocalMaxima : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    fHeight     : LREAL;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT, UINT, INT, REAL, or LREAL, 1 channel)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
fHeight	LREAL	Maximum height of the maxima to be removed (must be greater than 0)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_RemoveLocalMaxima从输入图像中删除局部极大值。

参数

输入图像

输入图像ipSrcImage只能有一个通道，并且必须为以下元素类型之一：USINT、UINT、INT、REAL或LREAL。

结果图像

结果图像ipDestImage返回已经去除局部极大值的图像。

最大高度

最大高度fHeight定义了输入图像中的特定高度，如果不超过该高度，极大值将被移除。高度描述的是最大值和有关像素被设定的值之间的强度差异。fHeight必须大于0。

应用

例如，去除不超过高度10的局部极大值如下所示：

```
hr := F_VN_RemoveLocalMaxima(
    ipSrcImage      :=  ipImageIn,
    ipDestImage     :=  ipImageRes,
    fHeight         :=  10,
    hrPrev          :=  hr
);
```

样本

- 消除局部极值

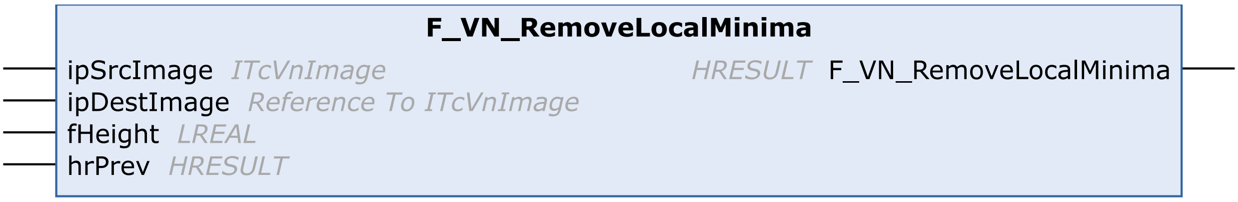
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.20 F_VN_RemoveLocalMinima



Remove local minima up to a given height from a gray-scale single-channel image.

Syntax

Definition:

```
FUNCTION F_VN_RemoveLocalMinima : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    fHeight     : LREAL;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (USINT, UINT, INT, REAL, or LREAL, 1 channel)
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
fHeight	LREAL	Maximum height of the minima to be removed (must be greater than 0)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_RemoveLocalMinima从输入图像中删除局部极小值。

参数

输入图像

输入图像ipSrcImage只能有一个通道，并且必须为以下元素类型之一：USINT、UINT、INT、REAL或LREAL。

结果图像

结果图像ipDestImage返回已经去除局部极小值的图像。

最大高度

最大高度fHeight定义了输入图像中的特定高度，如果不超过该高度，极小值将被移除。高度描述的是最小值和有关像素被设定的值之间的强度差异。fHeight必须大于0。

应用

例如，去除不超过高度10的局部极小值如下所示：

```
hr := F_VN_RemoveLocalMinima(
    ipSrcImage      :=  ipImageIn,
    ipDestImage     :=  ipImageRes,
    fHeight         :=  10,
    hrPrev          :=  hr
);
```

样本

- 消除局部极值

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.21 F_VN_ScharrFilter

Calculates the first order derivative in x or y direction using a Scharr filter.

Syntax

Definition:

```
FUNCTION F_VN_ScharrFilter : HRESULT
VAR_INPUT
    ipSrcImage       : ITcVnImage;
    ipDestImage      : Reference To ITcVnImage;
    eDestDepth       : ETcVnElementType;
    eFilterDirection : ETcVnFilterDirection;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
eDestDepth	ETcVnElementType [▶ 178]	Destination image depth
eFilterDirection	ETcVnFilterDirection [▶ 182]	Filter direction
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_ScharrFilter对输入图像应用夏尔过滤器。夏尔过滤器的功能类似于索贝尔过滤器 [▶ 1276]，而区别仅在于内核的不同。

算法

夏尔过滤器与索贝尔过滤器密切相关。然而，它提供的结果比 3 x 3 索贝尔过滤器更准确。

$$G_x = \begin{bmatrix} -3 & 0 & +3 \\ -10 & 0 & +10 \\ -3 & 0 & +3 \end{bmatrix} \qquad G_y = \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ +3 & +10 & +3 \end{bmatrix}$$

参数

原始图像

原始图像ipSrcImage可以有任何格式。

结果图像

结果图像ipDestImage返回过滤结果，且其格式与原始图像ipSrcImage相同。

结果深度

结果深度eDestDepth定义了结果图像ipDestImage的元素类型。一个更大的元素类型可以显示更多的信息。

过滤方向

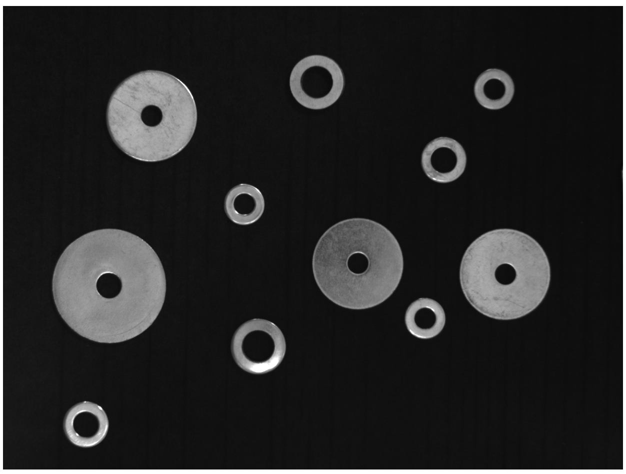
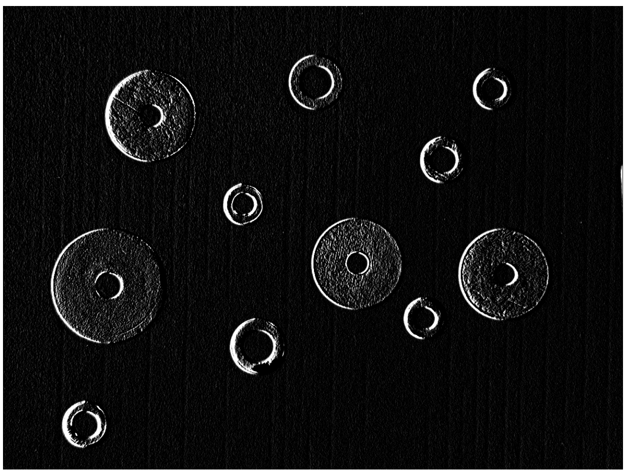
过滤方向eFilterDirection定义了夏尔过滤器是在 X 方向（TCVN_FD_X）还是在 Y 方向（TCVN_FD_Y）应用于图像。

专家参数

关于更多参数，可查看专家版F_VN_ScharrFilterExp [▶ 1270]。

应用

```
hr := F_VN_ScharrFilter(
    ipSrcImage       :=  ipImageIn,
    ipDestImage      :=  ipImageRes,
    eDestDepth       :=  TCVN_ET_USINT,
    eFilterDirection:=  TCVN_FD_X,
    hrPrev           :=  hr,
);
```

原始图像	结果图像
	

样本

- 边缘检测过滤器 [▶ 2690]

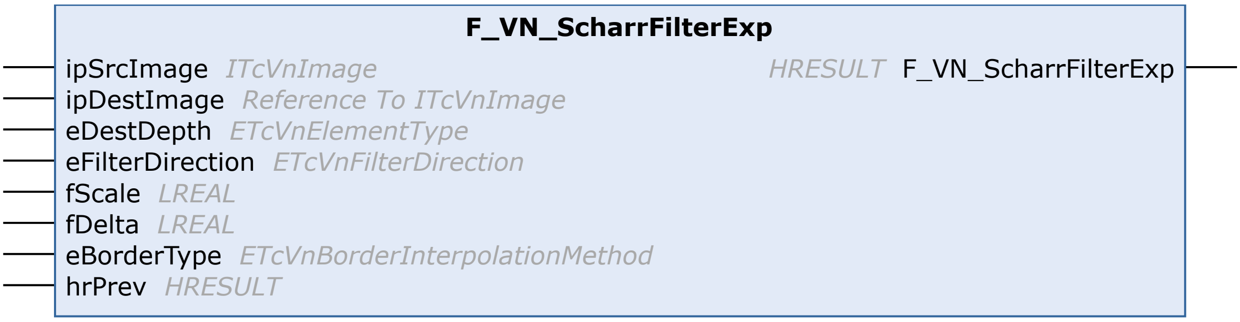
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.22 F_VN_ScharrFilterExp



Calculates the first order derivative in x or y direction using a Scharr filter. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_ScharrFilterExp : HRESULT
VAR_INPUT
    ipSrcImage       : ITcVnImage;
    ipDestImage      : Reference To ITcVnImage;
    eDestDepth       : ETcVnElementType;
    eFilterDirection : ETcVnFilterDirection;
    fScale           : LREAL;
    fDelta           : LREAL;
    eBorderType      : ETcVnBorderInterpolationMethod;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
eDestDepth	ETcVnElementType [▸ 178]	Destination image depth
eFilterDirection	ETcVnFilterDirection [▸ 182]	Filter direction
fScale	LREAL	Scale factor for the computed derivative values
fDelta	LREAL	Delta value that is added to the results prior to storing them in dest
eBorderType	ETcVnBorderInterpolationMethod [▸ 145]	Image border handling
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_ScharrFilterExp是F_VN_ScharrFilter [▶ 1268]的专家版本。它包含额外的参数。

参数

原始图像

原始图像ipSrcImage可以有任何格式。

结果图像

结果图像ipDestImage返回过滤结果，且其格式与原始图像ipSrcImage相同。

结果深度

结果深度eDestDepth定义了结果图像ipDestImage的元素类型。一个更大的元素类型可以显示更多的信息。

过滤方向

过滤方向eFilterDirection定义了夏尔过滤器是在 X 方向（TCVN_FD_X）还是在 Y 方向（TCVN_FD_Y）应用于图像。

缩放

夏尔运算的结果乘以比例系数fScale。

Delta

在通过fScale缩放后，常量 deltafDelta被添加到夏尔运算的结果中。

边界推断

边界推断eBorderType的方法定义了如何将不存在的像素外推到图像边界之外，以计算图像边界的过滤值。更多细节，请参见ETcVnBorderInterpolationMethod [▶ 145]。

应用

```
hr := F_VN_ScharrFilterExp(
    ipSrcImage       :=  ipImageIn,
    ipDestImage      :=  ipImageRes,
    eDestDepth       :=  TCVN_ET_USINT,
    eFilterDirection:=  TCVN_FD_X,
    fScale           :=  1,
    fDelta           :=  0,
    eBorderType      :=  TCVN_BIM_DEFAULT,
    hrPrev           :=  hr,
);
```

样本

- 边缘检测过滤器 [▶ 2690]

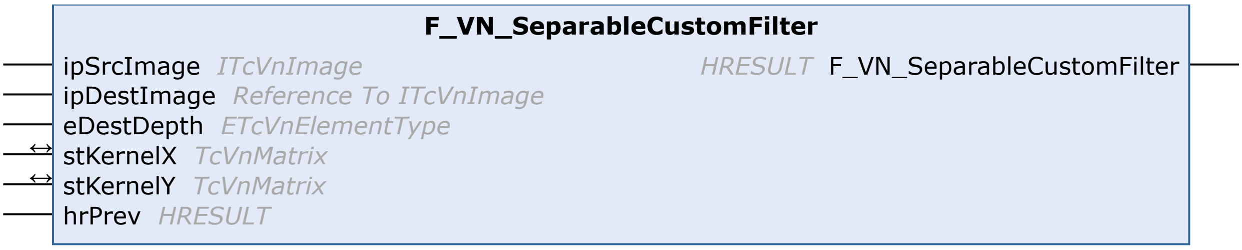
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.23 F_VN_SeparableCustomFilter



Apply a separable custom filter to the image.

Syntax

Definition:

```
FUNCTION F_VN_SeparableCustomFilter : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    eDestDepth  : ETcVnElementType;
END_VAR
VAR_IN_OUT
    stKernelX   : TcVnMatrix;
    stKernelY   : TcVnMatrix;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
eDestDepth	ETcVnElementType [▶ 178]	Destination image depth
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stKernelX	TcVnMatrix [▶ 211]	1D custom row-filter kernel with values of type REAL or LREAL
stKernelY	TcVnMatrix [▶ 211]	1D custom column-filter kernel with values of type REAL or LREAL

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_SeparableCustomFilter对输入图像应用可分离的二维过滤器。

算法

可分离的二维过滤器可以由两个一维过滤器组合而成。对一个图像应用两次一维过滤器比应用一次二维过滤器更有效率。例如，著名的索贝尔过滤器 [▶ 1276]可以由两个独立的一维过滤器构建如下：

$$stKernelX = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \qquad stKernelY = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$G = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

参数

输入图像

输入图像ipSrcImage可以有任何可用的格式。

结果图像

结果图像ipDestImage大小与输入图像ipSrcImage相同，但像素格式由结果深度eDestDepth决定。

X 和 Y 内核

两个一维过滤器内核stKernelX和stKernelY作为一个矩阵被转移。允许的矩阵元素类型为TCVN_ET_REAL和TCVN_ET_LREAL。相应类型的矩阵TcVnMatrix [▸ 211]可以通过函数F_VN_InitMatrixStruct [▸ 1484]进行创建。

专家参数

专家版F_VN_SeparableCustomFilterExp [▸ 1274]包含额外的参数。

应用

例如，可分离的二维过滤器的创建和应用如下所示：

```
hr := F_VN_SeparableCustomFilter(
    ipSrcImage   :=  ipImageIn,
    ipDestImage  :=  ipImageRes,
    eDestDepth   :=  TCVN_ET_USINT,
    stKernelX    :=  stKernelMatrixX,
    stKernelY    :=  stKernelMatrixY,
    hrPrev       :=  hr
);
```

样本

-

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.24 F_VN_SeparableCustomFilterExp

Apply a separable custom filter to the image. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_SeparableCustomFilterExp : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    eDestDepth  : ETcVnElementType;
END_VAR
VAR_IN_OUT
    stKernelX   : TcVnMatrix;
    stKernelY   : TcVnMatrix;
    aAnchor     : TcVnPoint;
END_VAR
VAR_INPUT
    fDelta      : LREAL;
    eBorderType : ETcVnBorderInterpolationMethod;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
eDestDepth	ETcVnElementType [▶ 178]	Destination image depth
fDelta	LREAL	Value added to each pixel after filtering
eBorderType	ETcVnBorderInterpolationMethod [▶ 145]	Image border handling
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stKernelX	TcVnMatrix [▶ 211]	1D custom row-filter kernel with values of type REAL or LREAL
stKernelY	TcVnMatrix [▶ 211]	1D custom column-filter kernel with values of type REAL or LREAL
aAnchor	TcVnPoint [▶ 139]	Anchor point of the kernel

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_SeparableCustomFilterExp是F_VN_SeparableCustomFilter [▶ 1272]的专家版本。它包含额外的参数。

参数

输入图像

输入图像ipSrcImage可以有任何可用的格式。

结果图像

结果图像ipDestImage大小与输入图像ipSrcImage相同，但像素格式由结果深度eDestDepth决定。

X 和 Y 内核

两个一维过滤器内核stKernelX和stKernelY作为一个矩阵被转移。允许的矩阵元素类型为TCVN_ET_REAL和TCVN_ET_LREAL。相应类型的矩阵TcVnMatrix [▶ 211]可以通过函数F_VN_InitMatrixStruct [▶ 1484]进行创建。

锚点

锚点aAnchor定义了在输入图像ipSrcImage的每个像素上移动过滤器内核的哪个点。

Delta

常量 deltafDelta被添加到过滤操作的结果中。

边界推断

边界推断eBorderType的方法定义了如何将不存在的像素外推到图像边界之外，以计算图像边界的过滤值。更多细节，请参见ETcVnBorderInterpolationMethod [▶ 145]。

应用

```
hr := F_VN_SeparableCustomFilterExp(
    ipSrcImage  :=  ipImageIn,
    ipDestImage :=  ipImageRes,
    eDestDepth  :=  TCVN_ET_USINT,
    stKernelX   :=  stKernelMatrixX,
    stKernelY   :=  stKernelMatrixY,
    aAnchor     :=  aAnchorPoint,
    fDelta      :=  0,
    eBorderType :=  TCVN_BIM_DEFAULT,
    hrPrev      :=  hr
);
```

样本

-

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.25 F_VN_SobelFilter

F_VN_SobelFilter

ipSrcImage *ITcVnImage* — *HRESULT* F_VN_SobelFilter
ipDestImage *Reference To ITcVnImage*
eDestDepth *ETcVnElementType*
nXOrder *UDINT*
nYOrder *UDINT*
hrPrev *HRESULT*

Calculates the first, second, or mixed image derivatives using an extended Sobel filter.

Syntax

Definition:

```
FUNCTION F_VN_SobelFilter : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    eDestDepth  : ETcVnElementType;
    nXOrder     : UDINT;
    nYOrder     : UDINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
eDestDepth	ETcVnElementType [▸ 178]	Destination image depth
nXOrder	UDINT	Order of the x-derivative (0, 1, 2)
nYOrder	UDINT	Order of the y-derivative (0, 1, 2)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_SobelFilter对输的图像应用索贝尔过滤器。索贝尔过滤器用于边缘检测。在结果图像中，输入图像中的边缘以深色背景上的亮线突出显示。

算法

扩展的索贝尔过滤器可用于通过近似相邻像素强度转换的第 n 次导数来检测边缘。由于导数也有负值，建议对结果图像使用有符号的元素类型（例如TCVN_ET_INT）。

过滤器的内核由函数的参数决定。通常情况下，调用该函数时，nXOrder = 1，nYOrder = 0，nKernelSize = 3 或 nXOrder = 0，nYOrder = 1，nKernelSize = 3。这些参数导致了内核 G_x和G_y。

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

对于这些内核，该过滤器将高斯平滑与第一导数的计算结合起来，因此，其结果具有相当强的抗噪性。G_x强调垂直边缘，G_y强调水平边缘。

参数

原始图像

原始图像ipSrcImage可以有任何格式。

结果图像

结果图像ipDestImage返回过滤结果，且其格式与原始图像ipSrcImage相同。

结果深度

结果深度eDestDepth定义了结果图像ipDestImage的元素类型。一个更大的元素类型可以显示更多的信息。

导数阶数

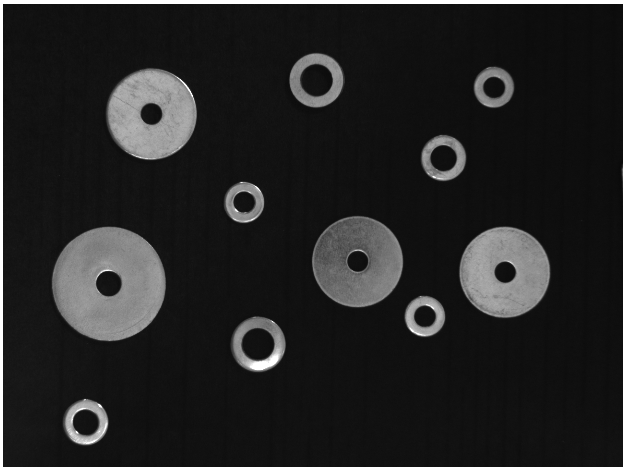
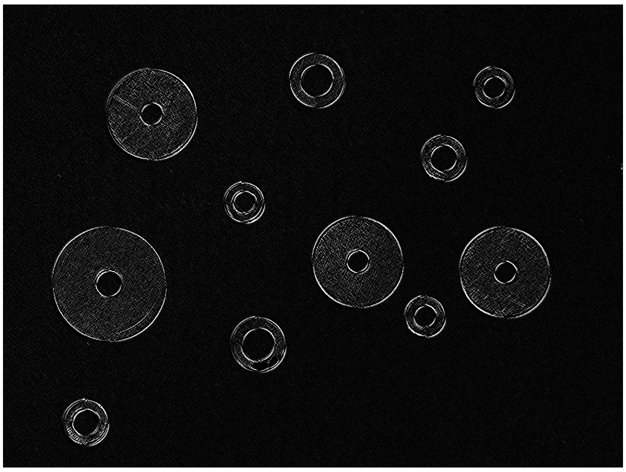
导数阶数nXOrder和nYOrder定义在 X 和 Y 方向上哪个导数用于创建过滤器。

专家参数

关于更多参数，可查看专家版F_VN_SobelFilterExp [▶ 1278]。

应用

```
hr := F_VN_SobelFilter(
    ipSrcImage  :=  ipImageIn,
    ipDestImage :=  ipImageRes,
    eDestDepth  :=  TCVN_ET_USINT,
    nXOrder     :=  1,
    nYOrder     :=  1,
    hrPrev      :=  hr,
);
```

原始图像	结果图像
	

样本

- 边缘检测过滤器 [▶ 2690]

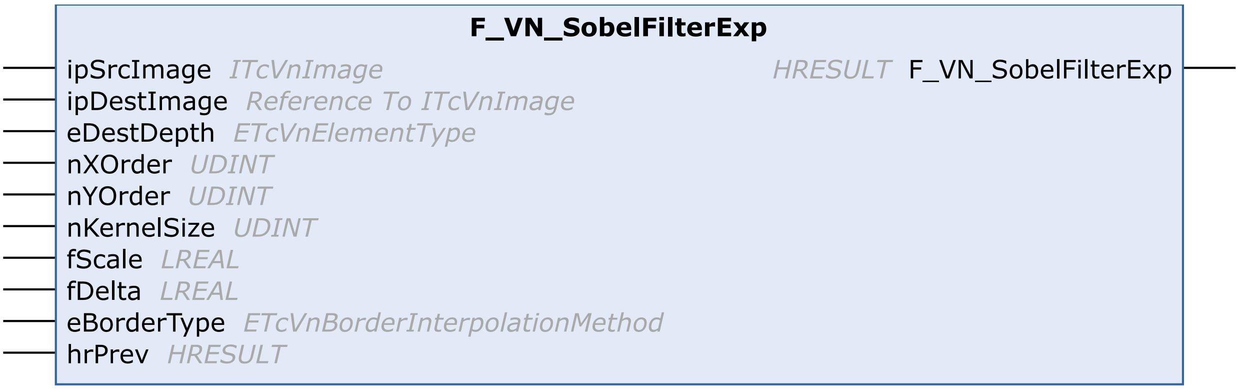
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.26 F_VN_SobelFilterExp



Calculates the first, second, third, or mixed image derivatives using an extended Sobel filter. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_SobelFilterExp : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    eDestDepth  : ETcVnElementType;
    nXOrder     : UDINT;
    nYOrder     : UDINT;
    nKernelSize : UDINT;
    fScale      : LREAL;
    fDelta      : LREAL;
    eBorderType : ETcVnBorderInterpolationMethod;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
eDestDepth	ETcVnElementType [▸ 178]	Destination image depth
nXOrder	UDINT	Order of the x-derivative (must be < nKernelSize)
nYOrder	UDINT	Order of the y-derivative (must be < nKernelSize)
nKernelSize	UDINT	Size of the extended Sobel kernel (3, 5, 7, ..., 31)
fScale	LREAL	Scale factor for the computed derivative values
fDelta	LREAL	Delta value that is added to the results prior to storing them in dest
eBorderType	ETcVnBorderInterpolationMethod [▸ 145]	Image border handling
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_SobelFilterExp是F_VN_SobelFilter [▶ 1276]的专家版本。它包含额外的参数。

参数

原始图像

原始图像ipSrcImage可以有任何格式。

结果图像

结果图像ipDestImage返回过滤结果，且其格式与原始图像ipSrcImage相同。

结果深度

结果深度eDestDepth定义了结果图像ipDestImage的元素类型。一个更大的元素类型可以显示更多的信息。

导数阶数

导数阶数nXOrder和nYOrder定义在 X 和 Y 方向上哪个导数用于创建过滤器。

内核大小

索贝尔内核的大小 nKernelSize 决定了影响一个像素结果值的区域有多大。

缩放

索贝尔运算的结果乘以缩放系数fScale。

Delta

在通过fScale缩放后，常量 deltafDelta被添加到索贝尔运算的结果中。

边界推断

边界推断eBorderType的方法定义了如何将不存在的像素外推到图像边界之外，以计算图像边界的过滤值。更多细节，请参见ETcVnBorderInterpolationMethod [▶ 145]。

应用

```
hr := F_VN_SobelFilterExp(
    ipSrcImage   :=  ipImageIn,
    ipDestImage  :=  ipImageRes,
    eDestDepth   :=  TCVN_ET_USINT,
    nXOrder      :=  1,
    nYOrder      :=  1,
    nKernelSize  :=  3,
    fScale       :=  1,
    fDelta       :=  0,
    eBorderType  :=  TCVN_BIM_DEFAULT,
    hrPrev       :=  hr,
);
```

样本

- 边缘检测过滤器 [▶ 2690]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.27 F_VN_VarianceFilter

F_VN_VarianceFilter

ipSrcImage *ITcVnImage* — *HRESULT* F_VN_VarianceFilter
ipDestImage *Reference To ITcVnImage*
nKernelSize *UDINT*
hrPrev *HRESULT*

Calculates the local variance of an image.

Syntax

Definition:

```
FUNCTION F_VN_VarianceFilter : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    nKernelSize : UDINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image. Only TCVN_ET_USINT Type is supported.
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image of type TCVN_ET_UINT (An appropriate destination image will be created if required.)
nKernelSize	UDINT	Size of the kernel
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.16.28 F_VN_VarianceFilterExp

F_VN_VarianceFilterExp

ipSrcImage *ITcVnImage* — *HRESULT* F_VN_VarianceFilterExp
ipDestImage *Reference To ITcVnImage*
nKernelSize *UDINT*
fScale *LREAL*
eBorderType *ETcVnBorderInterpolationMethod*
hrPrev *HRESULT*

Calculates the local variance of an image. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_VarianceFilterExp : HRESULT
VAR_INPUT
    ipSrcImage   : ITcVnImage;
    ipDestImage  : Reference To ITcVnImage;
    nKernelSize  : UDINT;
    fScale       : LREAL;
    eBorderType  : ETcVnBorderInterpolationMethod;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image. Only TCVN_ET_USINT Type is supported.
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image of type TCVN_ET_UINT (An appropriate destination image will be created if required.)
nKernelSize	UDINT	Size of the kernel
fScale	LREAL	Scale factor for the computed variance values
eBorderType	ETcVnBorderInterpolationMethod [▶ 145]	Image border handling
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.17 Image Segmentation

该组包含用于分割图像的函数。

函数

分割

- F_VN_AdaptiveThreshold(Exp) [▶ 1283]
- F_VN_CheckColorRange [▶ 1285]
- F_VN_Threshold [▶ 1287]
- F_VN_WatershedSegmentationExp [▶ 1289]

样本

图像分割 [▶ 2693]

还请参阅有关此

F_VN_DoubleThreshold [▸ 1286]

6.1.4.17.1 F_VN_AdaptiveThreshold

F_VN_AdaptiveThreshold	
ipSrcImage *ITcVnImage*	*HRESULT* F_VN_AdaptiveThreshold
ipDestImage *Reference To ITcVnImage*	
fMaxValue *LREAL*	
hrPrev *HRESULT*	

Apply an adaptive threshold to a gray level image.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_AdaptiveThreshold : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    fMaxValue   : LREAL;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (1 channel, elements of type USINT)
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
fMaxValue	LREAL	Value assigned to pixels for which the threshold condition is true
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_AdaptiveThreshold对输入图像应用一个自适应阈值。

参数

输入图像

输入图像ipSrcImage必须是位深为 8 位的单通道灰度图像（USINT）。

结果图像

二进制结果图像ipDestImage的格式与输入图像相同。像素的值是0或目标值fMaxValue。

目标值

目标值fMaxValue定义了应该为满足阈值标准的像素赋哪个值。如果结果图像只由黑白像素组成，那么必须在这里设置输入图像的元素类型的最大值。这对应于(2^d)-1，其中 d 代表元素类型的位深；例如，(2^8)-1=255，8 位图像。

专家参数

专家版F_VN_AdaptiveThresholdExp [▶ 1283]包含额外的参数。

应用

```
hr := F_VN_AdaptiveThreshold(
    ipSrcImage      :=  ipImageIn,
    ipDestImage     :=  ipImageRes,
    fMaxValue       :=  255, // max value of 8-bit image: (2^8)-1=255
    hrPrev          :=  hr
);
```

相关函数

- F_VN_Threshold [▶ 1287]用于在阈值的基础上进行二进制分割
- F_VN_AdaptiveThreshold [▶ 1282]用于二进制、自适应分割
- F_VN_CheckColorRange [▶ 1285]用于在色谱的基础上进行二进制分割
- F_VN_ReferenceColorSimilarity [▶ 1210]用于基于颜色模型的二进制分割

Required License

TC3 Vision Base

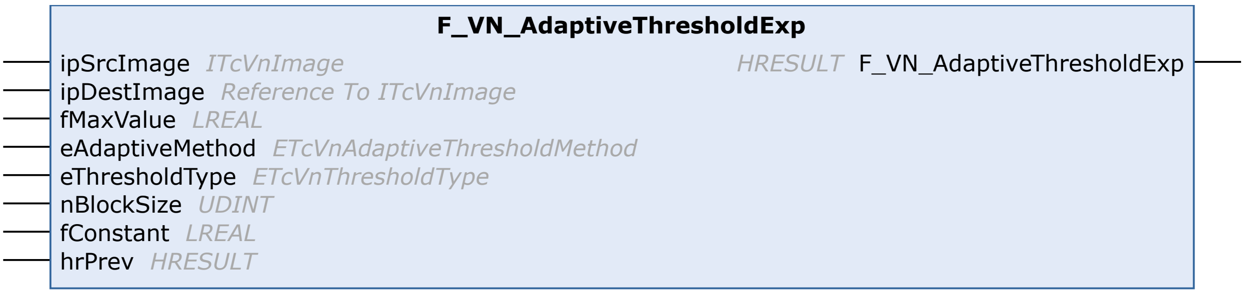
System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

还请参阅有关此

F_VN_DoubleThreshold [▶ 1286]

6.1.4.17.2 F_VN_AdaptiveThresholdExp



Apply an adaptive threshold to a gray level image. (expert function)
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_AdaptiveThresholdExp : HRESULT
VAR_INPUT
    ipSrcImage      : ITcVnImage;
    ipDestImage     : Reference To ITcVnImage;
    fMaxValue       : LREAL;
    eAdaptiveMethod : ETcVnAdaptiveThresholdMethod;
    eThresholdType  : ETcVnThresholdType;
    nBlockSize      : UDINT;
    fConstant       : LREAL;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel, elements of type USINT)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
fMaxValue	LREAL	Value assigned to pixels for which the threshold condition is true
eAdaptiveMethod	ETcVnAdaptiveThresholdMethod [▶ 142]	Adaptive threshold method to be applied (MEAN: threshold value will be calculated as a mean of the nBlockSize x nBlockSize neighborhood of (x,y) minus fConstant. GAUSSIAN: threshold value is a weighted sum (cross-correlation with a Gaussian window) of the nBlockSize x nBlockSize neighborhood of (x,y) minus fConstant.
eThresholdType	ETcVnThresholdType [▶ 200]	Threshold type to be applied (only BINARY and BINARY_INV are supported)
nBlockSize	UDINT	Size of the pixel neighborhood to calculate the local threshold (3, 5, 7, ...)
fConstant	LREAL	Constant that is subtracted from the weighted mean of the pixel neighborhood, which leads to the local threshold
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_AdaptiveThresholdExp是F_VN_AdaptiveThreshold [▶ 1282]的专家版本。它包含额外的参数。

参数

输入图像

输入图像ipSrcImage必须是位深为 8 位的单通道灰度图像（USINT）。

结果图像

二进制结果图像ipDestImage的格式与输入图像相同。像素的值是0或目标值fMaxValue。

目标值

目标值fMaxValue定义了应该为满足阈值标准的像素赋哪个值。如果结果图像只由黑白像素组成，那么必须在这里设置输入图像的元素类型的最大值。这对应于(2^d)-1，其中 d 代表元素类型的位深；例如，(2^8)-1=255，8 位图像。

阈值法

阈值方法 eAdaptiveMethod 定义了自适应阈值是作为平均数还是作为高斯加权和进行计算。在枚举ETcVnAdaptiveThresholdMethod [▶ 142]中，有两个值TCVN_ATM_MEAN和TCVN_ATM_GAUSSIAN可供使用。

阈值反转

参数 eThresholdType 指定结果图像是否应该被反转。因此，目前只支持来自枚举ETcVnThresholdType [▶ 200]的两个值TCVN_TT_BINARY和TCVN_TT_BINARY_INV。

阈值邻域大小

参数nBlockSize定义了计算局部阈值时考虑的邻域的大小。参数必须是一个奇数（3、5、7 等）。

阈值调整

从每个局部平均值中减去常量值fConstant。其结果是用于计算结果图像的局部阈值。

应用

```
hr := F_VN_AdaptiveThreshold(
    ipSrcImage      :=  ipImageIn,
    ipDestImage     :=  ipImageRes,
    fMaxValue       :=  255, // max value of 8-bit image: (2^8)-1=255
    eAdaptiveMethod :=  TCVN_ATM_MEAN,
    eThresholdType  :=  TCVN_TT_BINARY_INV,
    nBlockSize      :=  5,
    fConstant       :=  10,
    hrPrev          :=  hr
);
```

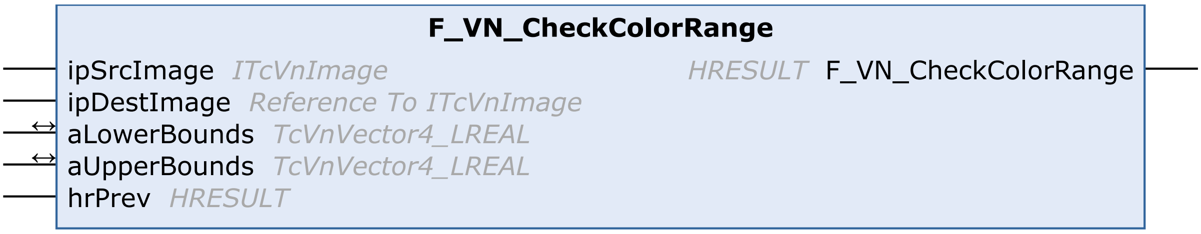
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.17.3 F_VN_CheckColorRange



Check if the pixel values of an image lie in a given range. The destination image has the same size as the source image but only one channel with a pixel element size of 8 bit. Its elements are set to 255 if the corresponding pixels of the source image are in the checked range and set to 0 otherwise.

Syntax

Definition:

```
FUNCTION F_VN_CheckColorRange : HRESULT
VAR_INPUT
    ipSrcImage   : ITcVnImage;
    ipDestImage  : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    aLowerBounds : TcVnVector4_LREAL;
    aUpperBounds : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aLowerBounds	TcVnVector4_LREAL [▶ 141]	Channel-wise lower bounds (Unused channels are ignored.)
aUpperBounds	TcVnVector4_LREAL [▶ 141]	Channel-wise upper bounds (Unused channels are ignored.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.17.4 F_VN_DoubleThreshold

Apply a double threshold (also called hysteresis threshold) using morphological reconstruction. Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_DoubleThreshold : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipDestImage    : Reference To ITcVnImage;
    fLowThreshold  : LREAL;
    fHighThreshold : LREAL;
    fMaxValue      : LREAL;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT, UINT, INT, REAL, or LREAL, 1 channel)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (An appropriate destination image will be created if required.)
fLowThreshold	LREAL	Low threshold determinining the shape of found regions
fHighThreshold	LREAL	High threshold selecting found regions (Each region must at least contain a single pixel if the high threshold is applied.)
fMaxValue	LREAL	Maximum pixel value
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

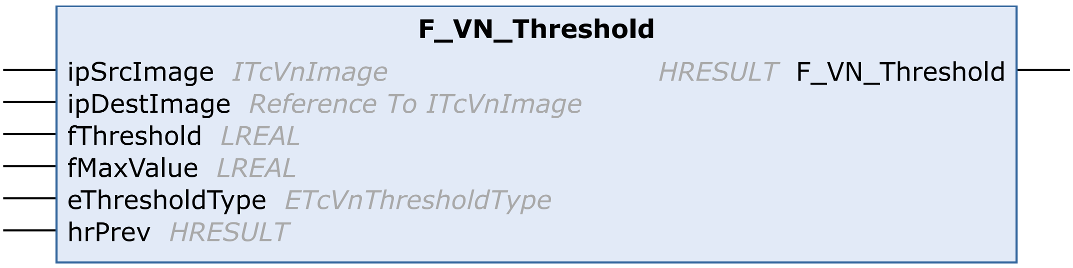
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.17.5 F_VN_Threshold



Apply a fixed threshold or a dynamic threshold according to Otsu.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_Threshold : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipDestImage    : Reference To ITcVnImage;
    fThreshold     : LREAL;
    fMaxValue      : LREAL;
    eThresholdType : ETcVnThresholdType;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (For Otsu and Triangle threshold types, only 1 channel of TCVN_ET_USINT is supported)
ipDestImage	Reference To ITcVnImage [▸ 390]	Destination image (An appropriate destination image will be created if required.)
fThreshold	LREAL	Fixed threshold (unused if dynamic thresholding is selected)
fMaxValue	LREAL	Maximum pixel value
eThresholdType	ETcVnThresholdType [▸ 200]	Threshold type to be applied
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_Threshold根据阈值对输入图像进行分割。根据这个阈值和使用的算法，二进制结果图像的每个像素都被分配了值0和一个指定的目标值。

算法

阈值将单色图像转化为二进制图像，从而形成分段。良好的分割是许多后续方法（如轮廓搜索）的先决条件。TwinCAT Vision 提供不同类型的阈值，适用于不同的应用。

固定的阈值

通过将数值大于 fThreshold 的像素设置为白色（255）并将其他像素设置为黑色（0），固定阈值创建二进制图像。fThreshold 是相应函数的一个参数。为了找到一个良好的数值，你很可能会通过不同的尝试来接近这个数值。

动态阈值

根据最大类间方差法，动态阈值对双模图像的效果非常好。双模图像的直方图有两个主要的峰值，而良好的阈值很可能正好在中间。对于噪声图像，可以使用高斯过滤器将其转向双模图像。

自适应阈值

自适应阈值对整个图像使用可变的阈值。见F_VN_AdaptiveThresholdExp [▸ 1283]。

参数

输入图像

输入图像ipSrcImage可以有一个或三个通道。

结果图像

结果图像ipDestImage返回分割的二进制图像。

阈值

阈值fThreshold定义了分割条件。它必须根据输入图像的属性来调整。

目标值

目标值fMaxValue定义了应该为满足阈值标准的像素赋哪个值。如果结果图像只由黑白像素组成，那么必须在这里设置输入图像的元素类型的最大值。这对应于(2^d)-1，其中d代表元素类型的位深；例如，(2^8)-1=255，8 位图像。

阈值类型

阈值类型eThresholdType定义了阈值fThreshold应用于输入图像的算法。枚举ETcVnThresholdType [▶ 200]的所有值都受到支持。

应用

例如，在数值范围中心的 8 位灰度图像的二进制分割如下所示：

```
hr := F_VN_Threshold(
    ipSrcImage       :=  ipImageIn,
    ipDestImage      :=  ipImageRes,
    fThreshold       :=  128, // half value range: (2^8)/2=128
    fMaxValue        :=  255, // max value of 8-bit image: (2^8)-1=255
    eThresholdType   :=  TCVN_TT_BINARY,
    hrPrev           :=  hr
);
```

相关函数

- F_VN_Threshold [▶ 1287]用于在阈值的基础上进行二进制分割
- F_VN_AdaptiveThreshold [▶ 1282]用于二进制、自适应分割
- F_VN_CheckColorRange [▶ 1285]用于在色谱的基础上进行二进制分割
- F_VN_ReferenceColorSimilarity [▶ 1210]用于基于颜色模型的二进制分割

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

还请参阅有关此

F_VN_DoubleThreshold [▶ 1286]

6.1.4.17.6 F_VN_WatershedSegmentationExp

Apply a marker-based watershed segmentation. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_WatershedSegmentationExp : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipDestImage : Reference To ITcVnImage;
    ipMarkers   : ITcVnImage;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT, 3 channels)
ipDestImage	Reference To ITcVnImage [▶ 390]	Destination image (DINT, 1 channel. An appropriate image will be created if required.)
ipMarkers	ITcVnImage [▶ 390]	Marker image (DINT, 1 channel)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18 Keypoint Features

这一组包含寻找和匹配图像中的独特特征的功能。

> **注意**
>
> **计算时间长**
>
> 许多寻找关键点特征的功能需要很长的计算时间。因此，请注意设定的周期时间，且必要时使用看门狗 [▶ 127]，以保证实时行为。

函数

关键点

- F_VN_KeyPointsAGAST(Exp) [▶ 1308]
- F_VN_KeyPointsFAST(Exp) [▶ 1317]
- F_VN_KeyPointsGFTT(Exp) [▶ 1319]
- F_VN_KeyPointsMSER(Exp) [▶ 1321]
- F_VN_KeyPointsSB(Exp) [▶ 1323]

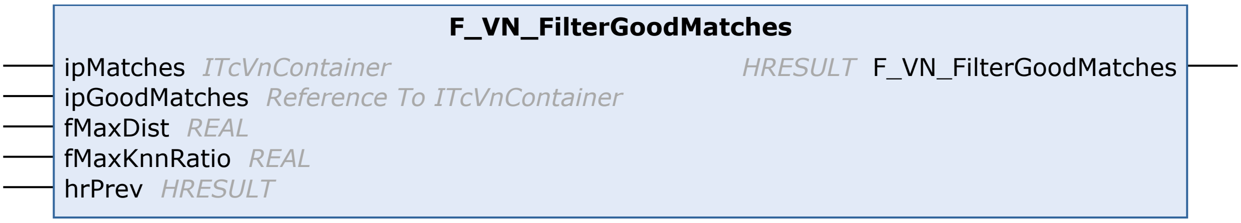
关键点和描述符

- F_VN_KeyPointsAndDescriptorsAKAZE(Exp) [▶ 1310]
- F_VN_KeyPointsAndDescriptorsBRISK(Exp) [▶ 1311]
- F_VN_KeyPointsAndDescriptorsaKAZE(Exp) [▶ 1313]
- F_VN_KeyPointsAndDescriptorsORB(Exp) [▶ 1315]

描述符匹配

- F_VN_FilterGoodMatches [▶ 1291]
- F_VN_GetMatchCoordinates [▶ 1307]
- F_VN_MatchDescriptorsBF(Exp) [▶ 1325]
- F_VN_MatchDescriptorsFlannLsh(Exp) [▶ 1327]

6.1.4.18.1 F_VN_FilterGoodMatches



Filter the descriptor matches and return only good ones.

Syntax

Definition:

```
FUNCTION F_VN_FilterGoodMatches : HRESULT
VAR_INPUT
    ipMatches      : ITcVnContainer;
    ipGoodMatches : Reference To ITcVnContainer;
    fMaxDist       : REAL;
    fMaxKnnRatio   : REAL;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMatches	ITcVnContainer [▸ 349]	Container with descriptor matches (ContainerType_Vector_TcVnDMatch or ContainerType_Vector_Vector_TcVnDMatch)
ipGoodMatches	Reference To ITcVnContainer [▸ 349]	Returns a container with good matches (ContainerType_Vector_TcVnDMatch)
fMaxDist	REAL	Maximum allowed descriptor distance (-1 disables this filter criterion)
fMaxKnnRatio	REAL	Maximum allowed distance ratio between first and second best match ([0..1], used for knn match results only, -1 disables this filter criterion)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

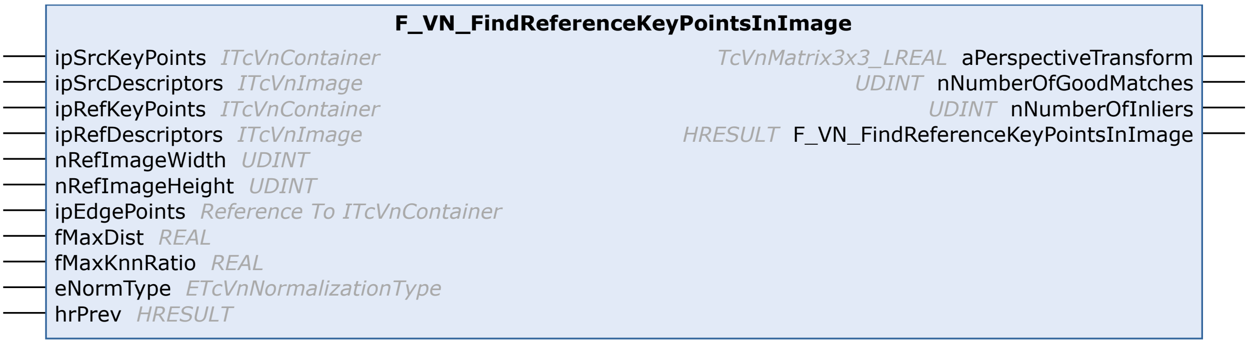
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.2 F_VN_FindReferenceKeyPointsInImage



Searches a reference image, represented by its keypoints and corresponding descriptors, in a given source image.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_FindReferenceKeyPointsInImage : HRESULT
VAR_INPUT
    ipSrcKeyPoints         : ITcVnContainer;
    ipSrcDescriptors       : ITcVnImage;
    ipRefKeyPoints         : ITcVnContainer;
    ipRefDescriptors       : ITcVnImage;
    nRefImageWidth         : UDINT;
    nRefImageHeight        : UDINT;
    ipEdgePoints           : Reference To ITcVnContainer;
    fMaxDist               : REAL;
    fMaxKnnRatio           : REAL;
    eNormType              : ETcVnNormalizationType;
    hrPrev                 : HRESULT;
END_VAR
VAR_OUTPUT
    aPerspectiveTransform  : TcVnMatrix3x3_LREAL;
    nNumberOfGoodMatches   : UDINT;
    nNumberOfInliers       : UDINT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcKeyPoints	ITcVnContainer [▸ 349]	KeyPoints of the source image, e.g. (ContainerType_Vector_TcVnKeyPoint)
ipSrcDescriptors	ITcVnImage [▸ 390]	Descriptors of the source image KeyPoints
ipRefKeyPoints	ITcVnContainer [▸ 349]	KeyPoints of the reference image, e.g. (ContainerType_Vector_TcVnKeyPoint)
ipRefDescriptors	ITcVnImage [▸ 390]	Descriptors of the reference image KeyPoints
nRefImageWidth	UDINT	Width of the reference image in pixels
nRefImageHeight	UDINT	Height of the reference image in pixels
ipEdgePoints	Reference To ITcVnContainer [▸ 349]	Returns a container with the 4 edge points of the reference image transformed into the coordinates of the source image (ContainerType_Vector_TcVnPoint2_REAL)
fMaxDist	REAL	Maximum allowed descriptor distance (-1 disables this filter criterion)
fMaxKnnRatio	REAL	Maximum allowed distance ratio between first and second best match ([0..1], -1 disables this filter criterion)
eNormType	ETcVnNormalizationType [▸ 189]	Normalization type used for descriptor matching (HAMMING recommended for AKAZE, ORB, and BRISK. HAMMING2 is recommended for ORB if the ORB nBriefPoints parameter is 3 or 4.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Outputs

Name	Type	Description
aPerspectiveTransform	TcVnMatrix3x3_LREAL [▸ 141]	Returns the perspective transformation matrix, which transforms the reference points to the source points
nNumberOfGoodMatches	UDINT	Return the number of good matches
nNumberOfInliers	UDINT	Return the number of inlier matches

Return value

HRESULT [▸ 122]

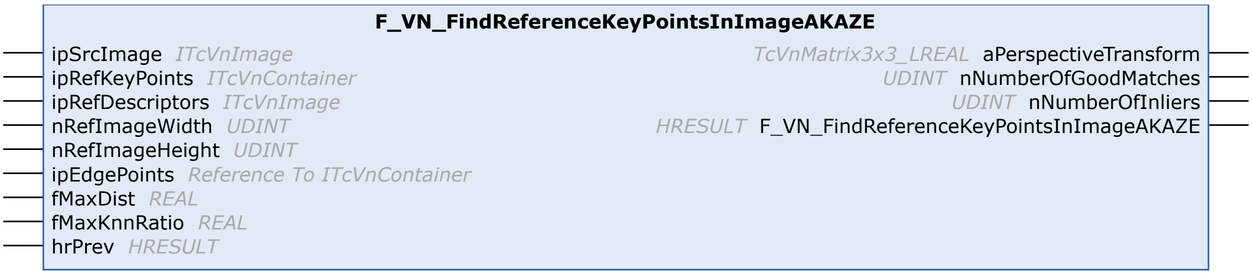
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.3 F_VN_FindReferenceKeyPointsInImageAKAZE



Searches a reference image, represented by its AKAZE keypoints and corresponding descriptors, in a given source image (uses the default TcVnParamsAKAZE parameters).
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_FindReferenceKeyPointsInImageAKAZE : HRESULT
VAR_INPUT
    ipSrcImage              : ITcVnImage;
    ipRefKeyPoints          : ITcVnContainer;
    ipRefDescriptors        : ITcVnImage;
    nRefImageWidth          : UDINT;
    nRefImageHeight         : UDINT;
    ipEdgePoints            : Reference To ITcVnContainer;
    fMaxDist                : REAL;
    fMaxKnnRatio            : REAL;
    hrPrev                  : HRESULT;
END_VAR
VAR_OUTPUT
    aPerspectiveTransform : TcVnMatrix3x3_LREAL;
    nNumberOfGoodMatches  : UDINT;
    nNumberOfInliers      : UDINT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipRefKeyPoints	ITcVnContainer [▸ 349]	KeyPoints of the reference image, e.g. computed with F_VN_KeyPointsAndDescriptorsAKAZE (ContainerType_Vector_TcVnKeyPoint)
ipRefDescriptors	ITcVnImage [▸ 390]	Descriptors of the reference image KeyPoints, e.g. computed with F_VN_KeyPointsAndDescriptorsAKAZE
nRefImageWidth	UDINT	Width of the reference image in pixels
nRefImageHeight	UDINT	Height of the reference image in pixels
ipEdgePoints	Reference To ITcVnContainer [▸ 349]	Returns a container with the 4 edge points of the reference image transformed into ipSrcImage coordinates (ContainerType_Vector_TcVnPoint2_REAL)
fMaxDist	REAL	Maximum allowed descriptor distance (-1 disables this filter criterion)
fMaxKnnRatio	REAL	Maximum allowed distance ratio between first and second best match ([0..1], -1 disables this filter criterion)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Outputs

Name	Type	Description
aPerspectiveTransform	TcVnMatrix3x3_LREAL [▶ 141]	Returns the perspective transformation matrix, which transforms the reference points to the source points
nNumberOfGoodMatches	UDINT	Return the number of good matches
nNumberOfInliers	UDINT	Return the number of inlier matches

Return value

HRESULT [▶ 122]

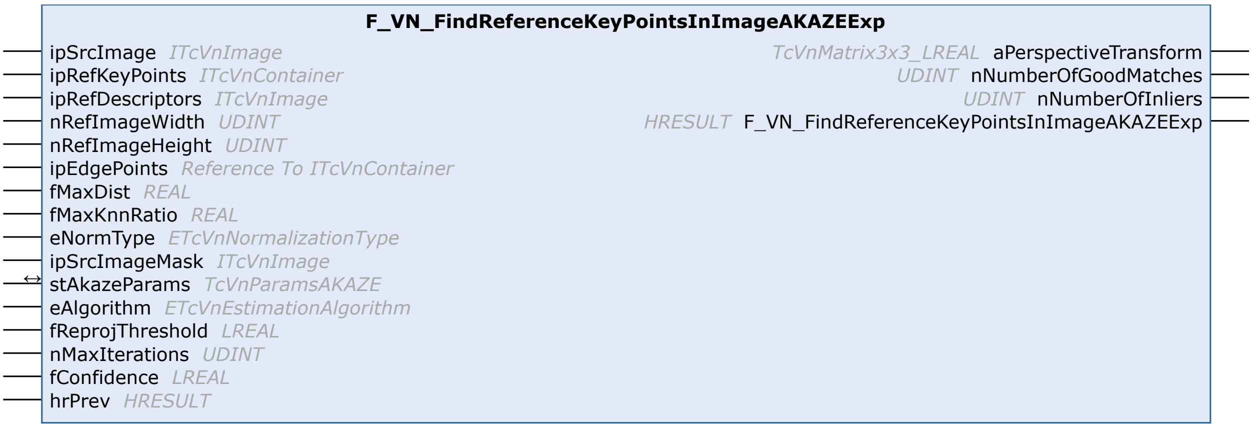
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.4 F_VN_FindReferenceKeyPointsInImageAKAZEExp



Searches a reference image, represented by its AKAZE keypoints and corresponding descriptors, in a given source image. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_FindReferenceKeyPointsInImageAKAZEExp : HRESULT
VAR_INPUT
    ipSrcImage              : ITcVnImage;
    ipRefKeyPoints          : ITcVnContainer;
    ipRefDescriptors        : ITcVnImage;
    nRefImageWidth          : UDINT;
    nRefImageHeight         : UDINT;
    ipEdgePoints            : Reference To ITcVnContainer;
    fMaxDist                : REAL;
    fMaxKnnRatio            : REAL;
    eNormType               : ETcVnNormalizationType;
    ipSrcImageMask          : ITcVnImage;
END_VAR
VAR_IN_OUT
    stAkazeParams           : TcVnParamsAKAZE;
END_VAR
```

```
VAR_INPUT
    eAlgorithm            : ETcVnEstimationAlgorithm;
    fReprojThreshold      : LREAL;
    nMaxIterations        : UDINT;
    fConfidence           : LREAL;
    hrPrev                : HRESULT;
END_VAR
VAR_OUTPUT
    aPerspectiveTransform : TcVnMatrix3x3_LREAL;
    nNumberOfGoodMatches  : UDINT;
    nNumberOfInliers      : UDINT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipRefKeyPoints	ITcVnContainer [▶ 349]	KeyPoints of the reference image, e.g. computed with F_VN_KeyPointsAndDescriptorsAKAZE (ContainerType_Vector_TcVnKeyPoint)
ipRefDescriptors	ITcVnImage [▶ 390]	Descriptors of the reference image KeyPoints, e.g. computed with F_VN_KeyPointsAndDescriptorsAKAZE
nRefImageWidth	UDINT	Width of the reference image in pixels
nRefImageHeight	UDINT	Height of the reference image in pixels
ipEdgePoints	Reference To ITcVnContainer [▶ 349]	Returns a container with the 4 edge points of the reference image transformed into ipSrcImage coordinates (ContainerType_Vector_TcVnPoint2_REAL)
fMaxDist	REAL	Maximum allowed descriptor distance (-1 disables this filter criterion)
fMaxKnnRatio	REAL	Maximum allowed distance ratio between first and second best match ([0..1], -1 disables this filter criterion)
eNormType	ETcVnNormalizationType [▶ 189]	Normalization type used for descriptor matching
ipSrcImageMask	ITcVnImage [▶ 390]	Mask to specify, where to look for keypoints in ipSrcImage (set to 0 if no mask required)
eAlgorithm	ETcVnEstimationAlgorithm [▶ 179]	Estimation algorithm used for computing the perspective transformation between the two point sets
fReprojThreshold	LREAL	Maximum allowed reprojection error to treat a point pair as an inlier (only used if eAlgorithm is RANSAC, RHO)
nMaxIterations	UDINT	Maximum number of RANSAC iterations
fConfidence	LREAL	Confidence (0..1)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stAkazeParams	TcVnParamsAKAZE [▶ 213]	Parameters to configure the keypoint and descriptor computation for ipSrcImage (resulting descriptors must be compatible to ipRefDescriptors!)

Outputs

Name	Type	Description
aPerspectiveTransform	TcVnMatrix3x3_LREAL [▸ 141]	Returns the perspective transformation matrix, which transforms the reference points to the source points
nNumberOfGoodMatches	UDINT	Return the number of good matches
nNumberOfInliers	UDINT	Return the number of inlier matches

Return value

HRESULT [▸ 122]

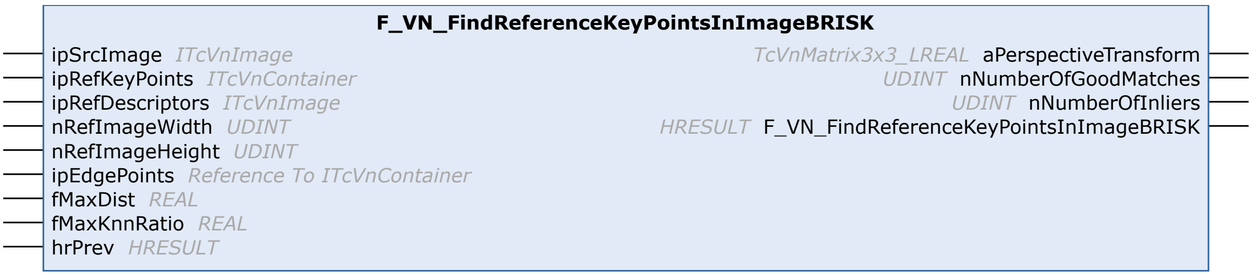
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.5 F_VN_FindReferenceKeyPointsInImageBRISK



Searches a reference image, represented by its BRISK keypoints and corresponding descriptors, in a given source image (uses the default TcVnParamsBRISK parameters).
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_FindReferenceKeyPointsInImageBRISK : HRESULT
VAR_INPUT
    ipSrcImage             : ITcVnImage;
    ipRefKeyPoints         : ITcVnContainer;
    ipRefDescriptors       : ITcVnImage;
    nRefImageWidth         : UDINT;
    nRefImageHeight        : UDINT;
    ipEdgePoints           : Reference To ITcVnContainer;
    fMaxDist               : REAL;
    fMaxKnnRatio           : REAL;
    hrPrev                 : HRESULT;
END_VAR
VAR_OUTPUT
    aPerspectiveTransform : TcVnMatrix3x3_LREAL;
    nNumberOfGoodMatches  : UDINT;
    nNumberOfInliers      : UDINT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipRefKeyPoints	ITcVnContainer [▶ 349]	KeyPoints of the reference image, e.g. computed with F_VN_KeyPointsAndDescriptorsBRISK (ContainerType_Vector_TcVnKeyPoint)
ipRefDescriptors	ITcVnImage [▶ 390]	Descriptors of the reference image KeyPoints, e.g. computed with F_VN_KeyPointsAndDescriptorsBRISK
nRefImageWidth	UDINT	Width of the reference image in pixels
nRefImageHeight	UDINT	Height of the reference image in pixels
ipEdgePoints	Reference To ITcVnContainer [▶ 349]	Returns a container with the 4 edge points of the reference image transformed into ipSrcImage coordinates (ContainerType_Vector_TcVnPoint2_REAL)
fMaxDist	REAL	Maximum allowed descriptor distance (-1 disables this filter criterion)
fMaxKnnRatio	REAL	Maximum allowed distance ratio between first and second best match ([0..1], -1 disables this filter criterion)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Outputs

Name	Type	Description
aPerspectiveTransform	TcVnMatrix3x3_LREAL [▶ 141]	Returns the perspective transformation matrix, which transforms the reference points to the source points
nNumberOfGoodMatches	UDINT	Return the number of good matches
nNumberOfInliers	UDINT	Return the number of inlier matches

Return value

HRESULT [▶ 122]

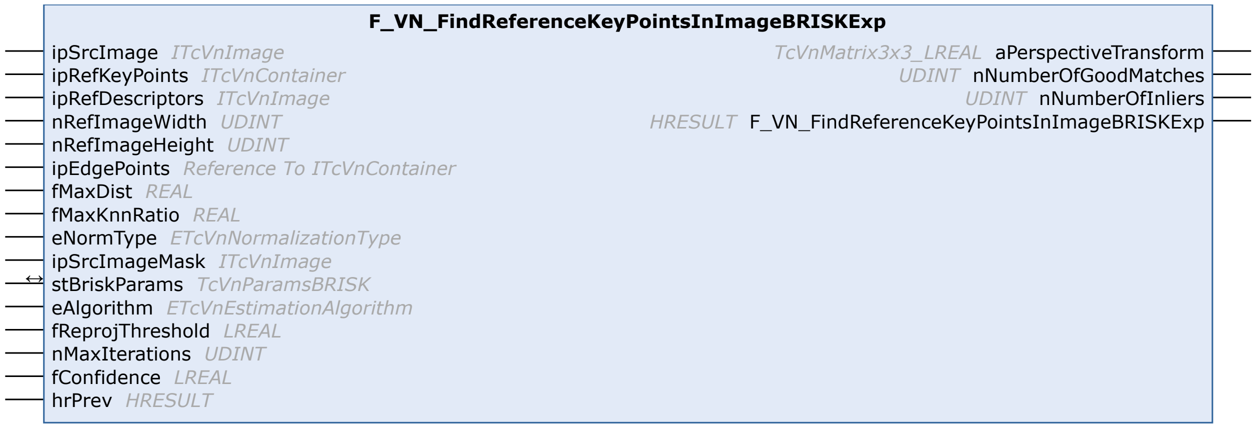
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.6 F_VN_FindReferenceKeyPointsInImageBRISKExp



Searches a reference image, represented by its BRISK keypoints and corresponding descriptors, in a given source image. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_FindReferenceKeyPointsInImageBRISKExp : HRESULT
VAR_INPUT
    ipSrcImage            : ITcVnImage;
    ipRefKeyPoints        : ITcVnContainer;
    ipRefDescriptors      : ITcVnImage;
    nRefImageWidth        : UDINT;
    nRefImageHeight       : UDINT;
    ipEdgePoints          : Reference To ITcVnContainer;
    fMaxDist              : REAL;
    fMaxKnnRatio          : REAL;
    eNormType             : ETcVnNormalizationType;
    ipSrcImageMask        : ITcVnImage;
END_VAR
VAR_IN_OUT
    stBriskParams         : TcVnParamsBRISK;
END_VAR
VAR_INPUT
    eAlgorithm            : ETcVnEstimationAlgorithm;
    fReprojThreshold      : LREAL;
    nMaxIterations        : UDINT;
    fConfidence           : LREAL;
    hrPrev                : HRESULT;
END_VAR
VAR_OUTPUT
    aPerspectiveTransform : TcVnMatrix3x3_LREAL;
    nNumberOfGoodMatches  : UDINT;
    nNumberOfInliers      : UDINT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipRefKeyPoints	ITcVnContainer [▸ 349]	KeyPoints of the reference image, e.g. computed with F_VN_KeyPointsAndDescriptorsBRISK (ContainerType_Vector_TcVnKeyPoint)
ipRefDescriptors	ITcVnImage [▸ 390]	Descriptors of the reference image KeyPoints, e.g. computed with F_VN_KeyPointsAndDescriptorsBRISK
nRefImageWidth	UDINT	Width of the reference image in pixels
nRefImageHeight	UDINT	Height of the reference image in pixels
ipEdgePoints	Reference To ITcVnContainer [▸ 349]	Returns a container with the 4 edge points of the reference image transformed into ipSrcImage coordinates (ContainerType_Vector_TcVnPoint2_REAL)
fMaxDist	REAL	Maximum allowed descriptor distance (-1 disables this filter criterion)
fMaxKnnRatio	REAL	Maximum allowed distance ratio between first and second best match ([0..1], -1 disables this filter criterion)
eNormType	ETcVnNormalizationType [▸ 189]	Normalization type used for descriptor matching
ipSrcImageMask	ITcVnImage [▸ 390]	Mask to specify, where to look for keypoints in ipSrcImage (set to 0 if no mask required)
eAlgorithm	ETcVnEstimationAlgorithm [▸ 179]	Estimation algorithm used for computing the perspective transformation between the two point sets
fReprojThreshold	LREAL	Maximum allowed reprojection error to treat a point pair as an inlier (only used if eAlgorithm is RANSAC, RHO)
nMaxIterations	UDINT	Maximum number of RANSAC iterations
fConfidence	LREAL	Confidence (0..1)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stBriskParams	TcVnParamsBRISK [▸ 217]	Parameters to configure the keypoint and descriptor computation for ipSrcImage (resulting descriptors must be compatible to ipRefDescriptors!)

Outputs

Name	Type	Description
aPerspectiveTransform	TcVnMatrix3x3_LREAL [▸ 141]	Returns the perspective transformation matrix, which transforms the reference points to the source points
nNumberOfGoodMatches	UDINT	Return the number of good matches
nNumberOfInliers	UDINT	Return the number of inlier matches

Return value

HRESULT [▶ 122]

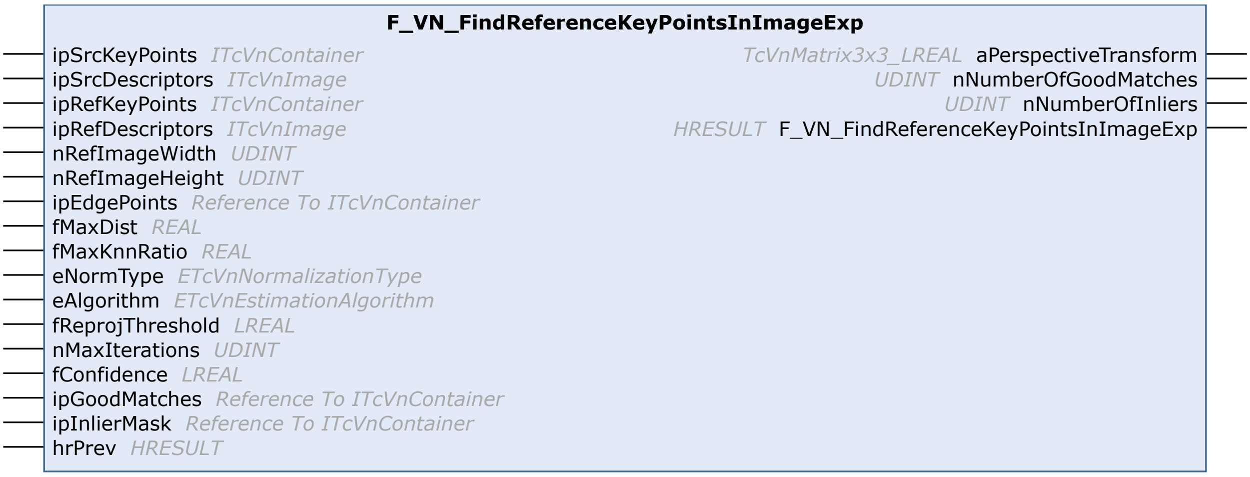
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.7 F_VN_FindReferenceKeyPointsInImageExp



Searches a reference image, represented by its keypoints and corresponding descriptors, in a given source image. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_FindReferenceKeyPointsInImageExp : HRESULT
VAR_INPUT
    ipSrcKeyPoints          : ITcVnContainer;
    ipSrcDescriptors        : ITcVnImage;
    ipRefKeyPoints          : ITcVnContainer;
    ipRefDescriptors        : ITcVnImage;
    nRefImageWidth          : UDINT;
    nRefImageHeight         : UDINT;
    ipEdgePoints            : Reference To ITcVnContainer;
    fMaxDist                : REAL;
    fMaxKnnRatio            : REAL;
    eNormType               : ETcVnNormalizationType;
    eAlgorithm              : ETcVnEstimationAlgorithm;
    fReprojThreshold        : LREAL;
    nMaxIterations          : UDINT;
    fConfidence             : LREAL;
    ipGoodMatches           : Reference To ITcVnContainer;
    ipInlierMask            : Reference To ITcVnContainer;
    hrPrev                  : HRESULT;
END_VAR
VAR_OUTPUT
    aPerspectiveTransform : TcVnMatrix3x3_LREAL;
    nNumberOfGoodMatches  : UDINT;
    nNumberOfInliers      : UDINT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcKeyPoints	ITcVnContainer [▸ 349]	KeyPoints of the source image, e.g. (ContainerType_Vector_TcVnKeyPoint)
ipSrcDescriptors	ITcVnImage [▸ 390]	Descriptors of the source image KeyPoints
ipRefKeyPoints	ITcVnContainer [▸ 349]	KeyPoints of the reference image (ContainerType_Vector_TcVnKeyPoint)
ipRefDescriptors	ITcVnImage [▸ 390]	Descriptors of the reference image KeyPoints
nRefImageWidth	UDINT	Width of the reference image in pixels
nRefImageHeight	UDINT	Height of the reference image in pixels
ipEdgePoints	Reference To ITcVnContainer [▸ 349]	Returns a container with the 4 edge points of the reference image transformed into the coordinates of the source image (ContainerType_Vector_TcVnPoint2_REAL)
fMaxDist	REAL	Maximum allowed descriptor distance (-1 disables this filter criterion)
fMaxKnnRatio	REAL	Maximum allowed distance ratio between first and second best match ([0..1], -1 disables this filter criterion)
eNormType	ETcVnNormalizationType [▸ 189]	Normalization type used for descriptor matching (HAMMING recommended for AKAZE, ORB, and BRISK. HAMMING2 is recommended for ORB if the ORB nBriefPoints parameter is 3 or 4.)
eAlgorithm	ETcVnEstimationAlgorithm [▸ 179]	Estimation algorithm used for computing the perspective transformation between the two point sets
fReprojThreshold	LREAL	Maximum allowed reprojection error to treat a point pair as an inlier (only used if eAlgorithm is RANSAC, RHO)
nMaxIterations	UDINT	Maximum number of RANSAC iterations
fConfidence	LREAL	Confidence (0..1)
ipGoodMatches	Reference To ITcVnContainer [▸ 349]	Returns a container with good matches (optional, set to 0 if not required; ContainerType_Vector_TcVnDMatch)
ipInlierMask	Reference To ITcVnContainer [▸ 349]	Returns a mask marking the inliers (optional, set to 0 if not required; ContainerType_Vector_SINT; only for RANSAC, LMEDS)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Outputs

Name	Type	Description
aPerspectiveTransform	TcVnMatrix3x3_LREAL [▸ 141]	Returns the perspective transformation matrix, which transforms the reference points to the source points
nNumberOfGoodMatches	UDINT	Return the number of good matches
nNumberOfInliers	UDINT	Return the number of inlier matches

Return value

HRESULT [▸ 122]

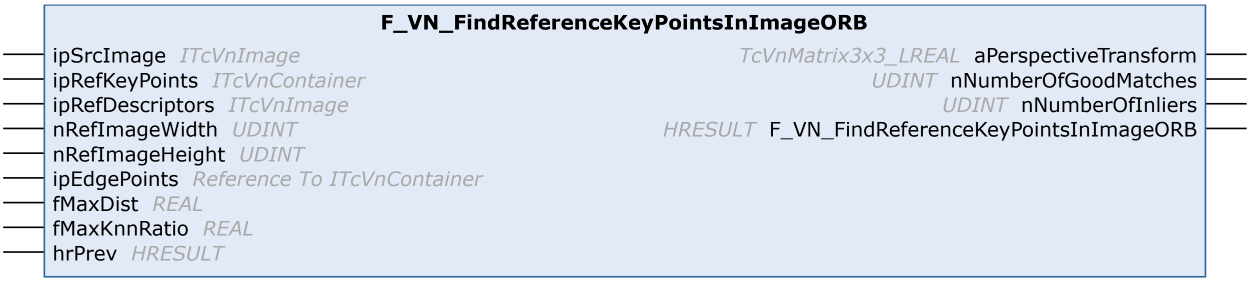
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.8 F_VN_FindReferenceKeyPointsInImageORB



Searches a reference image, represented by its ORB keypoints and corresponding descriptors, in a given source image (uses the default TcVnParamsORB parameters).
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_FindReferenceKeyPointsInImageORB : HRESULT
VAR_INPUT
    ipSrcImage            : ITcVnImage;
    ipRefKeyPoints        : ITcVnContainer;
    ipRefDescriptors      : ITcVnImage;
    nRefImageWidth        : UDINT;
    nRefImageHeight       : UDINT;
    ipEdgePoints          : Reference To ITcVnContainer;
    fMaxDist              : REAL;
    fMaxKnnRatio          : REAL;
    hrPrev                : HRESULT;
END_VAR
VAR_OUTPUT
    aPerspectiveTransform : TcVnMatrix3x3_LREAL;
    nNumberOfGoodMatches  : UDINT;
    nNumberOfInliers      : UDINT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipRefKeyPoints	ITcVnContainer [▶ 349]	KeyPoints of the reference image, e.g. computed with F_VN_KeyPointsAndDescriptorsORB (ContainerType_Vector_TcVnKeyPoint)
ipRefDescriptors	ITcVnImage [▶ 390]	Descriptors of the reference image KeyPoints, e.g. computed with F_VN_KeyPointsAndDescriptorsORB
nRefImageWidth	UDINT	Width of the reference image in pixels
nRefImageHeight	UDINT	Height of the reference image in pixels
ipEdgePoints	Reference To ITcVnContainer [▶ 349]	Returns a container with the 4 edge points of the reference image transformed into ipSrcImage coordinates (ContainerType_Vector_TcVnPoint2_REAL)
fMaxDist	REAL	Maximum allowed descriptor distance (-1 disables this filter criterion)
fMaxKnnRatio	REAL	Maximum allowed distance ratio between first and second best match ([0..1], -1 disables this filter criterion)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Outputs

Name	Type	Description
aPerspectiveTransform	TcVnMatrix3x3_LREAL [▶ 141]	Returns the perspective transformation matrix, which transforms the reference points to the source points
nNumberOfGoodMatches	UDINT	Return the number of good matches
nNumberOfInliers	UDINT	Return the number of inlier matches

Return value

HRESULT [▶ 122]

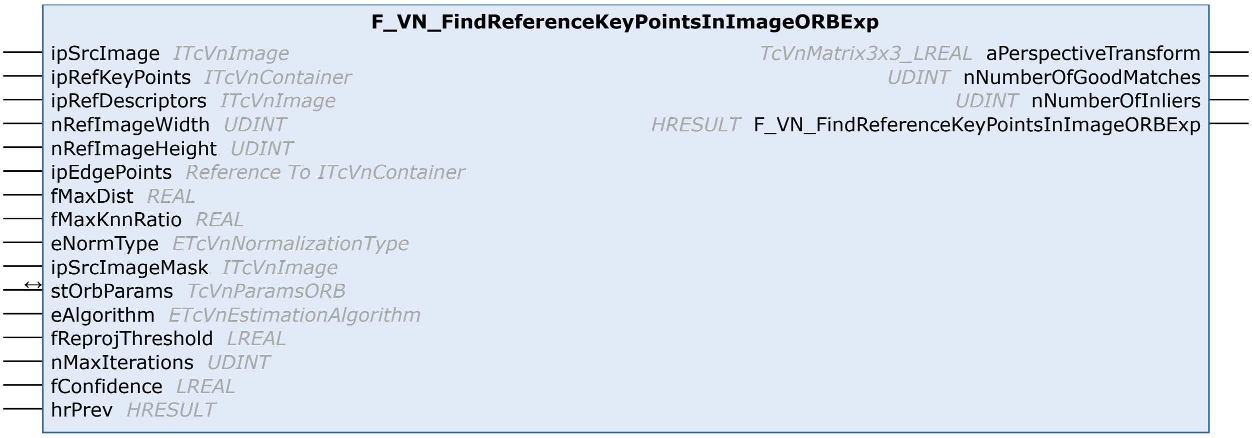
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.9 F_VN_FindReferenceKeyPointsInImageORBExp



Searches a reference image, represented by its ORB keypoints and corresponding descriptors, in a given source image. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_FindReferenceKeyPointsInImageORBExp : HRESULT
VAR_INPUT
    ipSrcImage              : ITcVnImage;
    ipRefKeyPoints          : ITcVnContainer;
    ipRefDescriptors        : ITcVnImage;
    nRefImageWidth          : UDINT;
    nRefImageHeight         : UDINT;
    ipEdgePoints            : Reference To ITcVnContainer;
    fMaxDist                : REAL;
    fMaxKnnRatio            : REAL;
    eNormType               : ETcVnNormalizationType;
    ipSrcImageMask          : ITcVnImage;
END_VAR
VAR_IN_OUT
    stOrbParams             : TcVnParamsORB;
END_VAR
VAR_INPUT
    eAlgorithm              : ETcVnEstimationAlgorithm;
    fReprojThreshold        : LREAL;
    nMaxIterations          : UDINT;
    fConfidence             : LREAL;
    hrPrev                  : HRESULT;
END_VAR
VAR_OUTPUT
    aPerspectiveTransform : TcVnMatrix3x3_LREAL;
    nNumberOfGoodMatches  : UDINT;
    nNumberOfInliers      : UDINT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipRefKeyPoints	ITcVnContainer [▶ 349]	KeyPoints of the reference image, e.g. computed with F_VN_KeyPointsAndDescriptorsORB (ContainerType_Vector_TcVnKeyPoint)
ipRefDescriptors	ITcVnImage [▶ 390]	Descriptors of the reference image KeyPoints, e.g. computed with F_VN_KeyPointsAndDescriptorsORB
nRefImageWidth	UDINT	Width of the reference image in pixels
nRefImageHeight	UDINT	Height of the reference image in pixels
ipEdgePoints	Reference To ITcVnContainer [▶ 349]	Returns a container with the 4 edge points of the reference image transformed into ipSrcImage coordinates (ContainerType_Vector_TcVnPoint2_REAL)
fMaxDist	REAL	Maximum allowed descriptor distance (-1 disables this filter criterion)
fMaxKnnRatio	REAL	Maximum allowed distance ratio between first and second best match ([0..1], -1 disables this filter criterion)
eNormType	ETcVnNormalizationType [▶ 189]	Normalization type used for descriptor matching (HAMMING recommended for ORB, HAMMING2 if the ORB nBriefPoints parameter is 3 or 4)
ipSrcImageMask	ITcVnImage [▶ 390]	Mask to specify, where to look for keypoints in ipSrcImage (set to 0 if no mask required)
eAlgorithm	ETcVnEstimationAlgorithm [▶ 179]	Estimation algorithm used for computing the perspective transformation between the two point sets
fReprojThreshold	LREAL	Maximum allowed reprojection error to treat a point pair as an inlier (only used if eAlgorithm is RANSAC, RHO)
nMaxIterations	UDINT	Maximum number of RANSAC iterations
fConfidence	LREAL	Confidence (0..1)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stOrbParams	TcVnParamsORB [▶ 220]	Parameters to configure the keypoint and descriptor computation for ipSrcImage (resulting descriptors must be compatible to ipRefDescriptors!)

Outputs

Name	Type	Description
aPerspectiveTransform	TcVnMatrix3x3_LREAL [▶ 141]	Returns the perspective transformation matrix, which transforms the reference points to the source points
nNumberOfGoodMatches	UDINT	Return the number of good matches
nNumberOfInliers	UDINT	Return the number of inlier matches

Return value

HRESULT [▶ 122]

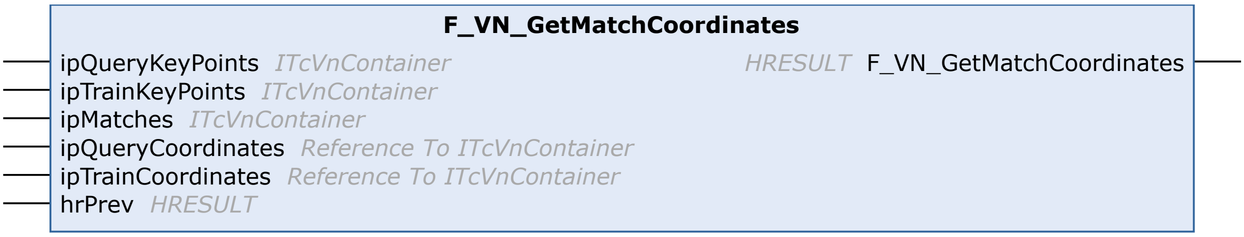
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.10 F_VN_GetMatchCoordinates



Return the coordinates of keypoints that match each other.

Syntax

Definition:

```
FUNCTION F_VN_GetMatchCoordinates : HRESULT
VAR_INPUT
    ipQueryKeyPoints   : ITcVnContainer;
    ipTrainKeyPoints   : ITcVnContainer;
    ipMatches          : ITcVnContainer;
    ipQueryCoordinates : Reference To ITcVnContainer;
    ipTrainCoordinates : Reference To ITcVnContainer;
    hrPrev             : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipQueryKeyPoints	ITcVnContainer [▶ 349]	Container with query Keypoints (i.e. keypoints of a reference/template image) of type CTcVnContainer_Vector_TcVnKeyPoint
ipTrainKeyPoints	ITcVnContainer [▶ 349]	Container with training Keypoints (i.e. keypoints of a source/input image) of type CTcVnContainer_Vector_TcVnKeyPoint
ipMatches	ITcVnContainer [▶ 349]	Container with matches between query and train Keypoints (CTcVnContainer_Vector_TcVnDMatch).
ipQueryCoordinates	Reference To ITcVnContainer [▶ 349]	Returns a container with the coordinates of keypoints from the query Keypoints that exist in ipMatches (CTcVnContainer_Vector_TcVnPoint2_REAL)
ipTrainCoordinates	Reference To ITcVnContainer [▶ 349]	Returns a container with the coordinates of keypoints from the training Keypoints that exist in ipMatches (CTcVnContainer_Vector_TcVnPoint2_REAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

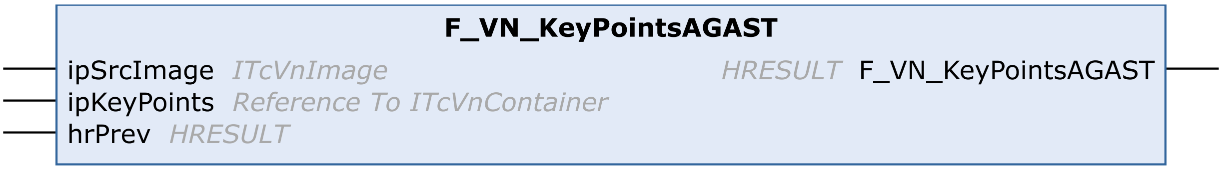
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.11 F_VN_KeyPointsAGAST



Detects keypoints using the AGAST method.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_KeyPointsAGAST : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipKeyPoints : Reference To ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipKeyPoints	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

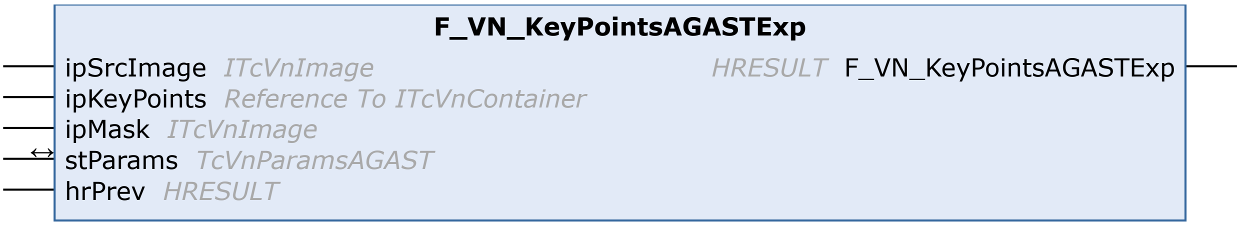
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.12 F_VN_KeyPointsAGASTExp



Detects keypoints using the AGAST method. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_KeyPointsAGASTExp : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipKeyPoints : Reference To ITcVnContainer;
    ipMask      : ITcVnImage;
END_VAR
VAR_IN_OUT
    stParams    : TcVnParamsAGAST;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipKeyPoints	Reference To ITcVnContainer [▸ 349]	Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipMask	ITcVnImage [▸ 390]	Mask to specify, where to look for keypoints (set to 0 if no mask required)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stParams	TcVnParamsAGAST [▸ 212]	Additional expert parameters

Return value

HRESULT [▸ 122]

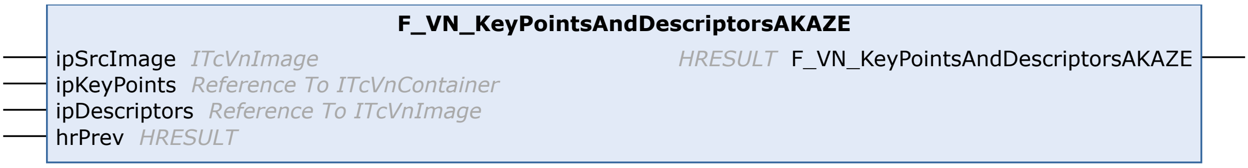
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.13 F_VN_KeyPointsAndDescriptorsAKAZE



Detects keypoints and compute descriptors using the AKAZE method.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_KeyPointsAndDescriptorsAKAZE : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipKeyPoints   : Reference To ITcVnContainer;
    ipDescriptors : Reference To ITcVnImage;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipKeyPoints	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipDescriptors	Reference To ITcVnImage [▶ 390]	Descriptor image (set to 0 if not required; 1 descriptor in each row; An appropriate image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

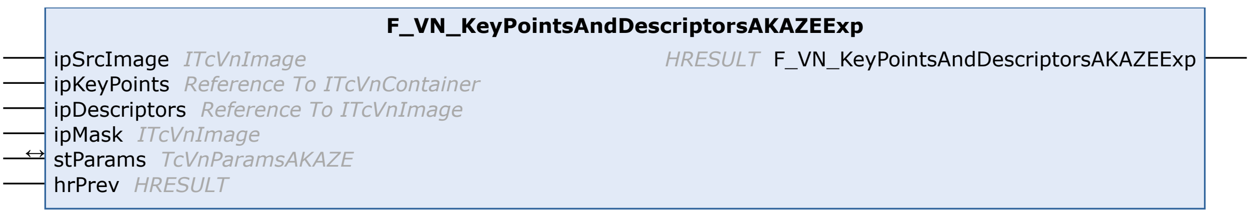
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.14 F_VN_KeyPointsAndDescriptorsAKAZEExp



Detects keypoints and compute descriptors using the AKAZE method. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_KeyPointsAndDescriptorsAKAZEExp : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipKeyPoints    : Reference To ITcVnContainer;
    ipDescriptors  : Reference To ITcVnImage;
    ipMask         : ITcVnImage;
END_VAR
VAR_IN_OUT
    stParams       : TcVnParamsAKAZE;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipKeyPoints	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipDescriptors	Reference To ITcVnImage [▶ 390]	Descriptor image (set to 0 if not required; 1 descriptor in each row; An appropriate image will be created if required.)
ipMask	ITcVnImage [▶ 390]	Mask to specify, where to look for keypoints (set to 0 if no mask required)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stParams	TcVnParamsAKAZE [▶ 213]	Additional expert parameters

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.15 F_VN_KeyPointsAndDescriptorsBRISK

Detects keypoints and compute descriptors using the BRISK method.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_KeyPointsAndDescriptorsBRISK : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipKeyPoints    : Reference To ITcVnContainer;
    ipDescriptors : Reference To ITcVnImage;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipKeyPoints	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipDescriptors	Reference To ITcVnImage [▶ 390]	Descriptor image (set to 0 if not required; 1 descriptor in each row; An appropriate image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.16 F_VN_KeyPointsAndDescriptorsBRISKExp

Detects keypoints and compute descriptors using the BRISK method. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_KeyPointsAndDescriptorsBRISKExp : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipKeyPoints    : Reference To ITcVnContainer;
```

```
    ipDescriptors : Reference To ITcVnImage;
    ipMask        : ITcVnImage;
END_VAR
VAR_IN_OUT
    stParams      : TcVnParamsBRISK;
END_VAR
VAR_INPUT
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipKeyPoints	Reference To ITcVnContainer [▸ 349]	Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipDescriptors	Reference To ITcVnImage [▸ 390]	Descriptor image (set to 0 if not required; 1 descriptor in each row; An appropriate image will be created if required.)
ipMask	ITcVnImage [▸ 390]	Mask to specify, where to look for keypoints (set to 0 if no mask required)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stParams	TcVnParamsBRISK [▸ 217]	Additional expert parameters

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.17 F_VN_KeyPointsAndDescriptorsKAZE

```
F_VN_KeyPointsAndDescriptorsKAZE
ipSrcImage  ITcVnImage                          HRESULT  F_VN_KeyPointsAndDescriptorsKAZE
ipKeyPoints  Reference To ITcVnContainer
ipDescriptors  Reference To ITcVnImage
hrPrev  HRESULT
```

Detects keypoints and compute descriptors using the KAZE method.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_KeyPointsAndDescriptorsKAZE : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipKeyPoints    : Reference To ITcVnContainer;
    ipDescriptors : Reference To ITcVnImage;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipKeyPoints	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipDescriptors	Reference To ITcVnImage [▶ 390]	Descriptor image (set to 0 if not required; 1 descriptor in each row; An appropriate image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

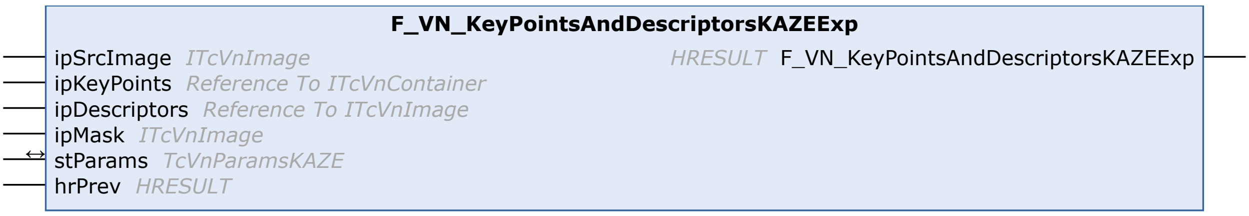
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.18 F_VN_KeyPointsAndDescriptorsKAZEExp



Detects keypoints and compute descriptors using the KAZE method. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_KeyPointsAndDescriptorsKAZEExp : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipKeyPoints    : Reference To ITcVnContainer;
    ipDescriptors : Reference To ITcVnImage;
    ipMask         : ITcVnImage;
END_VAR
VAR_IN_OUT
    stParams       : TcVnParamsKAZE;
END_VAR
```

```
VAR_INPUT
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipKeyPoints	Reference To ITcVnContainer [▸ 349]	Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipDescriptors	Reference To ITcVnImage [▸ 390]	Descriptor image (set to 0 if not required; 1 descriptor in each row; An appropriate image will be created if required.)
ipMask	ITcVnImage [▸ 390]	Mask to specify, where to look for keypoints (set to 0 if no mask required)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stParams	TcVnParamsKAZE [▸ 219]	Additional expert parameters

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.19 F_VN_KeyPointsAndDescriptorsORB

Detects keypoints and compute descriptors using the ORB method.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_KeyPointsAndDescriptorsORB : HRESULT
VAR_INPUT
    ipSrcImage    : ITcVnImage;
    ipKeyPoints   : Reference To ITcVnContainer;
    ipDescriptors : Reference To ITcVnImage;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipKeyPoints	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipDescriptors	Reference To ITcVnImage [▶ 390]	Descriptor image (set to 0 if not required; 1 descriptor in each row; An appropriate image will be created if required.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

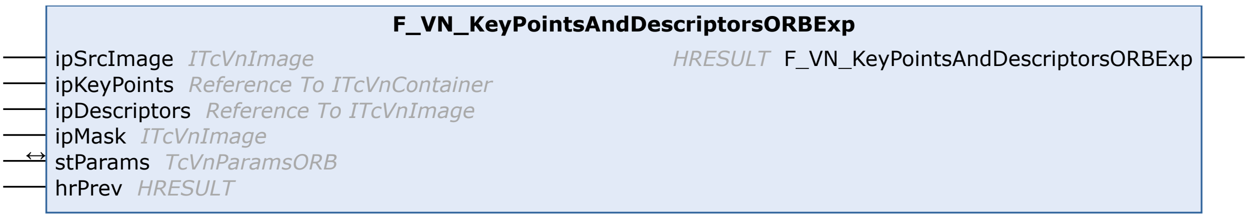
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.20 F_VN_KeyPointsAndDescriptorsORBExp



Detect keypoints and compute descriptors using the ORB method. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_KeyPointsAndDescriptorsORBExp : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
    ipKeyPoints    : Reference To ITcVnContainer;
    ipDescriptors  : Reference To ITcVnImage;
    ipMask         : ITcVnImage;
END_VAR
VAR_IN_OUT
    stParams       : TcVnParamsORB;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipKeyPoints	Reference To ITcVnContainer [▸ 349]	Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipDescriptors	Reference To ITcVnImage [▸ 390]	Descriptor image (set to 0 if not required; 1 descriptor in each row; An appropriate image will be created if required.)
ipMask	ITcVnImage [▸ 390]	Mask to specify, where to look for keypoints (set to 0 if no mask required)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stParams	TcVnParamsORB [▸ 220]	Additional expert parameters

Return value

HRESULT [▸ 122]

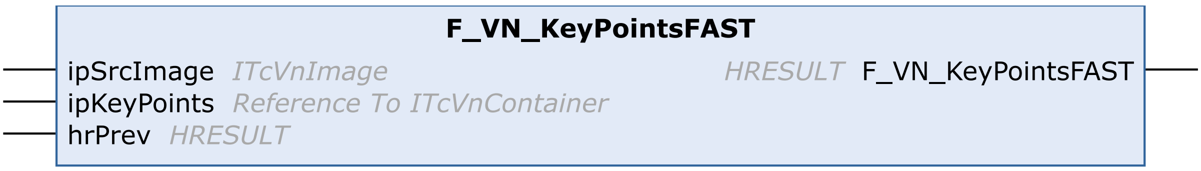
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.21 F_VN_KeyPointsFAST



Detects keypoints using the FAST method.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_KeyPointsFAST : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipKeyPoints : Reference To ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (single-channel).
ipKeyPoints	Reference To ITcVnContainer [▸ 349]	Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

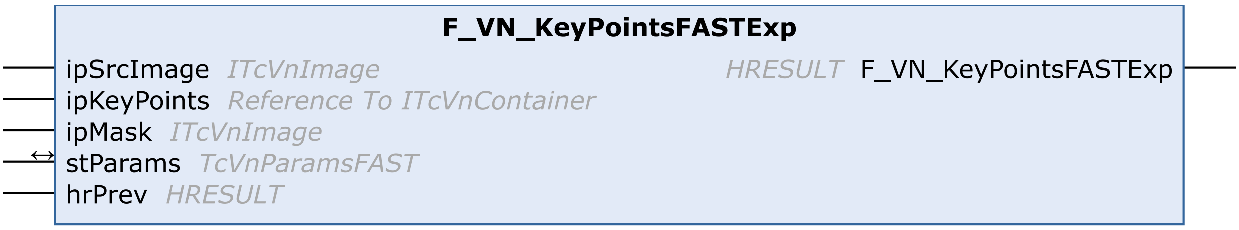
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.22 F_VN_KeyPointsFASTExp



Detects keypoints using the FAST method. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_KeyPointsFASTExp : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipKeyPoints : Reference To ITcVnContainer;
    ipMask      : ITcVnImage;
END_VAR
VAR_IN_OUT
    stParams    : TcVnParamsFAST;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipKeyPoints	Reference To ITcVnContainer [▸ 349]	Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipMask	ITcVnImage [▸ 390]	Mask to specify, where to look for keypoints (set to 0 if no mask required)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stParams	TcVnParamsFAST [▸ 218]	Additional expert parameters

Return value

HRESULT [▸ 122]

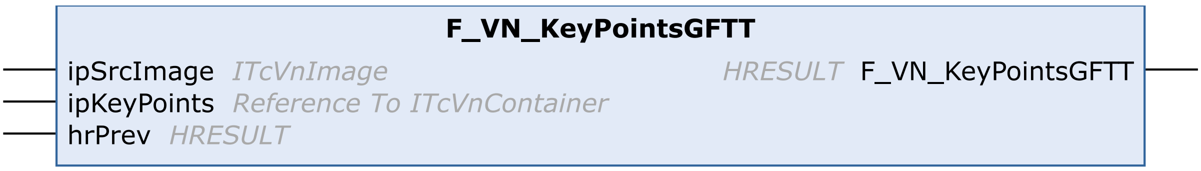
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.23 F_VN_KeyPointsGFTT



Detects keypoints using the GFTT method, which detects strong corners.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_KeyPointsGFTT : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipKeyPoints : Reference To ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipKeyPoints	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

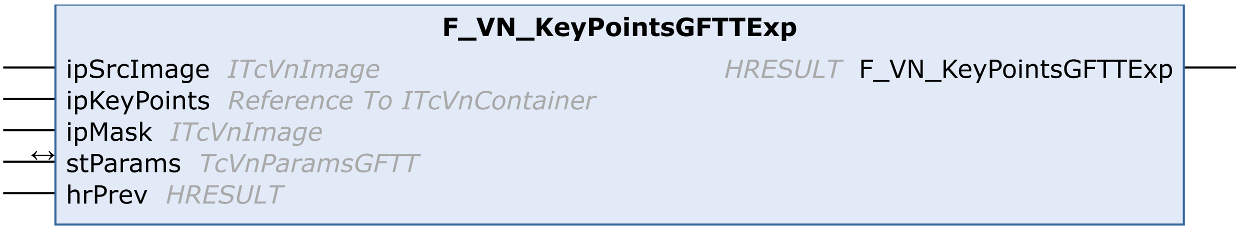
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.24 F_VN_KeyPointsGFTTExp



Detects keypoints using the GFTT method, which detects strong corners. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_KeyPointsGFTTExp : HRESULT
VAR_INPUT
    ipSrcImage   : ITcVnImage;
    ipKeyPoints  : Reference To ITcVnContainer;
    ipMask       : ITcVnImage;
END_VAR
VAR_IN_OUT
    stParams     : TcVnParamsGFTT;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipKeyPoints	Reference To ITcVnContainer [▸ 349]	Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipMask	ITcVnImage [▸ 390]	Mask to specify, where to look for keypoints (set to 0 if no mask required)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stParams	TcVnParamsGFTT [▸ 218]	Additional expert parameters

Return value

HRESULT [▸ 122]

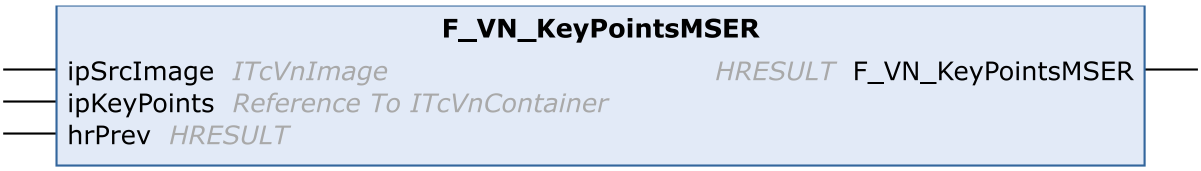
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.25 F_VN_KeyPointsMSER



Detects keypoints using the MSER method.

Syntax

Definition:

```
FUNCTION F_VN_KeyPointsMSER : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipKeyPoints : Reference To ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
ipKeyPoints	Reference To ITcVnContainer [▸ 349]	Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

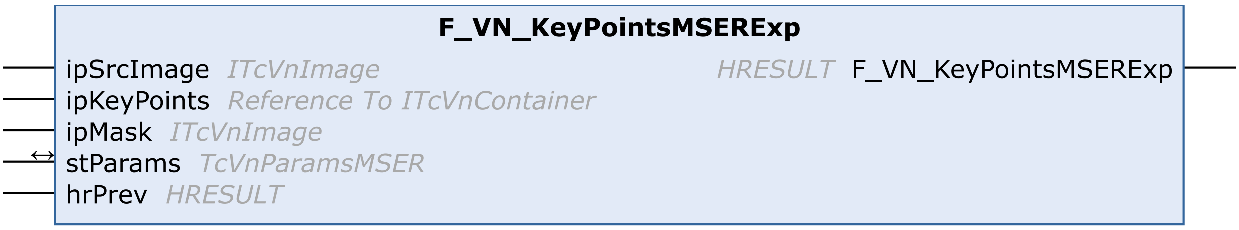
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.26 F_VN_KeyPointsMSERExp



Detects keypoints using the MSER method. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_KeyPointsMSERExp : HRESULT
VAR_INPUT
    ipSrcImage   : ITcVnImage;
    ipKeyPoints : Reference To ITcVnContainer;
    ipMask       : ITcVnImage;
END_VAR
VAR_IN_OUT
    stParams     : TcVnParamsMSER;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipKeyPoints	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipMask	ITcVnImage [▶ 390]	Mask to specify, where to look for keypoints (set to 0 if no mask required)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stParams	TcVnParamsMSER [▶ 220]	Additional expert parameters

Return value

HRESULT [▶ 122]

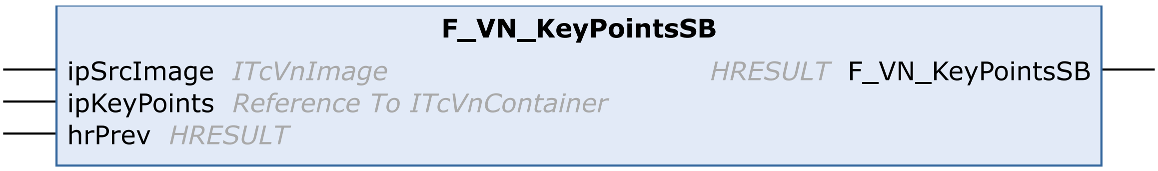
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.27 F_VN_KeyPointsSB



Detects keypoints using a Simple Blob method - several iterations apply different thresholds to source image - connected components (blobs) are detected - the center and radius of the blobs are returned as keypoints
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_KeyPointsSB : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipKeyPoints : Reference To ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipKeyPoints	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

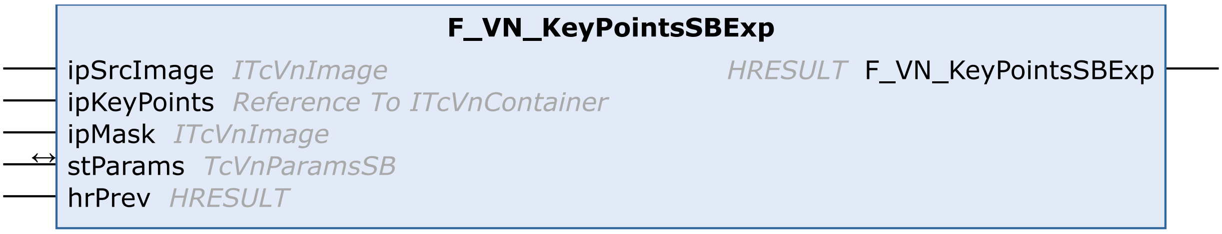
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.28 F_VN_KeyPointsSBExp



Detects keypoints using a Simple Blob method (expert function) - several iterations apply different thresholds to source image - connected components (blobs) are detected - the center and radius of the blobs are returned as keypoints
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_KeyPointsSBExp : HRESULT
VAR_INPUT
    ipSrcImage  : ITcVnImage;
    ipKeyPoints : Reference To ITcVnContainer;
    ipMask      : ITcVnImage;
END_VAR
VAR_IN_OUT
    stParams    : TcVnParamsSB;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipKeyPoints	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipMask	ITcVnImage [▶ 390]	Mask to specify, where to look for keypoints (optional, set to 0 if not required)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stParams	TcVnParamsSB [▶ 221]	Several parameters to filter the detected blobs

Return value

HRESULT [▶ 122]

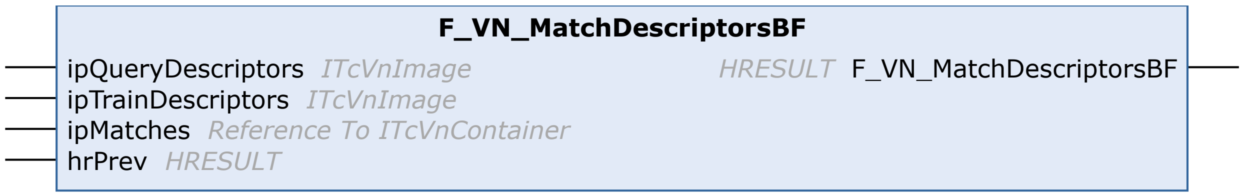
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.29 F_VN_MatchDescriptorsBF



Match descriptors using a brute force approach.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_MatchDescriptorsBF : HRESULT
VAR_INPUT
    ipQueryDescriptors : ITcVnImage;
    ipTrainDescriptors : ITcVnImage;
    ipMatches          : Reference To ITcVnContainer;
    hrPrev             : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipQueryDescriptors	ITcVnImage [▶ 390]	Query descriptors (i.e. descriptors of a reference/template image)
ipTrainDescriptors	ITcVnImage [▶ 390]	Training descriptors (i.e. descriptors of a source/input image)
ipMatches	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the descriptor matches (ContainerType_Vector_TcVnDMatch; Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

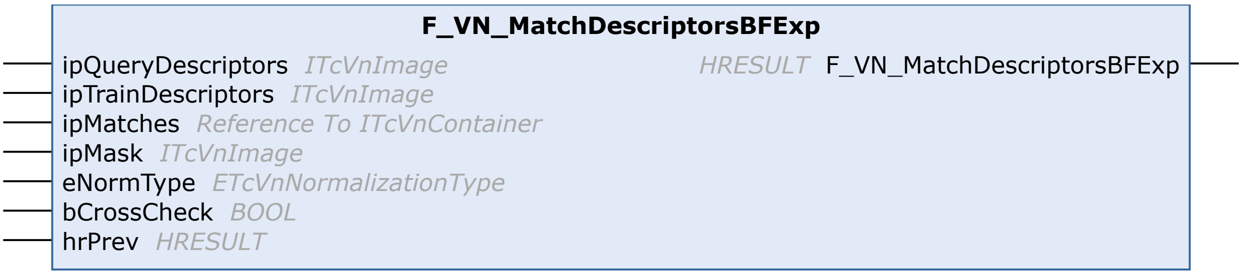
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.30 F_VN_MatchDescriptorsBFExp



Match descriptors using a brute force approach. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_MatchDescriptorsBFExp : HRESULT
VAR_INPUT
    ipQueryDescriptors : ITcVnImage;
    ipTrainDescriptors : ITcVnImage;
    ipMatches          : Reference To ITcVnContainer;
    ipMask             : ITcVnImage;
    eNormType          : ETcVnNormalizationType;
    bCrossCheck        : BOOL;
    hrPrev             : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipQueryDescriptors	ITcVnImage [▶ 390]	Query descriptors (i.e. descriptors of a reference/template image)
ipTrainDescriptors	ITcVnImage [▶ 390]	Training descriptors (i.e. descriptors of a source/input image)
ipMatches	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the descriptor matches (ContainerType_Vector_TcVnDMatch; Non-zero interface pointers are reused.)
ipMask	ITcVnImage [▶ 390]	Mask to specify permissable matches, i.e. query[i] can be matched with train[j] only if mask[i][j] != 0 (set to 0 if no mask required)
eNormType	ETcVnNormalizationType [▶ 189]	Normalization type (only L1, L2, L2SQR, HAMMING, HAMMING2 supported). HAMMING2 should be used for ORB descriptors if the ORB nBriefPoints parameter is 3 or 4)
bCrossCheck	BOOL	If true, only consistent matches are returned, i.e. query->train and train->query detect the same match
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

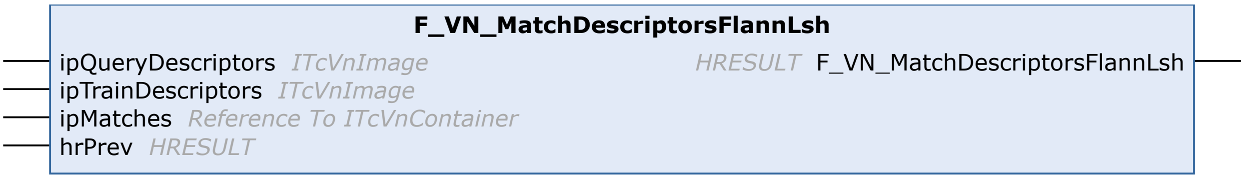
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.31 F_VN_MatchDescriptorsFlannLsh



Match descriptors using a FLANN based approach with LSH index.

Syntax

Definition:

```
FUNCTION F_VN_MatchDescriptorsFlannLsh : HRESULT
VAR_INPUT
    ipQueryDescriptors : ITcVnImage;
    ipTrainDescriptors : ITcVnImage;
    ipMatches          : Reference To ITcVnContainer;
    hrPrev             : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipQueryDescriptors	ITcVnImage [▶ 390]	Query descriptors (i.e. descriptors of a reference/template image)
ipTrainDescriptors	ITcVnImage [▶ 390]	Training descriptors (i.e. descriptors of a source/input image)
ipMatches	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the descriptor matches (ContainerType_Vector_TcVnDMatch; Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

匹配的数量

由于与方法有关的原因，在每次调用此函数时，匹配的数量可能会有所不同。

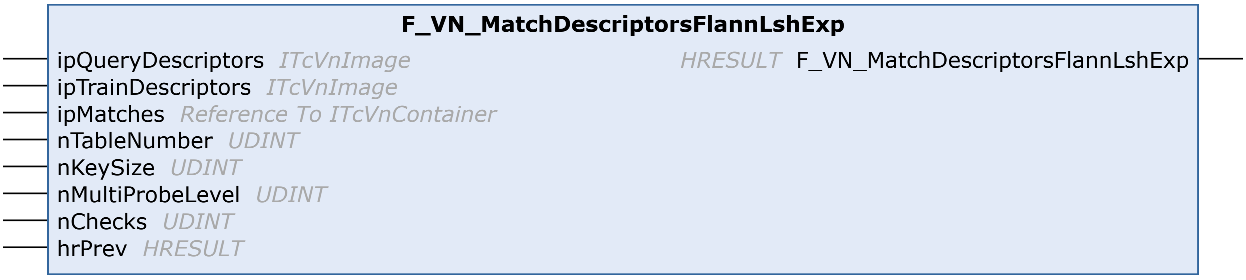
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.32 F_VN_MatchDescriptorsFlannLshExp



Match descriptors using a FLANN based approach with LSH index. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_MatchDescriptorsFlannLshExp : HRESULT
VAR_INPUT
    ipQueryDescriptors : ITcVnImage;
    ipTrainDescriptors : ITcVnImage;
    ipMatches          : Reference To ITcVnContainer;
    nTableNumber       : UDINT;
    nKeySize           : UDINT;
    nMultiProbeLevel   : UDINT;
```

```
    nChecks              : UDINT;
    hrPrev               : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipQueryDescriptors	ITcVnImage [▸ 390]	Query descriptors (i.e. descriptors of a reference/template image)
ipTrainDescriptors	ITcVnImage [▸ 390]	Training descriptors (i.e. descriptors of a source/input image)
ipMatches	Reference To ITcVnContainer [▸ 349]	Returns a container which is filled with the descriptor matches (ContainerType_Vector_TcVnDMatch; Non-zero interface pointers are reused.)
nTableNumber	UDINT	Number of tables
nKeySize	UDINT	Key size
nMultiProbeLevel	UDINT	Multi-probe level
nChecks	UDINT	Maximum number of visited leafs when searching for neighbors
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

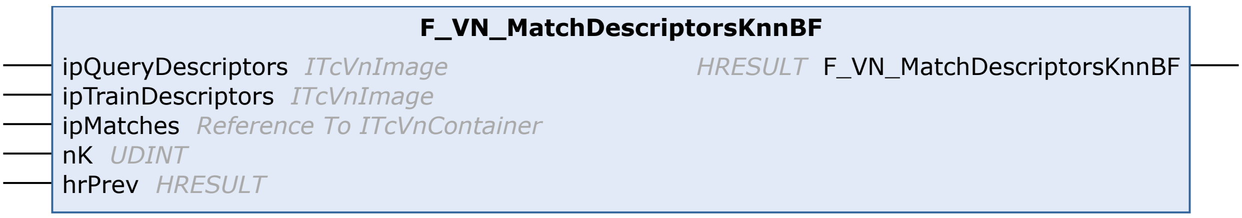
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.33 F_VN_MatchDescriptorsKnnBF



Match descriptors (k nearest neighbors) using a brute force approach.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_MatchDescriptorsKnnBF : HRESULT
VAR_INPUT
    ipQueryDescriptors : ITcVnImage;
    ipTrainDescriptors : ITcVnImage;
    ipMatches          : Reference To ITcVnContainer;
```

```
    nK                  : UDINT;
    hrPrev              : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipQueryDescriptors	ITcVnImage [▶ 390]	Query descriptors (i.e. descriptors of a reference/template image)
ipTrainDescriptors	ITcVnImage [▶ 390]	Training descriptors (i.e. descriptors of a source/input image)
ipMatches	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the descriptor matches (ContainerType_Vector_Vector_TcVnDMatch; Non-zero interface pointers are reused.)
nK	UDINT	Number of required best train matches for each query descriptor (i.e. nk := 2)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

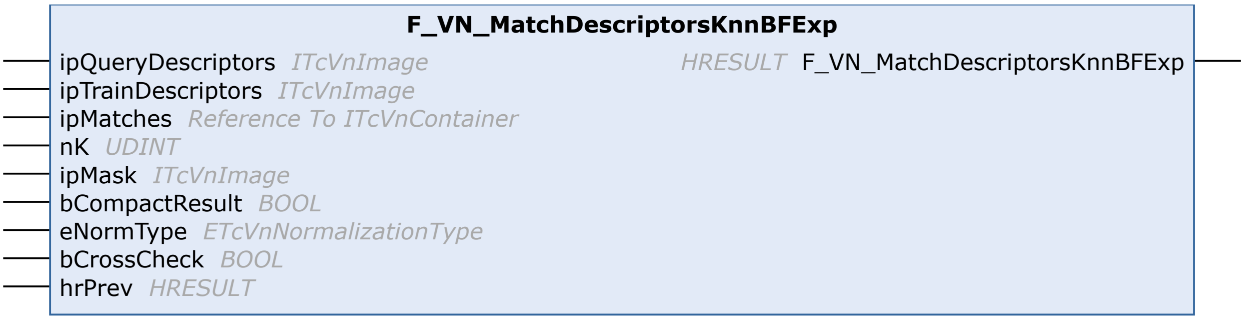
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.34 F_VN_MatchDescriptorsKnnBFExp



Match descriptors (k nearest neighbors) using a brute force approach. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_MatchDescriptorsKnnBFExp : HRESULT
VAR_INPUT
    ipQueryDescriptors : ITcVnImage;
    ipTrainDescriptors : ITcVnImage;
    ipMatches          : Reference To ITcVnContainer;
    nK                 : UDINT;
    ipMask             : ITcVnImage;
    bCompactResult     : BOOL;
```

```
    eNormType           : ETcVnNormalizationType;
    bCrossCheck         : BOOL;
    hrPrev              : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipQueryDescriptors	ITcVnImage [▶ 390]	Query descriptors (i.e. descriptors of a reference/template image)
ipTrainDescriptors	ITcVnImage [▶ 390]	Training descriptors (i.e. descriptors of a source/input image)
ipMatches	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the descriptor matches (ContainerType_Vector_Vector_TcVnDMatch; Non-zero interface pointers are reused.)
nK	UDINT	Number of required best train matches for each query descriptor (i.e. nk := 2)
ipMask	ITcVnImage [▶ 390]	Mask to specify permissable matches, i.e. query[i] can be matched with train[j] only if mask[i][j] != 0 (set to 0 if no mask required)
bCompactResult	BOOL	If true, matches vector does not contain entries for fully masked-out query descriptors
eNormType	ETcVnNormalizationType [▶ 189]	Normalization type (only L1, L2, L2SQR, HAMMING, HAMMING2 supported). HAMMING2 should be used for ORB descriptors if the ORB nBriefPoints parameter is 3 or 4)
bCrossCheck	BOOL	If true, only consistent matches are returned, i.e. query->train and train->query detect the same match (only used if nK = 1)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

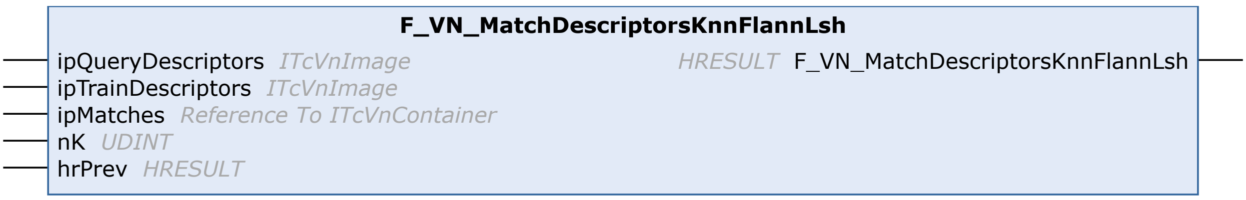
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.35 F_VN_MatchDescriptorsKnnFlannLsh



Match descriptors (k nearest neighbors) using a FLANN based approach with LSH index.

Syntax

Definition:

```
FUNCTION F_VN_MatchDescriptorsKnnFlannLsh : HRESULT
VAR_INPUT
    ipQueryDescriptors : ITcVnImage;
    ipTrainDescriptors : ITcVnImage;
    ipMatches          : Reference To ITcVnContainer;
    nK                 : UDINT;
    hrPrev             : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipQueryDescriptors	ITcVnImage [▶ 390]	Query descriptors (i.e. descriptors of a reference/template image)
ipTrainDescriptors	ITcVnImage [▶ 390]	Training descriptors (i.e. descriptors of a source/input image)
ipMatches	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the descriptor matches (ContainerType_Vector_Vector_TcVnDMatch; Non-zero interface pointers are reused.)
nK	UDINT	Number of required best train matches for each query descriptor (i.e. nk := 2)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

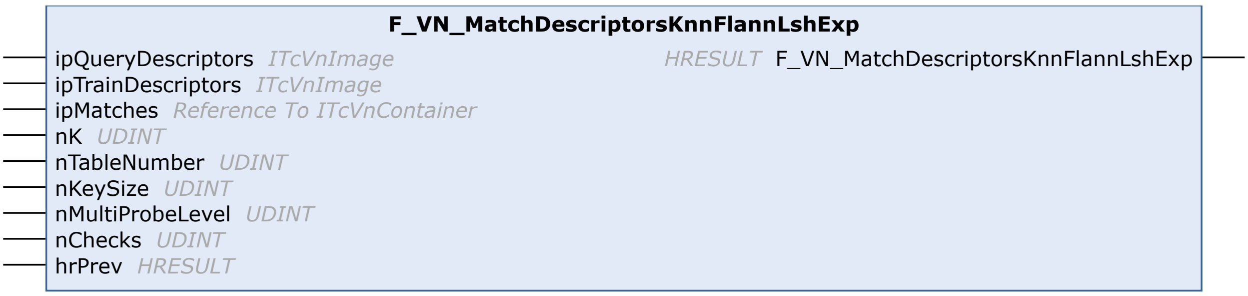
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.36 F_VN_MatchDescriptorsKnnFlannLshExp



Match descriptors (k nearest neighbors) using a FLANN based approach with LSH index. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_MatchDescriptorsKnnFlannLshExp : HRESULT
VAR_INPUT
    ipQueryDescriptors : ITcVnImage;
    ipTrainDescriptors : ITcVnImage;
    ipMatches          : Reference To ITcVnContainer;
    nK                 : UDINT;
    nTableNumber       : UDINT;
    nKeySize           : UDINT;
    nMultiProbeLevel   : UDINT;
    nChecks            : UDINT;
    hrPrev             : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipQueryDescriptors	ITcVnImage [▶ 390]	Query descriptors (i.e. descriptors of a reference/template image)
ipTrainDescriptors	ITcVnImage [▶ 390]	Training descriptors (i.e. descriptors of a source/input image)
ipMatches	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the descriptor matches (ContainerType_Vector_Vector_TcVnDMatch; Non-zero interface pointers are reused.)
nK	UDINT	Number of required best train matches for each query descriptor (i.e. nk := 2)
nTableNumber	UDINT	Number of tables
nKeySize	UDINT	Key size
nMultiProbeLevel	UDINT	Multi-probe level
nChecks	UDINT	Maximum number of visited leafs when searching for neighbors
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

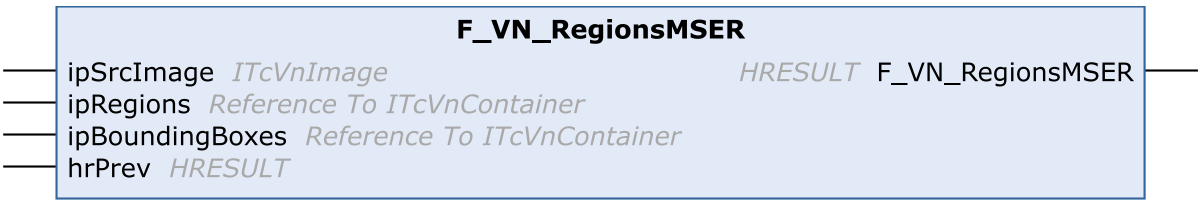
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.37 F_VN_RegionsMSER



Detects regions using the MSER method.

Syntax

Definition:

```
FUNCTION F_VN_RegionsMSER : HRESULT
VAR_INPUT
    ipSrcImage      : ITcVnImage;
    ipRegions       : Reference To ITcVnContainer;
    ipBoundingBoxes : Reference To ITcVnContainer;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipRegions	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the region points (ContainerType_Vector_Vector_TcVnPoint2_DINT; Non-zero interface pointers are reused.)
ipBoundingBoxes	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the region bounding boxes (ContainerType_Vector_TcVnRectangle_DINT; Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.18.38 F_VN_RegionsMSERExp

Detects regions using the MSER method. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_RegionsMSERExp : HRESULT
VAR_INPUT
    ipSrcImage      : ITcVnImage;
    ipRegions       : Reference To ITcVnContainer;
    ipBoundingBoxes : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    stParams        : TcVnParamsMSER;
```

```
END_VAR
VAR_INPUT
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image
ipRegions	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the region points (ContainerType_Vector_Vector_TcVnPoint2_DINT; Non-zero interface pointers are reused.)
ipBoundingBoxes	Reference To ITcVnContainer [▶ 349]	Returns a container which is filled with the region bounding boxes (ContainerType_Vector_TcVnRectangle_DINT; Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stParams	TcVnParamsMSER [▶ 220]	Additional expert parameters

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19 Machine Learning

该组包含用于创建、训练和应用机器学习模型的函数，以及用于特征预处理的函数。关于机器学习 [▶ 125] 的可能应用的简要概述和进一步说明可在软件概念 [▶ 119]下找到。

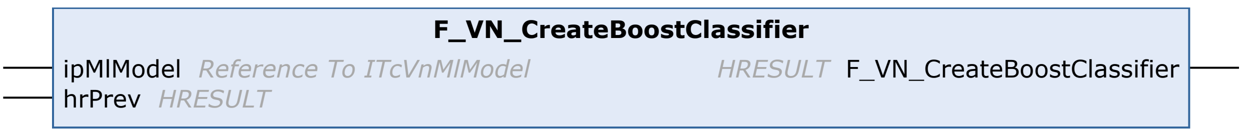
模型创建函数：

- F_VN_CreateKmppModel(Exp) [▶ 1340]
- F_VN_CreateKnnModel [▶ 1343]
- F_VN_CreateLbgModel(Exp) [▶ 1344]
- F_VN_CreateLdaTransform [▶ 1348]
- F_VN_CreateLdaTransformViaComponentNum [▶ 1350]
- F_VN_CreateNbcModel [▶ 1351]
- F_VN_CreatePcaTransform [▶ 1353]
- F_VN_CreatePcaTransformViaComponentNum [▶ 1354]
- F_VN_CreatePcaTransformViaVariance [▶ 1355]
- F_VN_CreateRTreesModel(Exp) [▶ 1357]
- F_VN_CreateStaModel(Exp) [▶ 1362]

还请参阅有关此

F_VN_CreateBoostClassifier [▶ 1336]

6.1.4.19.1 F_VN_CreateBoostClassifier



Create a Boost classifier using the default parameters. The initial reference count is set to one if a new model is created and kept, otherwise. The Boost classifier is only applicable to binary classification problems. It learns to distinguish between samples labelled with two user-defined class labels by incrementally adding weak classifiers to improve the classification results. Models of this type neither support on-line training (sample by sample) nor retraining.

Syntax

Definition:

```
FUNCTION F_VN_CreateBoostClassifier : HRESULT
VAR_INPUT
    ipMlModel : Reference To ITcVnMlModel;
    hrPrev    : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▶ 399]	Returns the created model (Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

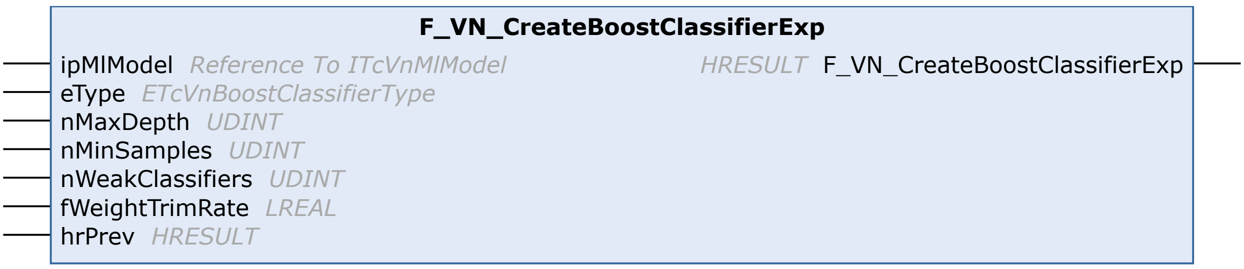
System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

还请参阅有关此

F_VN_CreateBoostClassifierExp [▶ 1337]
F_VN_CreateBoostClassifierExp2 [▶ 1338]

6.1.4.19.2 F_VN_CreateBoostClassifierExp



Create a Boost classifier. The initial reference count is set to one if a new model is created and kept, otherwise. The Boost classifier is only applicable to binary classification problems. It learns to distinguish between samples labelled with two user-defined class labels by incrementally adding weak classifiers to improve the classification results. Models of this type neither support on-line training (sample by sample) nor retraining. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_CreateBoostClassifierExp : HRESULT
VAR_INPUT
    ipMlModel          : Reference To ITcVnMlModel;
    eType              : ETcVnBoostClassifierType;
    nMaxDepth          : UDINT;
```

```
    nMinSamples       : UDINT;
    nWeakClassifiers  : UDINT;
    fWeightTrimRate   : LREAL;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▸ 399]	Returns the created model (Non-zero interface pointers are reused.)
eType	ETcVnBoostClassifierType [▸ 145]	Learning algorithm type (default: TCVN_BCT_REAL)
nMaxDepth	UDINT	Maximum tree depth (default: 1)
nMinSamples	UDINT	Minimum number of samples within a node required for splitting (default: 10)
nWeakClassifiers	UDINT	Number of weak classifiers (default: 100)
fWeightTrimRate	LREAL	Weight threshold used during training (off: 0; default: 0.95).
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Machine Learning

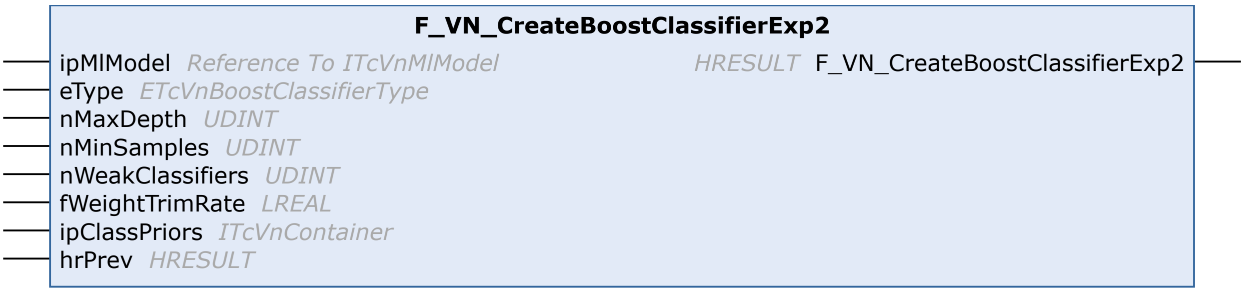
System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

还请参阅有关此

F_VN_CreateBoostClassifier [▸ 1336]

6.1.4.19.3 F_VN_CreateBoostClassifierExp2



Create a Boost classifier. The initial reference count is set to one if a new model is created and kept, otherwise. The Boost classifier is only applicable to binary classification problems. It learns to distinguish between samples labelled with two user-defined class labels by incrementally adding weak classifiers to improve the classification results. Models of this type neither support on-line training (sample by sample) nor retraining. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_CreateBoostClassifierExp2 : HRESULT
VAR_INPUT
    ipMlModel         : Reference To ITcVnMlModel;
    eType             : ETcVnBoostClassifierType;
    nMaxDepth         : UDINT;
    nMinSamples       : UDINT;
    nWeakClassifiers  : UDINT;
    fWeightTrimRate   : LREAL;
    ipClassPriors     : ITcVnContainer;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▸ 399]	Returns the created model (Non-zero interface pointers are reused.)
eType	ETcVnBoostClassifierType [▸ 145]	Learning algorithm type (default: TCVN_BCT_REAL)
nMaxDepth	UDINT	Maximum tree depth (default: 1)
nMinSamples	UDINT	Minimum number of samples within a node required for splitting (default: 10)
nWeakClassifiers	UDINT	Number of weak classifiers (default: 100)
fWeightTrimRate	LREAL	Weight threshold used during training (off: 0; default: 0.95).
ipClassPriors	ITcVnContainer [▸ 349]	Class priors (ContainerType_Vector_REAL or ContainerType_Vector_LREAL; only for classifiers; optional, set to 0 if not required or not allowed; default: 0)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

还请参阅有关此

F_VN_CreateBoostClassifier [▸ 1336]

6.1.4.19.4 F_VN_CreateKmppModel

F_VN_CreateKmppModel

ipMlModel *Reference To ITcVnMlModel* — *HRESULT* F_VN_CreateKmppModel
eKmppType *ETcVnPrototypeClusterer*
nK *UDINT*
hrPrev *HRESULT*

Create a k-means++ model of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type neither support on-line training (sample by sample) nor retraining.

Syntax

Definition:

```
FUNCTION F_VN_CreateKmppModel : HRESULT
VAR_INPUT
    ipMlModel : Reference To ITcVnMlModel;
    eKmppType : ETcVnPrototypeClusterer;
    nK        : UDINT;
    hrPrev    : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▶ 399]	Returns the created model (Non-zero interface pointers are reused.)
eKmppType	ETcVnPrototypeClusterer [▶ 194]	k-means++ model type
nK	UDINT	Parameter k (number of clusters)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_CreateKmppModel 可创建 K-Means++（KMPP）模型。

K-Means++ 模型

K-Means++ 可用于聚类或异常检测。它能发现数据点的聚类。需要同时对所有样本进行训练（批量训练），无法进行后期训练。

在不同训练中，集群名称的分配（如 DINT）可能会有所不同。如果该分配是确定性的，则必须事先使用 F_VN_SetRngSeed [▶ 1491]设置一个固定的随机种子。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

模型类型

eKmppType 指定 K-Means++ 是用于聚类（TCVN_PC_CLUSTERER）还是异常检测（TCVN_PC_NOVELTY_DETECTOR）。

聚类数

K-Means++ 模型中的聚类数由 nK 指定，一旦创建就不能更改。在训练过程中，样本数必须大于或等于聚类数。

专家参数

专家变体 F_VN_CreateKmppModelExp [▶ 1341] 包含附加参数。

应用

例如，一个包含 5 个聚类的 K-Means++ 模型可如下创建：

```
hr := F_VN_CreateKmppModel(
    ipMlModel    := ipMlModel,
    eKmppType    := TCVN_PC_CLUSTERER,
    nK           := 5,
    hrPrev       := hr);
```

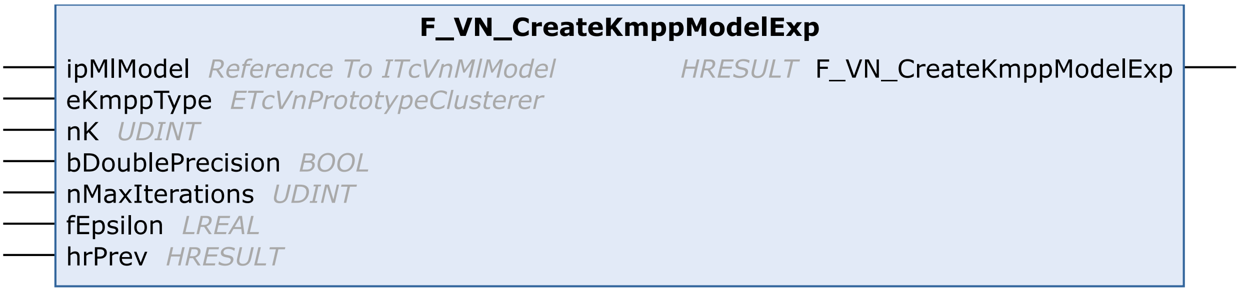
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.5 F_VN_CreateKmppModelExp



Create a k-means++ model of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type neither support on-line training (sample by sample) nor retraining. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_CreateKmppModelExp : HRESULT
VAR_INPUT
    ipMlModel          : Reference To ITcVnMlModel;
    eKmppType          : ETcVnPrototypeClusterer;
    nK                 : UDINT;
    bDoublePrecision : BOOL;
    nMaxIterations    : UDINT;
    fEpsilon           : LREAL;
    hrPrev             : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▶ 399]	Returns the created model (Non-zero interface pointers are reused.)
eKmppType	ETcVnPrototypeClusterer [▶ 194]	k-means++ model type
nK	UDINT	Parameter k (number of clusters)
bDoublePrecision	BOOL	If true, the model is generated with double precision (LREAL) instead of single precision (REAL). (default: FALSE)
nMaxIterations	UDINT	Maximum number of iterations (triggers the usage of the default value of 10 if it equals 0)
fEpsilon	LREAL	Maximum allowed difference of the error between two successive iterations (triggers the usage of the default value of 0.001 if it equals 0)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_CreateKmppModelExp 是F_VN_CreateKmppModel [▶ 1340] 的专家变体。它包含额外的参数。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

模型类型

eKmppType 指定 K-Means++ 是用于聚类（TCVN_PC_CLUSTERER）还是异常检测（TCVN_PC_NOVELTY_DETECTOR）。

聚类数

K-Means++ 模型中的聚类数由 nK 指定，一旦创建就不能更改。在训练过程中，样本数必须大于或等于聚类数。

更高精度

如果 bDoublePrecision 是 TRUE，则使用 LREAL 作为模型内部计算的数据类型；如果是 FALSE，则使用 REAL。

最大迭代

优化最多使用 nMaxIterations 中指定的迭代次数。如果值为 0，则使用相应的默认值。

终止限制

只要两次迭代之间的误差变化不超过 fEpsilon 中的规定，优化就会终止。如果值为 0，则使用相应的默认值。

应用

例如，一个包含 5 个聚类的 K-Means++ 模型可如下创建：

```
hr := F_VN_CreateKmppModelExp(
    ipMlModel           := ipMlModel,
    eKmppType           := TCVN_PC_CLUSTERER,
    nK                  := 5,
    bDoublePrecision    := FALSE,
    nMaxIterations      := 0,
    fEpsilon            := 0,
    hrPrev              := hr);
```

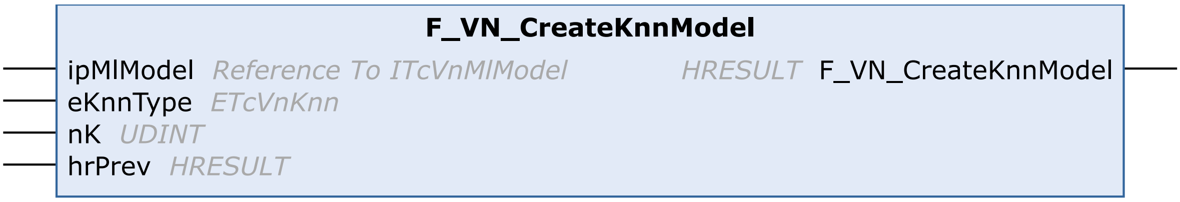
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.6 F_VN_CreateKnnModel



Create a k-nearest neighbors model of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type support on-line training (sample by sample) and retraining. Predictions can only be scalar.

Syntax

Definition:

```
FUNCTION F_VN_CreateKnnModel : HRESULT
VAR_INPUT
    ipMlModel : Reference To ITcVnMlModel;
    eKnnType  : ETcVnKnn;
    nK        : UDINT;
    hrPrev    : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▸ 399]	Returns the created model (Non-zero interface pointers are reused.)
eKnnType	ETcVnKnn [▸ 187]	k-nearest neighbors model type
nK	UDINT	Parameter k used for prediction (number of considered neighbors)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_CreateKnnModel 可创建 K 最近邻算法 (KNN) 模型。

K 最近邻算法模型

在训练 K 最近邻算法模型时，所有训练样本均以 1:1 的比例存储在模型中。多维空间（与样本的维度相对应）中的数据点这一概念有助于理解这一点。最接近输入样本（欧式距离）的数据点数 nK 会用于预测。这些数据点被称为“最近邻算法”。在预测中忽略所有其他数据点。预测结果由所有最近邻算法计算得出，无论输入样本的实际距离如何，权重均相同。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

模型类型

eKnnType 表示模型是用作分类器（TCVN_KNN_CLASSIFIER）、异常检测（TCVN_KNN_NOVELTY_DETECTOR）还是回归子（TCVN_KNN_REGRESSOR）。

使用的邻算法数量

nK 指定在预测中使用多少个数据点作为最近邻算法。对于异常检测，nK 并不重要，因为只需要最近邻算法。因此，在这种情况下应使用数值 1。

应用

例如，一个使用 5 个最近邻算法进行分类的 K 最近邻算法模型如下创建：

```
hr := F_VN_CreateKnnModel(
    ipMlModel    := ipMlModel,
    eKnnType     := TCVN_KNN_CLASSIFIER,
    nK           := 5,
    hrPrev       := hr);
```

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.7 F_VN_CreateLbgModel

Create a LBG model of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type neither support on-line training (sample by sample) nor retraining.

Syntax

Definition:

```
FUNCTION F_VN_CreateLbgModel : HRESULT
VAR_INPUT
    ipMlModel          : Reference To ITcVnMlModel;
    eLbgType           : ETcVnPrototypeClusterer;
    nMaxClusters       : UDINT;
```

```
    fMaxClusterRadius : LREAL;
    bSingleSplitSteps : BOOL;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▸ 399]	Returns the created model (Non-zero interface pointers are reused.)
eLbgType	ETcVnPrototypeClusterer [▸ 194]	LBG model type
nMaxClusters	UDINT	Maximum number of clusters
fMaxClusterRadius	LREAL	Maximum allowed radius (L2 norm) of a single cluster, i.e. clusters with a higher radius will be split into smaller ones, until a global number of nMaxClusters is reached.
bSingleSplitSteps	BOOL	If true, the global optimization is always run after a single cluster has been split. If false, several clusters are split within the same step before applying the global optimization. Applying the global optimization less often is faster, but can lead to less optimal results, especially having 2 nearby clusters that could be represented by 1.
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_CreateLbgModel 创建 Linde-Buzo-Gray（LBG）模型。

Linde-Buzo-Gray 模型

LBG 模型的工作原理与 K-Means++ 模型类似，都是根据样本迭代调整聚类。LBG 的主要特征是，在迭代过程中，只要新聚类大于 fMaxClusterRadius，就会创建新聚类（直到达到最大数 nMaxClusters）。另一方面，对于 K-Means++，聚类数为初始设置值。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

模型类型

eLbgType 指定 LBG 是用于聚类（TCVN_PC_CLUSTERER）还是异常检测（TCVN_PC_NOVELTY_DETECTOR）。

最大聚类数

如果由于拆分过大聚类而导致总计存在 nMaxClusters，则不会再创建新聚类。

最大聚类半径

在培训过程中，如果某个聚类的半径大于 fMaxClusterRadius，该聚类就会被分成两个较小聚类。

优化质量

迭代培训过程的一部分是全局优化步骤。如果 bSingleSplitSteps = TRUE，则在每个聚类拆分后执行该优化。这将提高聚类质量，但也会延长训练时间。如果 bSingleSplitSteps = FALSE，则只有在多个聚类被拆分后才会执行优化。

专家参数

专家变体 F_VN_CreateLbgModelExp [▶ 1346] 包含附加参数。

应用

例如，可通过以下方法创建一个具有 5 个聚类、最大聚类大小为 3.6 以及拥有较高优化质量的LBG 聚类模型：

```
hr := F_VN_CreateLbgModel(
    ipMlModel            := ipMlModel,
    eLbgType             := TCVN_PC_CLUSTERER,
    nMaxClusters         := 5,
    fMaxClusterRadius    := 3.6,
    bSingleSplitSteps    := TRUE,
    hrPrev               := hr);
```

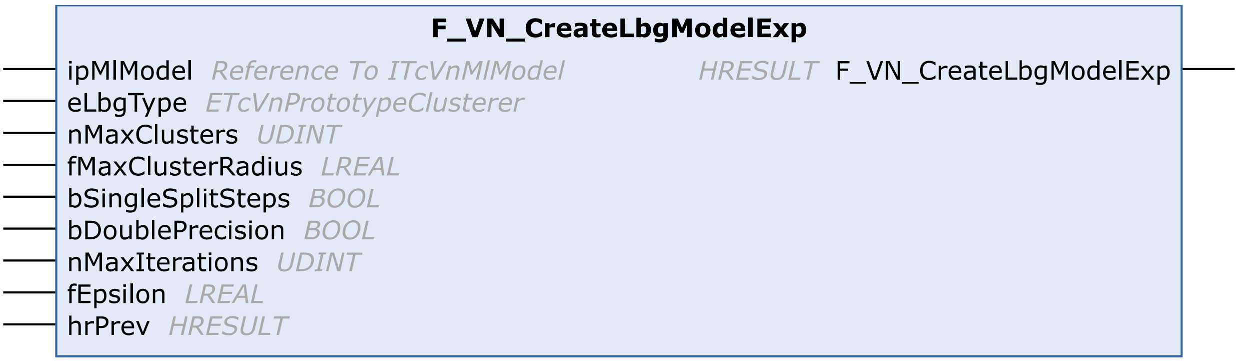
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.8 F_VN_CreateLbgModelExp



Create a LBG model of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type neither support on-line training (sample by sample) nor retraining. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_CreateLbgModelExp : HRESULT
VAR_INPUT
    ipMlModel          : Reference To ITcVnMlModel;
    eLbgType           : ETcVnPrototypeClusterer;
    nMaxClusters       : UDINT;
    fMaxClusterRadius  : LREAL;
    bSingleSplitSteps  : BOOL;
    bDoublePrecision   : BOOL;
    nMaxIterations     : UDINT;
    fEpsilon           : LREAL;
    hrPrev             : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▶ 399]	Returns the created model (Non-zero interface pointers are reused.)
eLbgType	ETcVnPrototypeClusterer [▶ 194]	LBG model type
nMaxClusters	UDINT	Maximum number of clusters
fMaxClusterRadius	LREAL	Maximum allowed radius (L2 norm) of a single cluster, i.e. clusters with a higher radius will be split into smaller ones, until a global number of nMaxClusters is reached.
bSingleSplitSteps	BOOL	If true, the global optimization is always run after a single cluster has been split. If false, several clusters are split within the same step before applying the global optimization. Applying the global optimization less often is faster, but can lead to less optimal results, especially having 2 nearby clusters that could be represented by 1.
bDoublePrecision	BOOL	If true, the model is generated with double precision (LREAL) instead of single precision (REAL). (default: FALSE)
nMaxIterations	UDINT	Maximum number of iterations (triggers the usage of the default value of 10 if it equals 0)
fEpsilon	LREAL	Maximum allowed difference of the error between two successive iterations (triggers the usage of the default value of 0.001 if it equals 0)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_CreateLbgModelExp 是 F_VN_CreateLbgModel [▶ 1344] 的专家变体。它包含额外的参数。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

模型类型

eLbgType 指定 LBG 是用于聚类（TCVN_PC_CLUSTERER）还是异常检测（TCVN_PC_NOVELTY_DETECTOR）。

最大聚类数

如果由于拆分过大聚类而导致总计存在 nMaxClusters，则不会再创建新聚类。

最大聚类半径

在培训过程中，如果某个聚类的半径大于 fMaxClusterRadius，该聚类就会被分成两个较小聚类。

优化质量

迭代培训过程的一部分是全局优化步骤。如果 bSingleSplitSteps = TRUE，则在每个聚类拆分后执行该优化。这将提高聚类质量，但也会延长训练时间。如果 bSingleSplitSteps = FALSE，则只有在多个聚类被拆分后才会执行优化。

更高精度

如果 bDoublePrecision 是 TRUE，则使用 LREAL 作为模型内部计算的数据类型；如果是 FALSE，则使用 REAL。

最大迭代

优化最多使用 nMaxIterations 中指定的迭代次数。如果值为 0，则使用相应的默认值。

终止限制

只要两次迭代之间的误差变化不超过 fEpsilon 中的规定，优化就会终止。如果值为 0，则使用相应的默认值。

应用

例如，可通过以下方法创建一个具有 5 个聚类、最大聚类大小为 3.6 以及拥有较高优化质量的LBG 聚类模型：

```
hr := F_VN_CreateLbgModelExp(
    ipMlModel            := ipMlModel,
    eLbgType             := TCVN_PC_CLUSTERER,
    nMaxClusters         := 5,
    fMaxClusterRadius    := 3.6,
    bSingleSplitSteps    := TRUE,
    bDoublePrecision     := TRUE,
    nMaxIterations       := 0,
    fEpsilon             := 0,
    hrPrev               := hr);
```

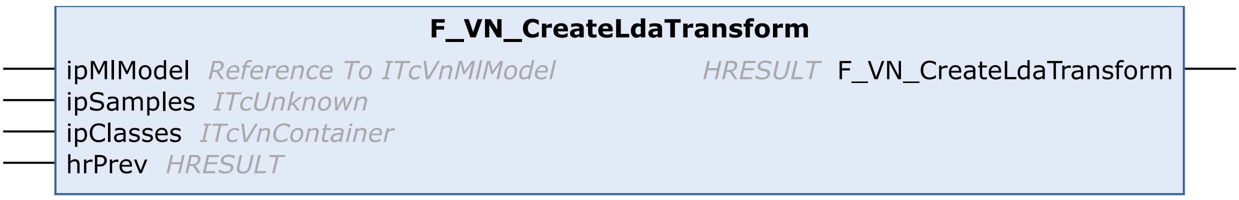
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.9 F_VN_CreateLdaTransform



Create an LDA-based feature transform from the provided data. The number of samples must be >= the number of features and the number of classes must be >= 2. The initial reference count is set to one if a new model is created and kept, otherwise.

Syntax

Definition:

```
FUNCTION F_VN_CreateLdaTransform : HRESULT
VAR_INPUT
    ipMlModel : Reference To ITcVnMlModel;
    ipSamples : ITcUnknown;
    ipClasses : ITcVnContainer;
    hrPrev    : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▶ 399]	Returns the created feature transform (Non-zero interface pointers are reused.)
ipSamples	ITcUnknown [▶ 407]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
ipClasses	ITcVnContainer [▶ 349]	Class labels corresponding to the input samples (ContainerType_Vector_DINT)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_CreateLdaTransform 可创建一个用于特征转换的线性判别分析（LDA）模型。

线性判别分析模型

线性判别分析创建一种特征转换，目的是利用转换后的特征尽可能容易地将类别区分开来。该模型可用于有效压缩样本。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

样本

所有样本均在一个容器中通过 ipSamples 发送至模型。

分类

容器中所有样本的类别分配通过 ipClasses 提供给模型。

应用

例如，一个用于特征降维的 LDA 模型可如下创建：

```
hr := F_VN_CreateLdaTransform(
    ipMlModel    := ipMlModel,
    ipSamples    := ipSamples,
    ipClasses    := ipClasses,
    hrPrev       := hr);
```

相关函数

- F_VN_CreateLdaTransformViaComponentNum [▶ 1350]
- F_VN_CreatePcaTransform [▶ 1353]
- F_VN_CreatePcaTransformViaComponentNum [▶ 1354]
- F_VN_CreatePcaTransformViaVariance [▶ 1355]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.10 F_VN_CreateLdaTransformViaComponentNum

Create an LDA-based feature transform from the provided data where the number of components to be used is predefined. The number of samples must be >= the number of features and the number of classes must be >= 2. The initial reference count is set to one if a new model is created and kept, otherwise. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_CreateLdaTransformViaComponentNum : HRESULT
VAR_INPUT
    ipMlModel     : Reference To ITcVnMlModel;
    ipSamples     : ITcUnknown;
    ipClasses     : ITcVnContainer;
    nComponentNum : UDINT;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▶ 399]	Returns the created feature transform (Non-zero interface pointers are reused.)
ipSamples	ITcUnknown [▶ 407]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
ipClasses	ITcVnContainer [▶ 349]	Class labels corresponding to the input samples (ContainerType_Vector_DINT)
nComponentNum	UDINT	Number of components (nComponentNum must be smaller than the number of classes. A value of 0 results in the automatic computation of the maximum number of components.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_CreateLdaTransformViaComponentNum 是 F_VN_CreateLdaTransform [▶ 1348] 的变体。它包含一个附加参数 nComponentNum，用于预先指定特征降维的组件数。这样就可以设置数据压缩级别。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

样本

所有样本均在一个容器中通过 ipSamples 发送至模型。

分类

容器中所有样本的类别分配通过 ipClasses 提供给模型。

组件数量

nComponentNum 组件用于特征降维。如为 0，将自动计算组件的最大数量。

应用

例如，一个包含 25 个组件的 LDA 模型可以如下创建：

```
hr := F_VN_CreateLdaTransformViaComponentNum(
    ipMlModel       := ipMlModel,
    ipSamples       := ipSamples,
    ipClasses       := ipClasses,
    nComponentNum   := 25,
    hrPrev          := hr);
```

相关函数

- F_VN_CreateLdaTransform [▶ 1348]
- F_VN_CreatePcaTransform [▶ 1353]
- F_VN_CreatePcaTransformViaComponentNum [▶ 1354]
- F_VN_CreatePcaTransformViaVariance [▶ 1355]

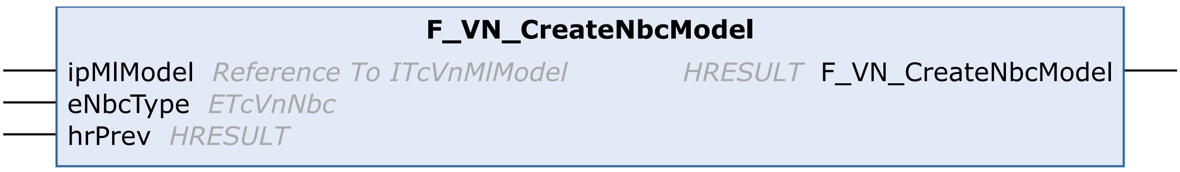
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.11 F_VN_CreateNbcModel



Create a normal Bayes classifier of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. In order to train normal Bayes classifiers, a sufficiently high number of samples is required for each class. It depends on the number of features and the distribution of the data. Hence, it needs to be tested for each application. Models of this type do not support on-line training (sample by sample). For the retraining of such classifier models, the set of presented classes must be identical to the previous learning steps. Otherwise, an exception is raised.

Syntax

Definition:

```
FUNCTION F_VN_CreateNbcModel : HRESULT
VAR_INPUT
    ipMlModel : Reference To ITcVnMlModel;
    eNbcType  : ETcVnNbc;
    hrPrev    : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▶ 399]	Returns the created model (Non-zero interface pointers are reused.)
eNbcType	ETcVnNbc [▶ 189]	Normal Bayes classifier type
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_CreateNbcModel 可创建一个一般贝叶斯分类器 (NBC) 模型。

一般贝叶斯分类器模型

一般贝叶斯分类器创建一个混合模型，其中每个类别均由高斯分布表示。这些高斯函数的参数在训练过程中确定。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

模型类型

eNbcType 指定 NBC 是用于分类（TCVN_NBC_CLASSIFIER）还是异常检测（TCVN_NBC_NOVELTY_DETECTOR）。

应用

例如，用于分类的 NBC 模型可如下创建：

```
hr := F_VN_CreateNbcModel(
    ipMlModel    := ipMlModel,
    eNbcType     := TCVN_NBC_CLASSIFIER,
    hrPrev       := hr);
```

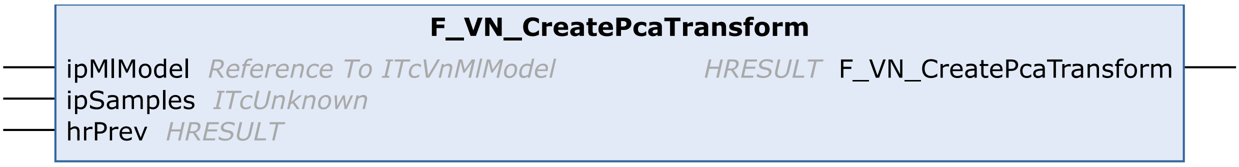
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.12 F_VN_CreatePcaTransform



Create a PCA-based feature transform from the provided data. The maximum number of principal components that can be computed equals the minimum of the number of samples and the number of features. The initial reference count is set to one if a new model is created and kept, otherwise.

Syntax

Definition:

```
FUNCTION F_VN_CreatePcaTransform : HRESULT
VAR_INPUT
    ipMlModel : Reference To ITcVnMlModel;
    ipSamples : ITcUnknown;
    hrPrev    : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▸ 399]	Returns the created feature transform (Non-zero interface pointers are reused.)
ipSamples	ITcUnknown [▸ 407]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_CreatePcaTransform 可创建主组件分析 (PCA) 模型。

主成分分析模型

在主成分分析中，要计算样本分布的主要组成成分。主成分是特征空间中显示样本方差最大的方向。通过将数据投影到选定主成分上，特征空间的维度会降低，其中尽可能保留了数据的方差。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

样本

所有样本均在一个容器中通过 ipSamples 发送至模型。

应用

例如，用于特征降维的 PCA 模型可如下创建：

```
hr := F_VN_CreatePcaTransform(
    ipMlModel    := ipMlModel,
    ipSamples    := ipSamples,
    hrPrev       := hr);
```

相关函数

- F_VN_CreatePcaTransformViaComponentNum [▶ 1354]
- F_VN_CreatePcaTransformViaVariance [▶ 1355]
- F_VN_CreateLdaTransform [▶ 1348]
- F_VN_CreateLdaTransformViaComponentNum [▶ 1350]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.13 F_VN_CreatePcaTransformViaComponentNum

Create a PCA-based feature transform from the provided data where the number of prinicpal components to be used is predefined. The maximum number of principal components that can be computed equals the minimum of the number of samples and the number of features. The initial reference count is set to one if a new model is created and kept, otherwise. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_CreatePcaTransformViaComponentNum : HRESULT
VAR_INPUT
    ipMlModel     : Reference To ITcVnMlModel;
    ipSamples     : ITcUnknown;
    nComponentNum : UDINT;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▶ 399]	Returns the created feature transform (Non-zero interface pointers are reused.)
ipSamples	ITcUnknown [▶ 407]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
nComponentNum	UDINT	Number of principal components (A value of 0 results in the automatic computation of the maximum number of components.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_CreatePcaTransformViaComponentNum 是F_VN_CreatePcaTransform [▶ 1353] 的变体。它包含一个附加参数 nComponentNum，用于提前定义特征降维的主成分数量。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

样本

所有样本均在一个容器中通过 ipSamples 发送至模型。

组件数量

PCA 计算 nComponentNum 主成分。如为 0，将自动计算成分的最大数量。

应用

例如，一个有 5 个主成分的 PCA 模型可如下创建：

```
hr := F_VN_CreatePcaTransformViaComponentNum(
    ipMlModel       := ipMlModel,
    ipSamples       := ipSamples,
    nComponentNum   := 5,
    hrPrev          := hr);
```

相关函数

- F_VN_CreatePcaTransform [▶ 1353]
- F_VN_CreatePcaTransformViaVariance [▶ 1355]
- F_VN_CreateLdaTransform [▶ 1348]
- F_VN_CreateLdaTransformViaComponentNum [▶ 1350]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.14 F_VN_CreatePcaTransformViaVariance

Create a PCA-based feature transform from the provided data based on a given fraction of variance to be retained. The maximum number of principal components that can be computed equals the minimum of the number of samples and the number of features. The initial reference count is set to one if a new model is created and kept, otherwise. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_CreatePcaTransformViaVariance : HRESULT
VAR_INPUT
    ipMlModel          : Reference To ITcVnMlModel;
    ipSamples          : ITcUnknown;
    fRetainedVariance  : LREAL;
    hrPrev             : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▶ 399]	Returns the created feature transform (Non-zero interface pointers are reused.)
ipSamples	ITcUnknown [▶ 407]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
fRetainedVariance	LREAL	Fraction of variance that is to be retained by the PCA (A value of 1.0 signifies 100%.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_CreatePcaTransformViaVariance 是F_VN_CreatePcaTransform [▶ 1353] 的变体。它包含用于定义保留方差的附加参数 fRetainedVariance。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

样本

所有样本均在一个容器中通过 ipSamples 发送至模型。

保留方差

在 PCA 中，选择主成分以使方差比例 fRetainedVariance 保留在样本中。如为 1，保留整个方差，因此特征数量不会减少。

应用

例如，保留方差为 65% 的 PCA 模型可如下创建：

```
hr := F_VN_CreatePcaTransformViaComponentNum(
    ipMlModel            := ipMlModel,
    ipSamples            := ipSamples,
    fRetainedVariance    := 0.65,
    hrPrev               := hr);
```

相关函数

- F_VN_CreatePcaTransform [▶ 1353]
- F_VN_CreatePcaTransformViaComponentNum [▶ 1354]
- F_VN_CreateLdaTransform [▶ 1348]

- F_VN_CreateLdaTransformViaComponentNum [▶ 1350]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.15 F_VN_CreateRTreesModel

Create a random trees model of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type neither support on-line training (sample by sample) nor retraining. Predictions can only be scalar.

Syntax

Definition:

```
FUNCTION F_VN_CreateRTreesModel : HRESULT
VAR_INPUT
    ipMlModel   : Reference To ITcVnMlModel;
    eRTreesType : ETcVnRTrees;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▶ 399]	Returns the created model (Non-zero interface pointers are reused.)
eRTreesType	ETcVnRTrees [▶ 195]	Random trees model type
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_CreateRTreesModel 可创建随机森林（RTrees）模型。

随机森林模型

随机森林模型使用决策树，其分支由随机过程决定。RTrees 模型的预测根据单个决策树的结果确定。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

模型类型

eRTreesType 指定随机森林模型是用于分类（TCVN_RT_CLASSIFIER）还是回归（TCVN_RT_REGRESSOR）。

专家参数

专家变体 F_VN_CreateRTreesModelExp [▶ 1358] 和 F_VN_CreateRTreesModelExp2 [▶ 1360] 包含附加参数。

应用

例如，用于回归的 RTrees 模型可如下创建：

```
hr := F_VN_CreateRTreesModel(
    ipMlModel    := ipMlModel,
    eRTreesType  := TCVN_RT_REGRESSOR,
    hrPrev       := hr);
```

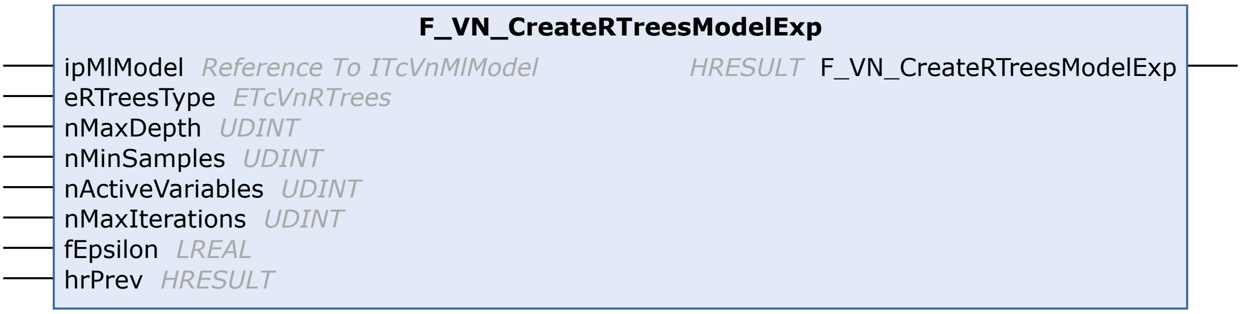
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.16 F_VN_CreateRTreesModelExp



Create a random trees model of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type neither support on-line training (sample by sample) nor retraining. Predictions can only be scalar. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_CreateRTreesModelExp : HRESULT
VAR_INPUT
    ipMlModel        : Reference To ITcVnMlModel;
    eRTreesType      : ETcVnRTrees;
    nMaxDepth        : UDINT;
    nMinSamples      : UDINT;
    nActiveVariables : UDINT;
    nMaxIterations   : UDINT;
    fEpsilon         : LREAL;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▶ 399]	Returns the created model (Non-zero interface pointers are reused.)
eRTreesType	ETcVnRTrees [▶ 195]	Random trees model type
nMaxDepth	UDINT	Maximum tree depth (default: 5)
nMinSamples	UDINT	Minimum number of samples within a node required for splitting (default: 10)
nActiveVariables	UDINT	Number of variables considered for splitting (0 means sqrt(total number of variables); default: 0)
nMaxIterations	UDINT	Maximum number of iterations (disabled if it equals 0 and fEpsilon is different from 0.0; triggers the usage of the default value of 50 if nMaxIterations and fEpsilon equal 0)
fEpsilon	LREAL	Maximum allowed difference of the error between two successive iterations (disabled if it equals 0.0 and nMaxIterations is different from 0; triggers the usage of the default value of 0.1 if nMaxIterations and fEpsilon equal 0)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_CreateRTReesModelExp 是F_VN_CreateRTReesModel [▶ 1357] 的专家变体。它包含额外的参数。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

模型类型

eRTreesType 指定随机森林模型是用于分类（TCVN_RT_CLASSIFIER）还是回归（TCVN_RT_REGRESSOR）。

最大树深度

nMaxDepth 表示树中决策层的最大数量。

节点内样本的最少数量

nMinSamples 是树形成过程中节点内必须保留的最小样本数。低于这一数字的拆分不会进行。

拆分特征

nActiveVariables 是用于拆分树的特征数量。如为 0，sqrt(N) 特征与所有可用特征数 N 一起使用。

最大迭代

优化最多使用 nMaxIterations 中指定的迭代次数。如果值为 0，则使用相应的默认值。

终止限制

只要两次迭代之间的误差变化不超过 fEpsilon 中的规定，优化就会终止。如果值为 0，则使用相应的默认值。

应用

例如，用于回归的 RTrees 模型可如下创建：

```
hr := F_VN_CreateRTreesModelExp(
    ipMlModel            := ipMlModel,
    eRTreesType          := TCVN_RT_REGRESSOR,
    nMaxDepth            := 8,
    nMinSamples          := 4,
    nActiveVariables     := 0,
    nMaxIterations       := 0,
    fEpsilon             := 0,
    hrPrev               := hr);
```

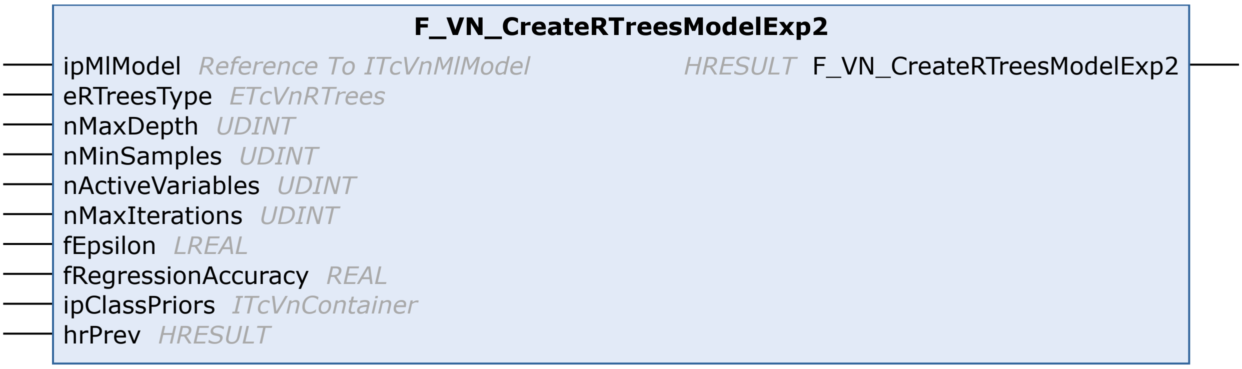
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.17 F_VN_CreateRTreesModelExp2



Create a random trees model of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type neither support on-line training (sample by sample) nor retraining. Predictions can only be scalar. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_CreateRTreesModelExp2 : HRESULT
VAR_INPUT
    ipMlModel            : Reference To ITcVnMlModel;
    eRTreesType          : ETcVnRTrees;
    nMaxDepth            : UDINT;
    nMinSamples          : UDINT;
    nActiveVariables     : UDINT;
    nMaxIterations       : UDINT;
    fEpsilon             : LREAL;
    fRegressionAccuracy  : REAL;
    ipClassPriors        : ITcVnContainer;
    hrPrev               : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▸ 399]	Returns the created model (Non-zero interface pointers are reused.)
eRTreesType	ETcVnRTrees [▸ 195]	Random trees model type
nMaxDepth	UDINT	Maximum tree depth (default: 5)
nMinSamples	UDINT	Minimum number of samples within a node required for splitting (default: 10)
nActiveVariables	UDINT	Number of variables considered for splitting (0 means sqrt(total number of variables); default: 0)
nMaxIterations	UDINT	Maximum number of iterations (disabled if it equals 0 and fEpsilon is different from 0.0; triggers the usage of the default value of 50 if nMaxIterations and fEpsilon equal 0)
fEpsilon	LREAL	Maximum allowed difference of the error between two successive iterations (disabled if it equals 0.0 and nMaxIterations is different from 0; triggers the usage of the default value of 0.1 if nMaxIterations and fEpsilon equal 0)
fRegressionAccuracy	REAL	Termination criterion for regressors (only for regressors; set to default if not allowed; default: 0.0)
ipClassPriors	ITcVnContainer [▸ 349]	Class priors (ContainerType_Vector_REAL or ContainerType_Vector_LREAL; only for classifiers; optional, set to 0 if not required or not allowed; default: 0)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_CreateRTReesModelExp2 是F_VN_CreateRTreesModel [▸ 1357] 的专家变体。它包含额外的参数。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

模型类型

eRTreesType 指定随机森林模型是用于分类（TCVN_RT_CLASSIFIER）还是回归（TCVN_RT_REGRESSOR）。

最大树深度

nMaxDepth 表示树中决策层的最大数量。

节点内样本的最少数量

nMinSamples 是树形成过程中节点内必须保留的最小样本数。低于这一数字的拆分不会进行。

拆分特征

nActiveVariables 是用于拆分树的特征数量。如为 0，sqrt(N) 特征与所有可用特征数 N 一起使用。

最大迭代

优化最多使用 nMaxIterations 中指定的迭代次数。如果值为 0，则使用相应的默认值。

终止限制

只要两次迭代之间的误差变化不超过 fEpsilon 中的规定，优化就会终止。如果值为 0，则使用相应的默认值。

回归精度

fRegressionAccuracy 是回归优化的终止准则。对于分类，该值必须设置为 0。

类别先验

ipClassPriors 是一个容器，可用来指定单个类别的先验概率。对于回归，该值应设置为 0。

应用

例如，用于回归的 RTrees 模型可如下创建：

```
hr := F_VN_CreateRTreesExp2(
    ipMlModel            := ipMlModel,
    eRTreesType          := TCVN_RT_REGRESSOR,
    nMaxDepth            := 8,
    nMinSamples          := 4,
    nActiveVariables     := 0,
    nMaxIterations       := 0,
    fEpsilon             := 0,
    fRegressionAccuracy := 0.3,
    ipClassPriors        := 0,
    hrPrev               := hr);
```

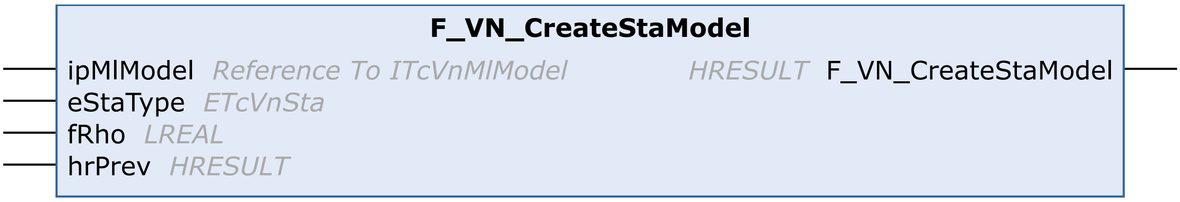
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.18 F_VN_CreateStaModel



Create a Simplified TopoART neural network of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type support on-line training (sample by sample), retraining, as well as scalar and vectorial predictions. It requires all input except class labels (i.e., samples and training outputs/predictions) to lie in the interval [0.0, 1.0]. The predictions of regressors need to be rescaled from the interval [0.0, 1.0] to their respective value range before usage. Depending on the parameter settings and the number of available training samples, repeated training with the same data may improve the results. Like other neural networks based on the Adaptive Resonance Theory (ART), Simplified TopoART neural networks are not prone to catastrophic inference and patricularly well-suited to incremental learning tasks.

Syntax

Definition:

```
FUNCTION F_VN_CreateStaModel : HRESULT
VAR_INPUT
    ipMlModel : Reference To ITcVnMlModel;
```

```
    eStaType   : ETcVnSta;
    fRho       : LREAL;
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▸ 399]	Returns the created model (Non-zero interface pointers are reused.)
eStaType	ETcVnSta [▸ 197]	Simpified TopoART model type
fRho	LREAL	Vigilance parameter (controls the number of neurons that are inserted and the maximum size of the formed categories; valid range: [0.0, 1.0]; suggested range: [0.8, 1.0))
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_CreateStaModel 可创建简化 TopoART（STA）模型。

简化 TopoART 模型

简化 TopoART 网络学习特征空间的子区域。根据不同的应用，它们会被分组为不同的聚类、分配到不同的类别或用于预测。由于增量结构，可进行后期培训，从而以最佳可能方式保留已学习信息。只接受 0.0 和 1.0 之间的数值作为输入（类别名称除外）。因此，建议使用特征归一化。

模型

创建的模型将在接口指针 ipMlModel 中返回。

模型类型

eStaType 指定 STA 模型是用于分类（TCVN_STA_CLASSIFIER）、回归（TCVN_STA_REGRESSOR）、聚类（TCVN_STA_CLUSTERER）还是异常检测（TCVN_STA_NOVELTY_DETECTOR）。

警戒参数

该参数对调整模型以适应相应任务至关重要，因为它控制着要学习的子区域的最大尺寸。

专家参数

专家变体 F_VN_CreateStaModelExp [▸ 1364] 和 F_VN_CreateStaModelExp2 [▸ 1366] 包含附加参数。

应用

例如，用于分类的 STA 模型可如下创建：

```
hr := F_VN_CreateStaModel(
    ipMlModel    := ipMlModel,
    eStaType     := TCVN_STA_CLASSIFIER,
    fRho         := 0.9,
    hrPrev       := hr);
```

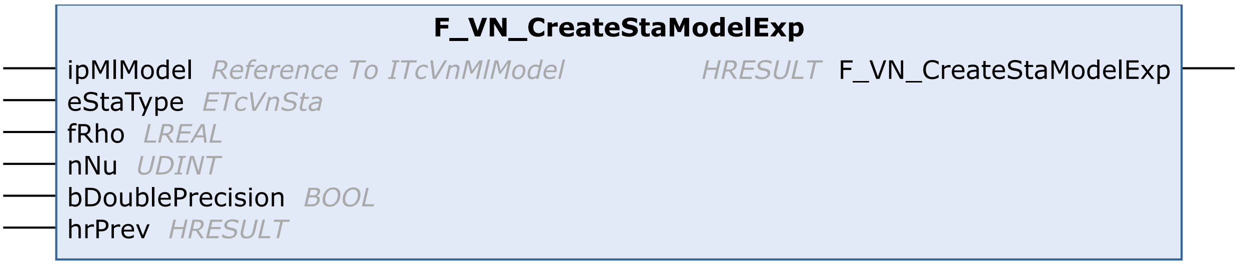
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.19 F_VN_CreateStaModelExp



Create a Simplified TopoART neural network of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type support on-line training (sample by sample), retraining, as well as scalar and vectorial predictions. It requires all input except class labels (i.e., samples and training outputs/predictions) to lie in the interval [0.0, 1.0]. The predictions of regressors need to be rescaled from the interval [0.0, 1.0] to their respective value range before usage. Depending on the parameter settings and the number of available training samples, repeated training with the same data may improve the results. Like other neural networks based on the Adaptive Resonance Theory (ART), Simplified TopoART neural networks are not prone to catastrophic inference and patricularly well-suited to incremental learning tasks. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_CreateStaModelExp : HRESULT
VAR_INPUT
    ipMlModel        : Reference To ITcVnMlModel;
    eStaType         : ETcVnSta;
    fRho             : LREAL;
    nNu              : UDINT;
    bDoublePrecision : BOOL;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▸ 399]	Returns the created model (Non-zero interface pointers are reused.)
eStaType	ETcVnSta [▸ 197]	Simpified TopoART model type
fRho	LREAL	Vigilance parameter (controls the number of neurons that are inserted and the maximum size of the formed categories; valid range: [0.0, 1.0]; suggested range: [0.8, 1.0))
nNu	UDINT	Number of neurons used for classification and prediction (must be larger than or equal to 1; suggested range: [1, 10); default: 3)
bDoublePrecision	BOOL	If true, the model is generated with double precision (LREAL) instead of single precision (REAL). (default: FALSE)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_CreateStaModelExp 是F_VN_CreateStaModel [▶ 1362] 的专家变体。它包含额外的参数。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

模型类型

eStaType 指定 STA 模型是用于分类（TCVN_STA_CLASSIFIER）、回归（TCVN_STA_REGRESSOR）、聚类（TCVN_STA_CLUSTERER）还是异常检测（TCVN_STA_NOVELTY_DETECTOR）。

警戒参数

该参数对调整模型以适应相应任务至关重要，因为它控制着要学习的子区域的最大尺寸。

使用的邻算法数量

nNu 指定用于分类和预测的相邻子区域的数量，且必须大于等于 1。

更高精度

如果 bDoublePrecision 是 TRUE，则使用 LREAL 作为模型内部计算的数据类型；如果是 FALSE，则使用 REAL。

应用

例如，用于分类的 STA 模型可如下创建：

```
hr := F_VN_CreateStaModel(
    ipMlModel          := ipMlModel,
    eStaType           := TCVN_STA_CLASSIFIER,
    fRho               := 0.9,
    nNu                := 3,
    bDoublePrecision := FALSE,
    hrPrev             := hr);
```

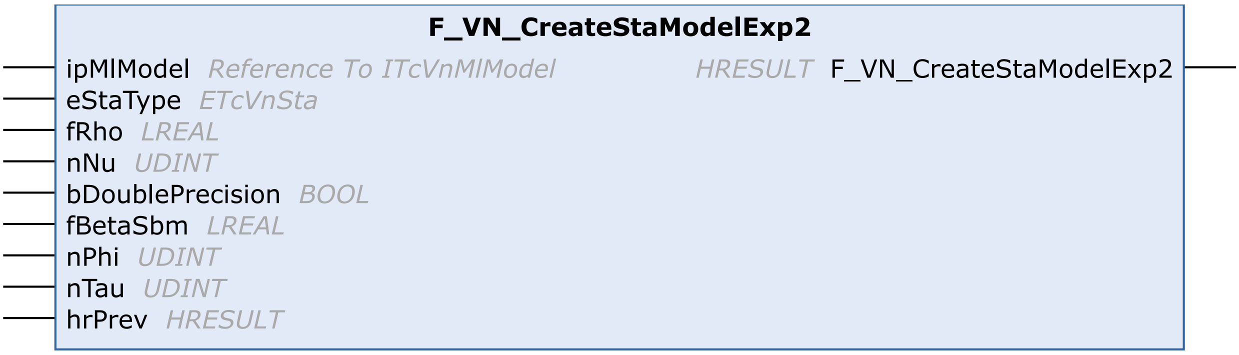
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.20 F_VN_CreateStaModelExp2



Create a Simplified TopoART neural network of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type support on-line training (sample by sample), retraining, as well as scalar and vectorial predictions. It requires all input except class labels (i.e., samples and training outputs/predictions) to lie in the interval [0.0, 1.0]. The predictions of regressors need to be rescaled from the interval [0.0, 1.0] to their respective value range before usage. Depending on the parameter settings and the number of available training samples, repeated training with the same data may improve the results. Like other neural networks based on the Adaptive Resonance Theory (ART), Simplified TopoART neural networks are not prone to catastrophic inference and patricularly well-suited to incremental learning tasks. (additional expert function providing parameters for fine-tuning and noise reduction)

Syntax

Definition:

```
FUNCTION F_VN_CreateStaModelExp2 : HRESULT
VAR_INPUT
    ipMlModel        : Reference To ITcVnMlModel;
    eStaType         : ETcVnSta;
    fRho             : LREAL;
    nNu              : UDINT;
    bDoublePrecision : BOOL;
    fBetaSbm         : LREAL;
    nPhi             : UDINT;
    nTau             : UDINT;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▶ 399]	Returns the created model (Non-zero interface pointers are reused.)
eStaType	ETcVnSta [▶ 197]	Simpified TopoART model type
fRho	LREAL	Vigilance parameter (controls the number of neurons that are inserted and the maximum size of the formed categories; valid range: [0.0, 1.0]; suggested range: [0.8, 1.0))
nNu	UDINT	Number of neurons used for classification and prediction (must be larger than or equal to 1; suggested range: [1, 10); default: 3)
bDoublePrecision	BOOL	If true, the model is generated with double precision (LREAL) instead of single precision (REAL). (default: FALSE)
fBetaSbm	LREAL	Learning rate of the second best-matching neuron (learning the second best-matching neuron keeps related categories closer together and might improve the results but may require a higher number of neurons and training runs; a value of 0.0 disables learning of the second best-matching neuron; valid range: [0.0, 1.0]; suggested range: [0.0, 0.5]); default: 0.0
nPhi	UDINT	Number of samples a neuron must have learnt to become permanent (required for noise reduction; works in conjunction with nTau; must be larger than or equal to 1; higher values intensify noise reduction; a value of 1 disables removal of neuron candidates; default: 1)
nTau	UDINT	Number of learning steps after which node removal is performed (required for noise reduction; works in conjunction with nPhi; must be large enough to allow neuron candidates representing non-noise samples to become the best-matching neuron at least nPhi times before node removal; default: 100)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_CreateStaModelExp2 是F_VN_CreateStaModel [▶ 1362] 的专家变体。它包含额外的参数。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

模型类型

eStaType 指定 STA 模型是用于分类（TCVN_STA_CLASSIFIER）、回归（TCVN_STA_REGRESSOR）、聚类（TCVN_STA_CLUSTERER）还是异常检测（TCVN_STA_NOVELTY_DETECTOR）。

警戒参数

该参数对调整模型以适应相应任务至关重要，因为它控制着要学习的子区域的最大尺寸。

使用的邻算法数量

nNu 指定用于分类和预测的相邻子区域的数量，且必须大于等于 1。

更高精度

如果 bDoublePrecision 是 TRUE，则使用 LREAL 作为模型内部计算的数据类型；如果是 FALSE，则使用 REAL。

调整分配

fBetaSbm 补充根据基本样本分配调整子区域的控制。

噪音抑制

nPhi 和 nTau 用于控制学习过程中对极少数样本（噪声、异常值）的抑制。

nPhi 指定在一个子区域内必须学习的样本数，以便永久考虑该子区域。

nTau 指定在多少个学习步骤后检查 nPhi 的条件，并删除不满足该条件的子区域

增加 nPhi 和减少 nTau 可以增加噪音抑制，因此 nPhi 与 nTau 的比值可用于估算强度。

应用

例如，用于分类的 STA 模型可如下创建：

```
hr := F_VN_CreateStaModel(
    ipMlModel         := ipMlModel,
    eStaType          := TCVN_STA_CLASSIFIER,
    fRho              := 0.9,
    nNu               := 3,
    bDoublePrecision  := FALSE,
    fBetaSbm          := 0.0,
    nPhi              := 1,
    nTau              := 100,
    hrPrev            := hr);
```

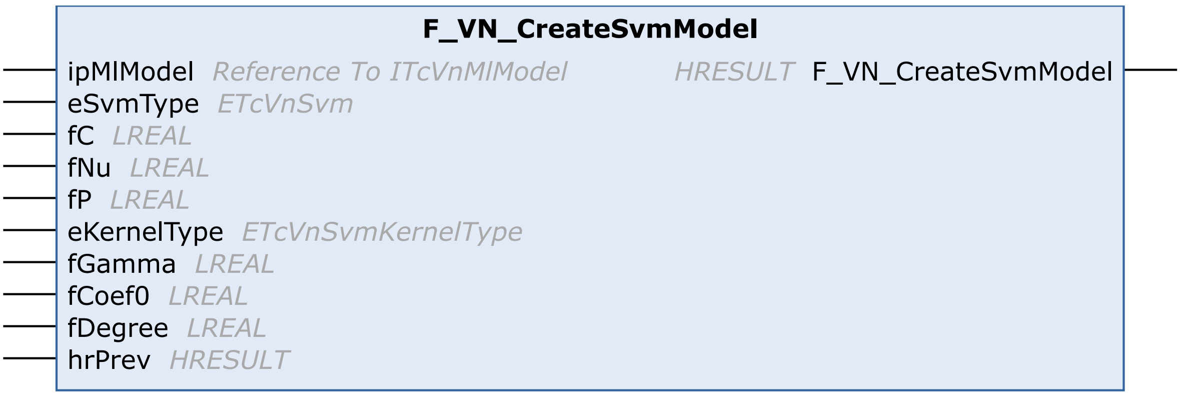
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.21 F_VN_CreateSvmModel



Create an SVM model of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type neither support on-line training (sample by sample) nor retraining. Predictions can only be scalar.

Syntax

Definition:

```
FUNCTION F_VN_CreateSvmModel : HRESULT
VAR_INPUT
    ipMlModel    : Reference To ITcVnMlModel;
    eSvmType     : ETcVnSvm;
    fC           : LREAL;
    fNu          : LREAL;
    fP           : LREAL;
    eKernelType  : ETcVnSvmKernelType;
    fGamma       : LREAL;
    fCoef0       : LREAL;
    fDegree      : LREAL;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▶ 399]	Returns the created model (Non-zero interface pointers are reused.)
eSvmType	ETcVnSvm [▶ 198]	SVM model type
fC	LREAL	Parameter C (required for TCVN_SVM_C_CLASSIFIER, TCVN_SVM_EPS_REGRESSOR, and TCVN_SVM_NU_REGRESSOR; ignored otherwise)
fNu	LREAL	Parameter nu (required for TCVN_SVM_NU_CLASSIFIER, TCVN_SVM_NOVELTY_DETECTOR, and TCVN_SVM_NU_REGRESSOR; ignored otherwise)
fP	LREAL	Parameter p (required for TCVN_SVM_EPS_REGRESSOR; ignored otherwise)
eKernelType	ETcVnSvmKernelType [▶ 198]	Kernel type
fGamma	LREAL	Parameter gamma (used by polynomial, RBF, sigmoid, and chi-squared kernels; ignored otherwise)
fCoef0	LREAL	parameter coef0 (used by polynomial and sigmoid kernels; ignored otherwise)
fDegree	LREAL	Degree (used by polynomial kernels; ignored otherwise)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_CreateSvmModel 可创建支持向量机（SVM）模型。

支持向量机模型

支持向量机还可用于复杂的非线性问题。预测完全基于所谓的支持向量，这就是 SVM 模型内存效率相对较高的原因。需要同时对所有样本进行训练（批量训练），无法进行后期训练。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

模型类型

eSvmType 指定 SVM 模型是用于分类、回归还是异常检测：

- TCVN_SVM_C_CLASSIFIER
- TCVN_SVM_NU_CLASSIFIER
- TCVN_SVM_NOVELTY_DETECTOR
- TCVN_SVM_EPS_REGRESSOR
- TCVN_SVM_NU_REGRESSOR

Model parameters

参数 fC、fNu 和 fP 的使用和含义取决于所选的 eSvmType。

内核类型

用于计算的内核类型通过 eKernelType 进行定义。内核类型取决于任务/数据分布，且必须进行相应调整。

内核参数

参数 fGamma、fCoef0 和 fDegree 的使用和含义取决于所选的 eKernelType。

专家参数

专家变体 F_VN_CreateSvmModelExp [▶ 1371] 和 F_VN_CreateSvmModelExp2 [▶ 1373] 包含附加参数。

应用

例如，用于分类的 SVM 模型可如下创建：

```
hr := F_VN_CreateSvmModel(
    ipMlModel    := ipMlModel,
    eSvmType     := TCVN_SVM_NU_CLASSIFIER,
    fC           := 0,
    fNu          := 0.1,
    fP           := 0,
    eKernelType := TCVN_SKT_RBF,
    fGamma       := 1,
    fCoef0       := 0,
    fDegree      := 0,
    hrPrev       := hr);
```

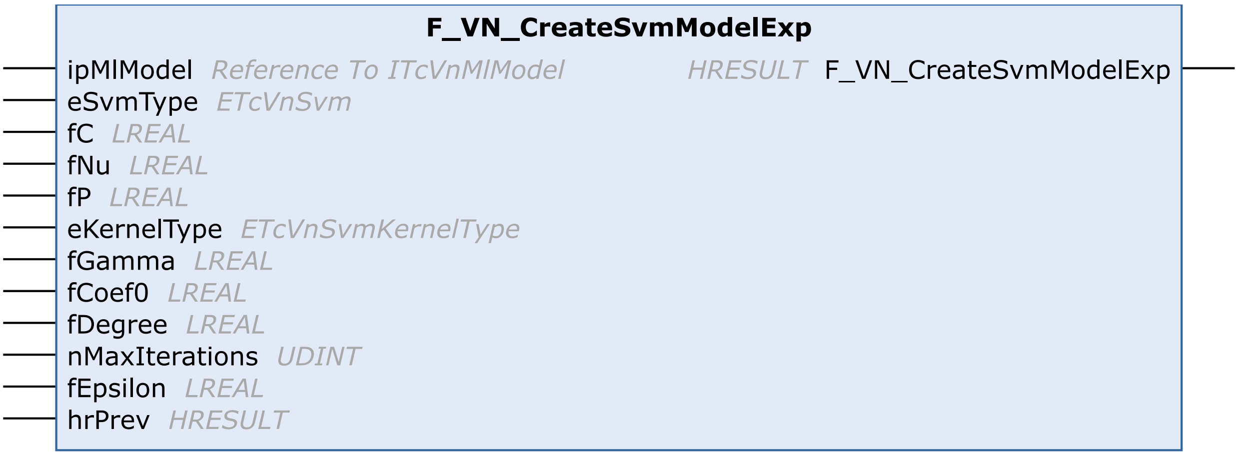
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.22 F_VN_CreateSvmModelExp



Create an SVM model of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type neither support on-line training (sample by sample) nor retraining. Predictions can only be scalar. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_CreateSvmModelExp : HRESULT
VAR_INPUT
    ipMlModel      : Reference To ITcVnMlModel;
    eSvmType       : ETcVnSvm;
    fC             : LREAL;
    fNu            : LREAL;
    fP             : LREAL;
    eKernelType    : ETcVnSvmKernelType;
    fGamma         : LREAL;
    fCoef0         : LREAL;
    fDegree        : LREAL;
    nMaxIterations : UDINT;
    fEpsilon       : LREAL;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▶ 399]	Returns the created model (Non-zero interface pointers are reused.)
eSvmType	ETcVnSvm [▶ 198]	SVM model type
fC	LREAL	Parameter C (required for TCVN_SVM_C_CLASSIFIER, TCVN_SVM_EPS_REGRESSOR, and TCVN_SVM_NU_REGRESSOR; ignored otherwise)
fNu	LREAL	Parameter nu (required for TCVN_SVM_NU_CLASSIFIER, TCVN_SVM_NOVELTY_DETECTOR, and TCVN_SVM_NU_REGRESSOR; ignored otherwise)
fP	LREAL	Parameter p (required for TCVN_SVM_EPS_REGRESSOR; ignored otherwise)
eKernelType	ETcVnSvmKernelType [▶ 198]	Kernel type
fGamma	LREAL	Parameter gamma (used by polynomial, RBF, sigmoid, and chi-squared kernels; ignored otherwise)
fCoef0	LREAL	Parameter coef0 (used by polynomial and sigmoid kernels; ignored otherwise)
fDegree	LREAL	Degree (used by polynomial kernels; ignored otherwise)
nMaxIterations	UDINT	Maximum number of iterations (disabled if it equals 0 and fEpsilon is different from 0.0; triggers the usage of the default value of 100000 if nMaxIterations and fEpsilon equal 0)
fEpsilon	LREAL	Maximum allowed difference of the error between two successive iterations (disabled if it equals 0.0 and nMaxIterations is different from 0; triggers the usage of the default value of 0.00001 if nMaxIterations and fEpsilon equal 0)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_CreateSvmModelExp 是F_VN_CreateSvmModel [▶ 1368] 的专家变体。它包含额外的参数。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

模型类型

eSvmType 指定 SVM 模型是用于分类、回归还是异常检测：

- TCVN_SVM_C_CLASSIFIER
- TCVN_SVM_NU_CLASSIFIER
- TCVN_SVM_NOVELTY_DETECTOR
- TCVN_SVM_EPS_REGRESSOR
- TCVN_SVM_NU_REGRESSOR

Model parameters

参数 fC、fNu 和 fP 的使用和含义取决于所选的 eSvmType。

内核类型

用于计算的内核类型通过 eKernelType 进行定义。内核类型取决于任务/数据分布，且必须进行相应调整。

内核参数

参数 fGamma、fCoef0 和 fDegree 的使用和含义取决于所选的 eKernelType。

最大迭代

优化最多使用 nMaxIterations 中指定的迭代次数。如果值为 0，则使用相应的默认值。

终止限制

只要两次迭代之间的误差变化不超过 fEpsilon 中的规定，优化就会终止。如果值为 0，则使用相应的默认值。

应用

例如，用于分类的 SVM 模型可如下创建：

```
hr := F_VN_CreateSvmModelExp(
    ipMlModel        := ipMlModel,
    eSvmType         := TCVN_SVM_NU_CLASSIFIER,
    fC               := 0,
    fNu              := 0.1,
    fP               := 0,
    eKernelType      := TCVN_SKT_RBF,
    fGamma           := 1,
    fCoef0           := 0,
    fDegree          := 0,
    nMaxIterations   := 0,
    fEpsilon         := 0,
    hrPrev           := hr);
```

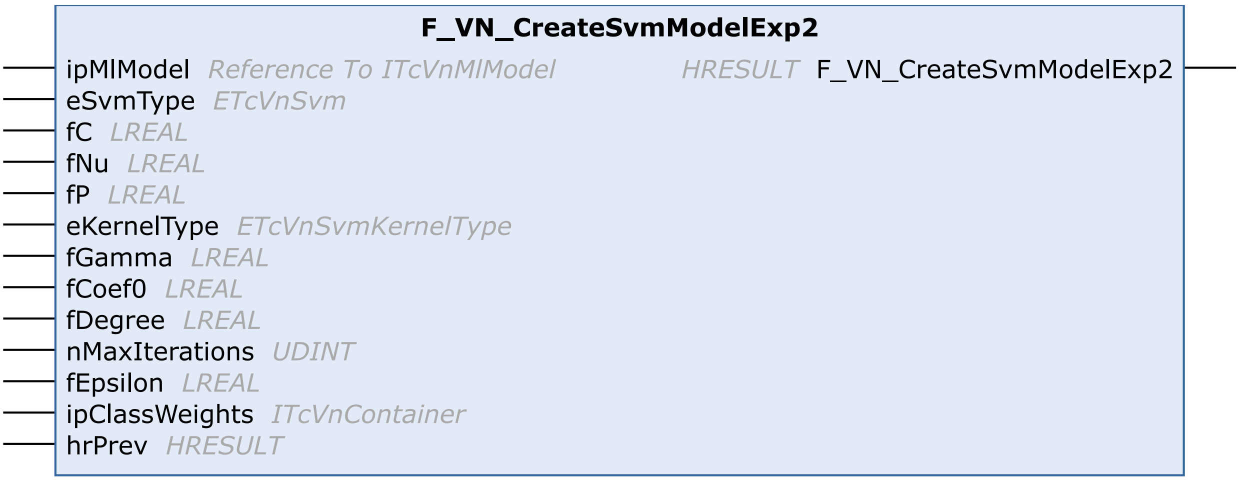
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.23 F_VN_CreateSvmModelExp2



Create an SVM model of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type neither support on-line training (sample by sample) nor retraining. Predictions can only be scalar. (additional expert function for C support vector classifiers)

Syntax

Definition:

```
FUNCTION F_VN_CreateSvmModelExp2 : HRESULT
VAR_INPUT
    ipMlModel      : Reference To ITcVnMlModel;
    eSvmType       : ETcVnSvm;
    fC             : LREAL;
    fNu            : LREAL;
    fP             : LREAL;
    eKernelType    : ETcVnSvmKernelType;
    fGamma         : LREAL;
    fCoef0         : LREAL;
    fDegree        : LREAL;
    nMaxIterations : UDINT;
    fEpsilon       : LREAL;
    ipClassWeights : ITcVnContainer;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▶ 399]	Returns the created model (Non-zero interface pointers are reused.)
eSvmType	ETcVnSvm [▶ 198]	SVM model type
fC	LREAL	Parameter C (required for TCVN_SVM_C_CLASSIFIER, TCVN_SVM_EPS_REGRESSOR, and TCVN_SVM_NU_REGRESSOR; ignored otherwise)
fNu	LREAL	Parameter nu (required for TCVN_SVM_NU_CLASSIFIER, TCVN_SVM_NOVELTY_DETECTOR, and TCVN_SVM_NU_REGRESSOR; ignored otherwise)
fP	LREAL	Parameter p (required for TCVN_SVM_EPS_REGRESSOR; ignored otherwise)
eKernelType	ETcVnSvmKernelType [▶ 198]	Kernel type
fGamma	LREAL	Parameter gamma (used by polynomial, RBF, sigmoid, and chi-squared kernels; ignored otherwise)
fCoef0	LREAL	Parameter coef0 (used by polynomial and sigmoid kernels; ignored otherwise)
fDegree	LREAL	Degree (used by polynomial kernels; ignored otherwise)
nMaxIterations	UDINT	Maximum number of iterations (disabled if it equals 0 and fEpsilon is different from 0.0; triggers the usage of the default value of 100000 if nMaxIterations and fEpsilon equal 0)
fEpsilon	LREAL	Maximum allowed difference of the error between two successive iterations (disabled if it equals 0.0 and nMaxIterations is different from 0; triggers the usage of the default value of 0.00001 if nMaxIterations and fEpsilon equal 0)
ipClassWeights	ITcVnContainer [▶ 349]	Class weights (ContainerType_Vector_REAL or ContainerType_Vector_LREAL; only valid if eSvmType equals TCVN_SVM_C_CLASSIFIER; optional, set to 0 if not required or not allowed; default: 0)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_CreateSvmModelExp2 是F_VN_CreateSvmModel [▶ 1368] 的专家变体。它包含只与 TCVN_SVM_C_CLASSIFIER 类型模型相关的附加参数。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

模型类型

eSvmType 指定 SVM 模型是用于分类、回归还是异常检测：

- TCVN_SVM_C_CLASSIFIER
- TCVN_SVM_NU_CLASSIFIER
- TCVN_SVM_NOVELTY_DETECTOR
- TCVN_SVM_EPS_REGRESSOR
- TCVN_SVM_NU_REGRESSOR

Model parameters

参数 fC、fNu 和 fP 的使用和含义取决于所选的 eSvmType。

内核类型

用于计算的内核类型通过 eKernelType 进行定义。内核类型取决于任务/数据分布，且必须进行相应调整。

内核参数

参数 fGamma、fCoef0 和 fDegree 的使用和含义取决于所选的 eKernelType。

最大迭代

优化最多使用 nMaxIterations 中指定的迭代次数。如果值为 0，则使用相应的默认值。

终止限制

只要两次迭代之间的误差变化不超过 fEpsilon 中的规定，优化就会终止。如果值为 0，则使用相应的默认值。

分类权重

ipClassWeights 是一个容器，可用于指定单个类别的权重。

应用

例如，使用 Nu 进行参数设置的 SVM 分类模型可如下创建：

```
hr := F_VN_CreateSvmModelExp2(
    ipMlModel       := ipMlModel,
    eSvmType        := TCVN_SVM_C_CLASSIFIER,
    fC              := 100,
    fNu             := 0,
    fP              := 0,
    eKernelType     := TCVN_SKT_RBF,
    fGamma          := 1,
    fCoef0          := 0,
    fDegree         := 0,
    nMaxIterations  := 0,
    fEpsilon        := 0,
    ipClassWeights  := ipClassWeights,
    hrPrev          := hr);
```

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.24 F_VN_CreateSvmSgdClassifier

F_VN_CreateSvmSgdClassifier	
ipMlModel *Reference To ITcVnMlModel*	*HRESULT* F_VN_CreateSvmSgdClassifier
hrPrev *HRESULT*	

Create a linear SVM classifier using stochastic gradient descent for training. The initial reference count is set to one if a new model is created and kept, otherwise. This SVM classifier is only applicable to binary classification problems. It learns a separating hyperplane between a class with label -1 and a class with label 1. These class labels are predefined. For training, any positive class labels are mapped to 1 and any negative class labels are mapped to -1. Models of this type neither support on-line training (sample by sample) nor retraining.

Syntax

Definition:

```
FUNCTION F_VN_CreateSvmSgdClassifier : HRESULT
VAR_INPUT
    ipMlModel : Reference To ITcVnMlModel;
    hrPrev    : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▶ 399]	Returns the created model (Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_CreateSvmSgdClassifier 可创建一个带有可用于随机梯度下降（SVM-SGD）训练的线性内核的支持向量机模型。

SVM-SGD 模型

这种 SVM 分类器模型只适用于二元分类问题。它在标签为 -1 的类和标签为 1 的类之间学习一个分离超平面。所有样本均需要同时进行训练（批量训练），不可能进行后期训练。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

专家参数

专家变体 F_VN_CreateSvmSgdClassifierExp [▶ 1377] 包含附加参数。

应用

例如，用于分类的 SVM-SGD 模型可如下创建：

```
hr := F_VN_CreateSvmSgdClassifier(
    ipMlModel    := ipMlModel,
    hrPrev       := hr);
```

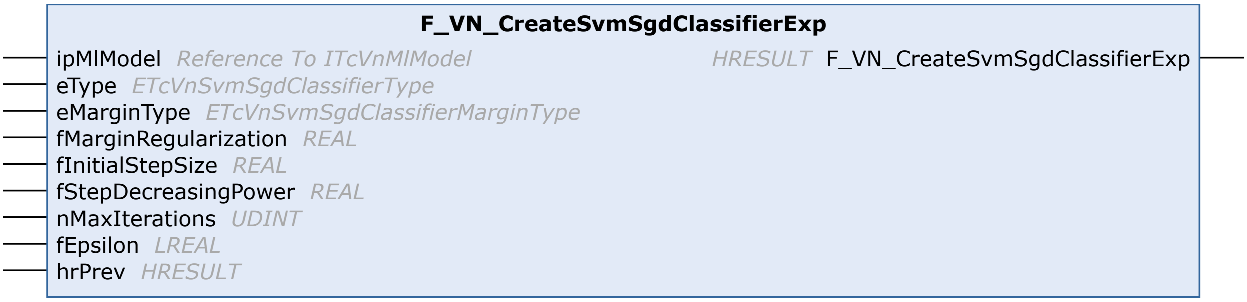
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.25 F_VN_CreateSvmSgdClassifierExp



Create a linear SVM classifier using stochastic gradient descent for training. The initial reference count is set to one if a new model is created and kept, otherwise. This SVM classifier is only applicable to binary classification problems. It learns a separating hyperplane between a class with label -1 and a class with label 1. These class labels are predefined. For training, any positive class labels are mapped to 1 and any negative class labels are mapped to -1. Models of this type neither support on-line training (sample by sample) nor retraining. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_CreateSvmSgdClassifierExp : HRESULT
VAR_INPUT
    ipMlModel              : Reference To ITcVnMlModel;
    eType                  : ETcVnSvmSgdClassifierType;
    eMarginType            : ETcVnSvmSgdClassifierMarginType;
    fMarginRegularization  : REAL;
    fInitialStepSize       : REAL;
    fStepDecreasingPower   : REAL;
    nMaxIterations         : UDINT;
    fEpsilon               : LREAL;
    hrPrev                 : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	Reference To ITcVnMlModel [▸ 399]	Returns the created model (Non-zero interface pointers are reused.)
eType	ETcVnSvmSgdClassifierType [▸ 199]	Learning algorithm type (default: TCVN_SSCT_ASGD)
eMarginType	ETcVnSvmSgdClassifierMarginType [▸ 199]	Margin type (default: TCVN_SSCMT_SOFT_MARGIN)
fMarginRegularization	REAL	Margin regularization parameter (default: 0.00001)
fInitialStepSize	REAL	Initial step size (default: 0.05)
fStepDecreasingPower	REAL	Power parameter (default: 0.75)
nMaxIterations	UDINT	Maximum number of iterations (disabled if it equals 0 and fEpsilon is different from 0.0; triggers the usage of the default value of 100000 if nMaxIterations and fEpsilon equal 0)
fEpsilon	LREAL	Maximum allowed difference of the error between two successive iterations (disabled if it equals 0.0 and nMaxIterations is different from 0; triggers the usage of the default value of 0.00001 if nMaxIterations and fEpsilon equal 0)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_CreateSvmSgdClassifierExp 是 F_VN_CreateSvmSgdClassifier [▸ 1376] 的专家变体。它包含额外的参数。

参数

模型

创建的模型将在接口指针 ipMlModel 中返回。

算法

eType 指定训练中使用的优化方法：

- TCVN_SSCT_SGD 随机梯度下降
- TCVN_SSCT_ASGD 平均随机梯度下降

分类界定

eMarginType 指定是严格（TCVN_SSCMT_HARD_MARGIN）还是使用离群值（TCVN_SSCMT_SOFT_MARGIN）进行类别界定。

正则化

fMarginRegularization 负责每一步的权重减少，并控制异常值影响的限制程度。参数越小，异常值被忽略的概率就越低。

步长

fInitialStepSize 表示培训开始时的步长。

减少步长

fStepDecreasingPower 指定用于减小步长的指数。

最大迭代

优化最多使用 nMaxIterations 中指定的迭代次数。如果值为 0，则使用相应的默认值。

终止限制

只要两次迭代之间的误差变化不超过 fEpsilon 中的规定，优化就会终止。如果值为 0，则使用相应的默认值。

应用

例如，用于分类的 SVM-SGD 模型可如下创建：

```
hr := F_VN_CreateSvmSgdClassifierExp(
    ipMlModel                   := ipMlModel,
    eType                       := TCVN_SSCT_ASGD,
    eMarginType                 := TCVN_SSCMT_SOFT_MARGIN,
    fMarginRegularization       := 0.00001,
    fInitialStepSize            := 0.05,
    fStepDecreasingPower        := 0.75,
    nMaxIterations              := 0,
    fEpsilon                    := 0,
    hrPrev                      := hr);
```

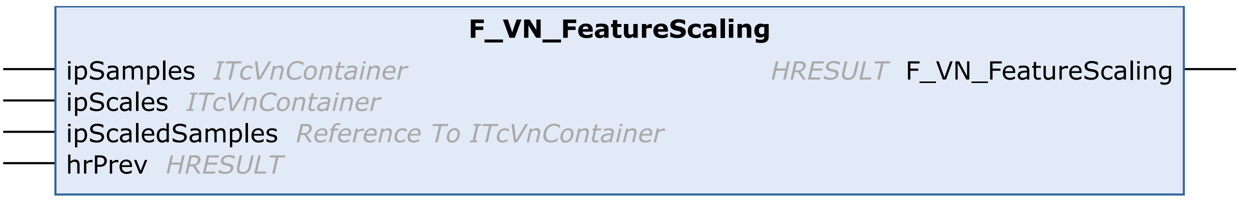
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.26 F_VN_FeatureScaling



Apply a feature scaling to one or more sample(s).

Syntax

Definition:

```
FUNCTION F_VN_FeatureScaling : HRESULT
VAR_INPUT
    ipSamples        : ITcVnContainer;
    ipScales         : ITcVnContainer;
    ipScaledSamples  : Reference To ITcVnContainer;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSamples	ITcVnContainer [▸ 349]	Container holding one or more input sample(s) (ContainerType_Vector_REAL, ContainerType_Vector_LREAL, ContainerType_Vector_Vector_REAL, or ContainerType_Vector_Vector_LREAL)
ipScales	ITcVnContainer [▸ 349]	Container with the scaling parameters of each feature (type ContainerType_Vector_REAL or ContainerType_Vector_LREAL).
ipScaledSamples	Reference To ITcVnContainer [▸ 349]	Returns the scaled sample(s) using the same type like ipSamples. If the same container ipSamples is uesd, the source data will be replaced.
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_FeatureScaling 对来自多个样本的特征进行缩放。函数 F_VN_InverseFeatureScaling [▸ 1397] 可用于反向缩放。

参数

样本

必须将一个或多个样本作为一个容器传输到 ipSamples。

缩放

必须将带有缩放类型和相应缩放参数的容器传输到 ipScales。容器可以使用函数 F_VN_GetFeatureScales [▸ 1391] 计算，也可以使用其中描述的结构创建。

缩放样本

缩放样本通过引用 ipScaledSamples 返回。

专家参数

专家变体 F_VN_FeatureScalingExp [▸ 1381] 包含附加参数。

应用

例如，F_VN_GetFeatureScales [▸ 1391] 使用MINMAX 方法事先计算缩放参数的特征缩放过程如下所示：

```
hr := F_VN_GetFeatureScales(ipSamples, ipScales, TCVN_FST_MINMAX, hr);

hr := F_VN_FeatureScaling(
    ipSamples       := ipSamples,
    ipScales        := ipScales,
    ipScaledSamples := ipScaledSamples,
    hrPrev          := hr);
```

相关函数

- F_VN_GetFeatureScales [▸ 1391]
- F_VN_InverseFeatureScaling [▸ 1397]
- F_VN_InverseFeatureScaling_REAL [▸ 1398]

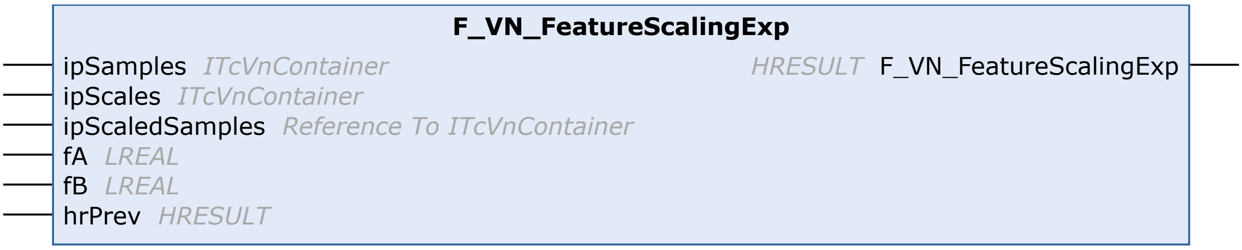
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.27 F_VN_FeatureScalingExp



Apply a feature scaling to one or more sample(s) and get output range between [fA, fB] in the case of TCVN_FST1_MINMAX. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_FeatureScalingExp : HRESULT
VAR_INPUT
    ipSamples        : ITcVnContainer;
    ipScales         : ITcVnContainer;
    ipScaledSamples : Reference To ITcVnContainer;
    fA               : LREAL;
    fB               : LREAL;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSamples	ITcVnContainer [▸ 349]	Container holding one or more input sample(s) (ContainerType_Vector_REAL, ContainerType_Vector_LREAL, ContainerType_Vector_Vector_REAL, or ContainerType_Vector_Vector_LREAL)
ipScales	ITcVnContainer [▸ 349]	Container with the scaling parameters of each feature (type ContainerType_Vector_REAL or ContainerType_Vector_LREAL).
ipScaledSamples	Reference To ITcVnContainer [▸ 349]	Returns the scaled sample(s) using the same type like ipSamples. If the same container ipSamples is uesd, the source data will be replaced.
fA	LREAL	represents the lower bound of the range for TCVN_FST1_MINMAX
fB	LREAL	represents the upper bound of the range for TCVN_FST1_MINMAX
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_FeatureScalingExp 是F_VN_FeatureScaling [▶ 1379] 的专家变体。它包含额外的参数。

参数

样本

必须将一个或多个样本作为一个容器传输到 ipSamples。

缩放

必须将带有缩放类型和相应缩放参数的容器传输到 ipScales。容器可以使用函数 F_VN_GetFeatureScales [▶ 1391] 计算，也可以使用其中描述的结构创建。

缩放样本

缩放样本通过引用 ipScaledSamples 返回。

定义输出范围

如果使用缩放类型 TCVN_FST1_MINMAX，还可以通过 fA 和 fB 指定输出值的值范围。

应用

例如，输出值范围为 0.1 至 0.9 的特征缩放如下所示：

```
hr := F_VN_FeatureScalingExp(
    ipSamples        := ipSamples,
    ipScales         := ipScales,
    ipScaledSamples := ipScaledSamples,
    fA               := 0.1,
    fB               := 0.9,
    hrPrev           := hr);
```

相关函数

- F_VN_GetFeatureScales [▶ 1391]
- F_VN_InverseFeatureScaling [▶ 1397]
- F_VN_InverseFeatureScaling_REAL [▶ 1398]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.28 F_VN_FeatureTransform

Apply a feature transform to one or more sample(s).

Syntax

Definition:

```
FUNCTION F_VN_FeatureTransform : HRESULT
VAR_INPUT
    ipFeatureTransform : ITcVnMlModel;
    ipSamples          : ITcUnknown;
    pipTransformed     : Pointer To ITcUnknown;
    hrPrev             : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipFeatureTransform	ITcVnMlModel [▶ 399]	Feature transform instance to be used
ipSamples	ITcUnknown [▶ 407]	Container holding one or more of input sample(s) (ContainerType_Vector_REAL, ContainerType_Vector_LREAL, ContainerType_Vector_Vector_REAL, or ContainerType_Vector_Vector_LREAL)
pipTransformed	Pointer To ITcUnknown [▶ 407]	Returns the transformed sample(s) using the same type like ipSamples
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_FeatureTransform 使用转换模型对一个或多个样本的特征进行转换。可以使用 F_VN_InverseFeatureTransform [▶ 1404] 函数进行逆转换。

参数

转换模型

先前创建模型必须传输到 ipFeatureTransform 进行特征转换。

样本

具有待转换样本的容器会被传输到 ipSamples。允许使用以下容器类型：

- ContainerType_Vector_REAL 或 ContainerType_Vector_LREAL，用于单个样本
- ContainerType_Vector_Vector_REAL 或 ContainerType_Vector_Vector_LREAL，用于具有多个样本的容器。

转换样本

通过指针 pipTransformed 返回具有转换样本的容器。容器类型取自 ipSamples。

应用

例如，特征转换如下所示进行：

```
hr := F_VN_FeatureTransform(
    ipFeatureTransform  := ipMlModel,
    ipSamples           := ipSamples,
    pipTransformed      := ADR(ipTransformed),
    hrPrev              := hr);
```

相关函数

- F_VN_InverseFeatureTransform [▶ 1404]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.29 F_VN_GetBatchClusters

Get the IDs of the best-matching clusters of a batch of samples.

Syntax

Definition:

```
FUNCTION F_VN_GetBatchClusters : HRESULT
VAR_INPUT
    ipClusterer : ITcVnMlModel;
    ipSamples   : ITcUnknown;
    ipClusters  : Reference To ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipClusterer	ITcVnMlModel [▶ 399]	Clusterer to be used
ipSamples	ITcUnknown [▶ 407]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
ipClusters	Reference To ITcVnContainer [▶ 349]	Returns the cluster IDs (ContainerType_Vector_DINT)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_GetBatchClusters 可确定容器中所有样本的最佳匹配聚类，并返回其 ID。

参数

聚类器模型

先前培训模型必须传输到 ipClusterer，以便分配至聚类。

样本

具有样本的容器被传输到 ipSamples。容器类型必须是 ContainerType_Vector_Vector_REAL 或 ContainerType_Vector_Vector_LREAL。

聚类 ID

将通过引用 ipClusters 返回具有已确定 ID 的容器。

专家参数

专家变体 F_VN_GetBatchClustersExp [▶ 1385] 包含附加参数。

应用

通过以下方法可以计算出样本至聚类的分配：

```
hr := F_VN_GetBatchClusters(
    ipClusterer := ipClusterer,
    ipSamples   := ipSamples,
    ipClusters  := ipClusters,
    hrPrev      := hr);
```

相关函数

- F_VN_GetSampleCluster [▶ 1392]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.30 F_VN_GetBatchClustersExp

Get the IDs of the best-matching clusters of a batch of samples.

Syntax

Definition:

```
FUNCTION F_VN_GetBatchClustersExp : HRESULT
VAR_INPUT
    ipClusterer : ITcVnMlModel;
    ipSamples   : ITcUnknown;
    ipClusters  : Reference To ITcVnContainer;
    ipNovelties : Reference To ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipClusterer	ITcVnMlModel [▶ 399]	Clusterer to be used
ipSamples	ITcUnknown [▶ 407]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
ipClusters	Reference To ITcVnContainer [▶ 349]	Returns the cluster IDs (ContainerType_Vector_DINT)
ipNovelties	Reference To ITcVnContainer [▶ 349]	Returns the degree of novelty (0.0 if a sample is completely known; > 0.0 otherwise) of each sample (ContainerType_Vector_REAL; optional, set to 0 if not required)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_GetBatchClustersExp 是F_VN_GetBatchClusters [▶ 1384] 的专家变体。它包含额外的参数。

参数

聚类器模型

先前培训模型必须传输到 ipClusterer，以便分配至聚类。

样本

具有样本的容器被传输到 ipSamples。容器类型必须是 ContainerType_Vector_Vector_REAL 或 ContainerType_Vector_Vector_LREAL。

聚类 ID

将通过引用 ipClusters 返回具有已确定 ID 的容器。

新颖性

具有每个样本 新颖性 的容器将通过引用 ipNovelties 返回。新颖性 表示样本与聚类的差异程度。0 的新颖性 表示样本正好位于聚类的中心。

应用

样本至聚类的分配（包括新颖性 ）可如下计算：

```
hr := F_VN_GetBatchClustersExp(
    ipClusterer := ipClusterer,
    ipSamples   := ipSamples,
    ipClusters  := ipClusters,
    ipNovelties := ipNovelties,
    hrPrev      := hr);
```

相关函数

- F_VN_GetSampleCluster [▶ 1392]
- F_VN_GetSampleNovelty [▶ 1395]
- F_VN_GetBatchNovelties [▶ 1387]

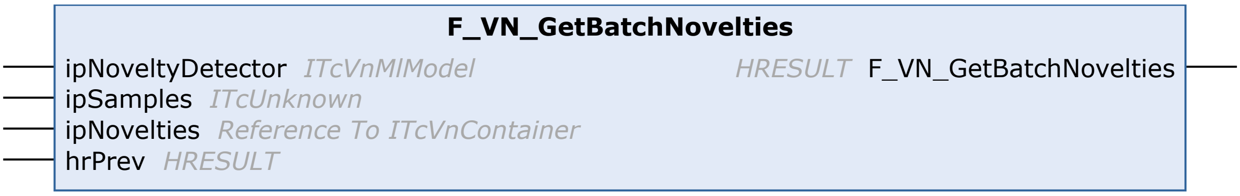
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.31 F_VN_GetBatchNovelties



Get the degree of novelty of multiple samples.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_GetBatchNovelties : HRESULT
VAR_INPUT
    ipNoveltyDetector : ITcVnMlModel;
    ipSamples         : ITcUnknown;
    ipNovelties       : Reference To ITcVnContainer;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipNoveltyDetector	ITcVnMlModel [▸ 399]	Novelty detector to be used
ipSamples	ITcUnknown [▸ 407]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
ipNovelties	Reference To ITcVnContainer [▸ 349]	Returns the degree of novelty (0.0 if a sample is completely known; > 0.0 otherwise) of each sample (ContainerType_Vector_REAL)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_GetBatchNovelties 可确定容器中所有样本的新颖性 。

参数

异常模型

要计算新颖性 ，必须将先前训练模型传输到 ipNoveltyDetector。

样本

具有样本的容器被传输到 ipSamples。

新颖性

具有计算新颖性 的容器将通过引用 ipNovelties 返回。

应用

多个样本的新颖性 可如下计算：

```
hr := F_VN_GetBatchNovelties(
    ipNoveltyDetector    := ipNoveltyDetector,
    ipSamples            := ipSamples,
    ipNovelties          := ipNovelties,
    hrPrev               := hr);
```

相关函数

- F_VN_GetSampleNovelty [▶ 1395]

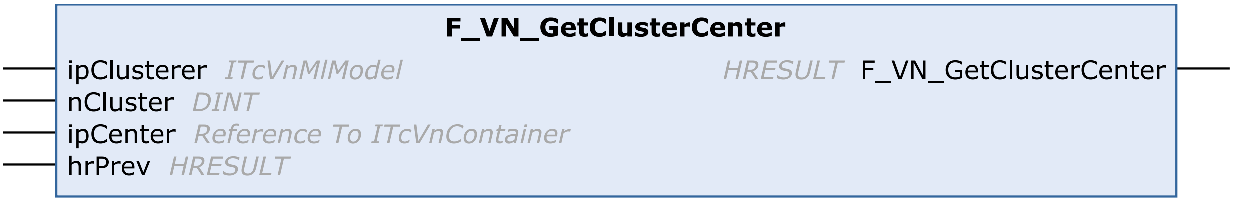
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.32 F_VN_GetClusterCenter



Get the center of a cluster.

Syntax

Definition:

```
FUNCTION F_VN_GetClusterCenter : HRESULT
VAR_INPUT
    ipClusterer : ITcVnMlModel;
    nCluster    : DINT;
    ipCenter    : Reference To ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipClusterer	ITcVnMlModel [▶ 399]	Clusterer to be used
nCluster	DINT	Cluster ID of the cluster the center of which is requested
ipCenter	Reference To ITcVnContainer [▶ 349]	Returns the cluster center (ContainerType_Vector_REAL or ContainerType_Vector_LREAL depending on the floating point type internally used by the clusterer)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_GetClusterCenter 返回聚类的中心。

参数

聚类器模型

要计算其中心的聚类必须包含在先前训练模型 ipClusterer 中。

聚类 ID

相应聚类的 ID 将传输到 nCluster。

聚类中心

聚类中心通过引用 ipCenter 返回。根据 ipClusterer 内部使用的数据类型，返回的容器类型是 ContainerType_Vector_REAL 或 ContainerType_Vector_LREAL。

应用

聚类模型中第三个聚类的中心可如下确定：

```
hr := F_VN_GetClusterCenter(
    ipClusterer := ipClusterer,
    nCluster    := 2,
    ipCenter    := ipCenter,
    hrPrev      := hr);
```

相关函数

- F_VN_GetClusterNum [▶ 1390]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.33 F_VN_GetClusterNum

Get the number of clusters used by this clusterer.

Syntax

Definition:

```
FUNCTION F_VN_GetClusterNum : HRESULT
VAR_INPUT
    ipClusterer : ITcVnMlModel;
END_VAR
VAR_IN_OUT
    nClusterNum : ULINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipClusterer	ITcVnMlModel [▶ 399]	Clusterer to be used
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nClusterNum	ULINT	Returns the number of clusters

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_GetClusterNum 返回 聚类器模型中的聚类数。

参数

聚类器模型

先前培训聚类器模型必须传输到 ipClusterer。

聚类数

聚类数通过 nClusterNum 返回。

应用

例如，聚类数可如下确定：

```
hr := F_VN_GetClusterNum(
    ipClusterer    := ipClusterer,
    nClusterNum    := nClusterNum,
    hrPrev         := hr);
```

相关函数

- F_VN_GetClusterCenter [▸ 1388]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.34 F_VN_GetFeatureScales

Calculate the scaling parameters for each feature in the input samples based on the scaling type.

Syntax

Definition:

```
FUNCTION F_VN_GetFeatureScales : HRESULT
VAR_INPUT
    ipSamples           : ITcVnContainer;
    ipScales            : Reference To ITcVnContainer;
    eFeatureScalingType : ETcVnFeatureScalingType;
    hrPrev              : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSamples	ITcVnContainer [▸ 349]	Container holding input sample(s) (ContainerType_Vector_REAL, ContainerType_Vector_LREAL, ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
ipScales	Reference To ITcVnContainer [▸ 349]	Returns a container with the scaling parameters for each feature (Vector_REAL or Vector_LREAL depending on the type of ipSamples).
eFeatureScalingType	ETcVnFeatureScalingType [▸ 181]	Feature scaling type
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_GetFeatureScales 可为传输样本的每个特征值计算缩放参数。然后可以使用 F_VN_FeatureScaling [▸ 1379] 函数进行相应的缩放，以实现特征归一化。同样的缩放比例也可以与 F_VN_InverseFeatureScaling [▸ 1397] 配合使用，以反转回归预测结果的缩放。

参数

样本

用于计算缩放参数的样本作为容器传输到 ipSamples。每个样本的特征值数必须相同。

缩放

计算缩放参数将通过引用 ipScales 作为一个容器返回。容器中元素的数量和顺序取决于缩放类型，始终存在于第一个元素中。如为 MAXABS，每个特征始终有一个缩放值。对于其他类型，每个特征始终有两个缩放值。

缩放类型

所需缩放类型将作为枚举 ETcVnFeatureScalingType [▶ 181] 传输到eFeatureScalingType。选项有：

- TCVN_FST1_MAXABS：使用特征绝对值的最大值从 -1 到 1 的范围内获得归一化。如果所有值都是正值，则从 0 到 1 进行归一化。
- TCVN_FST1_MINMAX：使用特征值的最小值和最大值从 0 到 1 的范围内获得归一化。
- TCVN_FST1_STANDARDIZATION：使用特征值的平均值和标准偏差将向量平均值归一化为 0，将向量标准偏差归一化为 1。

应用

例如，使用 MINMAX 方法计算缩放参数的过程如下所示：

```
hr := F_VN_GetFeatureScales(
    ipSamples            := ipSamples,
    ipScales             := ipScales,
    eFeatureScalingType  := TCVN_FST1_MINMAX,
    hrPrev               := hr);
```

例如，如果 ipSamples 中每个样本的特征值数为 12，则ipScales 针对此缩放类型将正好有1 + 2 * 12 = 25 个元素。

相关函数

- F_VN_FeatureScaling [▶ 1379]
- F_VN_InverseFeatureScaling [▶ 1397]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.35 F_VN_GetSampleCluster

Get the ID of the best-matching cluster of a single sample.

Syntax

Definition:

```
FUNCTION F_VN_GetSampleCluster : HRESULT
VAR_INPUT
    ipClusterer : ITcVnMlModel;
    ipSample    : ITcUnknown;
END_VAR
VAR_IN_OUT
    nCluster    : DINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipClusterer	ITcVnMlModel [▶ 399]	Clusterer to be used
ipSample	ITcUnknown [▶ 407]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nCluster	DINT	Returns the ID of the cluster the sample has been assigned to

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_GetSampleCluster 可确定样本的最佳匹配聚类，并返回其 ID。

参数

聚类器模型

先前训练模型必须传输到 ipClusterer，以便分配至聚类。

样本

带有样本的容器被传输到 ipSample。容器类型必须是 ContainerType_Vector_REAL 或 ContainerType_Vector_LREAL。

聚类 ID

确定的 ID 通过 nCluster 返回。

专家参数

专家变体 F_VN_GetSampleClusterExp [▶ 1394] 包含附加参数。

应用

样本到聚类的分配可如下计算：

```
hr := F_VN_GetSampleCluster(
    ipClusterer := ipClusterer,
    ipSample    := ipSample,
    nCluster    := nCluster,
    hrPrev      := hr);
```

相关函数

- F_VN_GetBatchClusters [▶ 1384]

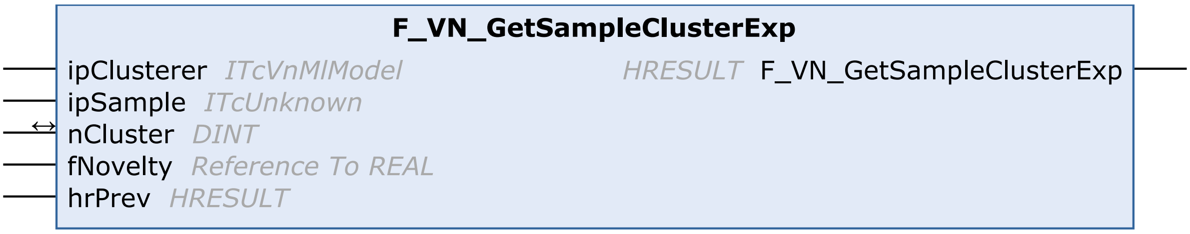
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.36 F_VN_GetSampleClusterExp



Get the ID of the best-matching cluster of a single sample. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_GetSampleClusterExp : HRESULT
VAR_INPUT
    ipClusterer : ITcVnMlModel;
    ipSample    : ITcUnknown;
END_VAR
VAR_IN_OUT
    nCluster    : DINT;
END_VAR
VAR_INPUT
    fNovelty    : Reference To REAL;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipClusterer	ITcVnMlModel [▶ 399]	Clusterer to be used
ipSample	ITcUnknown [▶ 407]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
fNovelty	Reference To REAL	Returns the degree of novelty (0.0 if a sample is completely known; > 0.0 otherwise) of the presented sample (optional, set to 0 if not required)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nCluster	DINT	Returns the ID of the cluster the sample has been assigned to

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_GetSampleClusterExp 是F_VN_GetSampleCluster [▶ 1392] 的专家变体。它包含额外的参数。

参数

聚类器模型

先前训练模型必须传输到 ipClusterer，以便分配至聚类。

样本

带有样本的容器被传输到 ipSample。容器类型必须是 ContainerType_Vector_REAL 或 ContainerType_Vector_LREAL。

聚类 ID

确定的 ID 通过 nCluster 返回。

新颖性

样本的新颖性 通过引用 fNovelty 返回。新颖性 表示样本与聚类的差异程度。新颖性 为 0 表示模型已知样本。

应用

样本的聚类和新颖性 可如下计算：

```
hr := F_VN_GetSampleClusterExp(
    ipClusterer := ipClusterer,
    ipSample    := ipSample,
    nCluster    := nCluster,
    fNovelty    := fNovelty,
    hrPrev      := hr);
```

相关函数

- F_VN_GetSampleNovelty [▶ 1395]
- F_VN_GetBatchClusters [▶ 1384]
- F_VN_GetBatchNovelties [▶ 1387]

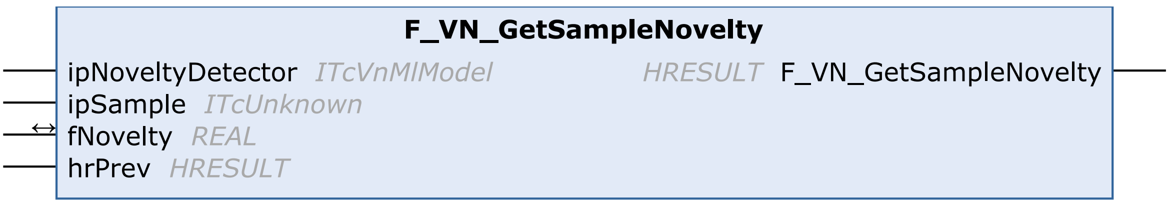
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.37 F_VN_GetSampleNovelty



Get the degree of novelty of a single sample.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_GetSampleNovelty : HRESULT
VAR_INPUT
    ipNoveltyDetector : ITcVnMlModel;
    ipSample          : ITcUnknown;
END_VAR
VAR_IN_OUT
    fNovelty          : REAL;
END_VAR
VAR_INPUT
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipNoveltyDetector	ITcVnMlModel [▸ 399]	Novelty detector to be used
ipSample	ITcUnknown [▸ 407]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fNovelty	REAL	Returns the degree of novelty (0.0 if a sample is completely known; > 0.0 otherwise) of the presented sample

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_GetSampleNovelty 可确定容器中样本的新颖性 。

参数

异常模型

要计算新颖性 ，必须将先前训练模型传输到 ipNoveltyDetector。

样本

带有样本的容器被传输到 ipSample

新颖性

新颖性 通过引用 fNovelty 返回。新颖性 表示样本与已学习数据的差异程度。新颖性 为 0 表示模型已知样本。

应用

样本的新颖性 可如下计算：

```
hr := F_VN_GetSampleNovelty(
    ipNoveltyDetector   := ipNoveltyDetector,
    ipSample            := ipSample,
    fNovelty            := fNovelty,
    hrPrev              := hr);
```

相关函数

- F_VN_GetBatchNovelties [▶ 1387]

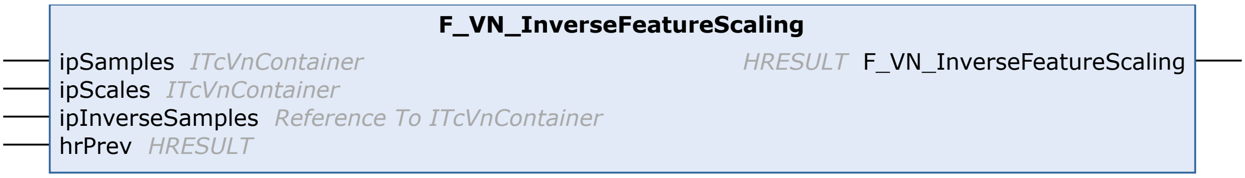
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.38 F_VN_InverseFeatureScaling



Apply the inverse of feature scaling to one or more sample(s).

Syntax

Definition:

```
FUNCTION F_VN_InverseFeatureScaling : HRESULT
VAR_INPUT
    ipSamples         : ITcVnContainer;
    ipScales          : ITcVnContainer;
    ipInverseSamples  : Reference To ITcVnContainer;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSamples	ITcVnContainer [▶ 349]	Container holding one or more input sample(s) (ContainerType_Vector_REAL, ContainerType_Vector_LREAL, ContainerType_Vector_Vector_REAL, or ContainerType_Vector_Vector_LREAL)
ipScales	ITcVnContainer [▶ 349]	Container with the scaling parameters of each feature (type ContainerType_Vector_REAL or ContainerType_Vector_LREAL).
ipInverseSamples	Reference To ITcVnContainer [▶ 349]	Returns the inversed sample(s) using the scaling type and parameters of ipScales. It has the same type of ipSamples. If the same container ipSamples is used, the source data will be replaced.
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_InverseFeatureScaling 反转一个或多个样本的缩放。函数 F_VN_FeatureScaling [▶ 1379] 可用于缩放。

参数

样本

必须将一个或多个样本作为一个容器传输到 ipSamples。

缩放

必须将带有缩放类型和相应缩放参数的容器传输到 ipScales。容器可以使用函数 F_VN_GetFeatureScales [▶ 1391] 计算，也可以使用其中描述的结构创建。

缩放样本

反缩放样本通过引用 ipInverseSamples 返回。

专家参数

专家变体 F_VN_InverseFeatureScalingExp [▶ 1400] 包含附加参数。

应用

例如，缩放比例之前使用 F_VN_FeatureScaling [▶ 1379] 计算的反特征缩放如下所示：

```
hr := F_VN_FeatureScaling(ipSamples, ipScales, ipSamplesScaled, hr);

hr := F_VN_InverseFeatureScaling(
    ipSamples        := ipSamplesScaled,
    ipScales         := ipScales,
    ipInverseSamples:= ipSamplesInvertedScaling,
    hrPrev           := hr);
```

相关函数

- F_VN_GetFeatureScales [▶ 1391]
- F_VN_FeatureScaling [▶ 1379]
- F_VN_InverseFeatureScaling_REAL [▶ 1398]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.39 F_VN_InverseFeatureScaling_REAL

Apply the inverse of feature scaling to a single value.

Syntax

Definition:

```
FUNCTION F_VN_InverseFeatureScaling_REAL : HRESULT
VAR_INPUT
    fSample        : REAL;
    ipScales       : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fInverseSample : REAL;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fSample	REAL	Sample value
ipScales	ITcVnContainer [▶ 349]	Container with the scaling parameters of each feature (type ContainerType_Vector_REAL or ContainerType_Vector_LREAL).
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fInverseSample	REAL	Returns the inversed sample.

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_InverseFeatureScaling_REAL 可逆缩放只有一个特征的样本或标量回归子的预测值。函数 F_VN_FeatureScaling [▶ 1379] 可用于缩放。

参数

样本

REAL 类型的样本必须传输到 fSample。

缩放

必须将带有缩放类型和相应缩放参数的容器传输到 ipScales。容器可以使用函数 F_VN_GetFeatureScales [▶ 1391] 计算，也可以使用其中描述的结构创建。

缩放样本

逆缩放特征通过 fInverseSample 返回。

专家参数

专家变体 F_VN_InverseFeatureScalingExp_REAL [▶ 1402] 包含附加参数。

应用

例如，一个样本的逆特征缩放如下所示：

```
hr := F_VN_InverseFeatureScaling_REAL(
    fSample        := fSample,
    ipScales       := ipScales,
    fInverseSample := fSampleInvertedScaling,
    hrPrev         := hr);
```

相关函数

- F_VN_GetFeatureScales [▶ 1391]
- F_VN_FeatureScaling [▶ 1379]
- F_VN_InverseFeatureScaling [▶ 1397]

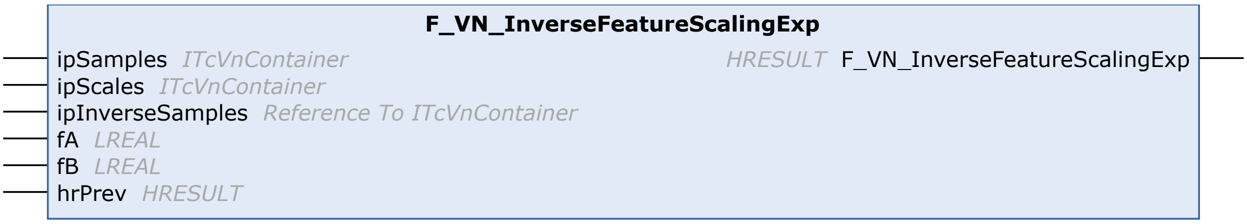
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.40 F_VN_InverseFeatureScalingExp



Apply the inverse of feature scaling to one or more sample(s). [fA, fB] should be used if the scaling has been performed using TCVN_FST1_MINMAX with a predetermined output range.. (expert function)

Syntax

Definition:

```
FUNCTION F_VN_InverseFeatureScalingExp : HRESULT
VAR_INPUT
    ipSamples        : ITcVnContainer;
    ipScales         : ITcVnContainer;
    ipInverseSamples : Reference To ITcVnContainer;
    fA               : LREAL;
    fB               : LREAL;
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSamples	ITcVnContainer [▶ 349]	Container holding one or more input sample(s) (ContainerType_Vector_REAL, ContainerType_Vector_LREAL, ContainerType_Vector_Vector_REAL, or ContainerType_Vector_Vector_LREAL)
ipScales	ITcVnContainer [▶ 349]	Container with the scaling parameters of each feature (type ContainerType_Vector_REAL or ContainerType_Vector_LREAL).
ipInverseSamples	Reference To ITcVnContainer [▶ 349]	Returns the inversed sample(s) using the scaling type and parameters of ipScales. It has the same type of ipSamples. If the same container ipSamples is used, the source data will be replaced.
fA	LREAL	represents the lower bound of the range for TCVN_FST1_MINMAX
fB	LREAL	represents the upper bound of the range for TCVN_FST1_MINMAX
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_InverseFeatureScalingExp 是F_VN_InverseFeatureScaling [▶ 1397] 的专家变体。它包含额外的参数。

参数

样本

必须将一个或多个样本作为一个容器传输到 ipSamples。

缩放

必须将带有缩放类型和相应缩放参数的容器传输到 ipScales。容器可以使用函数 F_VN_GetFeatureScales [▶ 1391] 计算，也可以使用其中描述的结构创建。

缩放样本

反缩放样本通过引用 ipInverseSamples 返回。

定义输出范围

如果使用缩放类型 TCVN_FST1_MINMAX，还可以通过 fA 和 fB 指定输出值的值范围。

应用

例如，先前由 F_VN_FeatureScalingExp [▶ 1381] 计算的逆特征缩放的输出值范围为 0.1 至 0.9，具体如下所示：

```
hr := F_VN_FeatureScalingExp(ipSamples, ipScales, ipSamplesScaled, 0.1, 0.9, hr);

hr := F_VN_InverseFeatureScalingExp(
    ipSamples        := ipSamplesScaled,
    ipScales         := ipScales,
    ipInverseSamples:= ipSamplesInvertedScaling,
    fA               := 0.1,
    fB               := 0.9,
    hrPrev           := hr);
```

相关函数

- F_VN_GetFeatureScales [▶ 1391]
- F_VN_FeatureScaling [▶ 1379]
- F_VN_InverseFeatureScaling_REAL [▶ 1398]

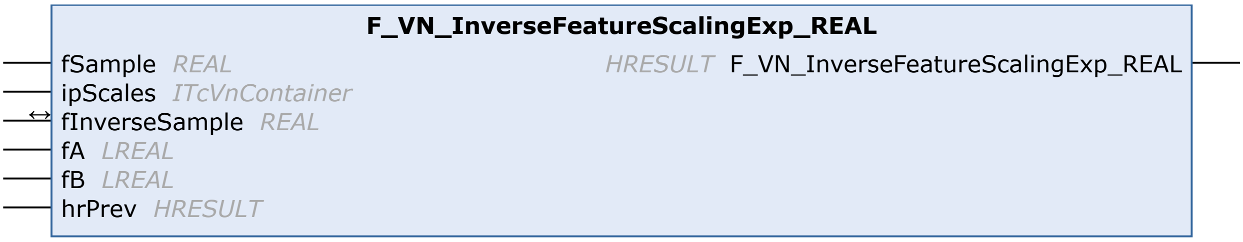
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.41 F_VN_InverseFeatureScalingExp_REAL



Apply the inverse of feature scaling to a single value. [fA, fB] should be used if the scaling has been performed using TCVN_FST1_MINMAX with a predetermined output range.

Syntax

Definition:

```
FUNCTION F_VN_InverseFeatureScalingExp_REAL : HRESULT
VAR_INPUT
    fSample        : REAL;
    ipScales       : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fInverseSample : REAL;
END_VAR
VAR_INPUT
    fA             : LREAL;
    fB             : LREAL;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
fSample	REAL	Sample value
ipScales	ITcVnContainer [▶ 349]	Container with the scaling parameters of each feature (type ContainerType_Vector_REAL or ContainerType_Vector_LREAL).
fA	LREAL	represents the lower bound of the range for TCVN_FST1_MINMAX
fB	LREAL	represents the upper bound of the range for TCVN_FST1_MINMAX
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fInverseSampl e	REAL	Returns the inversed sample.

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_InverseFeatureScalingExp_REAL 是F_VN_InverseFeatureScaling_REAL [▸ 1398] 的专家变体。它包含额外的参数。

参数

样本

REAL 类型的样本必须传输到 fSample。

缩放

必须将带有缩放类型和相应缩放参数的容器传输到 ipScales。容器可以使用函数 F_VN_GetFeatureScales [▸ 1391] 计算，也可以使用其中描述的结构创建。

缩放样本

逆缩放特征通过 fInverseSample 返回。

定义输出范围

如果使用缩放类型 TCVN_FST1_MINMAX，还可以通过 fA 和 fB 指定输出值的值范围。

应用

例如，一个样本的逆特征缩放如下所示：

```
hr := F_VN_InverseFeatureScalingExp_REAL(
    fSample        := fSample,
    ipScales       := ipScales,
    fInverseSample := fSampleInvertedScaling,
    fA             := 0.1,
    fB             := 0.9,
    hrPrev         := hr);
```

相关函数

- F_VN_GetFeatureScales [▸ 1391]
- F_VN_FeatureScaling [▸ 1379]
- F_VN_InverseFeatureScaling_REAL [▸ 1398]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.42 F_VN_InverseFeatureTransform

Apply an inverse feature transform to one or more sample(s).

Syntax

Definition:

```
FUNCTION F_VN_InverseFeatureTransform : HRESULT
VAR_INPUT
    ipFeatureTransform : ITcVnMlModel;
    ipSamples          : ITcUnknown;
    pipTransformed     : Pointer To ITcUnknown;
    hrPrev             : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipFeatureTransform	ITcVnMlModel [▸ 399]	Feature transform instance to be used
ipSamples	ITcUnknown [▸ 407]	Container holding one or more of input sample(s) (ContainerType_Vector_REAL, ContainerType_Vector_LREAL, ContainerType_Vector_Vector_REAL, or ContainerType_Vector_Vector_LREAL)
pipTransformed	Pointer To ITcUnknown [▸ 407]	Returns the transformed sample(s) using the same type like ipSamples
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_InverseFeatureTransform 利用转换模型逆转换一个或多个样本的特征。使用函数 F_VN_FeatureTransform [▸ 1382] 可以对同一模型进行直接转换。

参数

转换模型

先前创建模型必须传输到 ipFeatureTransform 进行特征转换。

样本

具有待转换样本的容器会被传输到 ipSamples。允许使用以下容器类型：

- ContainerType_Vector_REAL 或 ContainerType_Vector_LREAL，用于单个样本
- ContainerType_Vector_Vector_REAL 或 ContainerType_Vector_Vector_LREAL，用于具有多个样本的容器。

转换样本

通过指针 pipTransformed 返回具有转换样本的容器。容器类型取自 ipSamples。

应用

例如，逆特征转换如下所示进行：

```
hr := F_VN_InverseFeatureTransform(
    ipFeatureTransform  := ipMlModel,
    ipSamples           := ipSamplesTransformed,
    pipTransformed      := ADR(ipTransformed),
    hrPrev              := hr);
```

相关函数

- F_VN_FeatureTransform [▶ 1382]

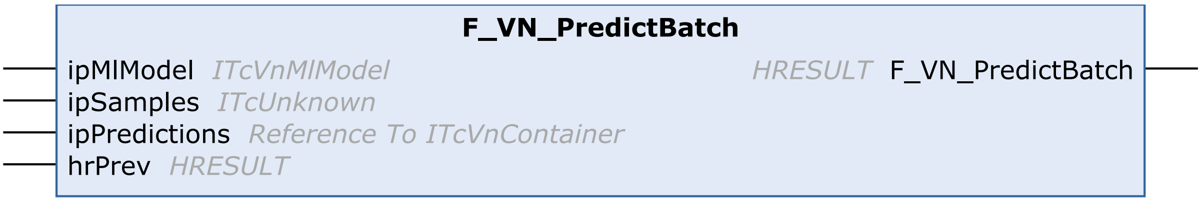
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.43 F_VN_PredictBatch



Compute predictions for a batch of samples.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_PredictBatch : HRESULT
VAR_INPUT
    ipMlModel     : ITcVnMlModel;
    ipSamples     : ITcUnknown;
    ipPredictions : Reference To ITcVnContainer;
    hrPrev        : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	ITcVnMlModel [▶ 399]	Classifier or regressor to be used
ipSamples	ITcUnknown [▶ 407]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
ipPredictions	Reference To ITcVnContainer [▶ 349]	Returns the predicted outputs (depending on ipSamples; class labels (for classification, ContainerType_Vector_DINT) or real-valued predictions (for regression with scalar output, ContainerType_Vector_REAL or ContainerType_Vector_LREAL; for regression with vectorial output, ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL))
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_PredictBatch 针对多个样本计算模型预测。根据模型类型的不同，预测的数据类型也不同（请参见 ipPredictions）。

参数

模型

要进行预测，必须将先前训练模型传输到 ipMlModel。

样本

样本在一个容器中被传输到 ipSamples。容器类型必须是 ContainerType_Vector_Vector_REAL 或 ContainerType_Vector_Vector_LREAL。

预测

预测在一个容器中通过引用 ipPredictions 返回。根据模型的不同，容器的类型也不同：

- 分类的类别标签：ContainerType_Vector_DINT
- 标量回归中的数值预测值：ContainerType_Vector_REAL 或ContainerType_Vector_LREAL,，具体取决于 ipSamples
- 向量回归中的数值预测值：ContainerType_Vector_Vector_REAL 或 ContainerType_Vector_Vector_LREAL,，具体取决于 ipSamples

专家参数

专家变体 F_VN_PredictBatchExp [▶ 1407] 包含附加参数。

应用

例如，可以同时对多个样本进行分类：

```
hr := F_VN_PredictBatch(
    ipMlModel       :=  ipMlModel,
    ipSamples       :=  ipSamples,
    ipPredictions   :=  ipPredictions,
    hrPrev          :=  hr);

hr := F_VN_GetAt_DINT(ipPredictions, nClassOfThirdSample, 2, hr);
```

F_VN_GetAt_DINT 的调用可显示如何从预测中获得第 3 个样本的类别标签。

相关函数

- F_VN_PredictSampleClass [▶ 1409]
- F_VN_PredictSampleScalar [▶ 1412]
- F_VN_PredictSampleVector [▶ 1415]

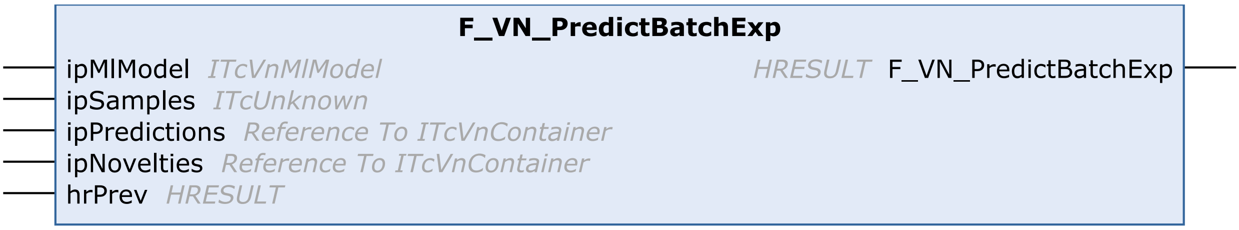
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.44 F_VN_PredictBatchExp



Compute predictions for a batch of samples. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_PredictBatchExp : HRESULT
VAR_INPUT
    ipMlModel      : ITcVnMlModel;
    ipSamples      : ITcUnknown;
    ipPredictions : Reference To ITcVnContainer;
    ipNovelties    : Reference To ITcVnContainer;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	ITcVnMlModel [▶ 399]	Classifier or regressor to be used
ipSamples	ITcUnknown [▶ 407]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
ipPredictions	Reference To ITcVnContainer [▶ 349]	Returns the predicted outputs (depending on ipSamples; class labels (for classification, ContainerType_Vector_DINT) or real-valued predictions (for regression with scalar output, ContainerType_Vector_REAL or ContainerType_Vector_LREAL; for regression with vectorial output, ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL))
ipNovelties	Reference To ITcVnContainer [▶ 349]	Returns the degree of novelty (0.0 if a sample is completely known; > 0.0 otherwise) of each sample (ContainerType_Vector_REAL; optional, set to 0 if not required)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_PredictBatchExp 是F_VN_PredictBatch [▶ 1405] 的专家变体。它包含额外的参数。

参数

模型

要进行预测，必须将先前训练模型传输到 ipMlModel。

样本

样本在一个容器中被传输到 ipSamples。容器类型必须是 ContainerType_Vector_Vector_REAL 或 ContainerType_Vector_Vector_LREAL。

预测

预测在一个容器中通过引用 ipPredictions 返回。根据模型的不同，容器的类型也不同：

- 分类的类别标签：ContainerType_Vector_DINT
- 标量回归中的数值预测值：ContainerType_Vector_REAL 或ContainerType_Vector_LREAL,，具体取决于 ipSamples
- 向量回归中的数值预测值：ContainerType_Vector_Vector_REAL 或 ContainerType_Vector_Vector_LREAL,，具体取决于 ipSamples

新颖性

样本的新颖性 通过引用 ipNovelties 作为容器返回。

应用

例如，可以同时对多个样本进行分类：

```
hr := F_VN_PredictBatchExp(
    ipMlModel       := ipMlModel,
    ipSamples       := ipSamples,
    ipPredictions   := ipPredictions,
```

```
    ipNovelties      := ipNovelties,
    hrPrev           := hr);

hr := F_VN_GetAt_DINT(ipPredictions, nClassOfThirdSample, 2, hr);
```

F_VN_GetAt_DINT 的调用可显示如何从预测中获得第 3 个样本的类别标签。

相关函数

- F_VN_PredictSampleClass [▸ 1409]
- F_VN_PredictSampleScalar [▸ 1412]
- F_VN_PredictSampleVector [▸ 1415]
- F_VN_GetBatchNovelties [▸ 1395]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.45 F_VN_PredictSampleClass

Classify a single sample.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_PredictSampleClass : HRESULT
VAR_INPUT
    ipClassifier : ITcVnMlModel;
    ipSample     : ITcUnknown;
END_VAR
VAR_IN_OUT
    nClass       : DINT;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipClassifier	ITcVnMlModel [▸ 399]	Classifier to be used
ipSample	ITcUnknown [▸ 407]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nClass	DINT	Returns the classification result

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_PredictSampleClass 根据分类模型对样本进行分类。

参数

分类模型

先前训练分类模型必须传输到 ipClassifier。

样本

样本在一个容器中被传输到 ipSample。容器类型必须是 ContainerType_Vector_REAL 或 ContainerType_Vector_LREAL。

类别

样本的类别作为分类结果通过 nClass 返回。

专家参数

专家变体 F_VN_PredictSampleClassExp [▶ 1411] 包含附加参数。

应用

例如，一个样本可如下分类：

```
hr := F_VN_PredictSampleClass(
    ipClassifier    := ipMlModel,
    ipSample        := ipSample,
    nClass          := nClassResult,
    hrPrev          := hr);
```

相关函数

- F_VN_PredictBatch [▶ 1405]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.46 F_VN_PredictSampleClassExp

F_VN_PredictSampleClassExp

ipClassifier *ITcVnMlModel* — *HRESULT* F_VN_PredictSampleClassExp
ipSample *ITcUnknown*
nClass *DINT*
fNovelty *Reference To REAL*
hrPrev *HRESULT*

Classify a single sample. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_PredictSampleClassExp : HRESULT
VAR_INPUT
    ipClassifier : ITcVnMlModel;
    ipSample     : ITcUnknown;
END_VAR
VAR_IN_OUT
    nClass       : DINT;
END_VAR
VAR_INPUT
    fNovelty     : Reference To REAL;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipClassifier	ITcVnMlModel [▸ 399]	Classifier to be used
ipSample	ITcUnknown [▸ 407]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
fNovelty	Reference To REAL	Returns the degree of novelty (0.0 if a sample is completely known; > 0.0 otherwise) of the presented sample (optional, set to 0 if not required)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nClass	DINT	Returns the classification result

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_PredictSampleClassExp 是F_VN_PredictSampleClass [▸ 1409] 的专家变体。它包含额外的参数。

参数

分类模型

先前训练分类模型必须传输到 ipClassifier。

样本

样本在一个容器中被传输到 ipSample。容器类型必须是 ContainerType_Vector_REAL 或 ContainerType_Vector_LREAL。

类别

样本的类别作为分类结果通过 nClass 返回。

新颖性

样本的新颖性 通过 fNovelty 返回。

应用

例如，一个样本可如下分类：

```
hr := F_VN_PredictSampleClassExp(
    ipClassifier    := ipMlModel,
    ipSample        := ipSample,
    nClass          := nClassResult,
    fNovelty        := fNovelty,
    hrPrev          := hr);
```

相关函数

- F_VN_GetSampleNovelty [▶ 1395]
- F_VN_PredictBatch [▶ 1405]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.47 F_VN_PredictSampleScalar

Compute a scalar prediction for a single sample.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_PredictSampleScalar : HRESULT
VAR_INPUT
    ipRegressor : ITcVnMlModel;
    ipSample    : ITcUnknown;
END_VAR
VAR_IN_OUT
    fPrediction : REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipRegressor	ITcVnMlModel [▶ 399]	Regressor to be used
ipSample	ITcUnknown [▶ 407]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fPrediction	REAL	Returns the predicted output

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_PredictSampleScalar 根据回归模型计算样本的标量预测值。

参数

回归模型

先前培训回归模型必须传输到 ipRegressor。

样本

样本容器作为 ipSample 传输。容器类型必须是 ContainerType_Vector_REAL 或 ContainerType_Vector_LREAL。

预测

计算预测值通过 fPrediction 返回。

专家参数

专家变体 F_VN_PredictSampleScalarExp [▶ 1414] 包含附加参数。

应用

例如，一个样本的预测值可如下计算：

```
hr := F_VN_PredictSampleScalar(
    ipRegressor := ipRegressor,
    ipSample    := ipSample,
    fPrediction := fPrediction,
    hrPrev      := hr);
```

相关函数

- F_VN_PredictBatch [▶ 1405]

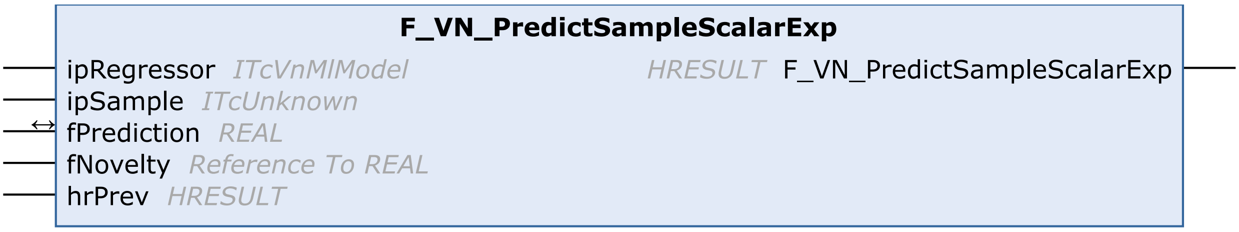
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.48 F_VN_PredictSampleScalarExp



Compute a scalar prediction for a single sample. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_PredictSampleScalarExp : HRESULT
VAR_INPUT
    ipRegressor : ITcVnMlModel;
    ipSample    : ITcUnknown;
END_VAR
VAR_IN_OUT
    fPrediction : REAL;
END_VAR
VAR_INPUT
    fNovelty    : Reference To REAL;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipRegressor	ITcVnMlModel [▸ 399]	Regressor to be used
ipSample	ITcUnknown [▸ 407]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
fNovelty	Reference To REAL	Returns the degree of novelty (0.0 if a sample is completely known; > 0.0 otherwise) of the presented sample (optional, set to 0 if not required)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fPrediction	REAL	Returns the predicted output

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_PredictSampleScalarExp 是 F_VN_PredictSampleScalar [▸ 1412] 的专家变体。它包含额外的参数。

参数

回归模型

先前培训回归模型必须传输到 ipRegressor。

样本

样本容器作为 ipSample 传输。容器类型必须是 ContainerType_Vector_REAL 或 ContainerType_Vector_LREAL。

预测

计算预测值通过 fPrediction 返回。

新颖性

样本的新颖性　通过 fNovelty 返回。

应用

例如，一个样本的回归值可如下计算：

```
hr := F_VN_PredictSampleScalarExp(
    ipRegressor := ipRegressor,
    ipSample    := ipSample,
    fPrediction := fPrediction,
    fNovelty    := fNovelty,
    hrPrev      := hr);
```

相关函数

- F_VN_PredictBatch [▶ 1405]
- F_VN_GetBatchNovelties [▶ 1395]

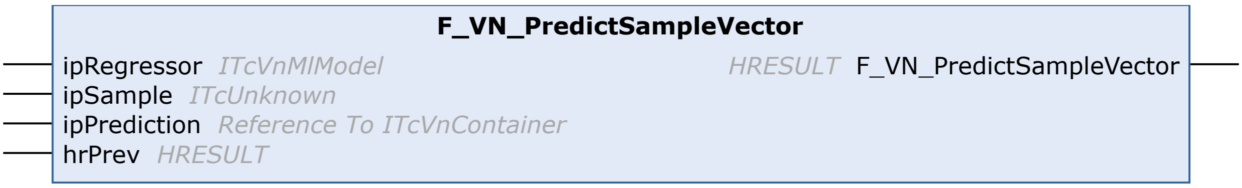
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.49 F_VN_PredictSampleVector



Compute a vectorial prediction for a single sample.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_PredictSampleVector : HRESULT
VAR_INPUT
    ipRegressor  : ITcVnMlModel;
    ipSample     : ITcUnknown;
    ipPrediction : Reference To ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipRegressor	ITcVnMlModel [▸ 399]	Regressor to be used
ipSample	ITcUnknown [▸ 407]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
ipPrediction	Reference To ITcVnContainer [▸ 349]	Returns the predicted output (ContainerType_Vector_REAL or ContainerType_Vector_LREAL, depending on ipSample)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_PredictSampleVector 根据回归模型计算样本的预测向量。

参数

回归模型

先前培训回归模型必须传输到 ipRegressor。

样本

样本容器作为 ipSample 传输。容器类型必须是 ContainerType_Vector_REAL 或 ContainerType_Vector_LREAL。

预测

计算出的预测向量作为一个容器通过引用 ipPrediction 返回。容器类型取自 ipSamples。

专家参数

专家变体 F_VN_PredictSampleVectorExp [▸ 1417] 包含附加参数。

应用

例如，一个样本的预测向量可如下计算：

```
hr := F_VN_PredictSampleVector(
    ipRegressor := ipRegressor,
    ipSample    := ipSample,
    ipPrediction:= ipPrediction,
    hrPrev      := hr);
```

相关函数

- F_VN_PredictBatch [▸ 1405]

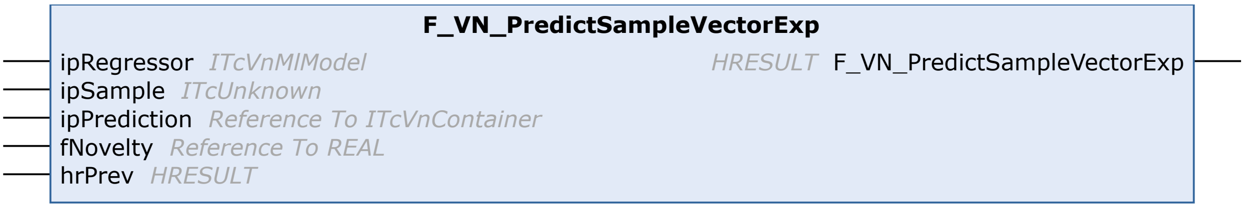
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.50 F_VN_PredictSampleVectorExp



Compute a vectorial prediction for a single sample. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_PredictSampleVectorExp : HRESULT
VAR_INPUT
    ipRegressor  : ITcVnMlModel;
    ipSample     : ITcUnknown;
    ipPrediction : Reference To ITcVnContainer;
    fNovelty     : Reference To REAL;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipRegressor	ITcVnMlModel [▸ 399]	Regressor to be used
ipSample	ITcUnknown [▸ 407]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
ipPrediction	Reference To ITcVnContainer [▸ 349]	Returns the predicted output (ContainerType_Vector_REAL or ContainerType_Vector_LREAL, depending on ipSample)
fNovelty	Reference To REAL	Returns the degree of novelty (0.0 if a sample is completely known; > 0.0 otherwise) of the presented sample (optional, set to 0 if not required)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_PredictSampleVectorExp 是F_VN_PredictSampleVector [▸ 1415] 的专家变体。它包含额外的参数。

参数

回归模型

先前培训回归模型必须传输到 ipRegressor。

样本

样本容器作为 ipSample 传输。容器类型必须是 ContainerType_Vector_REAL 或 ContainerType_Vector_LREAL。

预测

计算出的预测向量作为一个容器通过引用 ipPrediction 返回。容器类型取自 ipSamples。

新颖性

样本的新颖性 通过 fNovelty 返回。

应用

例如，一个样本的预测向量可如下计算：

```
hr := F_VN_PredictSampleVectorExp(
    ipRegressor := ipRegressor,
    ipSample    := ipSample,
    ipPrediction:= ipPrediction,
    fNovelty    := fNovelty,
    hrPrev      := hr);
```

相关函数

- F_VN_PredictBatch [▶ 1405]
- F_VN_GetBatchNovelties [▶ 1395]

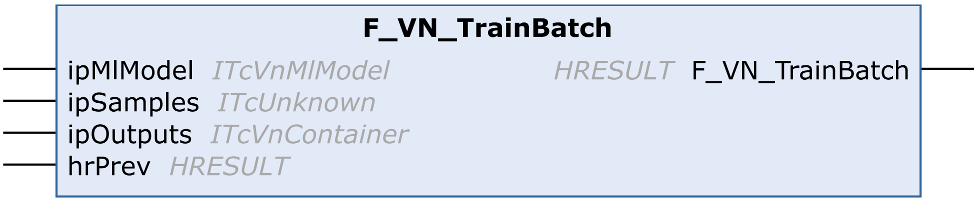
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.51 F_VN_TrainBatch



Train a classifier, a clusterer, a novelty detector, or a regressor with a batch of samples. Online trainable models are trained once with each sample. Depending on the application and the number of available training samples, repeated training of such models with the same data may improve the results.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_TrainBatch : HRESULT
VAR_INPUT
    ipMlModel : ITcVnMlModel;
    ipSamples : ITcUnknown;
    ipOutputs : ITcVnContainer;
    hrPrev    : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	ITcVnMlModel [▶ 399]	Classifier, clusterer, novelty detector, or regressor to be used
ipSamples	ITcUnknown [▶ 407]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
ipOutputs	ITcVnContainer [▶ 349]	Class labels (for classification, ContainerType_Vector_DINT) or target outputs (for regression with scalar output, ContainerType_Vector_REAL or ContainerType_Vector_LREAL; for regression with vectorial output, ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL) to be learnt (must be 0 for clusterers and for novelty detectors)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_TrainBatch 使用多个样本培训模型。根据模型类型，结果值必须具有不同的类型（请参见 ipOutputs）。分类器、聚类器、新颖性 检测器和回归子作为模型类型获得支持。

参数

模型

先前创建模型必须传输到 ipMlModel。

样本

样本在一个容器中被传输到 ipSamples。容器类型必须是 ContainerType_Vector_Vector_REAL 或 ContainerType_Vector_Vector_LREAL。

结果

样本的已知结果在一个容器中被传输到 ipOutputs。根据模型的类型，容器必须有不同的类型：

- 分类的类别标签：ContainerType_Vector_DINT
- 标量回归的数值结果值：ContainerType_Vector_REAL 或 ContainerType_Vector_LREAL
- 向量回归的数值结果值：ContainerType_Vector_Vector_REAL 或 ContainerType_Vector_Vector_LREAL

如果是聚类器模型或新颖性 检测器，则必须是 ipOutputs := 0。

应用

例如，用于分类的 ML 模型可如下训练：

```
hr := F_VN_TrainBatch(
    ipMlModel    := ipMlModel,
    ipSamples    := ipSamples,
    ipOutputs    := ipClassLabels,
    hrPrev       := hr);
```

相关函数

- F_VN_TrainBatchClusters [▶ 1420]
- F_VN_TrainSample [▶ 1421]

- F_VN_TrainSampleClass [▶ 1423]
- F_VN_TrainSampleCluster [▶ 1424]
- F_VN_TrainSampleScalar [▶ 1426]
- F_VN_TrainSampleVector [▶ 1427]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.52 F_VN_TrainBatchClusters

Train a clusterer with a batch of samples and return the IDs of the clusters the samples have been assigned to, if requested. On-line trainable clusterers are trained once with each sample. Depending on the application and the number of available training samples, repeated training of such models with the same data may improve the results.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_TrainBatchClusters : HRESULT
VAR_INPUT
    ipClusterer : ITcVnMlModel;
    ipSamples   : ITcUnknown;
    ipClusters  : Reference To ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipClusterer	ITcVnMlModel [▶ 399]	Clusterer to be used
ipSamples	ITcUnknown [▶ 407]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
ipClusters	Reference To ITcVnContainer [▶ 349]	Returns the IDs of the clusters the samples have been assigned to (ContainerType_Vector_DINT; optional, set to 0 if not required)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_TrainBatchClusters 使用多个样本训练聚类器模型。与 F_VN_TrainBatch [▶ 1418] 相比，还可以为每个样本返回分配的聚类。

参数

聚类器模型

先前创建聚类器模型必须传输到 ipClusterer。

样本

样本在一个容器中被传输到 ipSamples。容器类型必须是 ContainerType_Vector_Vector_REAL 或 ContainerType_Vector_Vector_LREAL。

聚类

分配聚类的 ID 通过引用 ipClusters 在一个 ContainerType_Vector_DINT 类型的容器中返回。

应用

例如，聚类器模型可如下训练：

```
hr := F_VN_TrainBatchClusters(
    ipClusterer := ipClusterModel,
    ipSamples   := ipSamples,
    ipClusters  := ipClusters,
    hrPrev      := hr);

hr := F_VN_GetAt_DINT(ipClusters, nClusterOfThirdSample, 2, hr);
```

F_VN_GetAt_DINT 的调用可显示如何在训练后直接查询第 3 个样本的聚类分配。

相关函数

- F_VN_TrainBatch [▶ 1418]
- F_VN_TrainSample [▶ 1421]
- F_VN_TrainSampleClass [▶ 1423]
- F_VN_TrainSampleCluster [▶ 1424]
- F_VN_TrainSampleScalar [▶ 1426]
- F_VN_TrainSampleVector [▶ 1427]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.53 F_VN_TrainSample

Train a clusterer or a novelty detector with a single sample.

Syntax

Definition:

```
FUNCTION F_VN_TrainSample : HRESULT
VAR_INPUT
    ipMlModel  : ITcVnMlModel;
    ipSample   : ITcUnknown;
    hrPrev     : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipMlModel	ITcVnMlModel [▶ 399]	Clusterer or novelty detector to be used
ipSample	ITcUnknown [▶ 407]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_TrainSample 使用样本训练聚类或新颖性 检测模型。

参数

模型

先前创建模型必须传输到 ipMlModel。

样本

样本容器传输到 ipSample。容器类型必须是 ContainerType_Vector_REAL 或 ContainerType_Vector_LREAL。

应用

例如，可以如下使用样本训练ML 模型：

```
hr := F_VN_TrainSample(
    ipMlModel    := ipMlModel,
    ipSample     := ipSample,
    hrPrev       := hr);
```

相关函数

- F_VN_TrainBatch [▶ 1418]
- F_VN_TrainBatchClusters [▶ 1420]
- F_VN_TrainSampleClass [▶ 1423]
- F_VN_TrainSampleCluster [▶ 1424]
- F_VN_TrainSampleScalar [▶ 1426]
- F_VN_TrainSampleVector [▶ 1427]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.54 F_VN_TrainSampleClass

Train a classifier with a single sample.

Syntax

Definition:

```
FUNCTION F_VN_TrainSampleClass : HRESULT
VAR_INPUT
    ipClassifier : ITcVnMlModel;
    ipSample     : ITcUnknown;
    nClass       : DINT;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipClassifier	ITcVnMlModel [▸ 399]	Classifier to be used
ipSample	ITcUnknown [▸ 407]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
nClass	DINT	Class label to be learnt
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_TrainSampleClass 使用样本训练分类器模型。

参数

分类器模型

先前创建分类器模型必须传输到 ipClassifier。

样本

样本容器传输到 ipSample。容器类型必须是 ContainerType_Vector_REAL 或 ContainerType_Vector_LREAL。

类别

样本的目标类别作为 nClass 转移。

应用

例如，分类器模型可以使用单个样本如下进行训练：

```
hr := F_VN_TrainSampleClass(
    ipClassifier      := ipMlModel,
    ipSample          := ipSample,
    nClass            := nClass,
    hrPrev            := hr);
```

相关函数

- F_VN_TrainBatch [▶ 1418]
- F_VN_TrainBatchClusters [▶ 1420]
- F_VN_TrainSample [▶ 1421]
- F_VN_TrainSampleCluster [▶ 1424]
- F_VN_TrainSampleScalar [▶ 1426]
- F_VN_TrainSampleVector [▶ 1427]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.55 F_VN_TrainSampleCluster

Train a clusterer with a single sample and return the ID of the cluster the sample has been assigned to, if requested. The clusterer must be online trainable.

Syntax

Definition:

```
FUNCTION F_VN_TrainSampleCluster : HRESULT
VAR_INPUT
    ipClusterer : ITcVnMlModel;
    ipSample    : ITcUnknown;
    nCluster    : Reference To DINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipClusterer	ITcVnMlModel [▶ 399]	Clusterer to be used
ipSample	ITcUnknown [▶ 407]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
nCluster	Reference To DINT	Returns the ID of the cluster the sample has been assigned to (If the pointer is 0, no cluster ID is requested.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_TrainSampleCluster 使用样本训练聚类器模型，并返回样本的对应聚类 ID。

参数

聚类器模型

先前创建聚类器模型必须传输到 ipClusterer。

样本

样本容器作为 ipSample 传输。容器类型必须是 ContainerType_Vector_REAL 或 ContainerType_Vector_LREAL。

聚类

与样本相关的聚类 ID 通过 nCluster 返回。

应用

例如，一个聚类器模型可以使用单个样本如下进行训练：

```
hr := F_VN_TrainSampleCluster(
    ipClusterer      := ipClusterModel,
    ipSample         := ipSample,
    nCluster         := nCluster,
    hrPrev           := hr);
```

相关函数

- F_VN_TrainBatch [▶ 1418]
- F_VN_TrainBatchClusters [▶ 1420]
- F_VN_TrainSample [▶ 1421]
- F_VN_TrainSampleClass [▶ 1423]
- F_VN_TrainSampleScalar [▶ 1426]
- F_VN_TrainSampleVector [▶ 1427]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.56 F_VN_TrainSampleScalar

Train a regressor with a single sample and scalar output.

Syntax

Definition:

```
FUNCTION F_VN_TrainSampleScalar : HRESULT
VAR_INPUT
    ipRegressor : ITcVnMlModel;
    ipSample    : ITcUnknown;
    fOutput     : REAL;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipRegressor	ITcVnMlModel [▶ 399]	Regressor to be used
ipSample	ITcUnknown [▶ 407]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
fOutput	REAL	Scalar output to be learnt
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_TrainSampleScalar 使用样本训练标量回归子模型。

参数

回归子模型

先前创建回归子模型必须传输到 ipRegressor。

样本

样本容器作为 ipSample 传输。容器类型必须是 ContainerType_Vector_REAL 或 ContainerType_Vector_LREAL。

训练值

样本的结果值被传输到 fOutput。

应用

例如，一个标量回归子模型可以使用单个样本如下进行训练：

```
hr := F_VN_TrainSampleScalar(
    ipRegressor      := ipRegressorModel,
    ipSample         := ipSample,
    fOutput          := fOutput,
    hrPrev           := hr);
```

相关函数

- F_VN_TrainBatch [▸ 1418]
- F_VN_TrainBatchClusters [▸ 1420]
- F_VN_TrainSample [▸ 1421]
- F_VN_TrainSampleClass [▸ 1423]
- F_VN_TrainSampleCluster [▸ 1424]
- F_VN_TrainSampleVector [▸ 1427]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.19.57 F_VN_TrainSampleVector

Train a regressor with a single sample and vectorial output.

Syntax

Definition:

```
FUNCTION F_VN_TrainSampleVector : HRESULT
VAR_INPUT
    ipRegressor : ITcVnMlModel;
    ipSample    : ITcUnknown;
    ipOutput    : ITcVnContainer;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipRegressor	ITcVnMlModel [▸ 399]	Regressor to be used
ipSample	ITcUnknown [▸ 407]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
ipOutput	ITcVnContainer [▸ 349]	Vectorial output to be learnt (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_TrainSampleVector 使用样本训练向量回归子模型。

参数

回归子模型

先前创建回归子模型必须传输到 ipRegressor。

样本

样本容器作为 ipSample 传输。容器类型必须是 ContainerType_Vector_REAL 或 ContainerType_Vector_LREAL。

训练向量

样本的结果向量被传输到 ipOutput。

应用

例如，向量回归子模型可以使用单个样本如下进行训练：

```
hr := F_VN_TrainSampleVector(
    ipRegressor      := ipRegressorModel,
    ipSample         := ipSample,
    ipOutput         := ipVector,
    hrPrev           := hr);
```

相关函数

- F_VN_TrainBatch [▸ 1418]
- F_VN_TrainBatchClusters [▸ 1420]
- F_VN_TrainSample [▸ 1421]
- F_VN_TrainSampleClass [▸ 1423]
- F_VN_TrainSampleCluster [▸ 1424]
- F_VN_TrainSampleScalar [▸ 1426]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.20 Measurement

该组包含用于测量图像中对象的函数。

函数

边缘定位

- LocateCircularArc [▸ 1435]，用于定位比例圆弧
- LocateEdge [▸ 1441]，用于定位线性边缘
- LocateEdges [▸ 1447]，用于定位几个线性边缘

- LocateEllipse [▶ 1453]，用于定位完整椭圆

几何特征

- ClosestPointsBF [▶ 1434]，用于计算两个点云中最近的点
- MeasureAngleBetweenEdges [▶ 1458]，用于测量两条边之间的角度
- MeasureEdgeDistance [▶ 1463]，用于测量两条边之间的平均距离
- MeasureMinEdgeDistance [▶ 1468]，用于测量两条边之间的最小距离

其他

- F_VN_AdjustSearchWindowOrientationToLinearEdge [▶ 1432]用于将搜索区域对准边缘

边缘定位的参数

图像对象的边界往往并不完全位于两个像素之间的边界上。相反，它们独立于像素网格运行。因此，需要具有亚像素精度的边缘定位。这不仅考虑了两个像素之间的过渡，而且还根据对象边界的整体强度梯度确定了一个像素内的边缘位置。

注意
请勿在测量前处理图像 在调用测量函数之前，不要处理图像。如果处理，会导致信息损失，并相应降低准确性。

这种边缘定位的基本功能被所有测量功能所使用。函数的不同之处主要在于它们进行边缘定位的几何排列（例如：F_VN_LocateEllipse [▶ 1453]，用于圆形物体）和你从中计算的属性（例如：F_VN_MeasureAngleBetweenEdges [▶ 1458]，用于角度测量）。

必须为每个使用边缘定位的函数定义某些参数。其中包括：

- 搜索区域的定义：在哪里寻找边缘？
- 边缘对比度的定义：什么被检测为边缘？
- 测量算法的定义：如何确定边缘的确切位置。

这些参数在下面进行描述。在专家参数的情况下，另外还规定了标准值。

搜索区域

参数	描述
搜索线的位置	搜索线的位置根据待定位的特征进行定义。 线性边缘的搜索线的位置被定义为单线，起点aStartPoint，终点aEndPoint。 椭圆的搜索线的位置由中心点aCenter和半径fSearchRadius定义。 角度的搜索线的位置由一个内点aInnerPoint和两个外点aOuterPoint1和aOuterPoint2定义。 搜索线的位置应始终设定为搜索线正交地切割待定位的边缘。矩形搜索区域可以通过函数F_VN_AdjustSearchWindowOrientationToLinearEdge [▶ 1432]根据边缘以正交方式调整。
搜索线的数量 nSearchLines : UDINT	这指定了多少条搜索线用于边缘定位。最简单的边缘定位有一条搜索线，必须在上面找到边缘。根据待确定的几何特征，多条搜索线分布在局部，以便对几个边缘转换进行定位。 在以下情况下，可能十分必要： • 不仅需要位置，而且还需要边缘的梯度或旋转。 • 几何特征（如椭圆）需要几条搜索线。 • 需要一个平均值。 当定位线性边缘时，搜索线在中心搜索线的两侧对称排列，而中心搜索线被包括在nSearchLines中。因此，总是需要奇数数量的搜索线。
搜索线之间的距离 fSearchlineDist : REAL *（专家）* 默认值：1	搜索线距离定义了相邻搜索线之间的距离，单位：像素（>0，默认为 1）。这允许在准确性和计算时间之间进行灵活的平衡。
搜索方向反转 bInvertSearchDirection : BOOL *（专家）* 默认值：FALSE	定位算法始终从搜索线一端到另一端搜索边缘。例如，搜索从 aStartPoint 到 aEndPoint 进行。如果由于图像的特性需要反转这个搜索方向（即从搜索线的另一端开始），这个参数可以提供帮助。

边缘对比

参数	描述
边缘方向 eEdgeDirection : ETcVnEdgeDirection	边缘方向决定了应该找到从明到暗或从暗到明的边缘过渡。下列数值可用于此： • TCVN_ED_DARK_TO_LIGHT • TCVN_ED_LIGHT_TO_DARK
最小强度 fMinStrength : LREAL	最小强度定义了边缘内所需的最小强度差异，以便找到边缘。 从而可以控制边缘必须具有多大的对比度才能被发现。
最大厚度 nMaxThickness : UDINT *（专家）* 默认值：10	最大厚度规定了必须达到的fMinStrength像素数。 这使你能够控制是否只发现尖锐的边缘或还发现非常模糊的边缘。

算法

参数	描述
算法 eAlgorithm : ETcVnEdgeDetectionAlgorithm （专家） 默认值：TCVN_EDA_INTERPOLATION	该参数定义了所使用的定位算法。可用的选项如下： • TCVN_EDA_INTERPOLATION • TCVN_EDA_APPROX_ERF • TCVN_EDA_APPROX_GAUSSIAN TCVN_EDA_APPROX_ERF和TCVN_EDA_APPROX_GAUSSIAN比标准算法TCVN_EDA_INTERPOLATION速度慢，但可以达到更高的精度。然而，这样的前提条件是，边缘的强度梯度也尽可能地对应于指定的模型。误差函数erf或高斯函数被作为模型使用。 如果从亮到暗的过渡延伸到几个像素，并且分别在亮区或暗区的两边保持近似恒定，那么erf函数就很适合用于边缘。  高斯函数非常适用于线条，即暗-亮-暗的过渡，或者反之亦然。然而，亮度高原可能只有几个像素宽，否则应使用 erf 函数。 
子像素/迭代 nSubpixelsIterations : UDINT （专家） 默认值：10	这个参数的含义取决于所使用的定位算法。 对于TCVN_EDA_INTERPOLATION： 所考虑的子像素的数量定义了搜索线评估基础像素强度时使用的精细粒度。例如，如果数值为 10，则指定在 1 个像素的长度上取 10 个测量值。 建议：5 - 10 对于TCVN_EDA_APPROX_ERF和TCVN_EDA_APPROX_GAUSSIAN： 迭代次数是一个终止标准，定义了对像素强度的边缘模型进行近似的最大迭代次数。更多的迭代会增加运行时间；如果这个时间太高，应该使用插值法。 建议：50 - 100
近似精度 fApproxPrecision : REAL （专家） 默认值：0.001	近似精度是边缘定位的另一个中止标准。只有当定位算法是TCVN_EDA_APPROX_ERF或TCVN_EDA_APPROX_GAUSSIAN，才会观察到它。 如果达到nSubPixelIterations的迭代次数或连续两次迭代的像素相对偏差小于fApproximationPrecision，则中止对边缘或直线的近似。 建议：0.01 - 0.0001

HRESULT 的解释

HRESULT [▶ 122]返回值含义如下：

代码	名称	描述
16#000	S_OK	函数已成功执行，并且在所有搜索线上都找到了一条边。
16#001	S_FALSE	函数已成功执行，但没有在每条搜索线上找到边。
16#256	S_WATCHDOGTIMEOUT	函数被看门狗中止。在这种情况下，所有测量函数都可以返回部分结果。
16#70C	NOTFOUND	函数已成功执行，但在任何搜索线上都找不到边。假设只有在确保图像包含具有参数化边缘属性的物体时才会调用测量函数。如果没有找到任何一条搜索线，这表明参数化不正确或对函数的调用不合适。因此，在这种情况下会输出一个错误代码。
16#7xx	除 NOTFOUND 外的所有错误代码	函数未成功执行。另请参见：ADS 返回代码 [▶ 2753]

例如，以下查询可用于处理返回值：

```
IF SUCCEEDED(hr) THEN
    // Process results
    IF hr <> S_OK THEN
        // Emit warning that not all search lines were found
    END_IF
ELSIF hr = Tc2_System.E_HRESULTAdsErr.NOTFOUND THEN
    // Emit warning for wrong parametrization or missing piece
ELSE
    // Error handling
END_IF
```

缩短执行时间

固定搜索窗口的执行时间根据组件在图像中的位置而略有不同。如果找到的边离起点较近，执行时间就会比离起点较远的边短。

两种近似算法会出现较大的波动，因为这些算法需要不同的迭代，以便利用周围的像素强度近似模型参数。

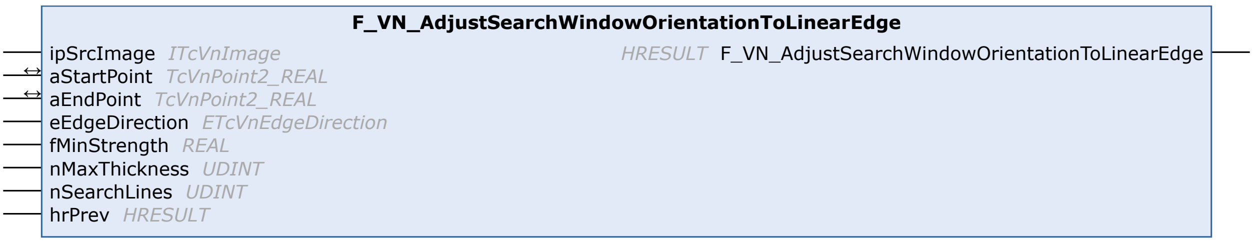
一般来说，如果搜索窗口的搜索线尽可能正交地击中边缘，则需要更少的迭代。如果图像中物体的位置和方向波动很小或没有波动，可以相应地调整搜索窗口以实现更短的执行时间。此外，可以通过减少最大迭代次数来减少所需的最大时间，但这可能会导致结果不太准确。

此外，可以使用外部看门狗限制时间。如有必要，看门狗会终止函数的执行并返回现有的部分结果。

样本

测量 [▶ 2695]

6.1.4.20.1 F_VN_AdjustSearchWindowOrientationToLinearEdge



Adjust the search window to be orthogonal to a linear edge. Might be helpful before calling edge localization and distance measurement functions that rely on the search window.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
FUNCTION F_VN_AdjustSearchWindowOrientationToLinearEdge : HRESULT
VAR_INPUT
    ipSrcImage      : ITcVnImage;
END_VAR
VAR_IN_OUT
    aStartPoint     : TcVnPoint2_REAL;
    aEndPoint       : TcVnPoint2_REAL;
END_VAR
```

```
VAR_INPUT
    eEdgeDirection : ETcVnEdgeDirection;
    fMinStrength   : REAL;
    nMaxThickness  : UDINT;
    nSearchLines   : UDINT;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image
eEdgeDirection	ETcVnEdgeDirection [▸ 178]	Specification of the edge direction to search for
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
nMaxThickness	UDINT	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels
nSearchLines	UDINT	Width of the search window, i.e. the number of search lines (3, 5, 7, ...), centered around the line specified by aStartPoint and aEndPoint
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aStartPoint	TcVnPoint2_REAL [▸ 139]	Position from which to start the search process (in the direction of aEndPoint), which is not changed
aEndPoint	TcVnPoint2_REAL [▸ 139]	Position where the search process ends, which is adjusted by this function

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_AdjustSearchWindowOrientationToLinearEdge旋转测量函数的搜索区域，使其与线性边缘正交。

参数

输入图像

输入图像ipSrcImage必须包含一个符合边缘检测参数的边缘。

搜索字段（返回值）

待调整搜索字段作为参考进行转移，以便可以调整。搜索字段包括中央搜索线（包括起点aStartPoint和终点aEndPoint）以及搜索线数量nSearchLines。

在函数调用之后，aStartPoint和aEndPoint包含调整后的中央搜索线。现在，这与输入图像ipSrcImage上的线性边缘正交。

边缘定位的参数

参数eEdgeDirection、fMinStrength和nMaxThickness决定了输入图像ipSrcImage中线性边缘的搜索特征。这些工作类似于F_VN_LocateEdge [▸ 1441]中的边缘定位。

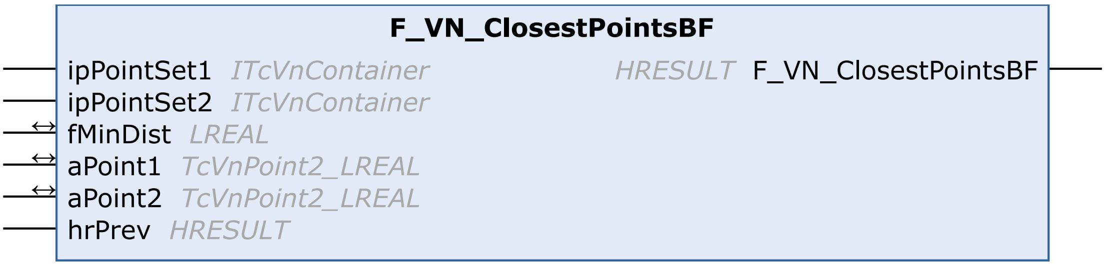
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.20.2 F_VN_ClosestPointsBF



Find the closest distance between two 2d point sets of the same type, using a brute force approach.

Syntax

Definition:

```
FUNCTION F_VN_ClosestPointsBF : HRESULT
VAR_INPUT
    ipPointSet1 : ITcVnContainer;
    ipPointSet2 : ITcVnContainer;
END_VAR
VAR_IN_OUT
    fMinDist    : LREAL;
    aPoint1     : TcVnPoint2_LREAL;
    aPoint2     : TcVnPoint2_LREAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipPointSet1	ITcVnContainer [▸ 349]	Container with 1st set of 2d points
ipPointSet2	ITcVnContainer [▸ 349]	Container with 2nd set of 2d points
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fMinDist	LREAL	Returns the minimum distance between 2 points of the different point sets
aPoint1	TcVnPoint2_LREAL [▸ 139]	Returns the point out of ipPointSet1, for which fMinDist is achieved
aPoint2	TcVnPoint2_LREAL [▸ 139]	Returns the point out of ipPointSet2, for which fMinDist is achieved

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_ClosestPointsBF定义两个点云中哪些点最接近对方。

参数

点云

在点云中寻找最接近的两个点ipPointSet1和ipPointSet2。

最小距离 （返回值）

参数fMinDist返回最近两点之间的距离。这对应于两个点云之间的最小距离。

最接近的点 （返回值）

参数aPoint1和aPoint2返回点云中最接近的两个点。

应用

例如，两条线之间最小距离的确定如下：

```
hr := F_VN_ClosestPointsBF(
    ipPointSet1 :=  ipLine1,
    ipPointSet2 :=  ipLine2,
    fMinDist    :=  fDistance,
    aPoint1     :=  aPoint1,
    aPoint2     :=  aPoint2,
    hrPrev      :=  hr
);
```

相关函数

- ClosestPointsBF [▶ 1434]，用于计算两个点云中最近的点
- MeasureAngleBetweenEdges [▶ 1458]，用于测量两条边之间的角度
- MeasureEdgeDistance [▶ 1463]，用于测量两条边之间的平均距离
- MeasureMinEdgeDistance [▶ 1468]，用于测量两条边之间的最小距离

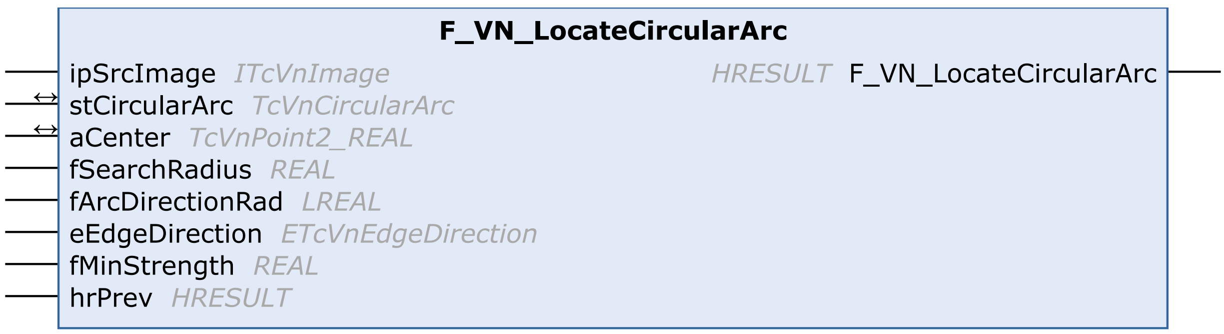
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.20.3 F_VN_LocateCircularArc



Locate a circular arc, using an interpolation method.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_LocateCircularArc : HRESULT
VAR_INPUT
    ipSrcImage        : ITcVnImage;
END_VAR
VAR_IN_OUT
    stCircularArc     : TcVnCircularArc;
    aCenter           : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    fSearchRadius     : REAL;
    fArcDirectionRad  : LREAL;
    eEdgeDirection    : ETcVnEdgeDirection;
    fMinStrength      : REAL;
    hrPrev            : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel)
fSearchRadius	REAL	Input search radius (starting from aCenter, should be greater than the actual circle radius but aCenter + fSearchRadius should be within the image borders)
fArcDirectionRad	LREAL	Input search starting direction in radian. Valid range is [-pi, +pi], where right is 0 rad, top is -pi/2 rad and bottom +pi/2 rad. The circular arc should at least be valid in range fArcDirectionRad +- 0.4 rad.
eEdgeDirection	ETcVnEdgeDirection [▶ 178]	Specification of the edge direction to search for
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stCircularArc	TcVnCircularArc [▶ 207]	Returns the circular arc parameters
aCenter	TcVnPoint2_REAL [▶ 139]	Input estimated circle center (only used as a starting point to search for the circle contour, not used for the circle center estimation)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_LocateCircularArc定位和测量一个比例圆弧。它是基于边缘定位 [▶ 1429]，并使用正常参数。它以 TcVnCircularArc（中心点、半径、角度范围）类型的结构返回定位的圆弧。

如果需要测量全圆而不是圆弧，请使用函数F_VN_LocateEllipse [▶ 1453]代替。

参数

输入图像

输入图像ipSrcImage必须为单通道灰度图像。

圆弧（返回值）

参数stCircularArc将定位的圆弧作为一个类型为TcVnCircularArc [▶ 207]的结构返回。

估计的中心点

必须指定圆弧的估计中心点aCenter，以确定搜索线的位置。

如果圆心点的位置未知，可能需要通过物体检测 [▶ 1137]事先确定。

搜索半径

搜索半径fSearchRadius定义了从圆心点开始的搜索线的长度，单位：像素。搜索半径的定义应该比圆的实际半径大一些，这样才能明确地进行边缘定位。

搜索方向

搜索方向fArcDirectionRad指定边缘搜索应开始的圆周方向（从圆心点开始）。该值以弧度为单位指定，且数值范围为 [-π， π]。右边对应 0，上面 -π/2，下面 π/2。从fArcDirectionRad开始，圆弧必须至少在 +-0.4 的范围内才能被检测到。建议以圆弧的中心为目标。

边缘定位的参数

关于其余参数的详细解释，请参见边缘定位 [▶ 1429]一章。

专家参数

专家版F_VN_LocateCircularArcExp [▶ 1438]包含额外的参数。

应用

例如，半径为270px且搜索角度为120° 的比例圆弧的定位和测量如下所示：

```
VAR
    stArc    :   TcVnCircularArc;
    aCenter :   TcVnPoint2_REAL := [420, 310];
END_VAR

hr := F_VN_LocateCircularArc(
    ipSrcImage          :=  ipImageIn,
    stCircularArc       :=  stArc,
    aCenterPoint        :=  aCenter,
    fSearchRadius       :=  270,
    fArcDirectionRad    :=  2.1, // 2.1 rad equals 120°
    eEdgeDirection      :=  TCVN_ED_LIGHT_TO_DARK,
    fMinStrength        :=  100,
    hrPrev              :=  hr
);
```

然后可以通过变量stArc.aCenter、stArc.fRadius、stArc.fStartAngle和stArc.fEndAngle检索已定位圆弧的属性。此外，圆弧可以通过函数F_VN_DrawCircularArc [▶ 963]实现可视化：

```
F_VN_DrawCircularArc(stArc, ipImageRes, aColorGreen, 5, hr);
```

样本

- 定位圆弧 [▶ 2695]

相关函数

- LocateCircularArc [▶ 1435]，用于定位比例圆弧
- LocateEdge [▶ 1441]，用于定位线性边缘
- LocateEdges [▶ 1447]，用于定位几个线性边缘
- LocateEllipse [▶ 1453]，用于定位完整椭圆

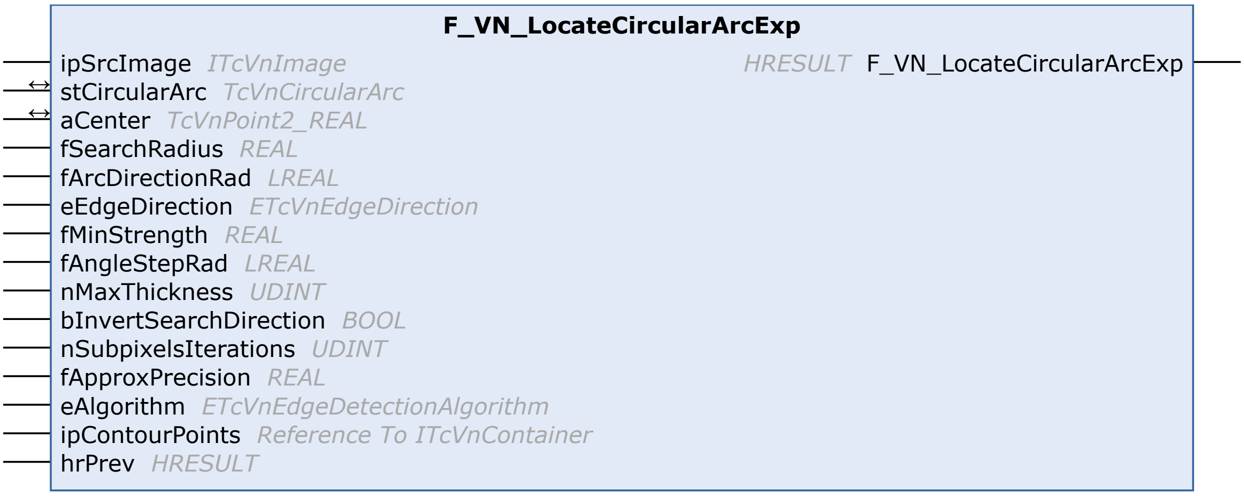
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.20.4 F_VN_LocateCircularArcExp



Locate a circular arc. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_LocateCircularArcExp : HRESULT
VAR_INPUT
    ipSrcImage               : ITcVnImage;
END_VAR
VAR_IN_OUT
    stCircularArc            : TcVnCircularArc;
    aCenter                  : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    fSearchRadius            : REAL;
    fArcDirectionRad         : LREAL;
    eEdgeDirection           : ETcVnEdgeDirection;
    fMinStrength             : REAL;
    fAngleStepRad            : LREAL;
    nMaxThickness            : UDINT;
    bInvertSearchDirection   : BOOL;
    nSubpixelsIterations     : UDINT;
    fApproxPrecision         : REAL;
    eAlgorithm               : ETcVnEdgeDetectionAlgorithm;
    ipContourPoints          : Reference To ITcVnContainer;
    hrPrev                   : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel)
fSearchRadius	REAL	Input search radius (starting from aCenter, should be greater than the actual circle radius but aCenter + fSearchRadius should be within the image borders)
fArcDirectionRad	LREAL	Input search starting direction in radian. Valid range is [-pi, +pi], where right is 0 rad, top is -pi/2 rad and bottom +pi/2 rad. The circular arc should at least be valid in range fArcDirectionRad +- 4 * fAngleStepRad)
eEdgeDirection	ETcVnEdgeDirection [▶ 178]	Specification of the edge direction to search for
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
fAngleStepRad	LREAL	Search step in rad (should be chosen so that about 20 - 60 steps are available for the whole arc. In most cases 0.1 rad (5.7 deg) is a good value)
nMaxThickness	UDINT	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels
bInvertSearchDirection	BOOL	If true, the search starts from outside the circular arc in direction of the center
nSubpixelsIterations	UDINT	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)
fApproxPrecision	REAL	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▶ 177]	Selection of the edge detection algorithm
ipContourPoints	Reference To ITcVnContainer [▶ 349]	Returns the subpixel accurate contour (optional, set to 0 if not required; ContainerType_Vector_TcVnPoint2_REAL; Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stCircularArc	TcVnCircularArc [▶ 207]	Returns the circular arc parameters
aCenter	TcVnPoint2_REAL [▶ 139]	Input estimated circle center (only used as a starting point to search for the circle contour, not used for the circle center estimation)

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_LocateCircularArcExp是F_VN_LocateCircularArc [▶ 1435]的专家版本。它包含额外的参数。

参数

输入图像

输入图像ipSrcImage必须为单通道灰度图像。

圆弧（返回值）

参数stCircularArc将定位的圆弧作为一个类型为TcVnCircularArc [▸ 207]的结构返回。

估计的中心点

必须指定圆弧的估计中心点aCenter，以确定搜索线的位置。

如果圆心点的位置未知，可能需要通过物体检测 [▸ 1137]事先确定。

搜索半径

搜索半径fSearchRadius定义了从圆心点开始的搜索线的长度，单位：像素。搜索半径的定义应该比圆的实际半径大一些，这样才能明确地进行边缘定位。

搜索方向

搜索方向fArcDirectionRad指定边缘搜索应开始的圆周方向（从圆心点开始）。该值以弧度为单位指定，且数值范围为 [-π， π]。右边对应 0，上面 -π/2，下面 π/2。从fArcDirectionRad开始，圆弧必须至少在 +-0.4 的范围内才能被检测到。建议以圆弧的中心为目标。

径向搜索线距离（专家）

径向搜索线距离fAngleStepRad定义了搜索线之间的角度步距有多大。它的单位是弧度。距离的选择应使整个圆弧被分为大约 20-60 步。在大多数情况下，0.1 rad （5.7°）是一个合适的值。

搜索方向反转（专家）

搜索方向可以通过布尔bInvertSearchDirection进行反转。在这种情况下，这意味着以下几点：

- FALSE：从圆心点到圆弧外搜索。
- TRUE：从圆弧外到圆心点的搜索

轮廓点（专家，返回值）

容器ipContourPoints返回找到的圆弧的边缘点，因此类型为ContainerType_Vector_TcVnPoint2_REAL。每成功找到一条搜索线容器中就包含一个点。

如果不需要返回边缘点，参数可以设置为0。

> **圆弧 stCircularArc**
>
> 圆弧 stCircularArc 由这些点的集合近似得出。

边缘定位的参数

关于其余参数的详细解释，请参见边缘定位 [▸ 1429]一章。

应用

例如，通过专家参数进行的圆弧定位如下所示：

```
hr := F_VN_LocateCircularArc(
    ipSrcImage          :=  ipImageIn,
    stCircularArc       :=  stArc,
    aCenterPoint        :=  aCenter,
    fSearchRadius       :=  270,
    fArcDirectionRad    :=  2.1,
    eEdgeDirection      :=  TCVN_ED_LIGHT_TO_DARK,
    fMinStrength        :=  100,
    fAngleStepRad       :=  40,
    nMaxThickness       :=  10,
    bInvertSearchDirection :=  FALSE,
    nSubpixelsIterations   :=  100,
    fApproxPrecision    :=  0.0001,
    eAlgorithm          :=  TCVN_EDA_APPROX_ERF,
```

```
    ipContourPoints    :=  ipEllipseContour,
    hrPrev             :=  hr
);
```

通过函数F_VN_DrawContours [▶ 967]，可将定位的点进行可视化处理。

```
hr := F_VN_DrawContours(ipEllipseContour, -1, ipImageRes, aColorGreen, 5, hr);
```

样本

- 定位圆弧 [▶ 2695]

相关函数

- LocateCircularArc [▶ 1435]，用于定位比例圆弧
- LocateEdge [▶ 1441]，用于定位线性边缘
- LocateEdges [▶ 1447]，用于定位几个线性边缘
- LocateEllipse [▶ 1453]，用于定位完整椭圆

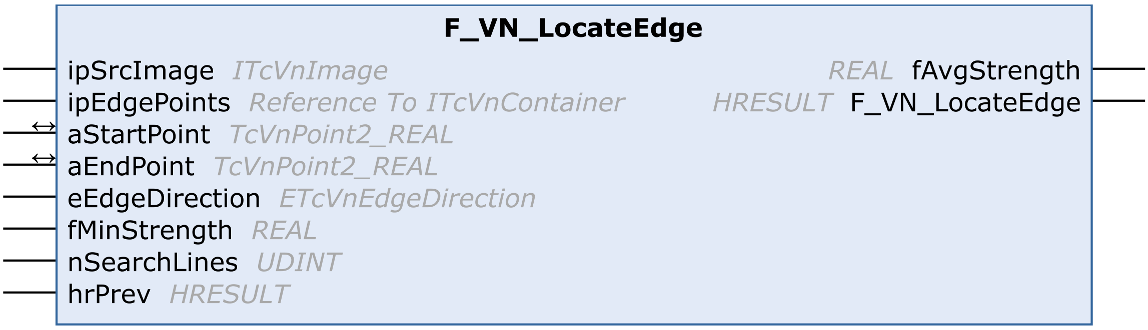
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.20.5 F_VN_LocateEdge



Locate the points of the first occuring edge inside a specified search window, using an interpolation method.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_LocateEdge : HRESULT
VAR_INPUT
    ipSrcImage      : ITcVnImage;
    ipEdgePoints    : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    aStartPoint     : TcVnPoint2_REAL;
    aEndPoint       : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    eEdgeDirection : ETcVnEdgeDirection;
    fMinStrength   : REAL;
    nSearchLines   : UDINT;
    hrPrev         : HRESULT;
END_VAR
```

```
VAR_OUTPUT
    fAvgStrength    : REAL;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel)
ipEdgePoints	Reference To ITcVnContainer [▶ 349]	Returns the detected edge points (ContainerType_Vector_TcVnPoint2_REAL)
eEdgeDirection	ETcVnEdgeDirection [▶ 178]	Specification of the edge direction to search for
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
nSearchLines	UDINT	Width of the search window, i.e. the number of search lines (1, 3, 5, 7, ...), centered around the line specified by aStartPoint and aEndPoint
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aStartPoint	TcVnPoint2_REAL [▶ 139]	Position from which to start the search process (in the direction of aEndPoint)
aEndPoint	TcVnPoint2_REAL [▶ 139]	Position where the search process ends

Outputs

Name	Type	Description
fAvgStrength	REAL	Returns the average strength of the detected edge

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_LocateEdge在定义的搜索窗口中定位一条边。

参数

输入图像

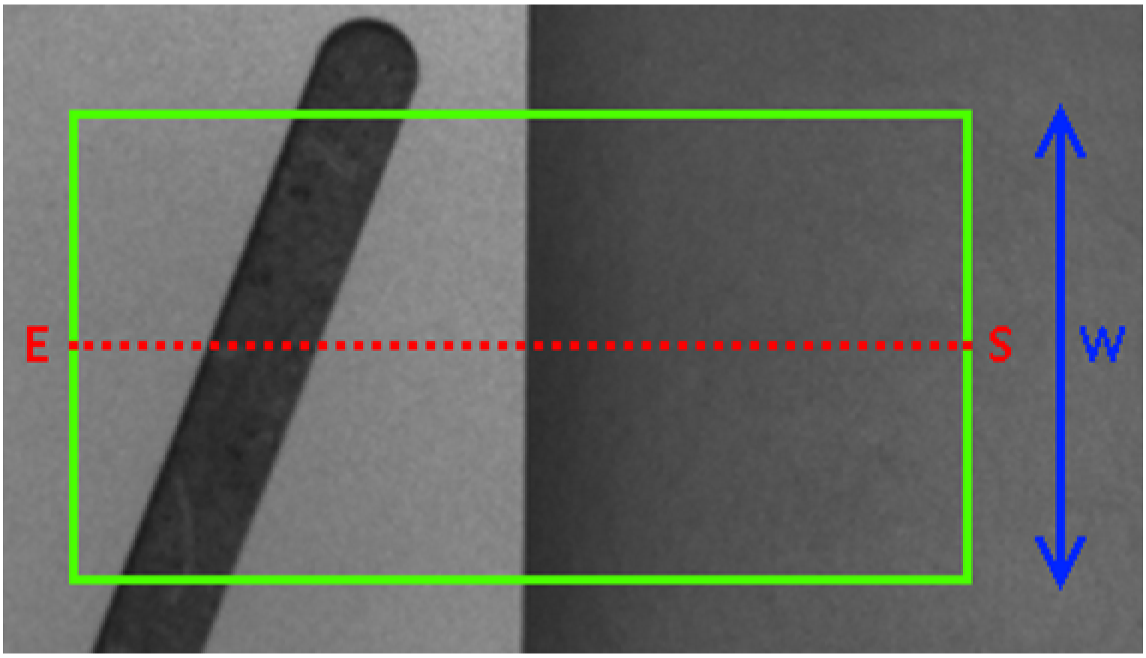
输入图像ipSrcImage必须为单通道灰度图像。

定位点（返回值）

容器ipEdgePoints返回线性边缘的定位点。

搜索窗口

首先，必须定义搜索窗口，以确定在哪里以及在哪个方向搜索边缘。通过参数 aStartPoint、aEndPoint 和 nSearchLines 完成。搜索窗口源自从 aStartPoint 到 aEndPoint 的搜索线并与之平行（nSearchLines - 1）/每个方向 2 条搜索线，均为 1 像素的距离。因此 nSearchLines 定义了搜索线的数量，且必须始终为奇数值。



附图 15: S：起始点，E：结束点，W：窗宽

边缘强度（返回值）

返回值fAvgStrength显示局部边缘的实际对比度（所有搜索线的平均数）。

边缘定位的参数

关于其余参数的详细解释，请参见边缘定位 [▸ 1429]一章。

专家参数

专家版F_VN_LocateEdgeExp包含额外的参数。

应用

例如，线性边缘的定位如下所示：

```
hr := F_VN_LocateEdge(
    ipSrcImage              :=  ipImageIn,
    ipEdgePoints            :=  ipEdgePoints,
    aStartPoint             :=  aStartPoint,
    aEndPoint               :=  aEndPoint,
    eEdgeDirection          :=  eDirection,
    fMinStrength            :=  fMinStrength,
    nSearchLines            :=  nSearchLines,
    hrPrev                  :=  hr,
    fAvgStrength            =>  fAvgStrength
);
```

所定位边缘点可以通过函数F_VN_DrawContours [▸ 967]进行可视化。

```
F_VN_DrawContours(ipEdgePoints, -1, ipImageRes, aColorGreen, 5, hr);
```

样本

- 定位边缘 [▸ 2699]

相关函数

- LocateCircularArc [▸ 1435]，用于定位比例圆弧
- LocateEdge [▸ 1441]，用于定位线性边缘
- LocateEdges [▸ 1447]，用于定位几个线性边缘
- LocateEllipse [▸ 1453]，用于定位完整椭圆

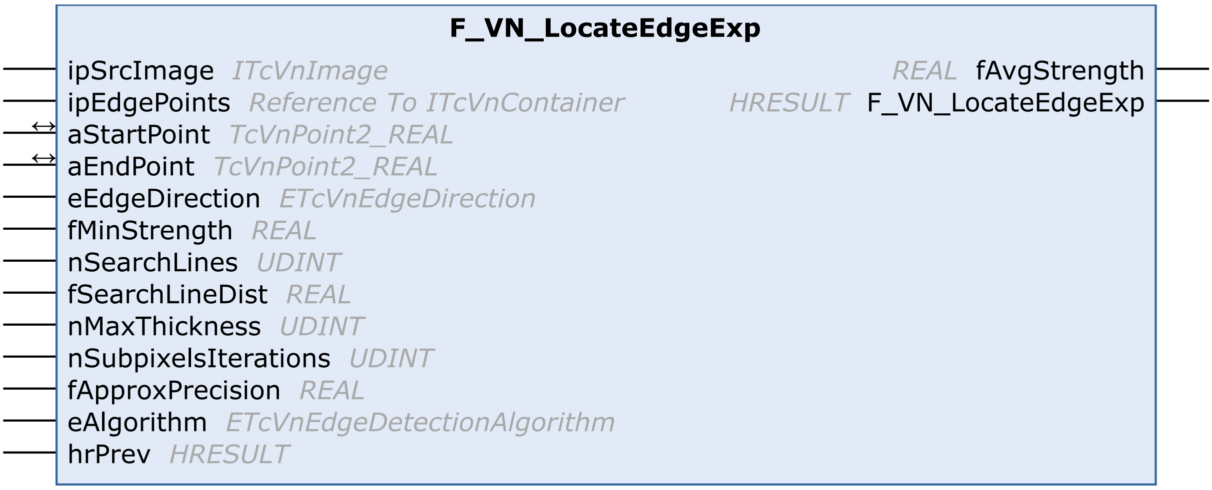
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.20.6 F_VN_LocateEdgeExp



Locate the points of the first occuring edge inside a specified search window with subpixel accuracy. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_LocateEdgeExp : HRESULT
VAR_INPUT
    ipSrcImage            : ITcVnImage;
    ipEdgePoints          : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    aStartPoint           : TcVnPoint2_REAL;
    aEndPoint             : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    eEdgeDirection        : ETcVnEdgeDirection;
    fMinStrength          : REAL;
    nSearchLines          : UDINT;
    fSearchLineDist       : REAL;
    nMaxThickness         : UDINT;
    nSubpixelsIterations  : UDINT;
    fApproxPrecision      : REAL;
    eAlgorithm            : ETcVnEdgeDetectionAlgorithm;
    hrPrev                : HRESULT;
END_VAR
VAR_OUTPUT
    fAvgStrength          : REAL;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel)
ipEdgePoints	Reference To ITcVnContainer [▶ 349]	Returns the detected edge points (ContainerType_Vector_TcVnPoint2_REAL)
eEdgeDirection	ETcVnEdgeDirection [▶ 178]	Specification of the edge direction to search for
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
nSearchLines	UDINT	Width of the search window, i.e. the number of search lines (1, 3, 5, 7, ...), centered around the line specified by aStartPoint and aEndPoint
fSearchLineDist	REAL	Distance between the search lines in pixels (> 0)
nMaxThickness	UDINT	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels
nSubpixelsIterations	UDINT	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)
fApproxPrecision	REAL	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▶ 177]	Selection of the edge detection algorithm
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aStartPoint	TcVnPoint2_REAL [▶ 139]	Position from which to start the search process (in the direction of aEndPoint)
aEndPoint	TcVnPoint2_REAL [▶ 139]	Position where the search process ends

Outputs

Name	Type	Description
fAvgStrength	REAL	Returns the average strength of the detected edge

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_LocateEdgeExp是F_VN_LocateEdge [▶ 1441]的专家版本。它包含额外的参数。

参数

输入图像

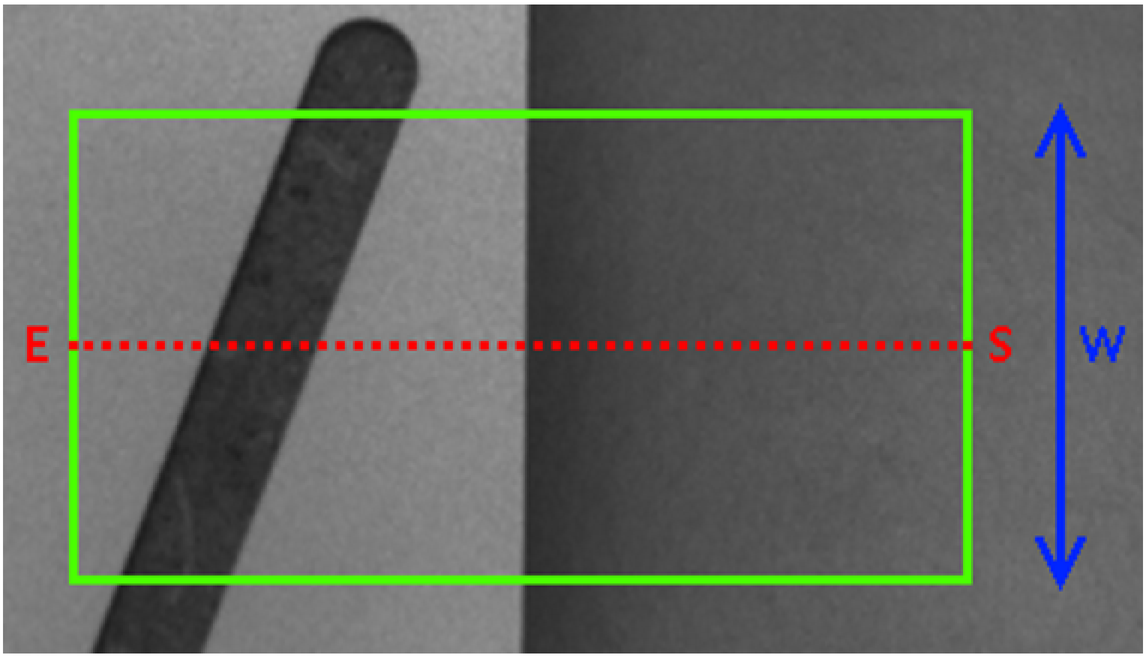
输入图像ipSrcImage必须为单通道灰度图像。

定位点（返回值）

容器ipEdgePoints返回线性边缘的定位点。

搜索窗口

首先，必须定义搜索窗口，以确定在哪里以及在哪个方向搜索边缘。通过参数 aStartPoint、aEndPoint 和 nSearchLines 完成。搜索窗口源自从 aStartPoint 到 aEndPoint 的搜索线并与之平行（nSearchLines – 1）/每个方向 2 条搜索线，均为 1 像素的距离。因此 nSearchLines 定义了搜索线的数量，且必须始终为奇数值。



附图 16: S: 起始点，E: 结束点，W: 窗宽

边缘强度（返回值）

返回值fAvgStrength显示局部边缘的实际对比度（所有搜索线的平均数）。

边缘定位的参数

关于其余参数的详细解释，请参见边缘定位 [▸ 1429]一章。

应用

例如，通过专家参数定位线性边缘如下所示：

```
hr := F_VN_LocateEdgeExp(
    ipSrcImage              :=  ipImageIn,
    ipEdgePoints            :=  ipEdgePoints,
    aStartPoint             :=  aStartPoint,
    aEndPoint               :=  aEndPoint,
    eEdgeDirection          :=  eDirection,
    fMinStrength            :=  fMinStrength,
    nSearchLines            :=  nSearchLines,
    fSearchLineDist         :=  fSearchLineDist
    nMaxThickness           :=  nMaxThickness,
    nSubpixelsIterations    :=  nSubpixIter,
    fApproxPrecision        :=  0.0001,
    eAlgorithm              :=  eAlgorithm,
    hrPrev                  :=  hr,
    fAvgStrength            =>  fAvgStrength
);
```

样本

- 定位边缘 [▸ 2699]

相关函数

- LocateCircularArc [▶ 1435]，用于定位比例圆弧
- LocateEdge [▶ 1441]，用于定位线性边缘
- LocateEdges [▶ 1447]，用于定位几个线性边缘
- LocateEllipse [▶ 1453]，用于定位完整椭圆

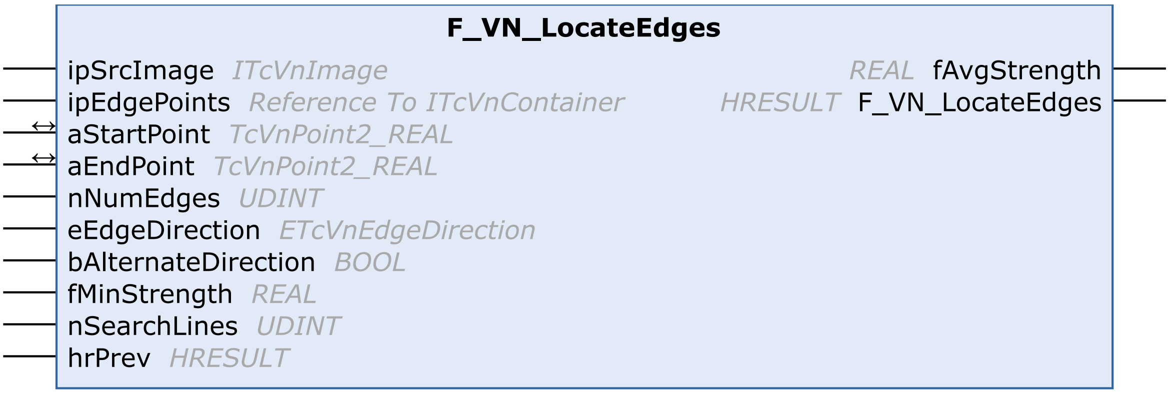
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.20.7 F_VN_LocateEdges



Locate the points of multiple occuring edges inside a specified search window, using an interpolation method.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_LocateEdges : HRESULT
VAR_INPUT
    ipSrcImage          : ITcVnImage;
    ipEdgePoints        : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    aStartPoint         : TcVnPoint2_REAL;
    aEndPoint           : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    nNumEdges           : UDINT;
    eEdgeDirection      : ETcVnEdgeDirection;
    bAlternateDirection : BOOL;
    fMinStrength        : REAL;
    nSearchLines        : UDINT;
    hrPrev              : HRESULT;
END_VAR
VAR_OUTPUT
    fAvgStrength        : REAL;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel)
ipEdgePoints	Reference To ITcVnContainer [▶ 349]	Returns the detected edge points (ContainerType_Vector_Vector_TcVnPoint2_REAL)
nNumEdges	UDINT	The (maximum) number of edges to search for
eEdgeDirection	ETcVnEdgeDirection [▶ 178]	Specification of the edge direction to search for
bAlternateDirection	BOOL	If true, eEdgeDirection is alternated after each detected edge. Else, only edges with eEdgeDirection are searched for.
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
nSearchLines	UDINT	Width of the search window, i.e. the number of search lines (1, 3, 5, 7, ...), centered around the line specified by aStartPoint and aEndPoint
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aStartPoint	TcVnPoint2_REAL [▶ 139]	Position from which to start the search process (in the direction of aEndPoint)
aEndPoint	TcVnPoint2_REAL [▶ 139]	Position where the search process ends

Outputs

Name	Type	Description
fAvgStrength	REAL	Returns the average strength of the detected edges

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_LocateEdges 在定义的搜索窗口中定位几个边。

参数

输入图像

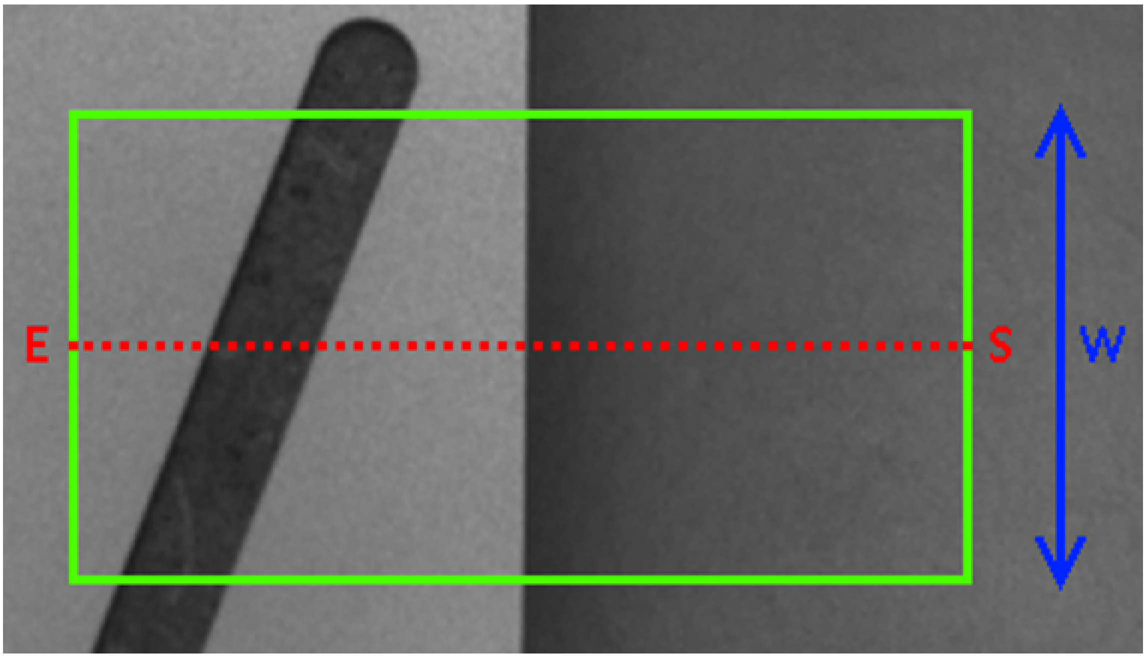
输入图像ipSrcImage必须为单通道灰度图像。

定位点（返回值）

容器ipEdgePoints返回线性边缘的定位点。

搜索窗口

首先，必须定义搜索窗口，以确定在哪里以及在哪个方向搜索边缘。通过参数 aStartPoint、aEndPoint 和 nSearchLines 完成。搜索窗口源自从 aStartPoint 到 aEndPoint 的搜索线并与之平行（nSearchLines - 1）/每个方向 2 条搜索线，均为 1 像素的距离。因此 nSearchLines 定义了搜索线的数量，且必须始终为奇数值。



附图 17: S：起始点，E：结束点，W：窗宽

边缘强度（返回值）

返回值fAvgStrength显示定位边缘的实际对比度（所有搜索线的平均数）。

边缘定位的参数

关于其余参数的详细解释，请参见边缘定位 [▶ 1429]一章。

专家参数

专家版 F_VN_LocateEdgeExp 包含额外的参数。

应用

例如，线性边缘的定位如下所示：

```
hr := F_VN_LocateEdges(
    ipSrcImage              :=  ipImageIn,
    ipEdgePoints            :=  ipEdgePoints,
    aStartPoint             :=  aStartPoint,
    aEndPoint               :=  aEndPoint,
    nNumEdges               :=  nNumEdges,
    eEdgeDirection          :=  eDirection,
    bAlternateDirection     :=  bAlternateDirection,
    fMinStrength            :=  fMinStrength,
    nSearchLines            :=  nSearchLines,
    hrPrev                  :=  hr,
    fAvgStrength            =>  fAvgStrength
);
```

所定位边缘点可以通过函数F_VN_DrawContours [▶ 968]进行可视化。

```
F_VN_DrawContours(ipEdgePoints, -1, ipImageRes, aColorGreen, 5, hr);
```

相关函数

- LocateCircularArc [▶ 1435]，用于定位比例圆弧
- LocateEdge [▶ 1441]，用于定位线性边缘
- LocateEdges [▶ 1447]，用于定位几个线性边缘
- LocateEllipse [▶ 1453]，用于定位完整椭圆

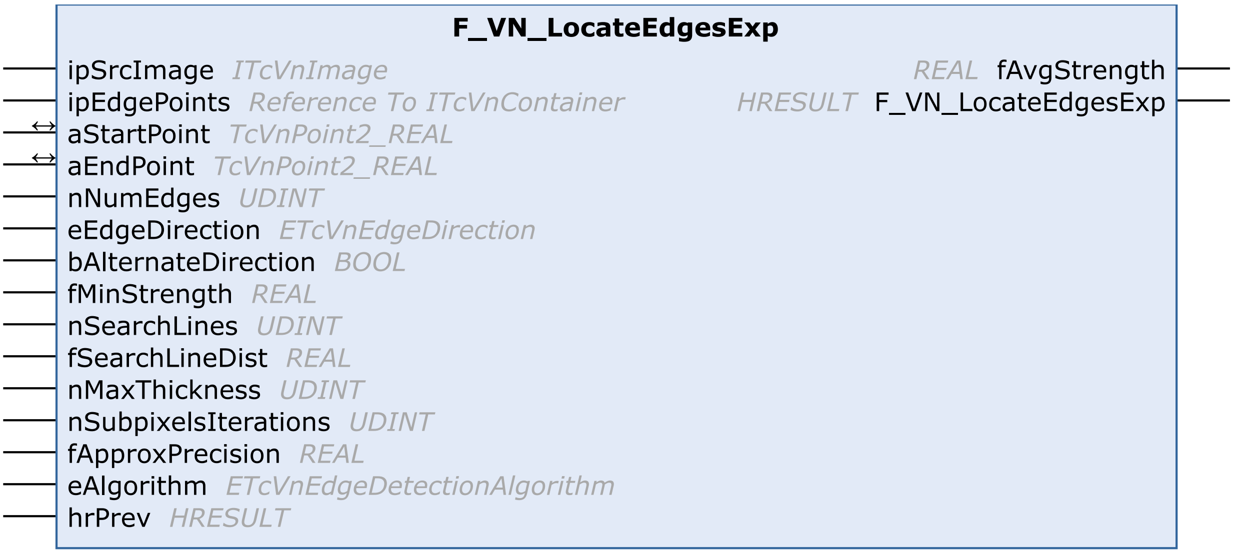
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.20.8 F_VN_LocateEdgesExp



Locate the points of multiple occuring edges inside a specified search window with subpixel accuracy. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_LocateEdgesExp : HRESULT
VAR_INPUT
    ipSrcImage            : ITcVnImage;
    ipEdgePoints          : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    aStartPoint           : TcVnPoint2_REAL;
    aEndPoint             : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    nNumEdges             : UDINT;
    eEdgeDirection        : ETcVnEdgeDirection;
    bAlternateDirection   : BOOL;
    fMinStrength          : REAL;
    nSearchLines          : UDINT;
    fSearchLineDist       : REAL;
    nMaxThickness         : UDINT;
    nSubpixelsIterations  : UDINT;
    fApproxPrecision      : REAL;
    eAlgorithm            : ETcVnEdgeDetectionAlgorithm;
    hrPrev                : HRESULT;
END_VAR
VAR_OUTPUT
    fAvgStrength          : REAL;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▸ 390]	Source image (1 channel)
ipEdgePoints	Reference To ITcVnContainer [▸ 349]	Returns the detected edge points (ContainerType_Vector_Vector_TcVnPoint2_REAL)
nNumEdges	UDINT	The (maximum) number of edges to search for
eEdgeDirection	ETcVnEdgeDirection [▸ 178]	Specification of the edge direction to search for
bAlternateDirection	BOOL	If true, eEdgeDirection is alternated after each detected edge. Else, only edges with eEdgeDirection are searched for.
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
nSearchLines	UDINT	Width of the search window, i.e. the number of search lines (1, 3, 5, 7, ...), centered around the line specified by aStartPoint and aEndPoint
fSearchLineDist	REAL	Distance between the search lines in pixels (> 0)
nMaxThickness	UDINT	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels
nSubpixelsIterations	UDINT	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)
fApproxPrecision	REAL	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▸ 177]	Selection of the edge detection algorithm
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aStartPoint	TcVnPoint2_REAL [▸ 139]	Position from which to start the search process (in the direction of aEndPoint)
aEndPoint	TcVnPoint2_REAL [▸ 139]	Position where the search process ends

Outputs

Name	Type	Description
fAvgStrength	REAL	Returns the average strength of the detected edges

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_LocateEdgesExp 是F_VN_LocateEdges [▸ 1447]的专家版本。它包含额外的参数。

参数

输入图像

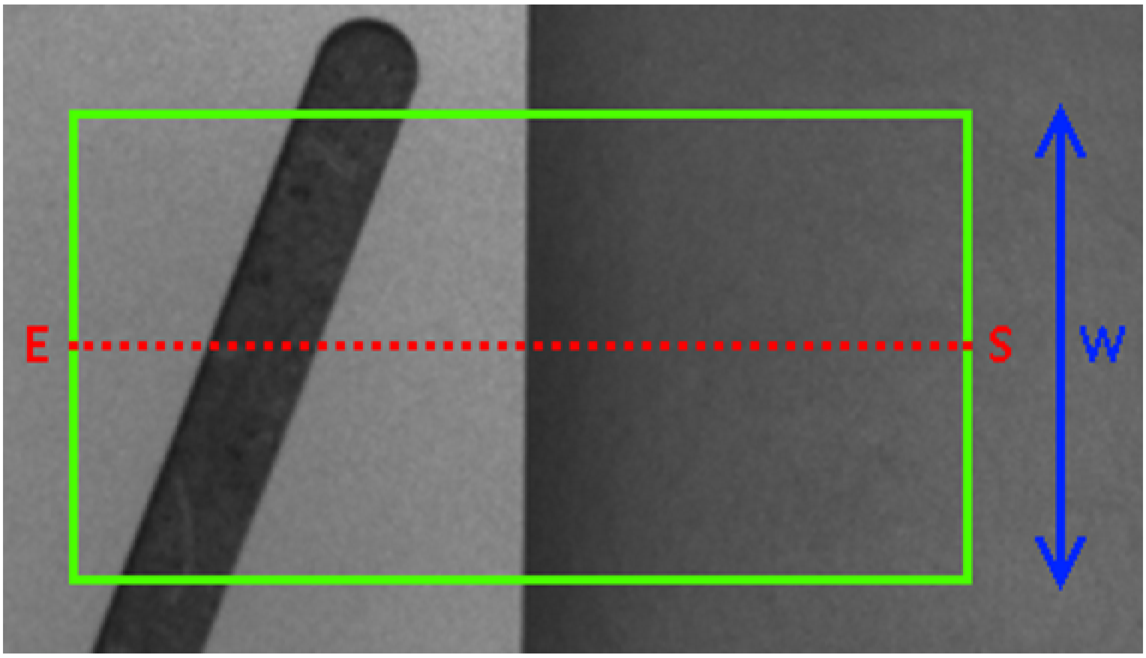
输入图像ipSrcImage必须为单通道灰度图像。

定位点（返回值）

容器ipEdgePoints返回线性边缘的定位点。

搜索窗口

首先，必须定义搜索窗口，以确定在哪里以及在哪个方向搜索边缘。通过参数 aStartPoint、aEndPoint 和 nSearchLines 完成。搜索窗口源自从 aStartPoint 到 aEndPoint 的搜索线并与之平行（nSearchLines - 1）/每个方向 2 条搜索线，均为 1 像素的距离。因此 nSearchLines 定义了搜索线的数量，且必须始终为奇数值。



附图 18: S: 起始点，E: 结束点，W: 窗宽

边缘强度（返回值）

返回值fAvgStrength显示定位边缘的实际对比度（所有搜索线的平均数）。

边缘定位的参数

关于其余参数的详细解释，请参见边缘定位 [▶ 1429]一章。

应用

例如，通过专家参数定位线性边缘如下所示：

```
hr := F_VN_LocateEdgesExp(
    ipSrcImage                  :=  ipImageIn,
    ipEdgePoints                :=  ipEdgePoints,
    aStartPoint                 :=  aStartPoint,
    aEndPoint                   :=  aEndPoint,
    nNumEdges                   :=  nNumEdges,
    eEdgeDirection              :=  eDirection,
    bAlternateDirection         :=  bAlternateDirection,
    fMinStrength                :=  fMinStrength,
    nSearchLines                :=  nSearchLines,
    fSearchLineDist             :=  fSearchLineDist
    nMaxThickness               :=  nMaxThickness,
    nSubpixelsIterations        :=  nSubpixIter,
    fApproxPrecision            :=  0.001,
    eAlgorithm                  :=  eAlgorithm,
```

```
    hrPrev                  :=  hr,
    fAvgStrength            =>  fAvgStrength
);
```

相关函数

- LocateCircularArc [▶ 1435]，用于定位比例圆弧
- LocateEdge [▶ 1441]，用于定位线性边缘
- LocateEdges [▶ 1447]，用于定位几个线性边缘
- LocateEllipse [▶ 1453]，用于定位完整椭圆

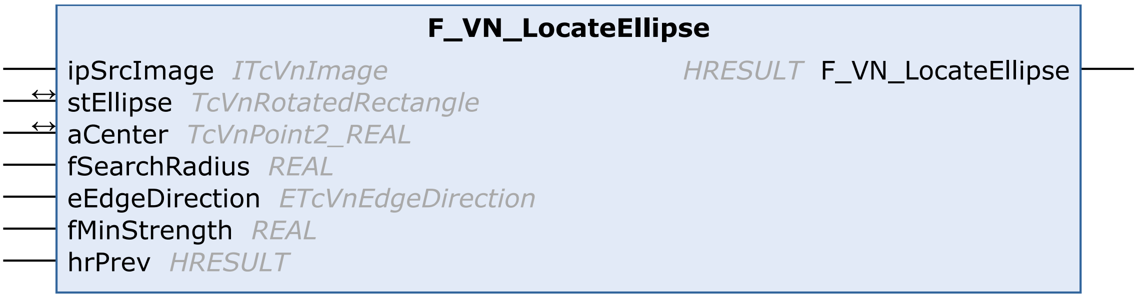
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.20.9 F_VN_LocateEllipse



Locate an ellipse, using an interpolation method for locating the edges.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_LocateEllipse : HRESULT
VAR_INPUT
    ipSrcImage      : ITcVnImage;
END_VAR
VAR_IN_OUT
    stEllipse       : TcVnRotatedRectangle;
    aCenter         : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    fSearchRadius   : REAL;
    eEdgeDirection  : ETcVnEdgeDirection;
    fMinStrength    : REAL;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel)
fSearchRadius	REAL	Radius around aCenter to search for edges (aCenter + fSearchRadius in all directions must be inside image borders!)
eEdgeDirection	ETcVnEdgeDirection [▶ 178]	Specification of the edge direction to search for (from center to outside ellipse)
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stEllipse	TcVnRotatedRectangle [▶ 225]	Returns the detected ellipse
aCenter	TcVnPoint2_REAL [▶ 139]	The expected ellipse center

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_LocateEllipse定位和测量椭圆。它是基于边缘定位 [▶ 1429]，并使用正常参数。它将定位的椭圆以类型TcVnRotatedRectangle [▶ 225]（中心点、大小和角度）的结构返回。

参数

输入图像

输入图像ipSrcImage必须为单通道灰度图像。

椭圆（返回值）

参数 stEllipse 将定位的椭圆以类型TcVnRotatedRectangle [▶ 225]（中心点、大小和角度）的结构返回。

搜索圈

圆形搜索区域通过预期中心点aCenter和搜索半径fSearchRadius定义。搜索半径应设置得比预期的椭圆半径大一些，以便安全地定位椭圆。

边缘定位的参数

关于其余参数的详细解释，请参见边缘定位 [▶ 1429]一章。

专家参数

专家版F_VN_LocateEllipse(Exp) [▶ 1455]包含额外的参数。

应用

例如，通过估计中心点[360, 240]和半径100 px定位和测量椭圆如下所示：

```
VAR
    aGuessedCenter          :   TcVnPoint2_REAL := [360, 240];
    fGuessedRadius          :   REAL := 100;
```

```
END_VAR

hr := F_VN_LocateEllipseExp(
    ipSrcImage              :=  ipImageIn,
    stEllipse               :=  stEllipse,
    aCenterPoint            :=  aGuessedCenter,
    fSearchRadius           :=  (fGuessedRadius + 15), // increase search radius to make sure
ellipse is found!
    eEdgeDirection          :=  TCVN_ED_LIGHT_TO_DARK,
    fMinStrength            :=  fMinStrength,
    hrPrev                  :=  hr
);
```

然后可以通过变量stEllipse.aCenter、stEllipse.stSize和stEllipse.fAngle检索所定位椭圆的属性。此外，椭圆可以通过函数F_VN_DrawEllipse [▸ 971]实现可视化：

```
hr := F_VN_DrawEllipse(stEllipse, ipImageRes, aColorGreen, 5, hr);
```

样本

- 定位椭圆 [▸ 2704]

相关函数

- LocateCircularArc [▸ 1435]，用于定位比例圆弧
- LocateEdge [▸ 1441]，用于定位线性边缘
- LocateEdges [▸ 1447]，用于定位几个线性边缘
- LocateEllipse [▸ 1453]，用于定位完整椭圆

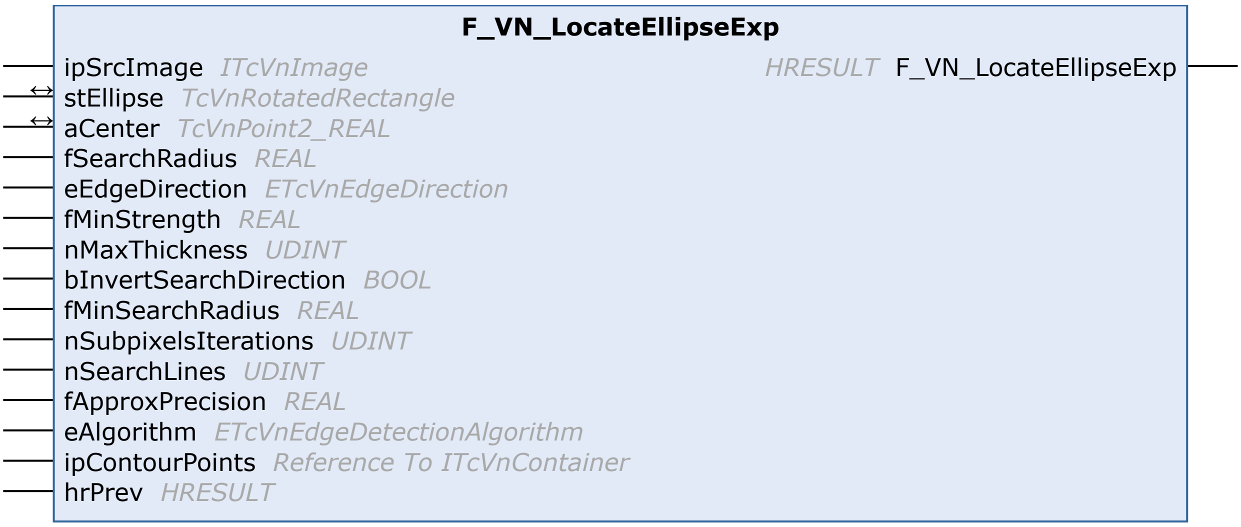
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.20.10 F_VN_LocateEllipseExp



Locate an ellipse. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_LocateEllipseExp : HRESULT
VAR_INPUT
    ipSrcImage              : ITcVnImage;
END_VAR
VAR_IN_OUT
    stEllipse               : TcVnRotatedRectangle;
    aCenter                 : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    fSearchRadius           : REAL;
    eEdgeDirection          : ETcVnEdgeDirection;
    fMinStrength            : REAL;
    nMaxThickness           : UDINT;
    bInvertSearchDirection  : BOOL;
    fMinSearchRadius        : REAL;
    nSubpixelsIterations    : UDINT;
    nSearchLines            : UDINT;
    fApproxPrecision        : REAL;
    eAlgorithm              : ETcVnEdgeDetectionAlgorithm;
    ipContourPoints         : Reference To ITcVnContainer;
    hrPrev                  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel)
fSearchRadius	REAL	Radius around aCenter to search for edges
eEdgeDirection	ETcVnEdgeDirection [▶ 178]	Specification of the edge direction to search for (from center to outside ellipse or other way round, if bInvertSearchDirection is true)
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
nMaxThickness	UDINT	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels
bInvertSearchDirection	BOOL	If true, the search starts from outside the ellipse in direction of the center
fMinSearchRadius	REAL	Radius around aCenter to skip before starting to search for edges (e.g. to save time or if the center contains edges that should be ignored)
nSubpixelsIterations	UDINT	Specifies the number of subpixels (for INTERPOLATION, 10 – 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 – 100 usually is sufficient)
nSearchLines	UDINT	Specifies the amount of search lines, which are equally distributed in all directions (must be >= 8 and a multiple of 4)
fApproxPrecision	REAL	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▶ 177]	Selection of the edge detection algorithm
ipContourPoints	Reference To ITcVnContainer [▶ 349]	Returns the subpixel accurate ellipse contour (optional, set to 0 if not required; ContainerType_Vector_TcVnPoint2_REAL; Non-zero interface pointers are reused.)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stEllipse	TcVnRotatedRectangle [▸ 225]	Returns the detected ellipse
aCenter	TcVnPoint2_REAL [▸ 139]	The expected ellipse center

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_LocateEllipseExp是F_VN_LocateEllipse [▸ 1453]的专家版本。

参数

输入图像

输入图像ipSrcImage必须为单通道灰度图像。

椭圆（返回值）

参数 stEllipse 将定位的椭圆以类型TcVnRotatedRectangle [▸ 225]（中心点、大小和角度）的结构返回。

搜索圈

圆形搜索区域通过预期中心点aCenter和搜索半径fSearchRadius定义。搜索半径应设置得比预期的椭圆半径大一些，以便安全地定位椭圆。

定位点（专家，返回值）

容器ipContourPoints返回定位的点。

如果不需要返回边缘点，该参数可以设置为 0。

边缘定位的参数

关于其余参数的详细解释，请参见边缘定位 [▸ 1429]一章。

应用

例如，通过专家参数定位和测量椭圆如下所示：

```
hr := F_VN_LocateEllipseExp(
    ipSrcImage                  :=  ipImageIn,
    stEllipse                   :=  stEllipse,
    aCenterPoint                :=  aGuessedCenter,
    fSearchRadius               :=  (fGuessedRadius + 15), // increase search radius to make sure 
ellipse is found!
    eEdgeDirection              :=  TCVN_ED_LIGHT_TO_DARK,
    fMinStrength                :=  fMinStrength,
    nMaxThickness               :=  7,
    bInvertSearchDirection      :=  TRUE,
    fMinSearchRadius            :=  fMinRadius,
    nSubpixelsIterations        :=  nSubpixIter,
    nSearchLines                :=  nSearchLines,
    fApproxPrecision            :=  0.0001,
    eAlgorithm                  :=  eAlgorithm,
    ipContourPoints             :=  ipContourPoints,
    hrPrev                      :=  hr
);
```

定位的椭圆点可以通过函数F_VN_DrawContours [▸ 967]实现可视化。

```
hr := F_VN_DrawContours(ipContourPoints, -1, ipImageRes, aColorGreen, 5, hr);
```

样本

- 定位椭圆 [▶ 2704]

相关函数

- LocateCircularArc [▶ 1435]，用于定位比例圆弧
- LocateEdge [▶ 1441]，用于定位线性边缘
- LocateEdges [▶ 1447]，用于定位几个线性边缘
- LocateEllipse [▶ 1453]，用于定位完整椭圆

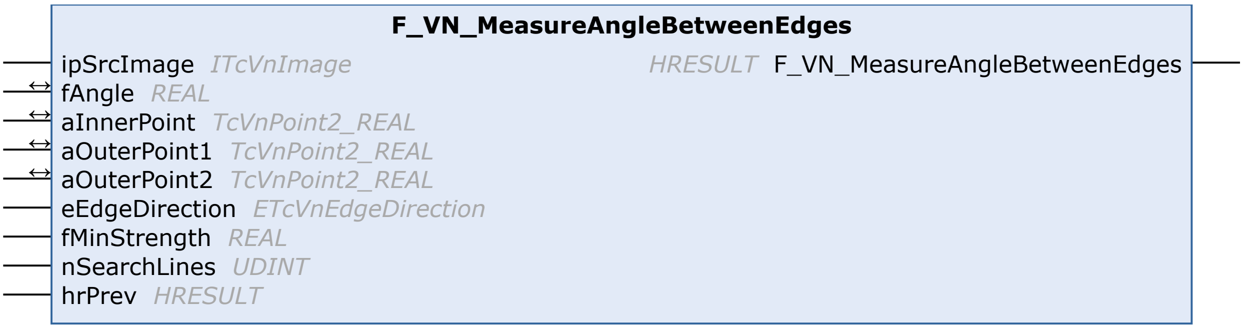
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.20.11 F_VN_MeasureAngleBetweenEdges



Measure the angle between 2 edges, using an interpolation method.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_MeasureAngleBetweenEdges : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
END_VAR
VAR_IN_OUT
    fAngle         : REAL;
    aInnerPoint    : TcVnPoint2_REAL;
    aOuterPoint1   : TcVnPoint2_REAL;
    aOuterPoint2   : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    eEdgeDirection : ETcVnEdgeDirection;
    fMinStrength   : REAL;
    nSearchLines   : UDINT;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel)
eEdgeDirection	ETcVnEdgeDirection [▶ 178]	Specification of the edge direction from aInnerPoint to aOuterPoint1 and aOuterPoint2 to search for
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
nSearchLines	UDINT	Width of the search windows, i.e. the number of search lines (3, 5, 7, ...), centered around the lines specified by aInnerPoint and aOuterPoint1, aOuterPoint2
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fAngle	REAL	Returns the angle between the detected edges (radians)
aInnerPoint	TcVnPoint2_REAL [▶ 139]	Position inside the angle to measure, from which to start the edge search process (in the direction of aOuterPoint1 and aOuterPoint2)
aOuterPoint1	TcVnPoint2_REAL [▶ 139]	Position where the search process for edge 1 ends
aOuterPoint2	TcVnPoint2_REAL [▶ 139]	Position where the search process for edge 2 ends

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_MeasureAngleBetweenEdges定位两个线性边缘并测量它们之间的角度。

参数

输入图像

输入图像ipSrcImage必须为单通道灰度图像。

角度（返回值）

参数fAngle返回定位边缘之间的角度。

根据参数bAngleInDegrees，角度fAngle以弧度（FALSE）或度数（TRUE）指定。

边缘定位的参数

关于其余参数的详细解释，请参见边缘定位 [▶ 1429]一章。

专家参数

专家版F_VN_MeasureAngleBetweenEdgesExp [▶ 1460]包含额外的参数。

应用

例如，两个线性边缘之间的角度测量如下所示：

```
hr := F_VN_MeasureAngleBetweenEdgesExp(
    ipSrcImage              :=    ipImageIn,
    fAngle                  :=    fAngle
```

```
    aInnerPoint            :=   aInnerPoint,
    aOuterPoint1           :=   aOuterPoint1
    aOuterPoint2           :=   aOuterPoint2,
    eEdgeDirection         :=   TCVN_ED_LIGHT_TO_DARK,
    fMinStrength           :=   fMinStrength,
    nSearchLines           :=   nSearchLines,
    hrPrev                 :=   hr
);
```

样本

- 测量边缘之间的角度 [▶ 2709]

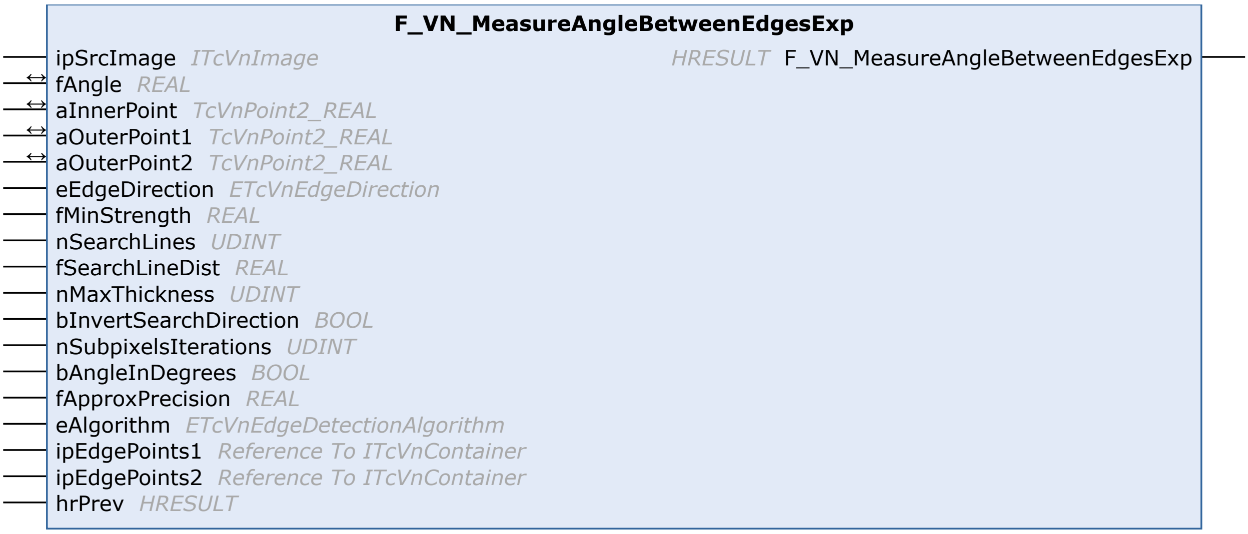
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.20.12 F_VN_MeasureAngleBetweenEdgesExp



Measure the angle between 2 edges.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_MeasureAngleBetweenEdgesExp : HRESULT
VAR_INPUT
    ipSrcImage              : ITcVnImage;
END_VAR
VAR_IN_OUT
    fAngle                  : REAL;
    aInnerPoint             : TcVnPoint2_REAL;
    aOuterPoint1            : TcVnPoint2_REAL;
    aOuterPoint2            : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    eEdgeDirection          : ETcVnEdgeDirection;
    fMinStrength            : REAL;
    nSearchLines            : UDINT;
    fSearchLineDist         : REAL;
    nMaxThickness           : UDINT;
    bInvertSearchDirection  : BOOL;
    nSubpixelsIterations    : UDINT;
```

```
    bAngleInDegrees         : BOOL;
    fApproxPrecision        : REAL;
    eAlgorithm              : ETcVnEdgeDetectionAlgorithm;
    ipEdgePoints1           : Reference To ITcVnContainer;
    ipEdgePoints2           : Reference To ITcVnContainer;
    hrPrev                  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel)
eEdgeDirection	ETcVnEdgeDirection [▶ 178]	Specification of the edge direction from aInnerPoint to aOuterPoint1 and aOuterPoint2 to search for
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
nSearchLines	UDINT	Width of the search window, i.e. the number of search lines (3, 5, 7, ...), centered around the line specified by aInnerPoint and aOuterPoint1, aOuterPoint2
fSearchLineDist	REAL	Distance between the search lines in pixels (> 0)
nMaxThickness	UDINT	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels
bInvertSearchDirection	BOOL	If true, the search starts from each aOuterPoint in direction of aInnerPoint
nSubpixelsIterations	UDINT	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)
bAngleInDegrees	BOOL	fAngle is in degrees, if true
fApproxPrecision	REAL	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▶ 177]	Selection of the edge detection algorithm
ipEdgePoints1	Reference To ITcVnContainer [▶ 349]	Returns the detected edge points between aInnerPoint and aOuterPoint1 (ContainerType_Vector_TcVnPoint2_REAL)
ipEdgePoints2	Reference To ITcVnContainer [▶ 349]	Returns the detected edge points between aInnerPoint and aOuterPoint2 (ContainerType_Vector_TcVnPoint2_REAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fAngle	REAL	Returns the angle between the detected edges (radians or degrees, depending on bAngleInDegrees)
aInnerPoint	TcVnPoint2_REAL [▶ 139]	Position inside the angle to measure, from which to start the edge search process (in the direction of aOuterPoint1 and aOuterPoint2)
aOuterPoint1	TcVnPoint2_REAL [▶ 139]	Position where the search process for edge 1 ends
aOuterPoint2	TcVnPoint2_REAL [▶ 139]	Position where the search process for edge 2 ends

Return value

HRESULT [▸ 122]

更多信息

函数F_VN_MeasureAngleBetweenEdgesExp是F_VN_MeasureAngleBetweenEdges [▸ 1458]的专家函数。

参数

输入图像

输入图像ipSrcImage必须为单通道灰度图像。

角度 （返回值）

参数fAngle返回定位边缘之间的角度。

根据参数bAngleInDegrees，角度fAngle以弧度（FALSE）或度数（TRUE）指定。

边缘定位的参数

关于其余参数的详细解释，请参见边缘定位 [▸ 1429]一章。

定位点 （专家，返回值）

容器ipEdgePoints1和ipEdgePoints2返回在每条搜索线基础上定位的确切点。如果不需要，可以传输值0，而不是接口指针

边缘定位的参数

关于其余参数的详细解释，请参见边缘定位 [▸ 1429]一章。

应用

例如，通过专家参数测量角度如下所示：

```
hr := F_VN_MeasureAngleBetweenEdgesExp(
    ipSrcImage              :=   ipImageIn,
    fAngle                  :=   fAngle
    aInnerPoint             :=   aInnerPoint,
    aOuterPoint1            :=   aOuterPoint1
    aOuterPoint2            :=   aOuterPoint2,
    eEdgeDirection          :=   TCVN_ED_LIGHT_TO_DARK,
    fMinStrength            :=   fMinStrength,
    nSearchLines            :=   nSearchLines,
    fSearchLineDist         :=   fSearchLineDist,
    nMaxThickness           :=   7,
    nSubpixelsIterations    :=   100,
    bAngleInDegrees         :=   TRUE,
    fApproxPrecision        :=   0.0001,
    eAlgorithm              :=   TCVN_EDA_APPROX_ERF,
    ipEdgePoints1           :=   ipEdgePoints1,
    ipEdgePoints2           :=   ipEdgePoints2,
    hrPrev                  :=   hr
);
```

样本

- 测量边缘之间的角度 [▸ 2709]

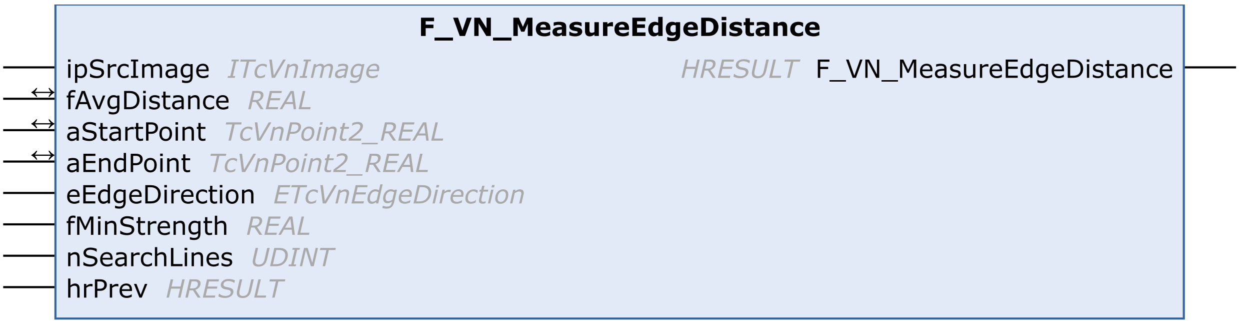
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.20.13 F_VN_MeasureEdgeDistance



Measure the distance between 2 parallel edges, using an interpolation method.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_MeasureEdgeDistance : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
END_VAR
VAR_IN_OUT
    fAvgDistance   : REAL;
    aStartPoint    : TcVnPoint2_REAL;
    aEndPoint      : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    eEdgeDirection : ETcVnEdgeDirection;
    fMinStrength   : REAL;
    nSearchLines   : UDINT;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel)
eEdgeDirectio n	ETcVnEdgeDirection [▶ 178]	Specification of the edge direction from aStartPoint to aEndPoint to search for
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
nSearchLines	UDINT	Width of the search window, i.e. the number of search lines (1, 3, 5, 7, ...), centered around the line specified by aStartPoint and aEndPoint
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fAvgDistance	REAL	Returns the average distance between the detected edges
aStartPoint	TcVnPoint2_REAL [▶ 139]	Position from which to start the search process (in the direction of aEndPoint)
aEndPoint	TcVnPoint2_REAL [▶ 139]	Position where the search process ends

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_MeasureEdgeDistance定位两条线性边缘，并测量它们之间的平均距离。

参数

输入图像

输入图像ipSrcImage必须为单通道灰度图像。

最小距离（返回值）

参数fAvgDistance返回两个定位边缘之间的最小距离。

边缘定位的参数

关于其余参数的详细解释，请参见边缘定位 [▶ 1429]一章。

专家参数

专家版F_VN_MeasureEdgeDistanceExp [▶ 1465]包含额外的参数。

应用

例如，边缘距离的测量如下所示：

```
hr := F_VN_MeasureEdgeDistance(
    ipSrcImage              :=   ipImageIn,
    fAvgDistance            :=   fDistance
    aStartPoint             :=   aStartPoint,
    aEndPoint               :=   aEndPoint,
    eEdgeDirection          :=   TCVN_ED_LIGHT_TO_DARK,
    fMinStrength            :=   fMinStrength,
    nSearchLines            :=   nSearchLines,
    hrPrev                  :=   hr
);
```

相关函数

- ClosestPointsBF [▶ 1434]，用于计算两个点云中最近的点
- MeasureAngleBetweenEdges [▶ 1458]，用于测量两条边之间的角度
- MeasureEdgeDistance [▶ 1463]，用于测量两条边之间的平均距离
- MeasureMinEdgeDistance [▶ 1468]，用于测量两条边之间的最小距离

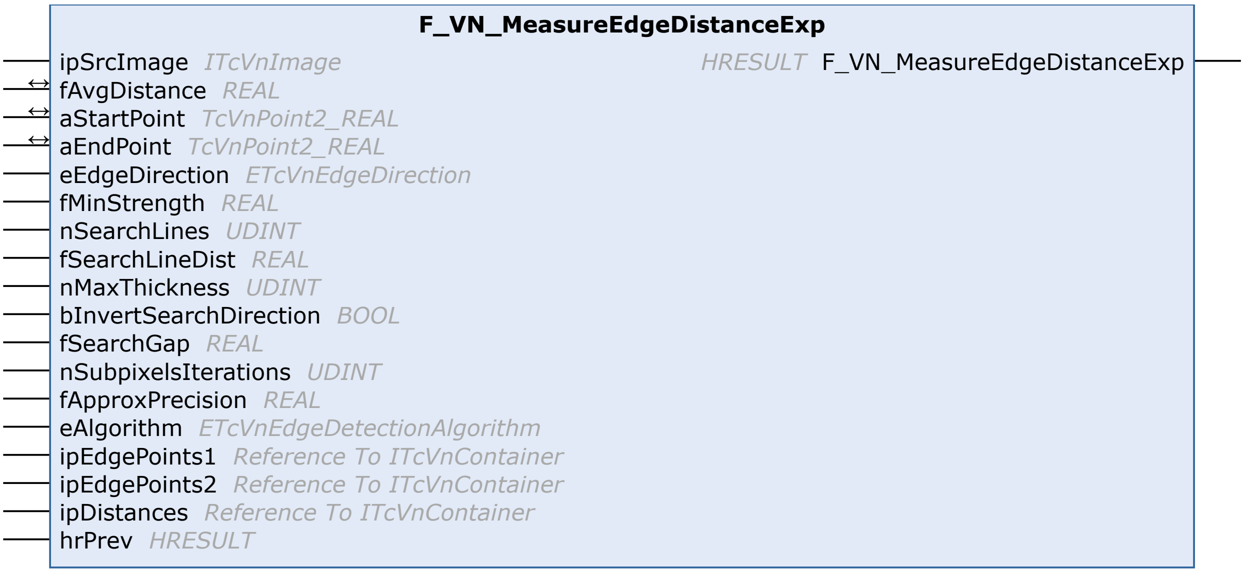
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.20.14 F_VN_MeasureEdgeDistanceExp



Measure the distance between 2 parallel edges. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_MeasureEdgeDistanceExp : HRESULT
VAR_INPUT
    ipSrcImage              : ITcVnImage;
END_VAR
VAR_IN_OUT
    fAvgDistance            : REAL;
    aStartPoint             : TcVnPoint2_REAL;
    aEndPoint               : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    eEdgeDirection          : ETcVnEdgeDirection;
    fMinStrength            : REAL;
    nSearchLines            : UDINT;
    fSearchLineDist         : REAL;
    nMaxThickness           : UDINT;
    bInvertSearchDirection  : BOOL;
    fSearchGap              : REAL;
    nSubpixelsIterations    : UDINT;
    fApproxPrecision        : REAL;
    eAlgorithm              : ETcVnEdgeDetectionAlgorithm;
    ipEdgePoints1           : Reference To ITcVnContainer;
    ipEdgePoints2           : Reference To ITcVnContainer;
    ipDistances             : Reference To ITcVnContainer;
    hrPrev                  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel)
eEdgeDirection	ETcVnEdgeDirection [▶ 178]	Specification of the edge direction from aStartPoint to aEndPoint to search for
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
nSearchLines	UDINT	Width of the search window, i.e. the number of search lines (1, 3, 5, 7, ...), centered around the line specified by aStartPoint and aEndPoint
fSearchLineDist	REAL	Distance between the search lines in pixels (> 0)
nMaxThickness	UDINT	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels
bInvertSearchDirection	BOOL	If true, the search starts from the center point between aStartPoint and aEndPoint in both directions
fSearchGap	REAL	Optional width of a gap (>= 0, centered between aStartPoint and aEndPoint), that is neglected for searching edges (can reduce execution time). The 2 edges to search for must be on different sides of the gap.
nSubpixelsIterations	UDINT	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)
fApproxPrecision	REAL	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▶ 177]	Selection of the edge detection algorithm
ipEdgePoints1	Reference To ITcVnContainer [▶ 349]	Returns the detected edge points of the edge near aStartPoint (optional, set to 0 if not required; ContainerType_Vector_TcVnPoint2_REAL)
ipEdgePoints2	Reference To ITcVnContainer [▶ 349]	Returns the detected edge points of the edge near aEndPoint (optional, set to 0 if not required; ContainerType_Vector_TcVnPoint2_REAL)
ipDistances	Reference To ITcVnContainer [▶ 349]	Returns the distances between the detected edge points (optional, set to 0 if not required; ContainerType_Vector_REAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fAvgDistance	REAL	Returns the average distance between the detected edges
aStartPoint	TcVnPoint2_REAL [▶ 139]	Position from which to start the search process (in the direction of aEndPoint)
aEndPoint	TcVnPoint2_REAL [▶ 139]	Position where the search process ends

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_MeasureEdgeDistanceExp是F_VN_MeasureEdgeDistance [▶ 1463]的专家版本。它包含额外的参数。

参数

输入图像

输入图像ipSrcImage必须为单通道灰度图像。

最小距离（返回值）

参数fAvgDistance返回两个定位边缘之间的最小距离。

定位的边缘点（专家，返回值）

容器ipEdgePoints1和ipEdgePoints2返回在每条搜索线基础上定位的确切点。如果不需要，可以传输值0，而不是接口指针。

测量的距离（专家，返回值）

容器ipDistances返回相对点之间的所有测量距离。如果不需要，可以传输值0，而不是接口指针。

搜索差距（专家）

定义的搜索线中心的区域，在该区域内的搜索被省略。因此可以减少执行时间。

边缘定位的参数

关于其余参数的详细解释，请参见边缘定位 [▶ 1429]一章。

应用

例如，通过专家参数测量边缘距离如下所示：

```
hr := F_VN_MeasureEdgeDistance(
    ipSrcImage                  :=   ipImageIn,
    fAvgDistance                :=   fDistance
    aStartPoint                 :=   aStartPoint,
    aEndPoint                   :=   aEndPoint,
    eEdgeDirection              :=   TCVN_ED_LIGHT_TO_DARK,
    fMinStrength                :=   fMinStrength,
    nSearchLines                :=   nSearchLines,
    fSearchLineDist             :=  3
    nMaxThickness               :=  7,
    bInvertSearchDirection      :=  TRUE,
    nSubpixelsIterations        :=  nSubpixIter,
    fApproxPrecision            :=  0.0001,
    eAlgorithm                  :=  eAlgorithm,
    ipEdgePoints1               :=  ipEdge1,
    ipEdgePoints2               :=  ipEdge2,
    ipDistances                 :=  ipDistances,
    hrPrev                     :=   hr
);
```

所定位边缘点可以通过函数F_VN_DrawContours [▶ 967]进行可视化。

```
hr := F_VN_DrawContours(ipEdgePoints1, -1, ipImageRes, aColorGreen, 5, hr);
hr := F_VN_DrawContours(ipEdgePoints2, -1, ipImageRes, aColorGreen, 5, hr);
```

相关函数

- ClosestPointsBF [▶ 1434]，用于计算两个点云中最近的点
- MeasureAngleBetweenEdges [▶ 1458]，用于测量两条边之间的角度
- MeasureEdgeDistance [▶ 1463]，用于测量两条边之间的平均距离
- MeasureMinEdgeDistance [▶ 1468]，用于测量两条边之间的最小距离

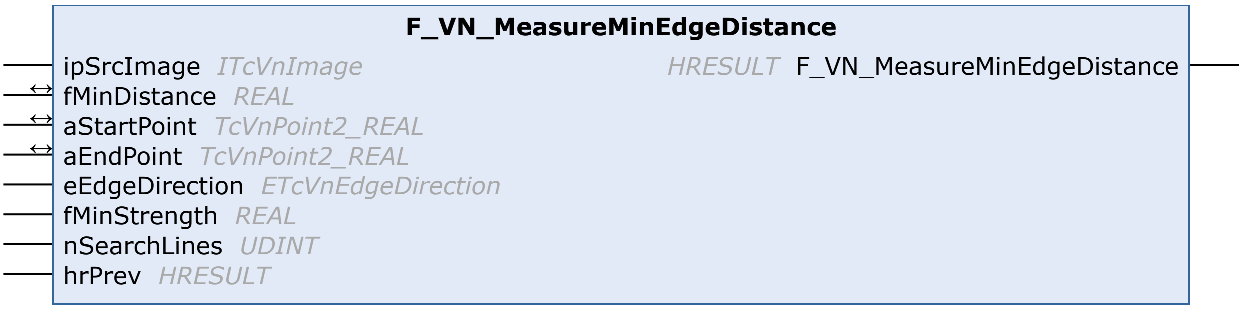
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.20.15 F_VN_MeasureMinEdgeDistance



Measure the minimum distance within the specified search window between 2 edges, using an interpolation method.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_MeasureMinEdgeDistance : HRESULT
VAR_INPUT
    ipSrcImage     : ITcVnImage;
END_VAR
VAR_IN_OUT
    fMinDistance   : REAL;
    aStartPoint    : TcVnPoint2_REAL;
    aEndPoint      : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    eEdgeDirection : ETcVnEdgeDirection;
    fMinStrength   : REAL;
    nSearchLines   : UDINT;
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel)
eEdgeDirection	ETcVnEdgeDirection [▶ 178]	Specification of the edge direction from aStartPoint to aEndPoint to search for
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
nSearchLines	UDINT	Width of the search window, i.e. the number of search lines (1, 3, 5, 7, ...), centered around the line specified by aStartPoint and aEndPoint
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fMinDistance	REAL	Returns the minimum distance between the detected edges
aStartPoint	TcVnPoint2_REAL [▶ 139]	Position from which to start the search process (in the direction of aEndPoint)
aEndPoint	TcVnPoint2_REAL [▶ 139]	Position where the search process ends

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_MeasureMinEdgeDistance定位两条线性边缘，并测量它们之间的最小距离。它以REAL返回最小距离。

参数

输入图像

输入图像ipSrcImage必须为单通道灰度图像。

最小距离（返回值）

参数fMinDistance返回两个定位边缘之间的最小距离。

边缘定位的参数

关于其余参数的详细解释，请参见边缘定位 [▶ 1429]一章。

专家参数

专家版F_VN_MeasureMinEdgeDistanceExp [▶ 1470]包含额外的参数。

应用

例如，边缘距离的测量如下所示：

```
hr := F_VN_MeasureMinEdgeDistance(
    ipSrcImage              :=   ipImageIn,
    fMinDistance            :=   fDistance
    aStartPoint             :=   aStartPoint,
    aEndPoint               :=   aEndPoint,
    eEdgeDirection          :=   TCVN_ED_LIGHT_TO_DARK,
    fMinStrength            :=   fMinStrength,
    nSearchLines            :=   nSearchLines,
    hrPrev                  :=   hr
);
```

相关函数

- ClosestPointsBF [▶ 1434]，用于计算两个点云中最近的点
- MeasureAngleBetweenEdges [▶ 1458]，用于测量两条边之间的角度
- MeasureEdgeDistance [▶ 1463]，用于测量两条边之间的平均距离
- MeasureMinEdgeDistance [▶ 1468]，用于测量两条边之间的最小距离

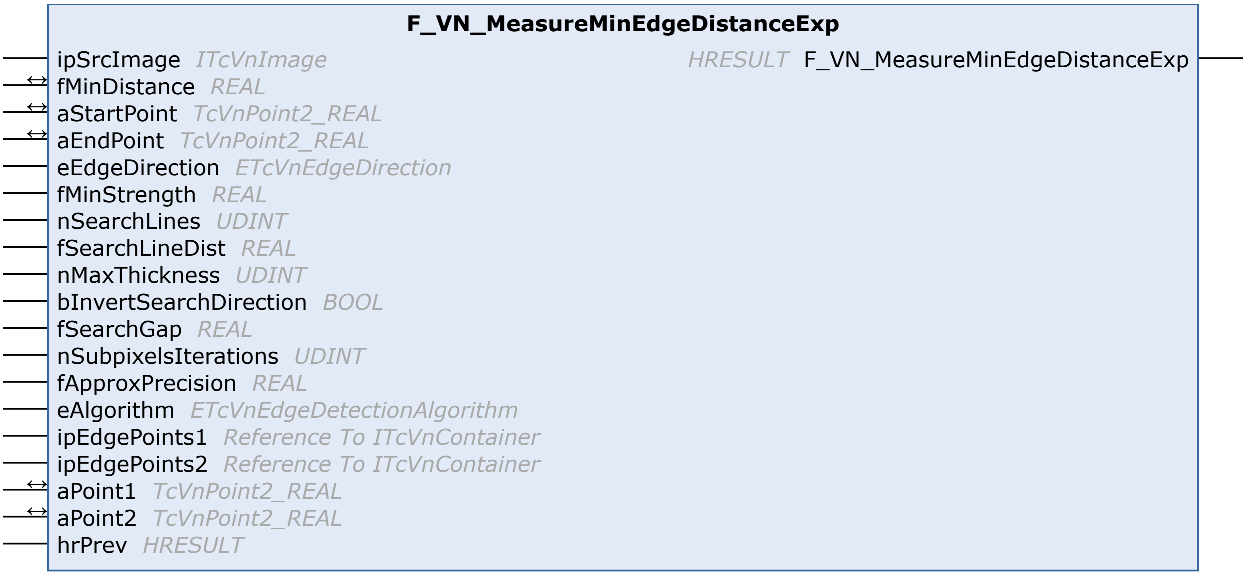
Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.20.16 F_VN_MeasureMinEdgeDistanceExp



Measure the minimum distance within the specified search window between 2 edges. (expert function)
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
FUNCTION F_VN_MeasureMinEdgeDistanceExp : HRESULT
VAR_INPUT
    ipSrcImage             : ITcVnImage;
END_VAR
VAR_IN_OUT
    fMinDistance           : REAL;
    aStartPoint            : TcVnPoint2_REAL;
    aEndPoint              : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    eEdgeDirection         : ETcVnEdgeDirection;
    fMinStrength           : REAL;
    nSearchLines           : UDINT;
    fSearchLineDist        : REAL;
    nMaxThickness          : UDINT;
    bInvertSearchDirection : BOOL;
    fSearchGap             : REAL;
    nSubpixelsIterations   : UDINT;
    fApproxPrecision       : REAL;
    eAlgorithm             : ETcVnEdgeDetectionAlgorithm;
    ipEdgePoints1          : Reference To ITcVnContainer;
    ipEdgePoints2          : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    aPoint1                : TcVnPoint2_REAL;
    aPoint2                : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    hrPrev                 : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (1 channel)
eEdgeDirection	ETcVnEdgeDirection [▶ 178]	Specification of the edge direction from aStartPoint to aEndPoint to search for
fMinStrength	REAL	Specification of the minimum strength (intensity difference) of the edge to search for
nSearchLines	UDINT	Width of the search window, i.e. the number of search lines (1, 3, 5, 7, ...), centered around the line specified by aStartPoint and aEndPoint
fSearchLineDist	REAL	Distance between the search lines in pixels (> 0)
nMaxThickness	UDINT	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels
bInvertSearchDirection	BOOL	If true, the search starts from the center point between aStartPoint and aEndPoint in both directions
fSearchGap	REAL	Optional width of a gap (>= 0, centered between aStartPoint and aEndPoint), that is neglected for searching edges (can reduce execution time). The 2 edges to search for must be on different sides of the gap.
nSubpixelsIterations	UDINT	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)
fApproxPrecision	REAL	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▶ 177]	Selection of the edge detection algorithm
ipEdgePoints1	Reference To ITcVnContainer [▶ 349]	Returns the detected edge points of the edge near aStartPoint (optional, set to 0 if not required; ContainerType_Vector_TcVnPoint2_REAL)
ipEdgePoints2	Reference To ITcVnContainer [▶ 349]	Returns the detected edge points of the edge near aEndPoint (optional, set to 0 if not required; ContainerType_Vector_TcVnPoint2_REAL)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
fMinDistance	REAL	Returns the minimum distance between the detected edges
aStartPoint	TcVnPoint2_REAL [▶ 139]	Position from which to start the search process (in the direction of aEndPoint)
aEndPoint	TcVnPoint2_REAL [▶ 139]	Position where the search process ends
aPoint1	TcVnPoint2_REAL [▶ 139]	Returns the point on the edge near aStartPoint, for which the minimum distance is achieved
aPoint2	TcVnPoint2_REAL [▶ 139]	Returns the point on the edge near aEndPoint, for which the minimum distance is achieved

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_MeasureMinEdgeDistanceExp是F_VN_MeasureMinEdgeDistance [▶ 1468]的专家版本。它包含额外的参数。

参数

输入图像

输入图像ipSrcImage必须为单通道灰度图像。

最小距离（返回值）

参数fMinDistance返回两个定位边缘之间的最小距离。

定位的边缘点（返回值）

容器ipEdgePoints1和ipEdgePoints2返回在每条搜索线基础上定位的确切点。如果不需要，可以传输值 0，而不是接口指针。

此外，点aPoint1和aPoint2返回两条边之间距离最小的点。

搜索差距（专家）

定义的搜索线中心的区域，在该区域内的搜索被省略。因此可以减少执行时间。

边缘定位的参数

关于其余参数的详细解释，请参见边缘定位 [▶ 1429]一章。

应用

例如，使用专家参数测量最小边缘距离如下所示：

```
hr := F_VN_MeasureEdgeDistance(
    ipSrcImage                 :=  ipImageIn,
    fMinDistance               :=  fDistance
    aStartPoint                :=  aStartPoint,
    aEndPoint                  :=  aEndPoint,
    eEdgeDirection             :=  TCVN_ED_LIGHT_TO_DARK,
    fMinStrength               :=  fMinStrength,
    nSearchLines               :=  nSearchLines,
    fSearchLineDist            :=  3
    nMaxThickness              :=  7,
    bInvertSearchDirection     :=  TRUE,
    nSubpixelsIterations       :=  100,
    fApproxPrecision           :=  0.0001,
    eAlgorithm                 :=  TCVN_EDA_APPROX_ERF,
    ipEdgePoints1              :=  ipEdge1,
    ipEdgePoints2              :=  ipEdge2,
    aPoint1                    :=  aPoint1,
    aPoint2                    :=  aPoint2,
    hrPrev                     :=  hr
);
```

定位的边缘点可以通过函数F_VN_DrawContours [▶ 967]和F_VN_DrawPoint [▶ 990]实现可视化：

```
hr := F_VN_DrawContours(ipEdgePoints1, -1, ipImageRes, aColorGreen, 5, hr);
hr := F_VN_DrawContours(ipEdgePoints2, -1, ipImageRes, aColorGreen, 5, hr);

F_VN_DrawPoint(aPoint1[0], aPoint1[1], ipImageRes, TCVN_DS_X, aColorGreen, hr);
F_VN_DrawPoint(aPoint2[0], aPoint2[1], ipImageRes, TCVN_DS_X, aColorGreen, hr);
```

相关函数

- ClosestPointsBF [▶ 1434]，用于计算两个点云中最近的点
- MeasureAngleBetweenEdges [▶ 1458]，用于测量两条边之间的角度
- MeasureEdgeDistance [▶ 1463]，用于测量两条边之间的平均距离

- MeasureMinEdgeDistance [▶ 1468]，用于测量两条边之间的最小距离

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21 Miscellaneous

该组包含可跨组使用的进一步功能。

函数

几何辅助函数

- F_VN_ClipLineToBoundary [▶ 1474]
- F_VN_ClipLineToBoundary_ITcVnImage [▶ 1476]
- F_VN_ClipLineToBoundary_TcVnRectangle_DINT [▶ 1477]
- F_VN_LineIntersectionPoint [▶ 1487]
- F_VN_LineIntersectionPointAndAngle [▶ 1488]
- F_VN_RotatedRectangleCorners [▶ 1490]
- F_VN_RotatedRectangleIntersection [▶ 1490]

时间戳

- F_VN_GetTimestamp [▶ 1482]
- F_VN_UpdateTimestamp [▶ 1492]

矩阵辅助函数

- F_VN_InitMatrixStruct [▶ 1484]
- F_VN_InvertMatrix3x3 [▶ 1486]
- F_VN_MultiplyMatrices [▶ 1489]

转换接口

- F_VN_ConvertITcUnknownToITcVnBitmapExport [▶ 1478]
- F_VN_ConvertITcUnknownToITcVnContainer [▶ 1479]
- F_VN_ConvertITcUnknownToITcVnImage [▶ 1480]
- F_VN_ConvertITcUnknownToITcVnMlModel [▶ 1481]

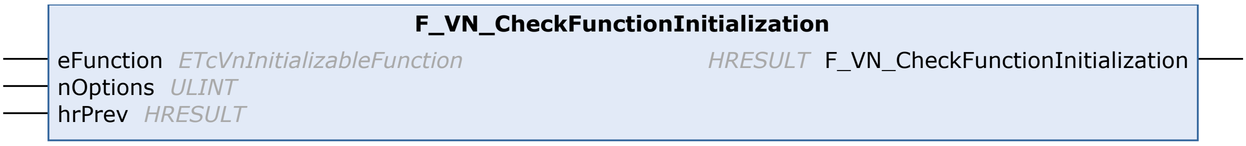
初始化函数

- F_VN_CheckFunctionInitialization [▶ 1474]
- F_VN_DeinitializeFunction [▶ 1481]

其它

- F_VN_HuMomentInvariants [▶ 1483]
- F_VN_SetRngSeed [▶ 1491]

6.1.4.21.1 F_VN_CheckFunctionInitialization



Check if a function is initialized with the specified options.

Syntax

Definition:

```
FUNCTION F_VN_CheckFunctionInitialization : HRESULT
VAR_INPUT
    eFunction : ETcVnInitializableFunction;
    nOptions  : ULINT;
    hrPrev    : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
eFunction	ETcVnInitializableFunction [▶ 185]	Initializable function
nOptions	ULINT	Initialization options for the function
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

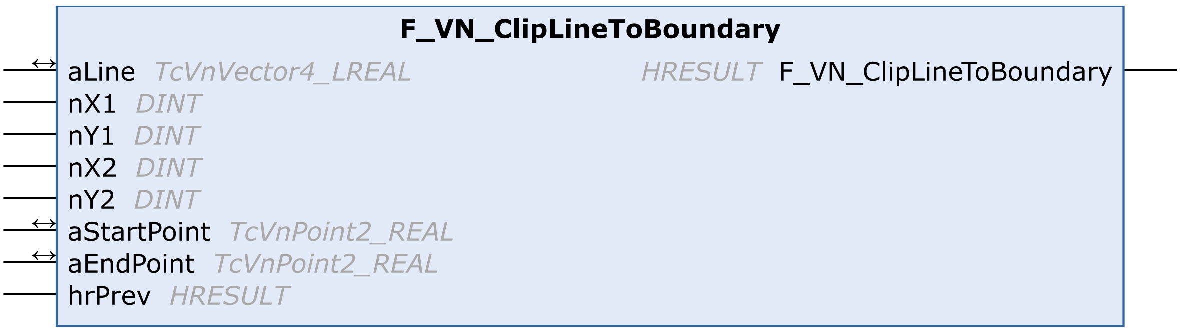
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.2 F_VN_ClipLineToBoundary



Clips a line to a rectangular boundary.

Syntax

Definition:

```
FUNCTION F_VN_ClipLineToBoundary : HRESULT
VAR_IN_OUT
    aLine       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    nX1         : DINT;
    nY1         : DINT;
    nX2         : DINT;
    nY2         : DINT;
END_VAR
VAR_IN_OUT
    aStartPoint : TcVnPoint2_REAL;
    aEndPoint   : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nX1	DINT	x of top left rectangle point
nY1	DINT	y of top left rectangle point
nX2	DINT	x of bottom right rectangle point
nY2	DINT	Y of bottom right rectangle point
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aLine	TcVnVector4_LREAL [▶ 141]	The first and second element describe the x and y component of the direction vector. The third and fourth element describe the x and y component of the position vector.
aStartPoint	TcVnPoint2_REAL [▶ 139]	Returns the clipped starting point of the line.
aEndPoint	TcVnPoint2_REAL [▶ 139]	Returns the clipped end point of the line.

Return value

HRESULT [▶ 122]

更多信息

该函数根据指定区域调整线条。

对于这一区域的各种定义形式，存在不同的函数：

矩形区域的定义	待使用的函数
由单个 X/Y 坐标指定	F_VN_ClipLineToBoundary [▶ 1474]
按图像大小	F_VN_ClipLineToBoundary_ITcVnImage [▶ 1476]
按矩形结构	F_VN_ClipLineToBoundary_TcVnRectangle_DINT [▶ 1477]

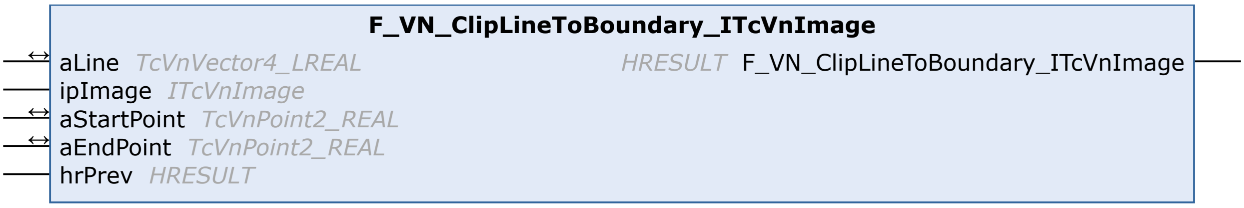
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.3 F_VN_ClipLineToBoundary_ITcVnImage



Clips a line to the boundary of an image.

Syntax

Definition:

```
FUNCTION F_VN_ClipLineToBoundary_ITcVnImage : HRESULT
VAR_IN_OUT
    aLine       : TcVnVector4_LREAL;
END_VAR
VAR_INPUT
    ipImage     : ITcVnImage;
END_VAR
VAR_IN_OUT
    aStartPoint : TcVnPoint2_REAL;
    aEndPoint   : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▸ 390]	Image, from which to derive the boundary
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aLine	TcVnVector4_LREAL [▸ 141]	The first and second element describe the x and y component of the direction vector. The third and fourth element describe the x and y component of the position vector.
aStartPoint	TcVnPoint2_REAL [▸ 139]	Returns the clipped starting point of the line.
aEndPoint	TcVnPoint2_REAL [▸ 139]	Returns the clipped end point of the line.

Return value

HRESULT [▶ 122]

更多信息

该函数根据指定区域调整线条。

对于这一区域的各种定义形式，存在不同的函数：

矩形区域的定义	待使用的函数
由单个 X/Y 坐标指定	F_VN_ClipLineToBoundary [▶ 1474]
按图像大小	F_VN_ClipLineToBoundary_ITcVnImage [▶ 1476]
按矩形结构	F_VN_ClipLineToBoundary_TcVnRectangle_DINT [▶ 1477]

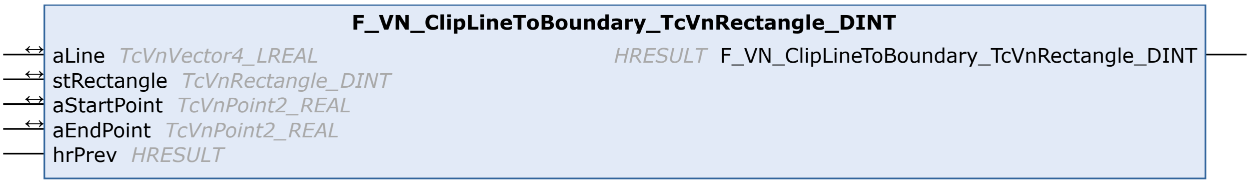
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.4 F_VN_ClipLineToBoundary_TcVnRectangle_DINT



Clips a line to a rectangular boundary.

Syntax

Definition:

```
FUNCTION F_VN_ClipLineToBoundary_TcVnRectangle_DINT : HRESULT
VAR_IN_OUT
    aLine       : TcVnVector4_LREAL;
    stRectangle : TcVnRectangle_DINT;
    aStartPoint : TcVnPoint2_REAL;
    aEndPoint   : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aLine	TcVnVector4_LREAL [▸ 141]	The first and second element describe the x and y component of the direction vector. The third and fourth element describe the x and y component of the position vector.
stRectangle	TcVnRectangle_DINT [▸ 224]	Rectangular boundary
aStartPoint	TcVnPoint2_REAL [▸ 139]	Returns the clipped starting point of the line.
aEndPoint	TcVnPoint2_REAL [▸ 139]	Returns the clipped end point of the line.

Return value

HRESULT [▸ 122]

更多信息

该函数根据指定区域调整线条。

对于这一区域的各种定义形式，存在不同的函数：

矩形区域的定义	待使用的函数
由单个 X/Y 坐标指定	F_VN_ClipLineToBoundary [▸ 1474]
按图像大小	F_VN_ClipLineToBoundary_ITcVnImage [▸ 1476]
按矩形结构	F_VN_ClipLineToBoundary_TcVnRectangle_DINT [▸ 1477]

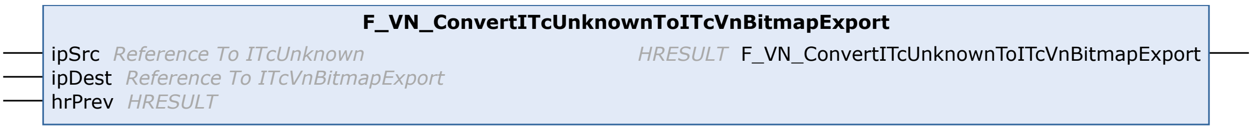
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.5 F_VN_ConvertITcUnknownToITcVnBitmapExport



Convert an ITcUnknown interface pointer to an ITcVnBitmapExport interface pointer. The destination pointer will be released (if existing) and set to the source pointer. The source pointer will be set to 0.

Syntax

Definition:

```
FUNCTION F_VN_ConvertITcUnknownToITcVnBitmapExport : HRESULT
VAR_INPUT
    ipSrc  : Reference To ITcUnknown;
    ipDest : Reference To ITcVnBitmapExport;
    hrPrev : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrc	Reference To ITcUnknown [▶ 407]	Source pointer
ipDest	Reference To ITcVnBitmapExport [▶ 373]	Destination pointer
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

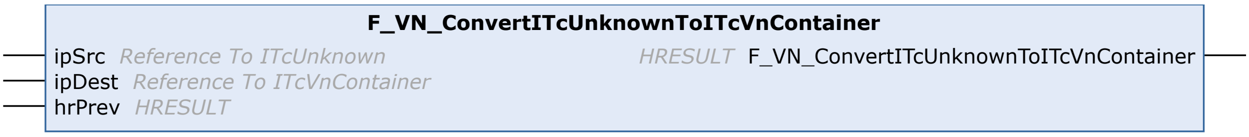
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.6 F_VN_ConvertITcUnknownToITcVnContainer



Convert an ITcUnknown interface pointer to an ITcVnContainer interface pointer. The destination pointer will be released (if existing) and set to the source pointer. The source pointer will be set to 0.

Syntax

Definition:

```
FUNCTION F_VN_ConvertITcUnknownToITcVnContainer : HRESULT
VAR_INPUT
    ipSrc  : Reference To ITcUnknown;
    ipDest : Reference To ITcVnContainer;
    hrPrev : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrc	Reference To ITcUnknown [▶ 407]	Source pointer
ipDest	Reference To ITcVnContainer [▶ 349]	Destination pointer
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.7 F_VN_ConvertITcUnknownToITcVnImage

F_VN_ConvertITcUnknownToITcVnImage
ipSrc *Reference To ITcUnknown* — *HRESULT* F_VN_ConvertITcUnknownToITcVnImage
ipDest *Reference To ITcVnImage*
hrPrev *HRESULT*

Convert an ITcUnknown interface pointer to an ITcVnImage interface pointer. The destination pointer will be released (if existing) and set to the source pointer. The source pointer will be set to 0.

Syntax

Definition:

```
FUNCTION F_VN_ConvertITcUnknownToITcVnImage : HRESULT
VAR_INPUT
    ipSrc  : Reference To ITcUnknown;
    ipDest : Reference To ITcVnImage;
    hrPrev : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrc	Reference To ITcUnknown [▶ 407]	Source pointer
ipDest	Reference To ITcVnImage [▶ 390]	Destination pointer
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

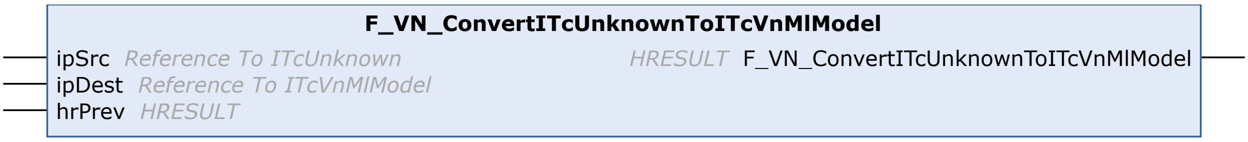
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.8 F_VN_ConvertITcUnknownToITcVnMlModel



Convert an ITcUnknown interface pointer to an ITcVnMlModel interface pointer. The destination pointer will be released (if existing) and set to the source pointer. The source pointer will be set to 0.

Syntax

Definition:

```
FUNCTION F_VN_ConvertITcUnknownToITcVnMlModel : HRESULT
VAR_INPUT
    ipSrc  : Reference To ITcUnknown;
    ipDest : Reference To ITcVnMlModel;
    hrPrev : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrc	Reference To ITcUnknown [▸ 407]	Source pointer
ipDest	Reference To ITcVnMlModel [▸ 399]	Destination pointer
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.9 F_VN_DeinitializeFunction

Deinitialize the specified options for the function.

Syntax

Definition:

```
FUNCTION F_VN_DeinitializeFunction : HRESULT
VAR_INPUT
    eFunction : ETcVnInitializableFunction;
    nOptions  : ULINT;
    hrPrev    : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
eFunction	ETcVnInitializableFunction [▸ 185]	Initializable function
nOptions	ULINT	Initialization options for the function
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.10 F_VN_GetTimestamp

Get the timestamp (actual DC time) from the last change of an object (e.g. ITcVnImage or ITcVnContainer)

Syntax

Definition:

```
FUNCTION F_VN_GetTimestamp : HRESULT
VAR_INPUT
    ipUnknown   : ITcUnknown;
END_VAR
VAR_IN_OUT
    nTimestamp : LINT;
END_VAR
VAR_INPUT
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipUnknown	ITcUnknown [▶ 407]	An object implementing ITcUnknown and ITcVnTimestamp (e.g. ITcVnImage or ITcVnContainer)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
nTimestamp	LINT	Returns the timestamp

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.11 F_VN_HuMomentInvariants

Computes the Hu moment invariants.

Syntax

Definition:

```
FUNCTION F_VN_HuMomentInvariants : HRESULT
VAR_IN_OUT
    stMoments      : TcVnMoments;
    aHuInvariants : TcVnArray7_LREAL;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stMoments	TcVnMoments [▶ 211]	Moments from which to compute the Hu invariants
aHuInvariants	TcVnArray7_LREAL [▶ 141]	Returns the Hu moment invariants

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.12 F_VN_InitMatrixStruct

Initialize a struct of the type TcVnMatrix extending a buffer with metainformation so that it can be used as a matrix.

Syntax

Definition:

```
FUNCTION F_VN_InitMatrixStruct : HRESULT
VAR_INPUT
    pSrcBuffer   : PVOID;
END_VAR
VAR_IN_OUT
    stDestMatrix : TcVnMatrix;
END_VAR
VAR_INPUT
    nRows        : UDINT;
    nCols        : UDINT;
    eElementType : ETcVnElementType;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
pSrcBuffer	PVOID	Source buffer
nRows	UDINT	Matrix rows
nCols	UDINT	Matrix columns
eElementType	ETcVnElementType [▶ 178]	Type of the matrix elements
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stDestMatrix	TcVnMatrix [▶ 211]	Returns completed matrix struct

Return value

HRESULT [▶ 122]

更多信息

函数F_VN_InitMatrixStruct从任何内存区域创建类型为TcVnMatrix [▶ 211]的矩阵结构。

参数

数据缓冲区

指针pSrcBuffer必须指向包含矩阵数据的内存区域。例如，这可以是一个数组。

矩阵描述

矩阵必须由行数nRows、列数nCols、以及元素的数据类型eElementType进行描述。

结果矩阵

TcVnMatrix [▶ 211]类型的矩阵结构stDestMatrix作为结果返回。

应用

例如，3×3矩阵创建如下所示：

```
VAR
    aMatrixArray    :   ARRAY [0..2, 0..2] OF USINT := [
        -1, 0, 1,
        -2, 0, 2,
        -1, 0, 1
    ];
END_VAR

hr := F_VN_InitMatrixStruct(
    pSrcBuffer      :=  ADR(aMatrixArray),
    stDestMatrix    :=  stMatrix,
    nRows           :=  3, // must match with array dimensions
    nCols           :=  3, // must match with array dimensions
    eElementType    :=  TCVN_ET_USINT, // must match with array type
    hrPrev          :=  hr
);
```

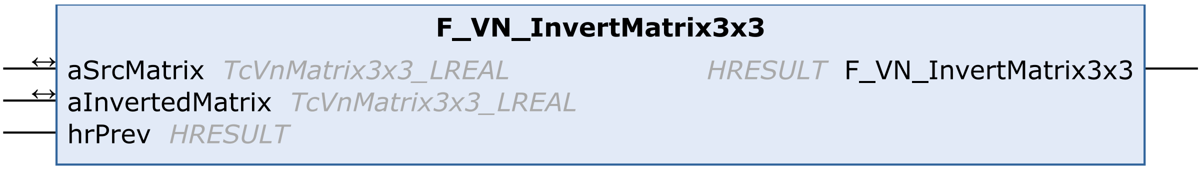
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.13 F_VN_InvertMatrix3x3



Invert a 3x3 matrix.

Syntax

Definition:

```
FUNCTION F_VN_InvertMatrix3x3 : HRESULT
VAR_IN_OUT
    aSrcMatrix       : TcVnMatrix3x3_LREAL;
    aInvertedMatrix : TcVnMatrix3x3_LREAL;
END_VAR
VAR_INPUT
    hrPrev           : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aSrcMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Source matrix
aInvertedMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns inverted matrix

Return value

HRESULT [▶ 122]

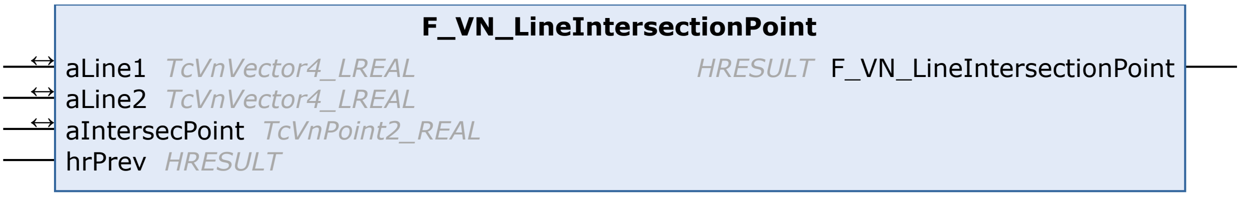
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.14 F_VN_LineIntersectionPoint



Computes the intersection point between two lines (returns S_FALSE if the provided lines are parallel).

Syntax

Definition:

```
FUNCTION F_VN_LineIntersectionPoint : HRESULT
VAR_IN_OUT
    aLine1         : TcVnVector4_LREAL;
    aLine2         : TcVnVector4_LREAL;
    aIntersecPoint : TcVnPoint2_REAL;
END_VAR
VAR_INPUT
    hrPrev         : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aLine1	TcVnVector4_LREAL [▶ 141]	First line. The first and second element describe the x and y component of a vector collinear to the line. The third and fourth element describe the x and y component of a point on the line.
aLine2	TcVnVector4_LREAL [▶ 141]	Second line. The first and second element describe the x and y component of a vector collinear to the line. The third and fourth element describe the x and y component of a point on the line.
aIntersecPoint	TcVnPoint2_REAL [▶ 139]	Returns the intersection point of aLine1 and aLine2.

Return value

HRESULT [▶ 122]

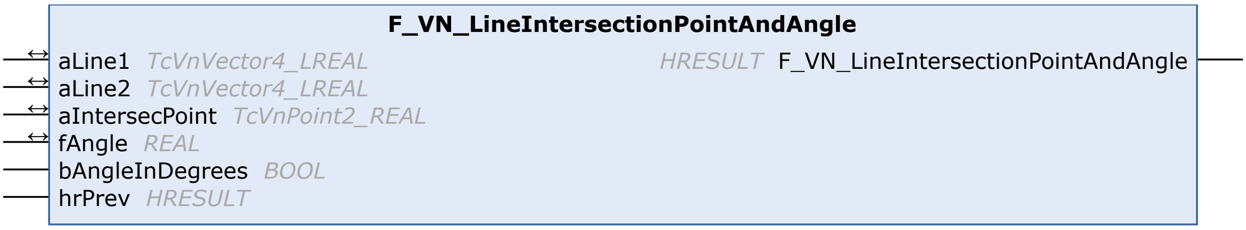
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.15 F_VN_LineIntersectionPointAndAngle



Computes the intersection point and angle between two lines (returns S_FALSE if the provided lines are parallel).

Syntax

Definition:

```
FUNCTION F_VN_LineIntersectionPointAndAngle : HRESULT
VAR_IN_OUT
    aLine1          : TcVnVector4_LREAL;
    aLine2          : TcVnVector4_LREAL;
    aIntersecPoint  : TcVnPoint2_REAL;
    fAngle          : REAL;
END_VAR
VAR_INPUT
    bAngleInDegrees : BOOL;
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
bAngleInDegrees	BOOL	If TRUE, fAngle is in degrees. If FALSE, fAngle is in radians.
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
aLine1	TcVnVector4_LREAL [▸ 141]	First line. The first and second element describe the x and y component of a vector collinear to the line. The third and fourth element describe the x and y component of a point on the line.
aLine2	TcVnVector4_LREAL [▸ 141]	Second line. The first and second element describe the x and y component of a vector collinear to the line. The third and fourth element describe the x and y component of a point on the line.
aIntersecPoint	TcVnPoint2_REAL [▸ 139]	Returns the intersection point of aLine1 and aLine2.
fAngle	REAL	Returns the intersection angle of aLine1 and aLine2.

Return value

HRESULT [▸ 122]

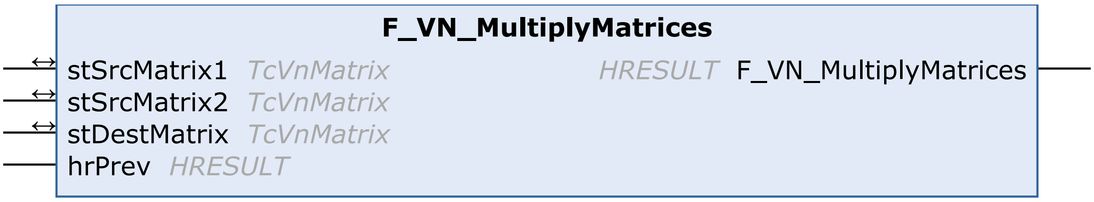
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.16 F_VN_MultiplyMatrices



Matrix multiplication of two matrices (A * B = C). All matrices are represented by structs holding a pointer to an array of the appropriate size.

Syntax

Definition:

```
FUNCTION F_VN_MultiplyMatrices : HRESULT
VAR_IN_OUT
    stSrcMatrix1 : TcVnMatrix;
    stSrcMatrix2 : TcVnMatrix;
    stDestMatrix : TcVnMatrix;
END_VAR
VAR_INPUT
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stSrcMatrix1	TcVnMatrix [▶ 211]	First source matrix (A)
stSrcMatrix2	TcVnMatrix [▶ 211]	Second source matrix (B)
stDestMatrix	TcVnMatrix [▶ 211]	Destination matrix (C) (The destination matrix is filled by this function, but the required memory needs to be provided.)

Return value

HRESULT [▶ 122]

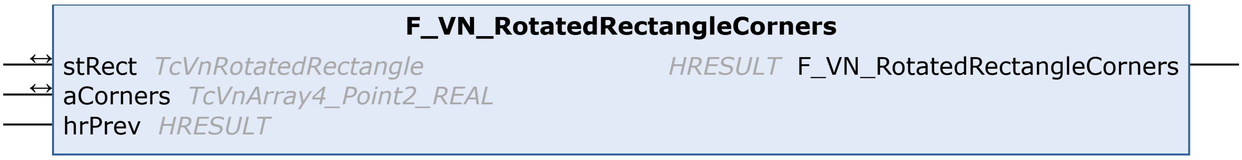
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.17 F_VN_RotatedRectangleCorners



Computes the 4 corner points of a rotated rectangle.

Syntax

Definition:

```
FUNCTION F_VN_RotatedRectangleCorners : HRESULT
VAR_IN_OUT
    stRect   : TcVnRotatedRectangle;
    aCorners : TcVnArray4_Point2_REAL;
END_VAR
VAR_INPUT
    hrPrev   : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stRect	TcVnRotatedRectangle [▶ 225]	Rotated rectangle
aCorners	TcVnArray4_Point2_REAL [▶ 141]	Returns the 4 corner points

Return value

HRESULT [▶ 122]

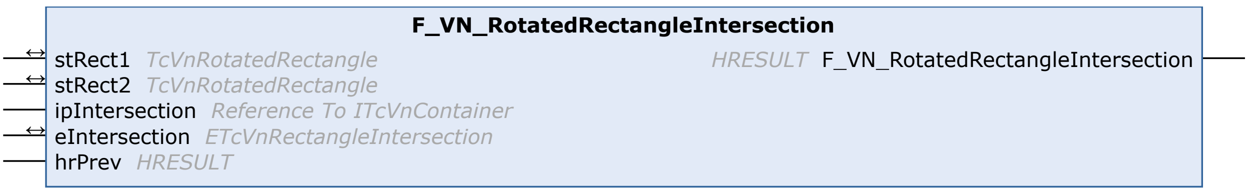
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.18 F_VN_RotatedRectangleIntersection



Finds the intersection of 2 rotated rectangles.

Syntax

Definition:

```
FUNCTION F_VN_RotatedRectangleIntersection : HRESULT
VAR_IN_OUT
    stRect1         : TcVnRotatedRectangle;
    stRect2         : TcVnRotatedRectangle;
END_VAR
VAR_INPUT
    ipIntersection : Reference To ITcVnContainer;
END_VAR
VAR_IN_OUT
    eIntersection  : ETcVnRectangleIntersection;
END_VAR
VAR_INPUT
    hrPrev          : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipIntersection	Reference To ITcVnContainer [▶ 349]	Returns the intersection points (ContainerType_Vector_TcVnPoint2_REAL, non-zero interface pointers are reused)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

In/Outputs

Name	Type	Description
stRect1	TcVnRotatedRectangle [▶ 225]	Rotated rectangle 1
stRect2	TcVnRotatedRectangle [▶ 225]	Rotated rectangle 2
eIntersection	ETcVnRectangleIntersection [▶ 194]	Returns the intersection type

Return value

HRESULT [▶ 122]

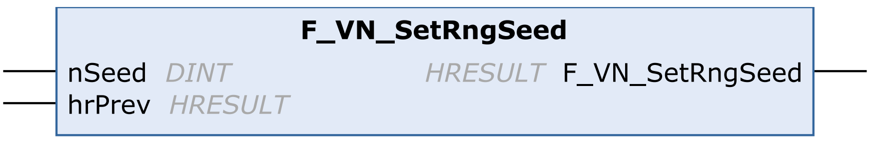
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.19 F_VN_SetRngSeed



Sets the internal pseudo random number generator seed (intended for testing purposes only).

Syntax

Definition:

```
FUNCTION F_VN_SetRngSeed : HRESULT
VAR_INPUT
    nSeed   : DINT;
    hrPrev  : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
nSeed	DINT	seed (0 sets the generator back to its initial state)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.21.20 F_VN_UpdateTimestamp

Update the timestamp of an object to the actual DC time (e.g. ITcVnImage or ITcVnContainer)

Syntax

Definition:

```
FUNCTION F_VN_UpdateTimestamp : HRESULT
VAR_INPUT
    ipUnknown : ITcUnknown;
    hrPrev    : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipUnknown	ITcUnknown [▶ 407]	An object implementing ITcUnknown and ITcVnTimestamp (e.g. ITcVnImage or ITcVnContainer)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.22 OCR

该组包含**光学**字符**识别**函数。

OCR 函数 [▶ 1495]识别图像中的字符，并将识别出的字符作为字符串返回。分类基于经典机器学习模型。这些模型以训练形式提供，因此可以直接使用函数。无需额外设置或训练。支持的字符和字体取决于训练中使用的数据。因此，有多种模型，其函数范围以及一般前提条件和要求如下所述。

对于图像中识别出的所有字符，模型会将每个字符分配到匹配度最高的已知类别。因此，结果总是包含所使用模型的已知字符。不包括拒绝或删除未知字符。

函数仅支持单行字符串；必须将多行字符串分为单独的 ROI，并通过多次函数调用读取。

识别字符的一般要求

- 字符高度最小为 20 像素
- 线宽最小为 3 像素
- 点最小为 3 x 3 像素
- 线最小为 3 x 6 像素
- 字符间距/分隔最小为 4 像素
- 字符不得重叠
- 仅水平对齐/字符排列最大 ± 6° 偏差
- 字符线/轮廓不得中断

图像的一般要求：

- ROI 只包含待识别字符和字符周围的无干扰区
- 在 ROI 内，字符不得被其他轮廓（如矩形）包围
- 字符与背景要对比鲜明
- 背景均匀、无噪音或干扰、不透明

对字体的要求

- 仅使用字宽相等的比例字体
- 较大的间隙始终会被识别为单个空格
- 仅使用无衬线字体，如 Arial、Tahoma、Courier、Univers、Frutiger、Verdana、OCR-B 等
- 无混合字体
- 无圆点打印或斜体字体

模型：

枚举 ETcVnOcrModelType [▶ 190] 允许访问以下模型：

- TCVN_OMT_NUMBERS
  - 支持数字分类
  - 包含 0-9 的字符

- TCVN_OMT_NUMBERS_SC
    - 支持数字和特殊字符分类
    - 包含 0-9 的字符和五个特殊字符 . / - : =
- TCVN_OMT_UCLETTERS
    - 支持大写字母分类
    - 包含 A-Z 的字符
- TCVN_OMT_NUMBERS_SC_UCLETTERS
    - 支持数字、特殊字符和大写字母分类
    - 包含 0-9 的字符、五个特殊字符 . / - : = 和 A-Z

在使用组合模型（TCVN_OMT_NUMBERS_SC_UCLETTERS）时，由于某些字符非常相似，可能会出现混淆。例如 O 和 0、S 和 5 以及 B 和 8。

如果数字或字母的位置已知，则应使用具有默认格式的专家函数（sPattern）和单独模型的组合作为替代。另外，也可以单独分析结果字符串（ipCharacters）的字符，这样就可以单独决定是否接受 O 而非 0。

初始化函数

为了能够将 OCR 函数与一个或多个模型一起使用，必须首先使用功能块 FB_VN_InitializeFunction [▶ 1521] 初始化该函数以及相应的模型。

加载模型所需的时间取决于所使用 IPC 的性能和模型大小，可能需要几百毫秒甚至几秒钟。因此，在任务循环时间较短的情况下，预计会出现循环时间超限。

如果要使用组合模型 TCVN_OMT_NUMBERS_SC_UCLETTERS，由于文件大小的原因，路由器内存必须至少设置为 512 MB。

具体大小取决于路由器内存的其他用途和任何现有分段储存。例如，如果功能块 FB_VN_InitializeFunction 返回返回代码 0x80004005，则表示路由器内存不足。因此，应首先检查路由器内存，必要时增加内存。加载模型后，大部分内存会重新释放，供应用使用。

HRESULT 的解释

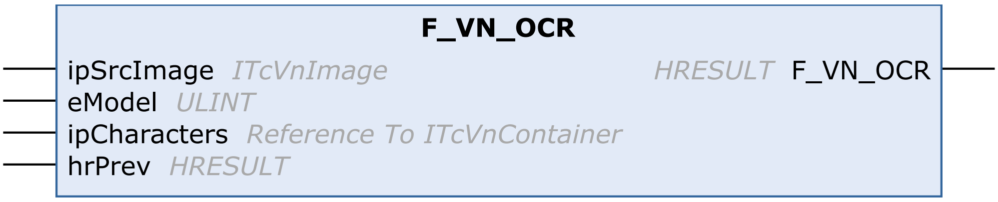
- 如果识别到图像上的字符，标准函数返回 S_OK。对于 Exp 函数，这取决于是否指定了 sPattern。如果没有定义格式规格，则返回对应于标准函数。如果已传递格式规格，只会在识别字符与模板规格相匹配的情况下才会返回 S_OK。
- 如果标准函数执行成功，但未找到任何字符，或者 Exp 函数中的字符与模板规格不匹配，则返回 S_FALSE。
- 如果在专家函数中输入了 sPattern，但缺少转移模型，则会返回 HRESULT = ADSERR_DEVICE_NOTINIT。
- 如果向标准函数传递了多个模型，则始终返回 HRESULT = ADSERR_DEVICE_INVALIDPARM，因为该函数仅支持每次调用一个模型。如为专家函数，这取决于是否指定了 sPattern。如果 sPattern 为空，则只使用 TCVN_OMT_NUMBERS_SC_UCLETTERS 模型。如果没有传递该模型，则返回 HRESULT = ADSERR_DEVICE_INVALIDPARM。

使用 sPattern 的示例

F_VN_OCRExp [▶ 1496] 通过参数 sPattern 和 eOcrOptions 提供其他选项。根据输入图像上的字符组合、来自 sPattern 和 eOcrOptions 的信息以及识别的字符，如果出现不匹配，则返回 S_FALSE。然后，还可以对结果字符串（ipCharacters）中的字符进行单独分析，例如，如果长度不同，也会返回 S_FALSE。这些示例涉及 TCVN_OMT_NUMBERS_SC_UCLETTERS 模型的使用。

ipSrcImage 上的字符	sPattern	eOcrOptions	ipCharacters	HRESULT
12/34	dd.dd		12/34	S_OK
12534	dd.dd		12534	S_OK
12/34	dd!dd		1234	S_OK
12 34	dd!d		124	S_OK
12534	dd!dd		1234	S_OK
12 34	dd_dd		12 34	S_OK
AB12/	uudd#		AB12/	S_OK
12 34	dddd	WITHBLANKS	12 34	S_OK
12/4	dd#d	WITHBLANKS	12/4	S_OK
12 34	dd_dd	WITHBLANKS	12 34	S_OK
12 34 56	dd_dddd	WITHBLANKS	12 34 56	S_OK
12 3 4	dd.dd	WITHBLANKS	12 3 4	S_FALSE
12 34	dd!dd		124	S_FALSE
12 34	dd.dd		1234	S_FALSE
1234	!dddd		234	S_FALSE
12 34	dd_dd		12.34	S_FALSE
12 3	dd_dd		123	S_FALSE
12 3 4	dd_d_d		123 4	S_FALSE
AB12/	uddd#		AB12/	S_FALSE
AB12/	uuddd		AB12/	S_FALSE

6.1.4.22.1 F_VN_OCR



Detects and recognizes characters in a binary image (white characters on black background).

Syntax

Definition:

```
FUNCTION F_VN_OCR : HRESULT
VAR_INPUT
    ipSrcImage   : ITcVnImage;
    eModel       : ULINT;
    ipCharacters : Reference To ITcVnContainer;
    hrPrev       : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT, 1 channel, binary)
eModel	ULINT	Selection of OCR model types (ETcVnOcrModelType)
ipCharacters	Reference To ITcVnContainer [▶ 349]	Returns the recognized characters (ContainerType_Vector_String_SINT)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_OCR 检测和识别二进制图像中的字符，并以容器形式返回识别出的字符。

参数

输入图像

传输到 ipSrcImage 的输入图像必须是单通道二进制图像。只允许使用黑底白字。

模型

必须将 ETcVnOcrModelType [▶ 190] 的模型类型传输到 eModel，该类型将用于对字符进行分类的函数调用。该函数仅支持每次调用使用一个模型。

要使用模型，首先必须使用功能块 FB_VN_InitializeFunction [▶ 1521] 对其进行初始化。

识别字符

识别字符通过引用 ipCharacters 返回。

专家参数

专家变体 F_VN_OCRExp [▶ 1497] 包含附加参数。

应用

例如，调用 OCR 函数，使用模型识别数字并在随后导出字符串的过程如下所示：

```
hr := F_VN_OCR(
    ipSrcImage   := ipBinaryImage,
    eModel       := TCVN_OMT_NUMBERS,
    ipCharacters := ipCharactersResults,
    hrPrev       := hr);

// Export character to string
hr := F_VN_ExportSubContainer_String(ipCharactersResults, 0, sText, 255, hr);
```

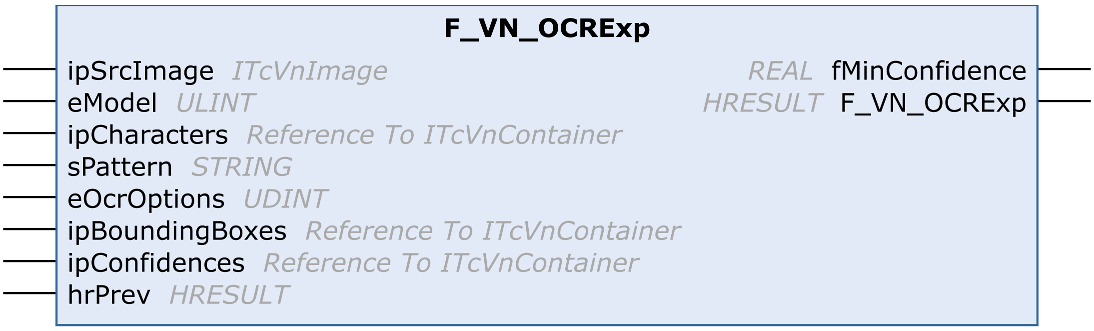
Required License

TC3 Vision OCR

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.22.2 F_VN_OCRExp



Detects and recognizes characters in a binary image (white characters on black background). (expert function)

Syntax

Definition:

```
FUNCTION F_VN_OCRExp : HRESULT
VAR_INPUT
    ipSrcImage       : ITcVnImage;
    eModel           : ULINT;
    ipCharacters     : Reference To ITcVnContainer;
    sPattern         : STRING;
    eOcrOptions      : UDINT;
    ipBoundingBoxes : Reference To ITcVnContainer;
    ipConfidences    : Reference To ITcVnContainer;
    hrPrev           : HRESULT;
END_VAR
VAR_OUTPUT
    fMinConfidence   : REAL;
END_VAR
```

Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT, 1 channel, binary)
eModel	ULINT	Selection of OCR model types (ETcVnOcrModelType)
ipCharacters	Reference To ITcVnContainer [▶ 349]	Returns the recognized characters (ContainerType_Vector_String_SINT)
sPattern	STRING	String pattern containing the format in which characters are presented
eOcrOptions	UDINT	Specifies which options must be applied to the function (ETcVnOcrOptions)
ipBoundingBoxes	Reference To ITcVnContainer [▶ 349]	Returns the corresponding bounding boxes of the recognized characters (ContainerType_Vector_TcVnRectangle_DINT, optional, set to 0 if not required)
ipConfidences	Reference To ITcVnContainer [▶ 349]	Returns the corresponding classification confidences of the recognized characters (ContainerType_Vector_REAL, optional, set to 0 if not required)
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Outputs

Name	Type	Description
fMinConfidence	REAL	Returns the minimum value of the confidences

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_OCRExp 是F_VN_OCR [▶ 1495] 的专家变体。它包含额外的参数。

参数

输入图像

传输到 ipSrcImage 的输入图像必须是单通道二进制图像。只允许使用黑底白字。

模型

在函数调用中用于对字符进行分类的 ETcVnOcrModelType [▶ 190] 模型类型必须传输到 eModel。要在一次函数调用中使用多个模型，必须通过 sPattern 指定字符串模式。这些模型可通过 OR 进行链接。

要使用单个模型，必须先使用功能块 FB_VN_InitializeFunction [▶ 1521] 对其进行初始化。

识别字符

识别字符通过引用 ipCharacters 返回。

字符串模式

带有待识别预期格式字符串的字符串模式可通过 sPattern 传输。

支持的格式及其含义：

- !（感叹号）表示忽略字符（不适用于空格）
- .（点）表示任何字符（不适用于空格）
  只有 TCVN_OMT_NUMBERS_SC_UCLETTERS 模型支持使用点，否则点的效果与感叹号相同，并会输出 S_FALSE。
- _（下划线）表示空格（不依赖 eOcrOptions）
- d（数字）表示 0-9 的预期数字
- #（特殊字符）表示预期特殊字符，支持 . / - : =
- u（大写字母）表示 A-Z 的预期大写字母

选项

通过 eOcrOptions 可以指定影响功能和结果输出的选项（ETcVnOcrOptions [▶ 191]）。如果不需要任何选项，则必须传递 0。

边界框

引用 ipBoundingBoxes 可选用于返回识别字符的相应边界框。如果不需要边界框，则可以传输值 0。

置信度

引用 ipConfidences 可选用于返回识别字符的相应分类置信度。如果不需要置信度，则可以传输值 0。

最小置信度

可选返回值 fMinConfidence 返回找到的所有字符的最小置信度值。

应用

例如，调用 OCRExp 函数，使用两个模型识别数字和大写字母并随后导出字符串的过程如下所示：

```
sPattern := 'dd!uuu!dddd'; // e.g. 02-FEB-2024

hr := F_VN_OCRExp(
    ipSrcImage      := ipBinaryImage,
    eModel          := TCVN_OMT_NUMBERS OR TCVN_OMT_UCLETTERS,
    ipCharacters    := ipCharactersResults,
    sPattern        := sPattern,
    eOcrOptions     := eOcrOptions,
    ipBoundingBoxes := ipBoundingBoxes,
    ipConfidences   := ipConfidences,
    hrPrev          := hr,
    fMinConfidence  => fMinConfidence);

// Export character to string
hr := F_VN_ExportSubContainer_String(ipCharactersResults, 0, sText, 255, hr);

// Get the bounding box of the first element
hr := F_VN_GetAt_TcVnRectangle_DINT(ipBoundingBoxes, stRectangle, 0, hr);

// Get the Confidence value of the first element
hr := F_VN_GetAt_REAL(ipConfidences, fConfidence, 0, hr);
```

要评估置信度值或绘制边界框，可以通过 GetAt 访问容器的各个元素。但是，如果要访问容器的几个或所有元素，建议编写一个循环程序或将 ipConfidences 容器完全导出为数组。

Required License

TC3 Vision OCR

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.23 Texture Analysis

该组包含用于分析图像纹理的函数。

函数

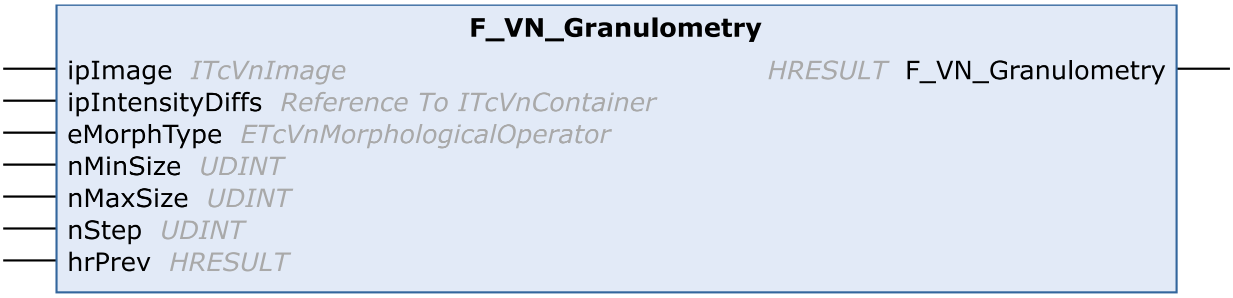
粒度属性

- F_VN_Granulometry [▶ 1499]

Haralick 特征

- F_VN_HaralickFeatures [▶ 1501]

6.1.4.23.1 F_VN_Granulometry



Computes the granulometry of a textured image using morphological operations.

Syntax

Definition:

```
FUNCTION F_VN_Granulometry : HRESULT
VAR_INPUT
    ipImage            : ITcVnImage;
    ipIntensityDiffs : Reference To ITcVnContainer;
    eMorphType         : ETcVnMorphologicalOperator;
    nMinSize           : UDINT;
    nMaxSize           : UDINT;
    nStep              : UDINT;
    hrPrev             : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	Source image (1 channel)
ipIntensityDiffs	Reference To ITcVnContainer [▶ 349]	Returns the intensity differences between successive structuring element sizes (CTcVnContainer_Vector_LREAL), which can be used as features for texture description.
eMorphType	ETcVnMorphologicalOperator [▶ 188]	Only opening or closing are supported.
nMinSize	UDINT	Min structuring element size (odd, >= 3)
nMaxSize	UDINT	Max structuring element size (>= nMinSize)
nStep	UDINT	Step, by which nMinSize is incremented until nMaxSize is exceeded (even, >= 2).
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▶ 122]

更多信息

函数 F_VN_Granulometry 计算图像的粒度属性。这些值可以显示不同图像区域的大小分布。

参数

图像

输入图像 ipImage。

特征

粒度特征通过引用 ipIntensityDiffs 作为容器返回。

算子类型

形态算子的类型通过 eMorphType 定义为 TCVN_MO_OPENING 或 TCVN_MO_CLOSING。

算子大小

形态算子的大小由最小 nMinSize、最大 nMaxSize 和步长 nStep 决定。算子的大小必须是奇数，步长必须是偶数。

应用

例如，可如下使用 3、7 和 11 大小的闭合算子计算粒度特征：

```
hr := F_VN_Granulometry(
    ipImage              :=  ipImage,
    ipIntensityDiffs     :=  ipIntensityDiffs,
    eMorphType           :=  TCVN_MO_CLOSING,
    nMinSize             :=  3,
    nMaxSize             :=  11,
    nStep                :=  4,
    hrPrev               :=  hr);
```

相关函数

- F_VN_HaralickFeatures [▶ 1501]

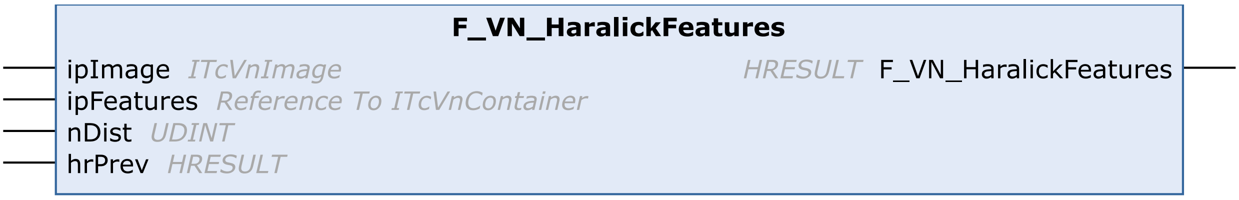
Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.4.23.2 F_VN_HaralickFeatures



Computes the Haralick features of a graylevel image, which describe the texture.

Syntax

Definition:

```
FUNCTION F_VN_HaralickFeatures : HRESULT
VAR_INPUT
    ipImage     : ITcVnImage;
    ipFeatures : Reference To ITcVnContainer;
    nDist       : UDINT;
    hrPrev      : HRESULT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▸ 390]	Source image (1 channel, USINT)
ipFeatures	Reference To ITcVnContainer [▸ 349]	Returns the computed Haralick features (CTcVnContainer_Vector_LREAL). The returned feature set contains the mean and standard deviation for each of the first 13 Haralick features, computed for 0, 45, 90, 135 degrees. So there will be 26 features in total (f1 mean, f1 stdDev, f2 mean, f2 stdDev, ..., f13 mean, f13 stdDev).
nDist	UDINT	Distance
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)

Return value

HRESULT [▸ 122]

更多信息

函数 F_VN_HaralickFeatures 计算图像的 Haralick 属性。这些值指示图像的结构。

参数

图像

输入图像 ipImage。

特征

Haralick 特征作为一个容器通过引用 ipFeatures 返回。

距离

灰度共生矩阵（GLCM）中相邻像素之间的距离指定为 nDist。

应用

例如，可如下计算距离为 5 的 Haralick 特征：

```
hr := F_VN_HaralickFeatures(
    ipImage     :=  ipImage,
    ipFeatures  :=  ipFeatures,
    nDist       :=  5,
    hrPrev      :=  hr);
```

相关函数

- F_VN_Granulometry [▸ 1499]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5 Function Blocks

与 GigE Vision 相机或文件系统的通信以及复杂的图像处理功能由功能块映射。功能块由前缀“FB_VN_”标识。

功能块的分组

TwinCAT Vision 库的所有功能块都按主题分为以下几组：

- 相机寄存器访问 [▸ 1503]
- 文件访问 [▸ 1519]
- 图像采集 [▸ 1540]
- 图像处理 [▸ 1581]

6.1.5.1 License Overview

以下列表显示了 TwinCAT Vision 功能块与可用许可证的分配情况（见许可证模型 [▸ 11]）。它显示了某些功能块的应用需要哪些许可证。

始终需要基本许可证 TC3 Vision Base（TF7100）。

TC3 Vision Base

FB_VN_FileSourceControl [▸ 1541]

FB_VN_GevCameraControl [▸ 1550]

FB_VN_InitializeFunction [▸ 1521]

FB_VN_ReadCalibrationPattern [▸ 1523]

FB_VN_ReadCalibrationResult [▸ 1525]

FB_VN_ReadContainer [▸ 1527]

6.1.5.2 Camera Register Access

该组包含用于读写相机寄存器或内存区域的功能块。

功能块

UDINT-寄存器（4 字节）

- FB_VN_ReadRegister_UDINT [▶ 1508]
- FB_VN_WriteRegister_UDINT [▶ 1515]

ULINT 寄存器（8 字节）

- FB_VN_ReadRegister_ULINT [▶ 1510]
- FB_VN_WriteRegister_ULINT [▶ 1517]

REAL-寄存器（4 字节）

- FB_VN_ReadRegister_REAL [▶ 1506]
- FB_VN_WriteRegister_REAL [▶ 1513]

更大的内存区域

- FB_VN_ReadMemory [▶ 1504]
- FB_VN_WriteMemory [▶ 1512]

参数

以下参数与本组中所有功能块的工作方式相似：

地址

地址 nAddress 指定要读取或保存的相机寄存器。GVCP_REGISTER_ADDRESS 类型只是 UDINT 的别名；因此，地址长度必须为 4 字节（或 32 位）。对于相机的较大内存区域（请参见 FB_VN_ReadMemory [▶ 1504]），地址指向要读取或保存区域的起点。

执行函数

写入和读取命令的触发器分别是bWrite和bRead。它们由一个上升沿触发（从FALSE更改为TRUE）。相关的输入参数被内部保存，因此在执行过程中参数的改变不会产生任何影响。触发器的下降沿也同样没有影响。

调用功能块

请注意，对于功能块的参数描述仅在功能块被实际调用时才有效果。

超时

为了在出现未知错误时不必长时间等待功能块的执行，可以通过nTimeout设置允许的最大执行时间。如果在触发指令触发器后超过这个时间而没有成功的结果，执行将被中止。然后，超时错误可由两个输出bError和nErrorId识别。

工作状态

输出bBusy显示功能块是否繁忙。如果bBusy = TRUE，则不能触发新的命令。

通过合适的IF查询，可以确定功能块的执行是否完成：

```
IF NOT fbReadOrWrite.bBusy THEN
    // Function block execution is complete or error occurred.
    // Next write-command can be issued.
END_IF
```

错误状态

如果适用，错误状态通过以下方式表示：bError以二进制方式表示错误，而nErrorId输出错误代码（ADS 返回代码 [▶ 2753]）。一旦有新的命令被触发，错误状态就会被重置。如果功能块繁忙，错误代码为PENDING (16#71E)（bBusy = TRUE）。

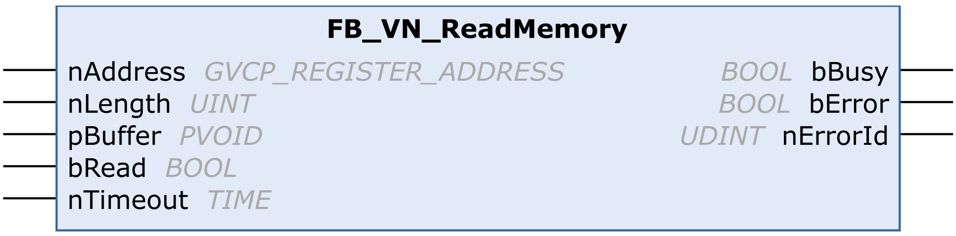
错误代码的十六进制值可以提取如下：

```
IF fbReadOrWrite.bError THEN
    // Show relevant error code for debugging purposes:
    nErrorCode := fbReadOrWrite.nErrorId AND 16#FFF;
END_IF
```

示例

- 读/写寄存器 [▶ 2714]

6.1.5.2.1 FB_VN_ReadMemory



This FB reads consecutive memory locations from the camera. Requires an open control channel (e.g. by calling FB_VN_GevCameraControl.OpenCamera() before)

Syntax

Definition:

```
FUNCTION_BLOCK FB_VN_ReadMemory
VAR_INPUT
    nAddress                  : GVCP_REGISTER_ADDRESS;
    nLength                   : UINT;
    pBuffer                   : PVOID;
    bRead                     : BOOL;
```

```
    nTimeout                : TIME;
END_VAR
VAR_OUTPUT
    bBusy                   : BOOL;
    bError                  : BOOL;
    nErrorId                : UDINT;
END_VAR
VAR
    oidITcVnGevImageProvider : OTCID;
END_VAR
```

Inputs

Name	Type	Default	Description
nAddress	GVCP_REGISTER_ADDRESS [▶ 139]		The address of the first byte to read from the camera (must be 32-bit aligned)
nLength	UINT		The number of bytes to read, starting with nAddress (must be a multiple of 4, not more than 536)
pBuffer	PVOID		Pointer to the buffer where the read memory content is written to (The buffer must have a size of at least nLength bytes!)
bRead	BOOL		Reading the memory is triggered by a rising edge at this input.
nTimeout	TIME	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	UDINT	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

更多信息

需要开放相机控制通道！

GigE Vision 相机的控制通道必须打开，才能使用此功能块。此外，有些相机参数只有在停止图像采集的情况下才能更改。因此，建议首先将相机置于 TCVN_CS_OPENED 状态。

参数

nAddress

寄存器地址标识要寻址的相机寄存器。它属于 GVCP_REGISTER_ADDRESS 类型，是 UDINT 类型的别名。因此，所有寄存器地址的大小都是 4 字节。相机参数的寄存器地址可在 GigE Vision 相机实例配置树中 [▶ 70]的相应点查看：

Property	Value
Name	PixelFormat
Value	BayerRG8
Address	0x00012120
Length	4 Bytes

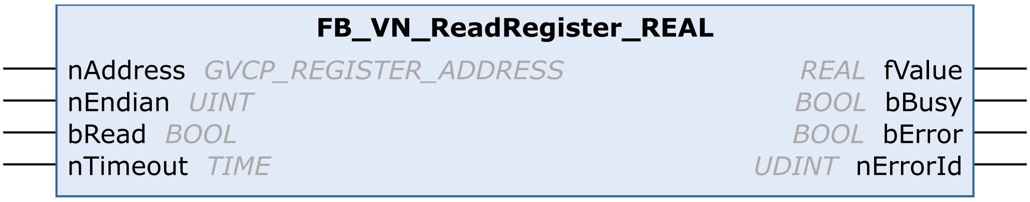
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.2.2 FB_VN_ReadRegister_REAL



This FB reads a REAL register from the camera. Requires an open control channel (e.g. by calling FB_VN_GevCameraControl.OpenCamera() before)

Syntax

Definition:

```
FUNCTION_BLOCK FB_VN_ReadRegister_REAL
VAR_INPUT
    nAddress                 : GVCP_REGISTER_ADDRESS;
    nEndian                  : UINT;
    bRead                    : BOOL;
    nTimeout                 : TIME;
END_VAR
VAR_OUTPUT
    fValue                   : REAL;
    bBusy                    : BOOL;
    bError                   : BOOL;
    nErrorId                 : UDINT;
END_VAR
VAR
    oidITcVnGevImageProvider : OTCID;
END_VAR
```

Inputs

Name	Type	Default	Description
nAddress	GVCP_REGISTER_ADDRESS [▶ 139]		The address of the register that should be read
nEndian	UINT		OPTIONAL: The endianness of the register. 0 = Big, 1 = Little. Default: 0
bRead	BOOL		Reading the register is triggered by a rising edge at this input.
nTimeout	TIME	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
fValue	REAL	The read value
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	UDINT	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

更多信息

需要开放相机控制通道！

GigE Vision 相机的控制通道必须打开，才能使用此功能块。此外，有些相机参数只有在停止图像采集的情况下才能更改。因此，建议首先将相机置于 TCVN_CS_OPENED 状态。

参数

nAddress

寄存器地址标识要寻址的相机寄存器。它属于 GVCP_REGISTER_ADDRESS 类型，是 UDINT 类型的别名。因此，所有寄存器地址的大小都是 4 字节。相机参数的寄存器地址可在 GigE Vision 相机实例配置树中 [▶ 70]的相应点查看：

Property	Value
Name	PixelFormat
Value	BayerRG8
Address	0x00012120
Length	4 Bytes

nEndian

字节顺序描述相机寄存器字节的排列顺序。这在应用中无需进一步考虑，可作为 GigE Vision 相机实例配置树中相应参数的属性：

Property	Value
Length	4 Bytes
Endianness	Big
Sign	Unsigned

大端序用 nEndian := 0 表示，小端序用 nEndian := 1 表示。默认使用大端序。

样本

- 读/写寄存器 [▶ 2714]

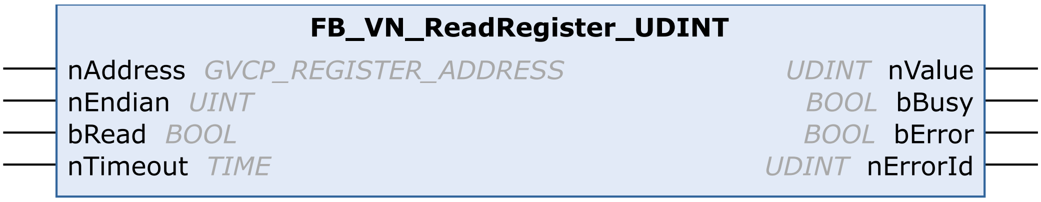
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.2.3 FB_VN_ReadRegister_UDINT



This FB reads a DINT register from the camera. Requires an open control channel (e.g. by calling FB_VN_GevCameraControl.OpenCamera() before)

Syntax

Definition:

```
FUNCTION_BLOCK FB_VN_ReadRegister_UDINT
VAR_INPUT
    nAddress                 : GVCP_REGISTER_ADDRESS;
    nEndian                  : UINT;
    bRead                    : BOOL;
    nTimeout                 : TIME;
END_VAR
VAR_OUTPUT
    nValue                   : UDINT;
    bBusy                    : BOOL;
    bError                   : BOOL;
    nErrorId                 : UDINT;
END_VAR
VAR
    oidITcVnGevImageProvider : OTCID;
END_VAR
```

Inputs

Name	Type	Default	Description
nAddress	GVCP_REGISTER_ADDRESS [▶ 139]		The address of the register that should be read
nEndian	UINT		OPTIONAL: The endianness of the register. 0 = Big, 1 = Little. Default: 0
bRead	BOOL		Reading the register is triggered by a rising edge at this input.
nTimeout	TIME	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
nValue	UDINT	The read value
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	UDINT	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

更多信息

需要开放相机控制通道!

GigE Vision 相机的控制通道必须打开，才能使用此功能块。此外，有些相机参数只有在停止图像采集的情况下才能更改。因此，建议首先将相机置于 TCVN_CS_OPENED 状态。

参数

nAddress

寄存器地址标识要寻址的相机寄存器。它属于 GVCP_REGISTER_ADDRESS 类型，是 UDINT 类型的别名。因此，所有寄存器地址的大小都是 4 字节。相机参数的寄存器地址可在 GigE Vision 相机实例配置树中 [▸ 70]的相应点查看：

Property	Value
Name	PixelFormat
Value	BayerRG8
Address	0x00012120
Length	4 Bytes

nEndian

字节顺序描述相机寄存器字节的排列顺序。这在应用中无需进一步考虑，可作为 GigE Vision 相机实例配置树中相应参数的属性：

Property	Value
Length	4 Bytes
Endianness	Big
Sign	Unsigned

大端序用 nEndian := 0 表示，小端序用 nEndian := 1 表示。默认使用大端序。

样本

- 读/写寄存器 [▸ 2714]

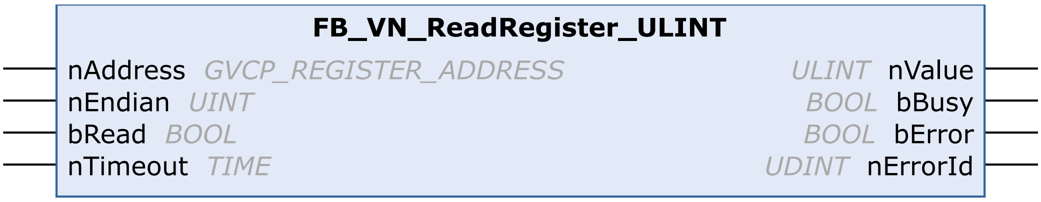
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.2.4 FB_VN_ReadRegister_ULINT



This FB reads a ULINT register from the camera. Requires an open control channel (e.g. by calling FB_VN_GevCameraControl.OpenCamera() before)

Syntax

Definition:

```
FUNCTION_BLOCK FB_VN_ReadRegister_ULINT
VAR_INPUT
    nAddress                 : GVCP_REGISTER_ADDRESS;
    nEndian                  : UINT;
    bRead                    : BOOL;
    nTimeout                 : TIME;
END_VAR
VAR_OUTPUT
    nValue                   : ULINT;
    bBusy                    : BOOL;
    bError                   : BOOL;
    nErrorId                 : UDINT;
END_VAR
VAR
    oidITcVnGevImageProvider : OTCID;
END_VAR
```

Inputs

Name	Type	Default	Description
nAddress	GVCP_REGISTER_ADDRESS [▶ 139]		The address of the register that should be read
nEndian	UINT		OPTIONAL: The endianness of the register. 0 = Big, 1 = Little. Default: 0
bRead	BOOL		Reading the register is triggered by a rising edge at this input.
nTimeout	TIME	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
nValue	ULINT	The read value
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	UDINT	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

更多信息

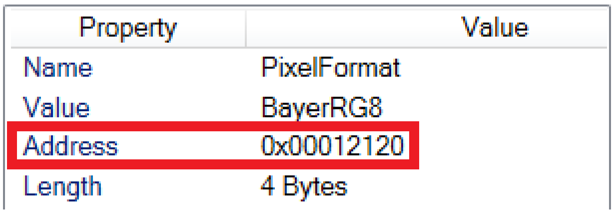
需要开放相机控制通道！

GigE Vision 相机的控制通道必须打开，才能使用此功能块。此外，有些相机参数只有在停止图像采集的情况下才能更改。因此，建议首先将相机置于 TCVN_CS_OPENED 状态。

参数

nAddress

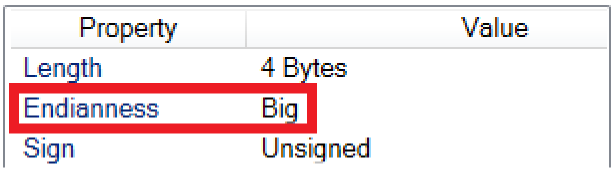
寄存器地址标识要寻址的相机寄存器。它属于 GVCP_REGISTER_ADDRESS 类型，是 UDINT 类型的别名。因此，所有寄存器地址的大小都是 4 字节。相机参数的寄存器地址可在 GigE Vision 相机实例配置树中 [▶ 70]的相应点查看：



Property	Value
Name	PixelFormat
Value	BayerRG8
Address	0x00012120
Length	4 Bytes

nEndian

字节顺序描述相机寄存器字节的排列顺序。这在应用中无需进一步考虑，可作为 GigE Vision 相机实例配置树中相应参数的属性：



Property	Value
Length	4 Bytes
Endianness	Big
Sign	Unsigned

大端序用 nEndian := 0 表示，小端序用 nEndian := 1 表示。默认使用大端序。

示例

读/写寄存器 [▶ 2714]

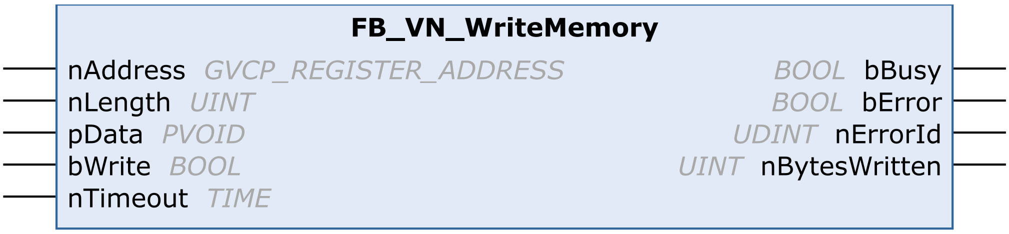
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.2.5 FB_VN_WriteMemory



This FB writes data to consecutive memory locations on the camera. Requires an open control channel (e.g. by calling FB_VN_GevCameraControl.OpenCamera() before)

Syntax

Definition:

```
FUNCTION_BLOCK FB_VN_WriteMemory
VAR_INPUT
    nAddress                 : GVCP_REGISTER_ADDRESS;
    nLength                  : UINT;
    pData                    : PVOID;
    bWrite                   : BOOL;
    nTimeout                 : TIME;
END_VAR
VAR_OUTPUT
    bBusy                    : BOOL;
    bError                   : BOOL;
    nErrorId                 : UDINT;
    nBytesWritten            : UINT;
END_VAR
VAR
    oidITcVnGevImageProvider : OTCID;
END_VAR
```

Inputs

Name	Type	Default	Description
nAddress	GVCP_REGISTER_ADDRESS [▶ 139]		The address of the first byte to write to the camera (must be 32-bit aligned)
nLength	UINT		The number of bytes to write, starting at nAddress (must be a multiple of 4, not more than 536)
pData	PVOID		Pointer to the data that should be written to the camera (must have a size of at least nLength bytes!)
bWrite	BOOL		Writing the memory is triggered by a rising edge at this input.
nTimeout	TIME	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	UDINT	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.
nBytesWritten	UINT	The number of bytes that have been successfully written to the camera.

更多信息

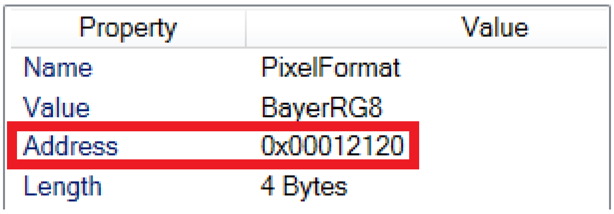
需要开放相机控制通道！

GigE Vision 相机的控制通道必须打开，才能使用此功能块。此外，有些相机参数只有在停止图像采集的情况下才能更改。因此，建议首先将相机置于 TCVN_CS_OPENED 状态。

参数

nAddress

寄存器地址标识要寻址的相机寄存器。它属于 GVCP_REGISTER_ADDRESS 类型，是 UDINT 类型的别名。因此，所有寄存器地址的大小都是 4 字节。相机参数的寄存器地址可在 GigE Vision 相机实例配置树中 [▶ 70]的相应点查看：



Property	Value
Name	PixelFormat
Value	BayerRG8
Address	0x00012120
Length	4 Bytes

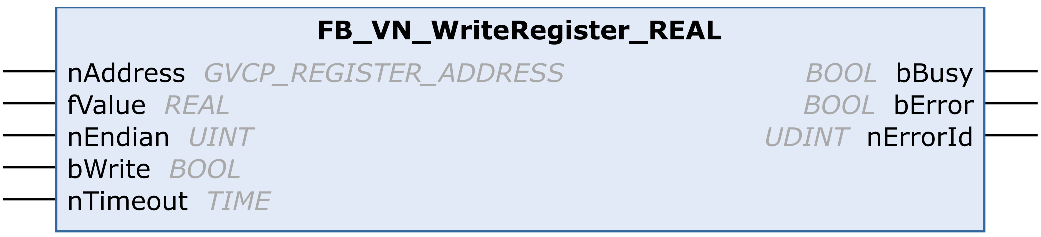
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.2.6 FB_VN_WriteRegister_REAL



This FB writes a REAL value into a register on the camera. Requires an open control channel (e.g. by calling FB_VN_GevCameraControl.OpenCamera() before)

Syntax

Definition:

```
FUNCTION_BLOCK FB_VN_WriteRegister_REAL
VAR_INPUT
    nAddress                 : GVCP_REGISTER_ADDRESS;
    fValue                   : REAL;
    nEndian                  : UINT;
    bWrite                   : BOOL;
    nTimeout                 : TIME;
END_VAR
VAR_OUTPUT
    bBusy                    : BOOL;
    bError                   : BOOL;
    nErrorId                 : UDINT;
END_VAR
VAR
    oidITcVnGevImageProvider : OTCID;
END_VAR
```

Inputs

Name	Type	Default	Description
nAddress	GVCP_REGISTER_ADDRESS [▸ 139]		The address of the register that should be written
fValue	REAL		The value to write
nEndian	UINT		OPTIONAL: The endianness of the register. 0 = Big, 1 = Little. Default: 0
bWrite	BOOL		Writing the register is triggered by a rising edge at this input.
nTimeout	TIME	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	UDINT	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

更多信息

需要开放相机控制通道！

GigE Vision 相机的控制通道必须打开，才能使用此功能块。此外，有些相机参数只有在停止图像采集的情况下才能更改。因此，建议首先将相机置于 TCVN_CS_OPENED 状态。

参数

nAddress

寄存器地址标识要寻址的相机寄存器。它属于 GVCP_REGISTER_ADDRESS 类型，是 UDINT 类型的别名。因此，所有寄存器地址的大小都是 4 字节。相机参数的寄存器地址可在 GigE Vision 相机实例配置树中 [▶ 70]的相应点查看：

Property	Value
Name	PixelFormat
Value	BayerRG8
Address	0x00012120
Length	4 Bytes

nEndian

字节顺序描述相机寄存器字节的排列顺序。这在应用中无需进一步考虑，可作为 GigE Vision 相机实例配置树中相应参数的属性：

Property	Value
Length	4 Bytes
Endianness	Big
Sign	Unsigned

大端序用 nEndian := 0 表示，小端序用 nEndian := 1 表示。默认使用大端序。

样本

- 读/写寄存器 [▶ 2714]

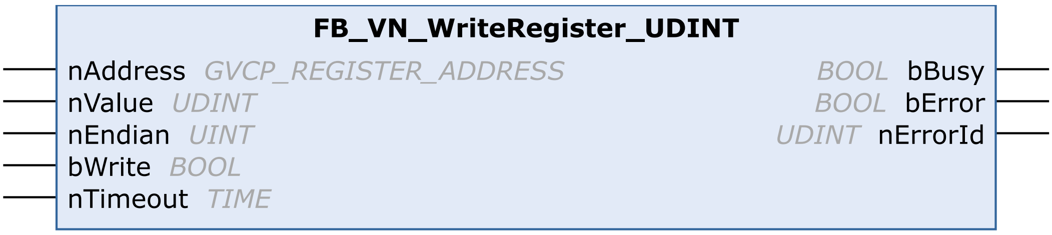
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.2.7 FB_VN_WriteRegister_UDINT



This FB writes a DINT value into a register on the camera. Requires an open control channel (e.g. by calling FB_VN_GevCameraControl.OpenCamera() before)

Syntax

Definition:

```
FUNCTION_BLOCK FB_VN_WriteRegister_UDINT
VAR_INPUT
    nAddress                    : GVCP_REGISTER_ADDRESS;
    nValue                      : UDINT;
    nEndian                     : UINT;
    bWrite                      : BOOL;
    nTimeout                    : TIME;
```

```
END_VAR
VAR_OUTPUT
    bBusy                     : BOOL;
    bError                    : BOOL;
    nErrorId                  : UDINT;
END_VAR
VAR
    oidITcVnGevImageProvider : OTCID;
END_VAR
```

Inputs

Name	Type	Default	Description
nAddress	GVCP_REGISTER_ADDRESS [▶ 139]		The address of the register that should be written
nValue	UDINT		The value to write
nEndian	UINT		OPTIONAL: The endianness of the register. 0 = Big, 1 = Little. Default: 0
bWrite	BOOL		Writing the register is triggered by a rising edge at this input.
nTimeout	TIME	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	UDINT	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

更多信息

需要开放相机控制通道!

GigE Vision 相机的控制通道必须打开，才能使用此功能块。此外，有些相机参数只有在停止图像采集的情况下才能更改。因此，建议首先将相机置于 TCVN_CS_OPENED 状态。

参数

nAddress

寄存器地址标识要寻址的相机寄存器。它属于 GVCP_REGISTER_ADDRESS 类型，是 UDINT 类型的别名。因此，所有寄存器地址的大小都是 4 字节。相机参数的寄存器地址可在 GigE Vision 相机实例配置树中 [▶ 70]的相应点查看：

Property	Value
Name	PixelFormat
Value	BayerRG8
Address	0x00012120
Length	4 Bytes

nEndian

字节顺序描述相机寄存器字节的排列顺序。这在应用中无需进一步考虑，可作为 GigE Vision 相机实例配置树中相应参数的属性：

Property	Value
Length	4 Bytes
Endianness	Big
Sign	Unsigned

大端序用 nEndian := 0 表示，小端序用 nEndian := 1 表示。默认使用大端序。

样本

- 读/写寄存器 [▶ 2714]

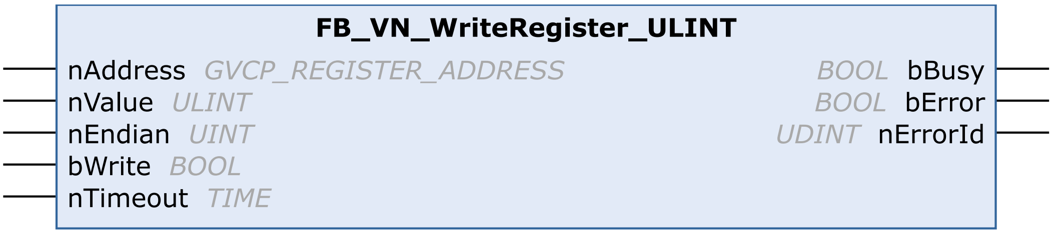
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.2.8 FB_VN_WriteRegister_ULINT



This FB writes a ULINT value into a register on the camera. Requires an open control channel (e.g. by calling FB_VN_GevCameraControl.OpenCamera() before)

Syntax

Definition:

```
FUNCTION_BLOCK FB_VN_WriteRegister_ULINT
VAR_INPUT
    nAddress                : GVCP_REGISTER_ADDRESS;
    nValue                  : ULINT;
    nEndian                 : UINT;
    bWrite                  : BOOL;
    nTimeout                : TIME;
END_VAR
VAR_OUTPUT
    bBusy                   : BOOL;
    bError                  : BOOL;
    nErrorId                : UDINT;
END_VAR
VAR
    oidITcVnGevImageProvider : OTCID;
END_VAR
```

Inputs

Name	Type	Default	Description
nAddress	GVCP_REGISTER_ADDRESS [▶ 139]		The address of the register that should be written
nValue	ULINT		The value to write
nEndian	UINT		OPTIONAL: The endianness of the register. 0 = Big, 1 = Little. Default: 0
bWrite	BOOL		Writing the register is triggered by a rising edge at this input.
nTimeout	TIME	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	UDINT	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

更多信息

需要开放相机控制通道！

GigE Vision 相机的控制通道必须打开，才能使用此功能块。此外，有些相机参数只有在停止图像采集的情况下才能更改。因此，建议首先将相机置于 TCVN_CS_OPENED 状态。

参数

nAddress

寄存器地址标识要寻址的相机寄存器。它属于 GVCP_REGISTER_ADDRESS 类型，是 UDINT 类型的别名。因此，所有寄存器地址的大小都是 4 字节。相机参数的寄存器地址可在 GigE Vision 相机实例配置树中 [▶ 70]的相应点查看：

Property	Value
Name	PixelFormat
Value	BayerRG8
Address	0x00012120
Length	4 Bytes

nEndian

字节顺序描述相机寄存器字节的排列顺序。这在应用中无需进一步考虑，可作为 GigE Vision 相机实例配置树中相应参数的属性：

Property	Value
Length	4 Bytes
Endianness	Big
Sign	Unsigned

大端序用 nEndian := 0 表示，小端序用 nEndian := 1 表示。默认使用大端序。

示例

读/写寄存器 [▸ 2714]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.3 File Access

该组包含用于在文件系统上读写数据的功能块。

功能块

图像

- FB_VN_ReadImage [▸ 1529]
- FB_VN_WriteImage [▸ 1536]

容器

- FB_VN_ReadContainer [▸ 1527]
- FB_VN_WriteContainer [▸ 1534]

机器学习模型

- FB_VN_ReadMlModel [▸ 1530]
- FB_VN_WriteMlModel [▸ 1538]

标定模板

- FB_VN_ReadCalibrationPattern [▸ 1523]
- FB_VN_ReadCalibrationResult [▸ 1526]
- FB_VN_WriteCalibrationResult [▸ 1533]

函数初始化

- FB_VN_InitializeFunction [▸ 1521]

综述

这些功能块异步工作；从第一次执行到所需结果的时间延长了几个周期。每个读/写操作的实际时间取决于几个因素，如图像大小和操作系统行为。以下参数对所有功能块都是一样，因此在此集中描述一次。

参数

文件路径

文件路径sFilePath指定保存文件的位置或加载哪个文件。可以使用绝对和相对路径。在相对路径的情况下，服务配置 [▸ 59]中设置的默认路径被作为基础。

路径长度

sFilePath 的类型被指定为 STRING，但可以传递长度不超过 255 个字符的路径。

文件路径在写入命令bWrite的上升沿时内部存储，因此，在以后的周期中对路径的改变没有影响。

执行函数

写入和读取命令的触发器分别是bWrite和bRead。它们由一个上升沿触发（从FALSE更改为TRUE）。相关的输入参数被内部保存，因此在执行过程中参数的改变不会产生任何影响。触发器的下降沿也同样没有影响。

调用功能块

请注意，对于功能块的参数描述仅在功能块被实际调用时才有效果。

超时

为了在出现未知错误时不必长时间等待功能块的执行，可以通过nTimeout设置允许的最大执行时间。如果在触发指令触发器后超过这个时间而没有成功的结果，执行将被中止。然后，超时错误可由两个输出bError和nErrorId识别。

工作状态

输出bBusy显示功能块是否繁忙。如果bBusy = TRUE，则不能触发新的命令。

通过合适的IF查询，可以确定功能块的执行是否完成：

```
IF NOT fbReadOrWrite.bBusy THEN
    // Function block execution is complete or error occurred.
    // Next write-command can be issued.
END_IF
```

错误状态

如果适用，错误状态通过以下方式表示：bError以二进制方式表示错误，而nErrorId输出错误代码（ADS 返回代码 [▶ 2753]）。一旦有新的命令被触发，错误状态就会被重置。如果功能块繁忙，错误代码为PENDING (16#71E)（bBusy = TRUE）。

错误代码的十六进制值可以提取如下：

```
IF fbReadOrWrite.bError THEN
    // Show relevant error code for debugging purposes:
    nErrorCode := fbReadOrWrite.nErrorId AND 16#FFF;
END_IF
```

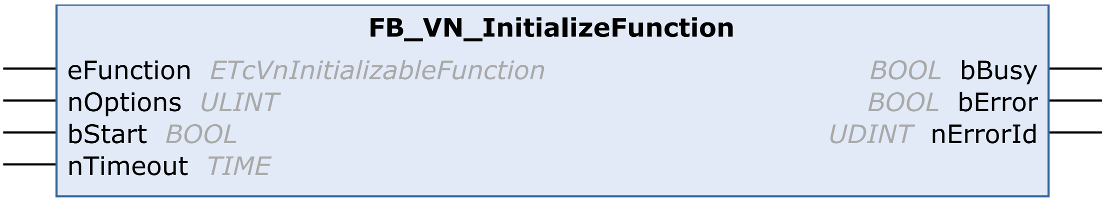
最常见的错误代码：

十六进制	十进制	名称	含义
16#006	6	TARGETPORTNOTFOUND	ADS 服务器无法访问。基本上是在 TwinCAT Vision 服务没有启动时发生。参见视觉服务不启动 [▶ 2739]。
16#700	1792	ERROR	如果保存或加载失败，则出现一般错误。这通常表明指定的文件格式不受支持或者文件路径无法访问。 该错误通常发生在TwinCAT Vision Service [▶ 59]中。
16#70B	1803	INVALIDPARM	无效参数。这通常表明接口指针 [▶ 119]无效。
16#719	1817	TIMEOUT	超时。加载或保存已超过最大执行时间nTimeout。一般来说，主要发生在 ADS 的负荷较重的情况下。
16#71E	1822	PENDING	并非真正的错误，因此这里没有设置输出bError。该代码发出信号，表明该功能块处于繁忙状态。如果在功能块繁忙时记录到新的上升沿，将在输出bBusy同时发生。

示例

- 保存 PLC 的图像 [▶ 2716]

6.1.5.3.1 FB_VN_InitializeFunction



This FB initializes selected functions. Before, it needs to initialize itself, i.e. call the FB until the output bInitialized is true before using it for function initialization.

Syntax

Definition:

```
FUNCTION_BLOCK FB_VN_InitializeFunction
VAR_INPUT
    eFunction : ETcVnInitializableFunction;
    nOptions  : ULINT;
    bStart    : BOOL;
    nTimeout  : TIME;
END_VAR
VAR_OUTPUT
    bBusy     : BOOL;
    bError    : BOOL;
    nErrorId  : UDINT;
END_VAR
```

Inputs

Name	Type	Default	Description
eFunction	ETcVnInitializableFunction [▶ 185]		Selects the function to initialize
nOptions	ULINT	0	Selects initialization options for the function
bStart	BOOL		Function initialization is triggered by a rising edge at this input.
nTimeout	TIME	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	UDINT	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

更多信息

功能块 FB_VN_InitializeFunction 初始化在 eFunction 指定的函数。所需数据在后台异步加载到 PLC。

参数

文件访问参数 [▶ 1519]部分对某些参数进行集中描述。与描述不同的是，必须为该功能块指定待初始化函数，而不是文件路径。

函数

待初始化的 eFunction。

选项

为指定函数选择初始化选项 nOptions。可能的枚举选项可在待初始化函数的描述中找到。这些选项可以与 OR 相链接，从而通过调用功能块加载。

应用

例如，带有大写字母分类选项的 OCR 函数的初始化过程如下所示：

```
fbInitOCR(
    eFunction   := TCVN_IF_OCR,
    nOptions    := ETcVnOcrModelType.TCVN_OMT_UCLETTERS,
    bStart      := TRUE,
    nTimeout    := T#500MS);

IF NOT fbInitOCR.bBusy AND NOT fbInitOCR.bError THEN
    // The function is initialized and can be used with the specified options
END_IF
```

相关函数

- F_VN_CheckFunctionInitialization [▶ 1474] 可用来检查函数是否已通过相应选项初始化。
- F_VN_DeinitializeFunction [▶ 1481] 提供取消初始化函数选项从而释放内存的可能性。

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.3.2 FB_VN_ReadCalibrationPattern

This FB reads calibration pattern reference points from an xml file on the targte pc.

Syntax

Definition:

```
FUNCTION_BLOCK FB_VN_ReadCalibrationPattern
VAR_INPUT
    sFilePath       : STRING;
    ipDestContainer : Reference To ITcVnContainer;
    bRead           : BOOL;
    nTimeout        : TIME;
END_VAR
VAR_OUTPUT
    bBusy           : BOOL;
    bError          : BOOL;
    nErrorId        : UDINT;
END_VAR
```

Inputs

Name	Type	Default	Description
sFilePath	STRING		Full path of the file or relative path to the default directory on the target pc
ipDestContainer	Reference To ITcVnContainer [▶ 349]		Returns a container with the pattern points (ContainerType_Vector_TcVnPoint3_REAL)
bRead	BOOL		Reading the file is triggered by a rising edge at this input.
nTimeout	TIME	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	UDINT	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

更多信息

功能块FB_VN_ReadCalibrationPattern从文件中异步加载单个校准模式 [▶ 105]的描述。

加载修改的容器

在加载容器时，视觉服务在内部缓存数据。因此，无法在标准系统中加载更改的容器。为此，请调整缓存设置 [▶ 59]或者参见串行加载图像到 PLC 的样本 [▶ 2717]。

参数

在文件访问参数 [▶ 1519]一节中集中描述了一些参数。

加载的容器（返回值）

在成功的读取操作之后，可以通过接口指针ipPatternReferencePoints访问加载了校准模式参考点的容器。

应用

例如，从路径C:\TcVision\CalibrationPatternReferencePoints.xml加载校准模式参考点如下所示：

```
fbReadCalibrationPattern(
    sFilePath       := 'C:\TcVision\CalibrationPatternReferencePoints.xml',
    ipDestContainer := ipPatternReferencePoints,
    bRead           := TRUE,
    nTimeout        := T#500MS);

IF NOT fbReadCalibrationPattern.bBusy AND NOT fbReadCalibrationPattern.bError THEN
    // reference points are accessible in ipPatternReferencePoints
END_IF
```

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.3.3 FB_VN_ReadCalibrationResult

FB_VN_ReadCalibrationResult

sFilePath *STRING* — *BOOL* bBusy
aCameraMatrix *Reference To TcVnMatrix3x3_LREAL* — *BOOL* bError
aDistortionCoefficients *Reference To TcVnArray8_LREAL* — *UDINT* nErrorId
aRotationMatrix *Reference To TcVnMatrix3x3_LREAL*
aTranslationVector *Reference To TcVnVector3_LREAL*
bRead *BOOL*
nTimeout *TIME*

This FB reads calibration results from a json file on the targte pc.

Syntax

Definition:

```
FUNCTION_BLOCK FB_VN_ReadCalibrationResult
VAR_INPUT
    sFilePath               : STRING;
    aCameraMatrix           : Reference To TcVnMatrix3x3_LREAL;
    aDistortionCoefficients : Reference To TcVnArray8_LREAL;
    aRotationMatrix         : Reference To TcVnMatrix3x3_LREAL;
    aTranslationVector      : Reference To TcVnVector3_LREAL;
    bRead                   : BOOL;
    nTimeout                : TIME;
END_VAR
VAR_OUTPUT
    bBusy                   : BOOL;
    bError                  : BOOL;
    nErrorId                : UDINT;
END_VAR
```

Inputs

Name	Type	Default	Description
sFilePath	STRING		Full path of the file or relative path to the default directory on the target pc
aCameraMatrix	Reference To TcVnMatrix3x3_LREAL [▸ 141]		Returns the camera matrix
aDistortionCoefficients	Reference To TcVnArray8_LREAL [▸ 141]		Returns the distortion coefficients
aRotationMatrix	Reference To TcVnMatrix3x3_LREAL [▸ 141]		Returns the rotation matrix
aTranslationVector	Reference To TcVnVector3_LREAL [▸ 141]		Returns the translation vector
bRead	BOOL		Reading the file is triggered by a rising edge at this input.
nTimeout	TIME	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	UDINT	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

更多信息

功能块FB_VN_ReadCalibrationResult将校准结果从文件中异步加载到 PLC。

加载更改后的校准结果。

在加载校准结果时，视觉服务在内部缓存了数据。因此，无法在标准系统中加载修改的结果。为此，请调整缓存设置 [▸ 59]或者参见串行加载图像到 PLC 的样本 [▸ 2717]。

参数

在文件访问参数 [▸ 1519]一节中集中描述了一些参数。

校准结果

读取操作成功后，可以通过以下变量访问加载的校准结果：

```
// Calibration results
aCameraMatrix            : TcVnMatrix3x3_LREAL;
aDistortionCoefficients  : TcVnArray8_LREAL;
aRotationMatrix          : TcVnMatrix3x3_LREAL;
aTranslationVector       : TcVnVector3_LREAL;
```

应用

例如，从路径C:\TcVision\CalibrationResult.json加载校准结果如下所示：

```
fbReadCalibrationResult(
        sFilePath                := 'C:\TcVision\ CalibrationResult.json',
        aCameraMatrix            := aCameraMatrix,
        aDistortionCoefficients  := aDistortionCoefficients,
        aRotationMatrix          := aRotationMatrix,
        aTranslationVector       := aTranslationVector,
        bRead                    := TRUE,
        nTimeout                 := T#500MS);

IF NOT fbReadCalibrationResult.bBusy AND NOT fbReadCalibrationResult.bError THEN
    // calibration results are accessible
END_IF
```

相关功能块：FB_VN_WriteCalibrationResult [▸ 1532]。

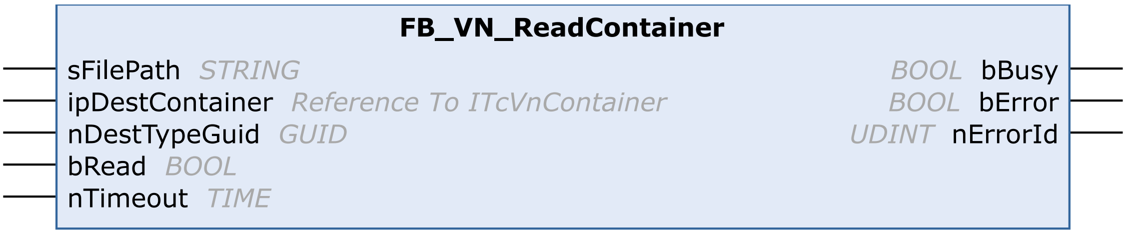
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.3.4 FB_VN_ReadContainer



This FB reads a container from an xml file on the target pc.

Syntax

Definition:

```
FUNCTION_BLOCK FB_VN_ReadContainer
VAR_INPUT
    sFilePath       : STRING;
    ipDestContainer : Reference To ITcVnContainer;
    nDestTypeGuid   : GUID;
    bRead           : BOOL;
    nTimeout        : TIME;
END_VAR
VAR_OUTPUT
    bBusy           : BOOL;
    bError          : BOOL;
    nErrorId        : UDINT;
END_VAR
```

Inputs

Name	Type	Default	Description
sFilePath	STRING		Full path of the file or relative path to the default directory on the target pc
ipDestContainer	Reference To ITcVnContainer [▶ 349]		Returns a container with the loaded content
nDestTypeGuid	GUID		The type id of the container to return. If this parameter is not set, a suitable container type will be determined automatically (except for csv files).
bRead	BOOL		Reading the file is triggered by a rising edge at this input.
nTimeout	TIME	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	UDINT	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

更多信息

功能块FB_VN_ReadContainer将一个容器 [▸ 132]从一个文件异步加载到 PLC 中。

加载修改的容器。

在加载容器时，视觉服务在内部缓存数据。因此，无法在标准系统中加载更改的容器。为此，请调整缓存设置 [▸ 59]或参见示例连续加载图像到 PLC [▸ 2717]代替。

参数

在文件访问参数 [▸ 1519]一节中集中描述了一些参数。

容器类型

可以选择性地指定待加载容器的容器类型 [▸ 139]。

加载的容器（返回值）

读取成功后，加载的容器可以通过接口指针ipDestContainer访问。

应用

例如，从路径C:\TcVision\Container.xml加载具有明确类型规范的容器ContainerType_Vector_TcVnPoint2_DINT如下所示：

```
fbReadContainer(
    sFilePath        := 'C:\TcVision\Container.xml',
    ipDestContainer  := ipContainer,
    nDestTypeGuid    := ContainerType_Vector_TcVnPoint2_DINT,
    bRead            := TRUE,
    nTimeout         := T#500MS);

IF NOT fbReadContainer.bBusy AND NOT fbReadContainer.bError THEN
    // container is accessible in ipContainer
END_IF
```

相关功能块：FB_VN_WriteContainer [▸ 1534]。

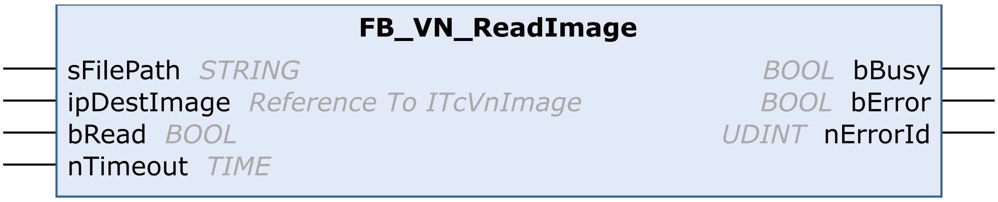
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.3.5 FB_VN_ReadImage



This FB reads an image from a file on the target pc.

Syntax

Definition:

```
FUNCTION_BLOCK FB_VN_ReadImage
VAR_INPUT
    sFilePath   : STRING;
    ipDestImage : Reference To ITcVnImage;
    bRead       : BOOL;
    nTimeout    : TIME;
END_VAR
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrorId    : UDINT;
END_VAR
```

Inputs

Name	Type	Default	Description
sFilePath	STRING		Full path of the file or relative path to the default image directory on the target pc
ipDestImage	Reference To ITcVnImage [▶ 390]		Returns an image with the loaded content
bRead	BOOL		Reading the file is triggered by a rising edge at this input.
nTimeout	TIME	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	UDINT	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

更多信息

功能块FB_VN_ReadImage将一个图像 [▶ 130]从一个文件异步加载到 PLC 中。

加载修改过的图像

当加载图像时，Vision 服务在内部缓存数据。因此，无法在标准系统中加载更改过的图像。为此，请调整缓存设置 [▸ 59]或参见示例连续加载图像到 PLC [▸ 2717]代替。

参数

在文件访问参数 [▸ 1519]一节中集中描述了一些参数。

目标图像（返回值）

读取成功后，加载的图像可以通过接口指针ipDestImage访问。

应用

例如，从路径C:\TcVision\Image.bmp加载图像如下所示：

```
fbReadImage(
    sFilePath   := 'C:\TcVision\Image.bmp',
    ipDestImage := ipImageIn,
    bRead       := TRUE,
    nTimeout    := T#500MS);

IF NOT fbReadImage.bBusy AND NOT fbReadImage.bError THEN
    // image is accessible in ipImageIn
END_IF
```

相关的功能块：FB_VN_WriteImage [▸ 1536]，用于保存图像。

样本

- 连续加载图像到 PLC [▸ 2717]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.3.6 FB_VN_ReadMlModel

This FB reads a machine learning model from a file on the target pc.

Syntax

Definition:

```
FUNCTION_BLOCK FB_VN_ReadMlModel
VAR_INPUT
    sFilePath     : STRING;
    ipDestMlModel : Reference To ITcVnMlModel;
    bRead         : BOOL;
    nTimeout      : TIME;
END_VAR
VAR_OUTPUT
    bBusy         : BOOL;
```

```
    bError          : BOOL;
    nErrorId        : UDINT;
END_VAR
```

Inputs

Name	Type	Default	Description
sFilePath	STRING		Full path of the file or relative path to the default directory on the target pc
ipDestMlModel	Reference To ITcVnMlModel [▸ 399]		Returns the loaded machine learning model
bRead	BOOL		Reading the file is triggered by a rising edge at this input.
nTimeout	TIME	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	UDINT	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

更多信息

功能块FB_VN_ReadMlModel将一个ML 模型 [▸ 399]从一个文件异步加载到 PLC 中。

加载修改的 ML 模型

在加载 ML 模型时，视觉服务在内部缓存数据。因此，无法在标准系统中加载修改过的 ML 模型。为此，请调整缓存设置 [▸ 59]或参见示例连续加载图像到 PLC [▸ 2717]代替。

参数

在文件访问参数 [▸ 1519]一节中集中描述了一些参数。

加载的 ML 模型（返回值）

读取成功后，加载的ITcVnMlModel可通过接口指针ipDestMlModel访问。

一个特殊的情况是作为ITcVnColorModel，因为在加载模型之后，仍然需要与下面的行进行转换。另外，ITcVnMlModel也可以通过相应的函数直接使用。

```
hr := ipReadMlModel.TcQueryInterface(IID_ITcVnColorModel, ADR(ipColorModel));
```

应用

例如，从路径C:\TcVision\MlModel.vmlm加载 ML 模型如下所示：

```
fbReadMlModel(
    sFilePath       := 'C:\TcVision\MlModel.vmlm',
    ipDestMlModel   := ipReadMlModel,
    bRead           := TRUE,
```

```
    nTimeout         := T#500MS);

IF NOT fbReadMlModel.bBusy AND NOT fbReadMlModel.bError THEN
    // Ml-Model is accessible in ipReadMlModel
END_IF
```

相关的功能块：FB_VN_WriteMlModel [▶ 1538]。

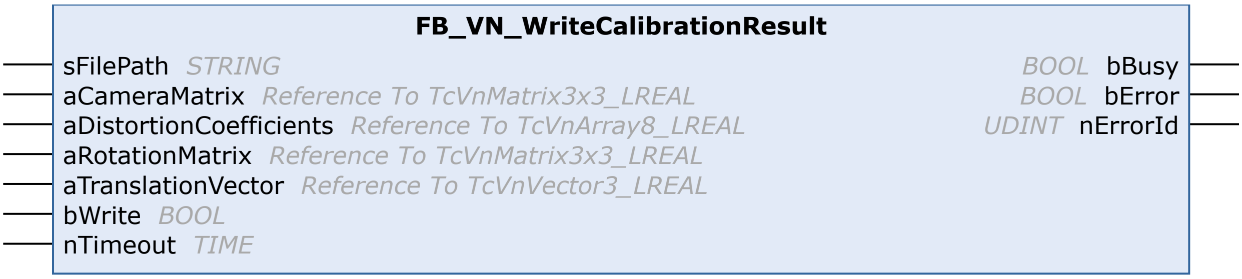
Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.3.7 FB_VN_WriteCalibrationResult



This FB writes calibration results as a json file to the hard drive.

Syntax

Definition:

```
FUNCTION_BLOCK FB_VN_WriteCalibrationResult
VAR_INPUT
    sFilePath               : STRING;
    aCameraMatrix           : Reference To TcVnMatrix3x3_LREAL;
    aDistortionCoefficients : Reference To TcVnArray8_LREAL;
    aRotationMatrix         : Reference To TcVnMatrix3x3_LREAL;
    aTranslationVector      : Reference To TcVnVector3_LREAL;
    bWrite                  : BOOL;
    nTimeout                : TIME;
END_VAR
VAR_OUTPUT
    bBusy                   : BOOL;
    bError                  : BOOL;
    nErrorId                : UDINT;
END_VAR
```

Inputs

Name	Type	Default	Description
sFilePath	STRING		Full path of the file or relative path to the default directory on the target pc (e.g. only a file name). If the string is empty, a file name will be generated (containing the current time).
aCameraMatrix	Reference To TcVnMatrix3x3_LREAL [▸ 141]		Returns the camera matrix
aDistortionCoefficients	Reference To TcVnArray8_LREAL [▸ 141]		Returns the distortion coefficients
aRotationMatrix	Reference To TcVnMatrix3x3_LREAL [▸ 141]		Returns the rotation matrix
aTranslationVector	Reference To TcVnVector3_LREAL [▸ 141]		Returns the translation vector
bWrite	BOOL		Writing the container is triggered by a rising edge at this input.
nTimeout	TIME	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	UDINT	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

更多信息

功能块FB_VN_WriteCalibrationResult将 PLC 的校准结果异步保存在一个文件中。

参数

在文件访问参数 [▸ 1519]一节中集中描述了一些参数。

校准结果

以下校准功能 [▸ 1031]的结果以JSON格式保存在指定的文件路径下。

```
// Calibration results
aCameraMatrix             : TcVnMatrix3x3_LREAL;
aDistortionCoefficients   : TcVnArray8_LREAL;
aRotationMatrix           : TcVnMatrix3x3_LREAL;
aTranslationVector        : TcVnVector3_LREAL;
```

文件路径和格式

如果sFilePath为空，将生成带有当前时间的文件名，并使用扩展名json。

应用

例如，将校准结果保存到C:\TcVision\CalibrationResult.json文件如下所示：

```
fbWriteCalibrationResult(
    sFilePath                := 'C:\TcVision\CalibrationResult.json',
    aCameraMatrix            := aCameraMatrix,
    aDistortionCoefficients := aDistortionCoefficients,
    aRotationMatrix          := aRotationMatrix,
    aTranslationVector       := aTranslationVector,
    bWrite                   := TRUE,
    nTimeout                 := T#500MS);

IF NOT fbWriteCalibrationResult.bBusy AND NOT fbWriteCalibrationResult.bError THEN
    // Calibration result was written successfully to file
END_IF
```

相关功能块：FB_VN_ReadCalibrationResult [▶ 1525]。

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.3.8 FB_VN_WriteContainer

This FB writes a container as an xml file to the hard drive.

Syntax

Definition:

```
FUNCTION_BLOCK FB_VN_WriteContainer
VAR_INPUT
    ipContainer   : Reference To ITcVnContainer;
    sFilePath     : STRING;
    eExportFormat : ETcVnContainerExportFormat;
    bWrite        : BOOL;
    nTimeout      : TIME;
END_VAR
VAR_OUTPUT
    bBusy         : BOOL;
    bError        : BOOL;
    nErrorId      : UDINT;
END_VAR
```

Inputs

Name	Type	Default	Description
ipContainer	Reference To ITcVnContainer [▶ 349]		The container to write
sFilePath	STRING		Full path of the file or relative path to the default directory on the target pc (e.g. only a file name). If the string is empty, a file name will be generated (containing the current time).
eExportFormat	ETcVnContainerExportFormat [▶ 167]	TCVN_CEF_XML	Container export format (default is human readable xml)
bWrite	BOOL		Writing the container is triggered by a rising edge at this input.
nTimeout	TIME	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	UDINT	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

更多信息

功能块FB_VN_WriteContainer将一个容器 [▶ 132]从一个文件异步保存到 PLC 中。

参数

在文件访问参数 [▶ 1519]一节中集中描述了一些参数。

容器

容器ipContainer以指定的文件路径保存为 XML 文件。

出口格式

导出格式eExportFormat通过枚举EtcVnContainerExportFormat [▶ 167]指定容器以人类可读的形式保存或进行序列化处理：

- TCVN_CEF_XML
- TCVN_CEF_XML_SERIALIZED
- TCVN_CEF_CSV
- TCVN_CEF_BINARY

人类可读的形式（XML、CSV）的优点是，可以简单地检查和手动编辑容器中的条目。然而，这可能导致四舍五入的不准确性。序列化格式的优点是不会出现四舍五入的不准确情况。然而，该格式不容易阅读或编辑。CSV 格式的优点是，数据可以很容易地在其他程序中进行外部处理。对于大量的数据，建议使用二进制格式。

应用

例如，在文件C:\TcVision\Container.xml中以序列化的 XML 格式保存一个容器如下所示：

```
fbWriteContainer(
    ipContainer     := ipContainer,
    sFilePath       := 'C:\TcVision\Container.xml',
    eExportFormat   := TCVN_CEF_XML_SERIALIZED,
    bWrite          := TRUE,
    nTimeout        := T#500MS);

IF NOT fbWriteContainer.bBusy AND NOT fbWriteContainer.bError THEN
    // container was written successfully to file
END_IF
```

相关功能块：FB_VN_ReadContainer [▸ 1527]。

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.3.9 FB_VN_WriteImage

This FB writes an image to the hard drive.

Syntax

Definition:

```
FUNCTION_BLOCK FB_VN_WriteImage
VAR_INPUT
    ipImage    : Reference To ITcVnImage;
    sFilePath  : STRING;
    bWrite     : BOOL;
    nTimeout   : TIME;
END_VAR
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrorId   : UDINT;
END_VAR
```

Inputs

Name	Type	Default	Description
ipImage	Reference To ITcVnImage [▸ 390]		The image to write
sFilePath	STRING		Full path of the file or relative path to the default directory on the target pc (e.g. only a file name). If the string is empty, a file name will be generated (containing the current time).
bWrite	BOOL		Writing the image is triggered by a rising edge at this input.
nTimeout	TIME	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	UDINT	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

更多信息

功能块FB_VN_WriteImage将文件中的图像 [▸ 130]异步地保存到 PLC 中。

参数

在文件访问参数 [▸ 1519]一节中集中描述了一些参数。

输入图像

输入图像ipImage的格式必须符合文件路径sFilePath中定义的文件格式。

在 bWrite 的上升沿时进行保存

在写命令的上升沿时，内部复制图像。因此，在接下来的周期中对图像的改变没有影响。在 bWrite 的上升沿时传输的图像始终是保存的图像。

文件路径和格式

文件路径sFilePath指定输入图像ipImage的保存位置以及应该使用哪种图像格式保存。图像格式根据文件扩展名自动识别。如果没有指定文件名，将会自动生成，并包含当前时间。如果没有指定文件扩展名（如.jpg），则自动使用扩展名.bmp，从而使用 BMP 格式。

支持以下图像格式的保存。

- BMP（.bmp）
- PNG（.png）
- JPEG（.jpeg或.jpg）

- TIF (.tif)
- TwinCAT 序列化 (.tcimg)

注意

发生错误的可能性

确保所选择的图像格式与实际输入图像的格式相一致。否则，可能会发生错误或保存的图像可能有问题。

注意

16 位图像

如果需要无损保存 16 位图像，仅可使用 TIF 格式和 TwinCAT 序列化。

应用

例如，将图像保存到文件C:\TcVision\Image.bmp如下所示：

```
fbWriteImage(
    ipImage     := ipImageRes,
    sFilePath   := 'C:\TcVision\Image.bmp',
    bWrite      := TRUE,
    nTimeout    := T#500MS);

IF NOT fbWriteImage.bBusy AND NOT fbWriteImage.bError THEN
    // ipImageRes was successfully written to given file path.
END_IF
```

相关功能块：FB_VN_ReadImage [▶ 1529]，用于加载图像。

样本

- 保存 PLC 的图像 [▶ 2716]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.3.10 FB_VN_WriteMlModel

This FB writes a machine learning model to the hard drive.

Syntax

Definition:

```
FUNCTION_BLOCK FB_VN_WriteMlModel
VAR_INPUT
    ipMlModel : Reference To ITcVnMlModel;
    sFilePath : STRING;
    bWrite    : BOOL;
    nTimeout  : TIME;
END_VAR
VAR_OUTPUT
    bBusy     : BOOL;
```

```
    bError    : BOOL;
    nErrorId  : UDINT;
END_VAR
```

Inputs

Name	Type	Default	Description
ipMlModel	Reference To ITcVnMlModel [▶ 399]		The machine learning model to write
sFilePath	STRING		Full path of the file or relative path to the default directory on the target pc (e.g. only a file name). If the string is empty, a file name will be generated (containing the current time).
bWrite	BOOL		Writing the machine learning model is triggered by a rising edge at this input.
nTimeout	TIME	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	BOOL	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	BOOL	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	UDINT	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

更多信息

功能块 FB_VN_WriteMlModel 以异步方式将 ML 模型 [▶ 399]从文件保存到 PLC 中。

参数

文件访问参数 [▶ 1519]部分对某些参数进行集中描述。

ML 模型

ML 模型 ipMlModel 以二进制格式保存在指定文件路径下。

ITcVnColorModel 的保存代表一个特例，因为这仍然必须首先将下面一行转换为 ML 模型。另外，也可以用带有相应函数的 ITcVnMlModel 替代颜色模型。

```
hr := ipColorModel.TcQueryInterface(IID_ITcVnMlModel, ADR(ipWriteMlModel));
```

文件路径和格式

如果 sFilePath 为空，则会生成包含当前时间的文件名，并使用扩展名 vmlm。

应用

例如，将一个 ML 模型保存到文件 C:\TcVision\MlModel.vmlm 的过程如下所示：

```
fbWriteMlModel(
    ipMlModel     := ipWriteMlModel,
    sFilePath     := 'C:\TcVision\MlModel.vmlm',
    bWrite        := TRUE,
    nTimeout      := T#500MS);

IF NOT fbWriteMlModel.bBusy AND NOT fbWriteMlModel.bError THEN
    // Ml-Model was written successfully to file
END_IF
```

相关功能块：FB_VN_ReadMlModel [▶ 1530]。

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.4 Image Acquisition

该组包含用于通过 Vision 设备采集图像的功能块（GigE Vision 相机 [▶ 63]和文件源 [▶ 110]对象）。

功能块

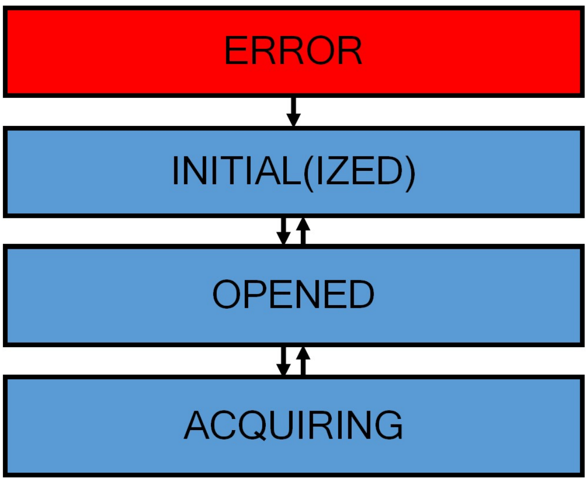
- FB_VN_SimpleCameraControl [▶ 1575]用于相机和文件源
- FB_VN_GevCameraControl [▶ 1550]用于 GigE Vision 相机
- FB_VN_FileSourceControl [▶ 1541]用于文件源

综述

该组中的功能块代表图像提供者 TcCOM 对象的代理。因此，它们不需要周期性地调用。相反，TcCOM 对象通过功能块的方法进行控制。

状态机

所有图像采集功能块均包括一个状态机，用于管理所代表设备的状态。基本上，功能块必须置于 ACQUIRING（采集中）状态，才能连续接收图像。INITIAL（初始）状态代表重启或复位后的起始点，可触发 INITIALIZING（初始化）。如为 OPENED（已打开）状态，与设备的通信通道会被打开，以读取或写入寄存器值。



所有状态都由枚举 ETcVnCameraState [▶ 147] 表示，并在下文中加以说明：

主要状态

主要的状态是静止状态。它们基本上表明是否有与 Vision 设备的连接（OPENED），该设备是否正在捕捉图像（ACQUIRING）以及与该设备的连接是否有错误（ERROR）。你可以通过 FB 的方法在这些状态之间切换。一般来说，两个主要状态之间的转换不会立即发生，而是至少经过一个周期。在这个中间时间，视觉设备处于中间状态 [▶ 1541]。

状态	描述
ERROR	错误状态，在这个状态下，每个设备只能通过调用方法Reset重新激活。
INITIAL	每个设备的初始状态。
INITIALIZED	与 INITIAL 类似，手动初始化相机（例如使用 Force-IP 和初始化命令）
OPENED	存在与相机的连接，可以触发单个图像，且可以读写相机寄存器。 **这个状态可以通过 StartAcquisition 方法跳过。**
ACQUIRING	设备处于采集状态，且正在发送图像，具体取决于设置。

中间状态

中间状态十分必要，因为大多数行动不能立即完成（如图像采集）。如果功能块处于这些中间状态之一，必须再次调用相应的方法，以完成向相应的主状态的过渡。

状态	描述
INITIALIZING	INITIAL > INITIALIZED
OPENING	INITIAL > OPENED 且 INITIALIZED > OPENED
STARTACQUISITION	OPENED > ACQUIRING.
STOPACQUISITION	ACQUIRING > OPENED
RESETTINGFEATURES	OPENED > OPENED
TRIGGERING	OPENED > OPENED 且 ACQUIRING > ACQUIRING
CLOSING	OPENED > INITIALIZED

必须在中间状态下调用方法！

在中间状态中也必须调用各自的状态转换方法。在相应的方法调用后使用GetState方法检测到静止状态时，状态转换才算完成，且不再需要调用该方法。触发一次方法，而随后的等待并不会导致所需的状态转换。

这样，PLC 中的状态机就可以用 CASE … OF 结构或 IF … ELSIF … 结构来实现。

样本

- 通过文件名触发图像 [▶ 2718]

6.1.5.4.1 FB_VN_FileSourceControl

This FB provides access to the images send by a File Source instance.

Do not call the main FB directly. Only use the available methods.

Methods

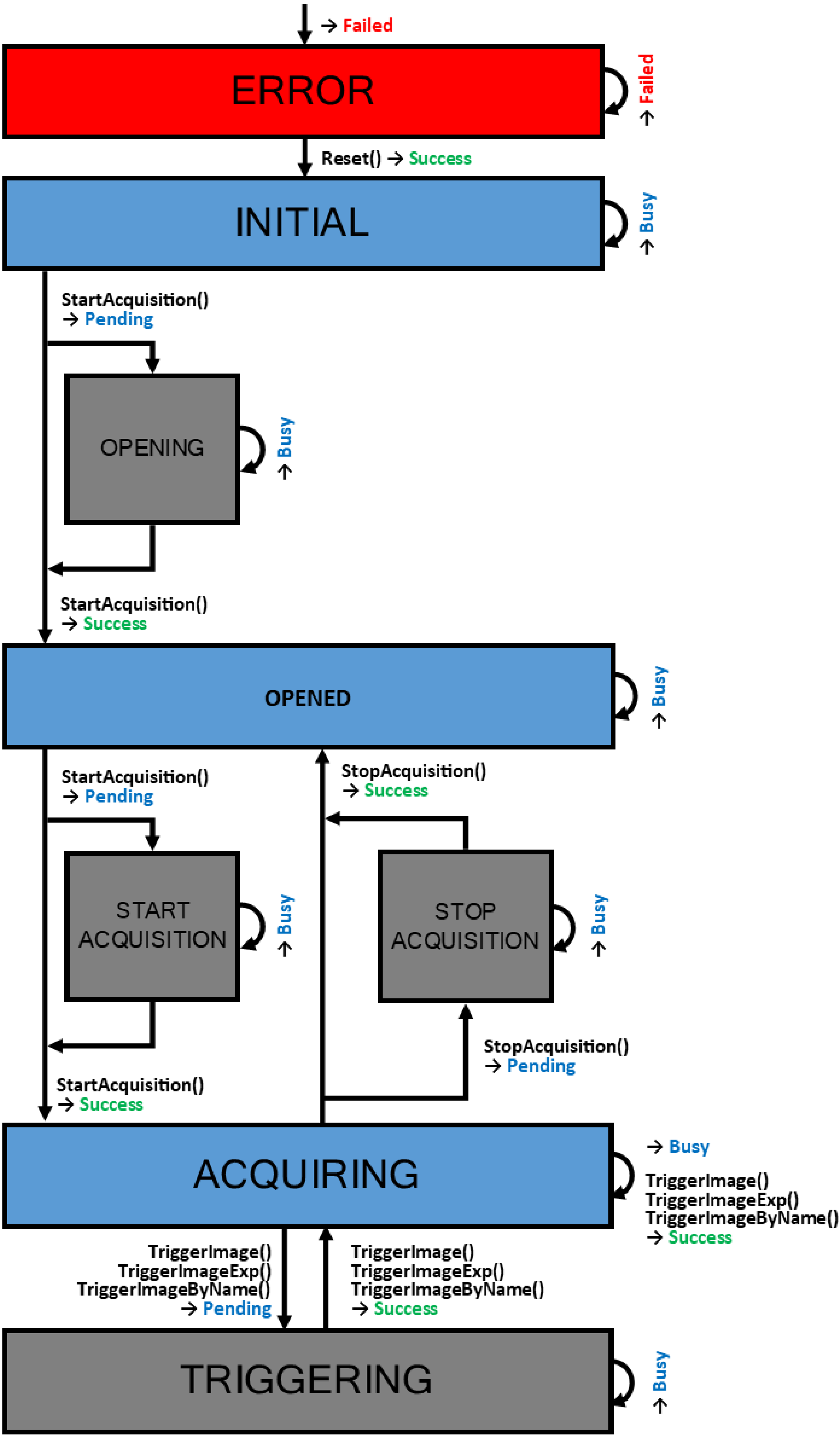
Name	Description
GetCurrentImage [▶ 1545]	Gets the current available image (if any).
GetCurrentImageAndFileName [▶ 1546]	Gets the current available image (if any) and the corresponding file name.
StartAcquisition [▶ 1546]	Activates the File Source. If not in trigger mode, it will start sending images.
StopAcquisition [▶ 1547]	Deactivates the File Source. It will stop sending images.
Reset [▶ 1547]	Reset the controller to initial state (might require multiple calls depending on current state, until S_OK is returned)
GetState [▶ 1548]	Gets the current state of the internal camera control state machine.
TriggerImage [▶ 1549]	If in trigger mode, this function triggers the next image in the list.
TriggerImageExp [▶ 1549]	If in trigger mode, this function triggers a specific image in the list.
TriggerImageByName [▶ 1550]	If in trigger mode, this function triggers a specific image in the list by name.

更多信息

功能块FB_VN_FileSourceControl可以使用 PLC 中的全部函数控制文件源 [▶ 110]实例。另外，也可以通过功能块FB_VN_SimpleCameraControl [▶ 1575]实现有限的功能。

状态机

下图显示了状态机。主要状态以蓝色标记，过渡状态为灰色，而错误状态为红色。



以下方法可用于控制状态机：

- GetState [▶ 1548]

- StartAcquisition [▶ 1546]
- StopAcquisition [▶ 1547]
- TriggerImage [▶ 1549]
- TriggerImageByName [▶ 1550]
- Reset [▶ 1547]

如果在非预期状态下调用方法，将返回错误代码INVALIDSTATE（16#712）。

带有连续图像采集的应用

例如，连续检索收到的图像的状态机如下所示：

```
VAR
    fbFileSource    :   FB_VN_FileSourceControl;
    eState          :   ETcVnCameraState;
    ipImageIn       :   ITcVnImage;
    ipImageInDisp   :   ITcVnDisplayableImage;
    hr              :   HRESULT;
    nNewImageCounter:   UINT;
END_VAR

eState := fbFileSource.GetState();

CASE eState OF

TCVN_CS_INITIAL, TCVN_CS_OPENING, TCVN_CS_OPENED, TCVN_CS_STARTACQUISITION:
    hr := fbFileSource.StartAcquisition();

TCVN_CS_ACQUIRING:
    hr := fbFileSource.GetCurrentImage(ipImageIn);

    // Check if new Image was received
    IF SUCCEEDED(hr) AND ipImageIn <> 0 THEN
        nNewImageCounter := nNewImageCounter + 1;

        // Place to call vision algorithms
        hr := F_VN_TransformIntoDisplayableImage(ipImageIn, ipImageInDisp, hr);
    END_IF

TCVN_CS_ERROR:
    hr := fbFileSource.Reset();

END_CASE
```

带有触发器的应用

例如，持续触发具有特殊文件名的图像的状态机如下所示：

```
VAR
    fbFileSource    :   FB_VN_FileSourceControl;
    eState          :   ETcVnCameraState;
    ipImageIn       :   ITcVnImage;
    ipImageInDisp   :   ITcVnDisplayableImage;
    bTrigger        :   BOOL := TRUE;
    sFileName       :   STRING := 'FileName.bmp';
    hr              :   HRESULT;
    nNewImageCounter:   UINT;
END_VAR

eState := fbFileSource.GetState();

CASE eState OF

TCVN_CS_INITIAL, TCVN_CS_OPENING, TCVN_CS_OPENED, TCVN_CS_STARTACQUISITION:
    hr := fbFileSource.StartAcquisition();

TCVN_CS_TRIGGERING:
    hr := fbFileSource.TriggerImage();

TCVN_CS_ACQUIRING:

    IF bTrigger THEN
        hr := fbFileSource.TriggerImageByName(sFileName);
        IF SUCCEEDED(hr) THEN
            bTrigger := FALSE;
```

```
        END_IF
    ELSE
        hr := fbFileSource.GetCurrentImage(ipImageIn);

        // Check if new Image was received
        IF SUCCEEDED(hr) AND ipImageIn <> 0 THEN
            nNewImageCounter := nNewImageCounter + 1;

            // Trigger next image
            bTrigger := TRUE;

            // Place to call vision algorithms
            hr := F_VN_TransformIntoDisplayableImage(ipImageIn, ipImageInDisp, hr);
        END_IF
    END_IF

TCVN_CS_ERROR:
    hr := fbFileSource.Reset();

END_CASE
```

作为TriggerImageByName(sFileName)方法的一个替代方法，TriggerImage()可以用于始终从文件源控制列表中获得下一个图像。

样本

- 通过文件名触发图像 [▶ 2718]

相关功能块

- FB_VN_SimpleCameraControl [▶ 1575]用于相机和文件源
- FB_VN_GevCameraControl [▶ 1550]用于 GigE Vision 相机
- FB_VN_FileSourceControl [▶ 1541]用于文件源

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.4.1.1 GetCurrentImage

GetCurrentImage	
ipImage *Reference To ITcVnImage*	*HRESULT* GetCurrentImage

Gets the current available image (if any).

Syntax

Definition:

```
METHOD GetCurrentImage : HRESULT
VAR_INPUT
    ipImage : Reference To ITcVnImage;
END_VAR
```

Inputs

Name	Type	Description
ipImage	Reference To ITcVnImage [▶ 390]	Returns the image

Return value

HRESULT [▶ 122]

6.1.5.4.1.2 GetCurrentImageAndFileName

Gets the current available image (if any) and the corresponding file name.

Syntax

Definition:

```
METHOD GetCurrentImageAndFileName : HRESULT
VAR_INPUT
    ipImage   : Reference To ITcVnImage;
    sFileName : STRING;
    nMaxLen   : UINT;
END_VAR
```

Inputs

Name	Type	Description
ipImage	Reference To ITcVnImage [▶ 390]	Returns the image
sFileName	STRING	Returns the file name
nMaxLen	UINT	Define the maximum allowed length for the file name

Return value

HRESULT [▶ 122]

6.1.5.4.1.3 StartAcquisition

Activates the File Source. If not in trigger mode, it will start sending images.

Syntax

Definition:

```
METHOD StartAcquisition : HRESULT
```

Return value

HRESULT [▶ 122]

更多信息

方法StartAcquisition开始加载图像。根据触发模式设置，这可能意味着直接提供图像或等待触发信号。图像加载可以从INITIAL和OPENED状态开始。

状态转换

通过StartAcquisition，文件源将置于ACQUIRING状态。这可在INITIAL和OPENED状态下进行。如果文件源处于INITIAL，则状态转换首先到OPENED。通常情况下，这通过中间状态OPENING进行。从OPENED到ACQUIRING的过渡通常通过中间状态STARTACQUISITION进行。

所有可能的状态转换和它们的返回代码都显示在以下的转换矩阵中：

<table>
<tr><th colspan="2" rowspan="2">返回代码（HRESULT）</th><th colspan="6">结果状态</th></tr>
<tr><th>INITIAL</th><th>OPENING</th><th>OPENED</th><th>STARTACQUISITION</th><th>ACQUIRING</th><th>ERROR</th></tr>
<tr><td rowspan="4">初始状态</td><td>INITIAL</td><td>BUSY</td><td>PENDING</td><td rowspan="2">SUCCESS</td><td rowspan="2">-</td><td rowspan="2">-</td><td rowspan="4">FAILED</td></tr>
<tr><td>OPENING</td><td rowspan="3">-</td><td>BUSY</td></tr>
<tr><td>OPENED</td><td rowspan="2">-</td><td>BUSY</td><td>PENDING</td><td rowspan="2">SUCCESS</td></tr>
<tr><td>STARTACQUISITION</td><td>-</td><td>BUSY</td></tr>
</table>

6.1.5.4.1.4 StopAcquisition

Deactivates the File Source. It will stop sending images.

Syntax

Definition:

```
METHOD StopAcquisition : HRESULT
```

Return value

HRESULT [▶ 122]

更多信息

方法StopAcquisition停止文件源的图像加载。

状态转换

通过StopAcquisition，会将文件源从ACQUIRING状态带入OPENED状态。通常情况下，这不会立即发生，而是通过中间状态STOPACQUISITION进行。

所有可能的状态转换和它们的返回代码都显示在以下的转换矩阵中：

<table>
<tr><th colspan="2" rowspan="2">返回代码（HRESULT）</th><th colspan="4">结果状态</th></tr>
<tr><th>ACQUIRING</th><th>STOPACQUISITION</th><th>OPENED</th><th>ERROR</th></tr>
<tr><td rowspan="2">初始状态</td><td>ACQUIRING</td><td>BUSY</td><td>PENDING</td><td rowspan="2">SUCCESS</td><td rowspan="2">FAILED</td></tr>
<tr><td>STOPACQUISITION</td><td>-</td><td>BUSY</td></tr>
</table>

6.1.5.4.1.5 Reset

Reset the controller to initial state (might require multiple calls depending on current state, until S_OK is returned)

Syntax

Definition:

```
METHOD Reset : HRESULT
```

Return value

HRESULT [▶ 122]

更多信息

如果文件源处于错误状态，方法Reset将重新设置文件源。

状态转换

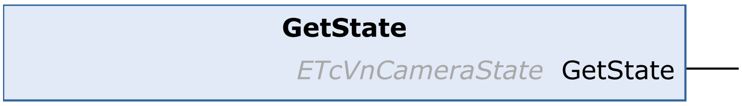
通过Reset，会将文件源从ERROR状态带入INITIAL状态。如果复位成功，这个转换立即发生，否则功能块仍处于ERROR状态。

所有可能的状态转换和它们的返回代码都显示在以下的转换矩阵中：

返回代码（HRESULT）		结果状态	
		ERROR	INITIAL
初始状态	ERROR	FAILED	SUCCESS

在成功的情况下，S_OK 或 S_FALSE（如果没有错误需要重置）将作为 SUCCESS 代码返回。

6.1.5.4.1.6 GetState



Gets the current state of the internal camera control state machine.

Syntax

Definition:

```
METHOD GetState : ETcVnCameraState
```

Return value

ETcVnCameraState [▶ 147]

更多信息

方法GetState将相机对象或文件源对象的状态作为枚举ETcVnCameraState [▶ 147]的值返回。可能出现以下状态：

- ERROR
- INITIAL
- OPENING
- OPENED
- STARTACQUISITION
- STOPACQUISITION
- ACQUIRING
- TRIGGERING

6.1.5.4.1.7 TriggerImage

TriggerImage	
HRESULT	TriggerImage

If in trigger mode, this function triggers the next image in the list.

Syntax

Definition:

```
METHOD TriggerImage : HRESULT
```

Return value

HRESULT [▶ 122]

更多信息

方法 TriggerImage 从文件源触发一个图像。

状态转换

通过 TriggerImage，文件源将仅在保持当前 ACQUIRING 状态的情况下加载一个图像。然而，由于图像采集通常需要几个周期，所以文件源通常会短暂地进入中间状态TRIGGERING。

所有可能的状态转换和它们的返回代码都显示在以下的转换矩阵中：

返回代码（HRESULT）		结果状态		
		TRIGGERING	ACQUIRING	ERROR
初始状态	ACQUIRING	PENDING	BUSY / SUCCESS	FAILED
	TRIGGERING	BUSY	SUCCESS	

6.1.5.4.1.8 TriggerImageExp

TriggerImageExp			
nSkipImages	DINT	HRESULT	TriggerImageExp

If in trigger mode, this function triggers a specific image in the list.

Syntax

Definition:

```
METHOD TriggerImageExp : HRESULT
VAR_INPUT
    nSkipImages : DINT;
END_VAR
```

Inputs

Name	Type	Description
nSkipImages	DINT	Amount of images to skip, relative to current

Return value

HRESULT [▶ 122]

更多信息

方法TriggerImageExp的状态转换与TriggerImage [▸ 1549]相同。

6.1.5.4.1.9 TriggerImageByName

If in trigger mode, this function triggers a specific image in the list by name.

Syntax

Definition:

```
METHOD TriggerImageByName : HRESULT
VAR_INPUT
    sImageName : STRING;
END_VAR
```

Inputs

Name	Type	Description
sImageName	STRING	Name of the image to trigger

Return value

HRESULT [▸ 122]

更多信息

方法TriggerImageByName的状态转换与TriggerImage [▸ 1549]相同。

6.1.5.4.2 FB_VN_GevCameraControl

This FB provides the basic functionality to control a GigE Vision camera and access its calibration data.

Do not call the main FB directly. Only use the available methods.

Methods

Name	Description
ClearImageQueue [▶ 1557]	Delete all images contained in the TcVnGevImageProvider TcCOM module receive queue and reset the corresponding omitted images counter.
CloseCamera [▶ 1557]	Close the control channel to the camera.
GetCalibPatternRef [▶ 1558]	Gets the reference calibration pattern points from the TcVnGevImageProvider TcCOM module (Can be set from the calibration assistant)
GetCameraMatrix [▶ 1559]	Gets the camera matrix from the TcVnGevImageProvider TcCOM module (requires intrinsic calibration parameters, e.g. set by the calibration assistant)
GetCurrentImage [▶ 1559]	Gets the current available image (first in receive queue).
GetCurrentImageUndistorted [▶ 1559]	Gets the current available image (first in receive queue) with undistortion applied. Incompatible with pixel formats that contain an encoding, e.g. a Bayer pattern. Requires intrinsic calibration parameters in the TcVnGevImageProvider TcCOM module, e.g. set by the calibration assistant.
GetCurrentImageWithGvspInfo [▶ 1560]	Gets the current available image (first in receive queue) and its corresponding GVSP info.
GetCurrentImageWithGvspInfoUndistorted [▶ 1561]	Gets the current available image (first in receive queue) with its corresponding GVSP info and undistortion applied. Incompatible with pixel formats that contain an encoding, e.g. a Bayer pattern. Requires intrinsic calibration parameters in the TcVnGevImageProvider TcCOM module, e.g. set by the calibration assistant.
GetCurrentImageWithTimestamps [▶ 1561]	Gets the current available image (first in receive queue) and corresponding timestamps.
GetDistortionCoefficients [▶ 1562]	Gets the distortion coefficients from the TcVnGevImageProvider TcCOM module (requires intrinsic calibration parameters, e.g. set by the calibration assistant)
GetLastImageFromQueue [▶ 1562]	Gets the last received image from the queue.
GetOmittedImagesNum [▶ 1563]	Gets the number of omitted images since the last call of ClearImageQueue(). If the image receive queue in the TcVnGevImageProvider TcCOM module is full and a new image arrives, the first one in the queue will be deleted and the omitted counter is increased.
GetRotationMatrix [▶ 1563]	Gets the rotation matrix from the TcVnGevImageProvider TcCOM module (requires extrinsic calibration parameters, e.g. set by the calibration assistant)
GetState [▶ 1564]	Gets the current state of the internal camera control state machine.
GetTranslationVector [▶ 1564]	Gets the translation vector from the TcVnGevImageProvider TcCOM module (requires extrinsic calibration parameters, e.g. set by the calibration assistant)
InitializeCamera [▶ 1565]	Initialize the camera to the intended state (includes sending the 'InitCameraCommands' defined in the GevImageAcquisition TcCOM module to the camera).
OpenCamera [▶ 1566]	Open a control channel to the camera.
Reset [▶ 1567]	Reset the camera controller to initial state (might require multiple calls depending on current state, until S_OK is returned)
ResetCameraFeatures [▶ 1568]	Reset the camera features to initial state (sends the 'InitCameraCommands' defined in the GevImageAcquisition TcCOM module to the camera).
SetCameraMatrix [▶ 1569]	Sets the camera matrix to the TcVnGevImageProvider TcCOM module

Name	Description
SetDistortionCoefficients [▶ 1569]	Sets the distortion coefficients to the TcVnGevImageProvider TcCOM module
SetRotationMatrix [▶ 1570]	Sets the rotation matrix to the TcVnGevImageProvider TcCOM module
SetTranslationVector [▶ 1570]	Sets the translation vector to the TcVnGevImageProvider TcCOM module
StartAcquisition [▶ 1570]	Send the 'StartAcquisitionCommands' defined in the TcVnGevImageProvider TcCOM module to the camera.
StopAcquisition [▶ 1573]	Send the 'StopAcquisitionCommands' defined in the TcVnGevImageProvider TcCOM module to the camera.
TriggerImage [▶ 1574]	Send the 'SoftwareTriggerCommands' defined in the TcVnGevImageProvider TcCOM module to the camera.

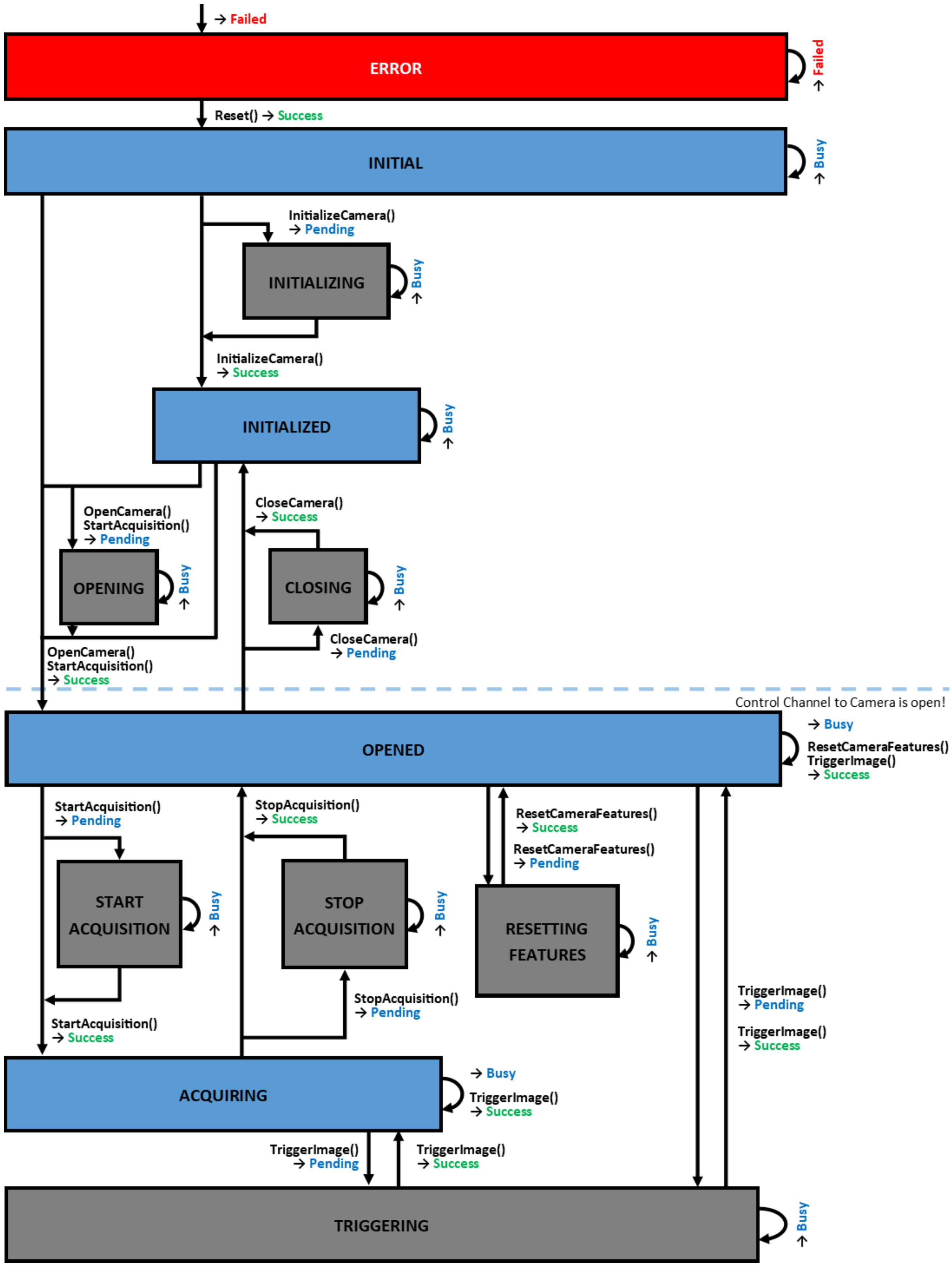
更多信息

功能块 FB_VN_GevCameraControl 在 PLC 中充分代表 GigE Vision 相机 [▶ 63]对象。另外，也可以使用功能块 FB_VN_SimpleCameraControl [▶ 1575] 限制访问相机对象。

此功能块不可直接调用。事实上，这些方法包含所有功能。

状态机

下图显示了状态机。主要状态用蓝色标记，过渡状态用灰色标记，错误状态用红色标记。GetState [▶ 1564] 方法可用于查询相机的状态。此外，还显示实现状态转换的方法以及每种方法的返回值。如果存在 Init 命令，并且选择了初始化自动模式 AUTOINIT_AFTER_SO，则还会简单地假设自动初始化发生的 INITIALIZING 状态。



您可以使用 GetState 方法查询当前状态。

以下方法可用于控制状态机：

- GetState [▸ 1564]
- InitializeCamera [▸ 1565]

- OpenCamera [▶ 1566]
- CloseCamera [▶ 1557]
- StartAcquisition [▶ 1570]
- StopAcquisition [▶ 1573]
- TriggerImage [▶ 1574]
- ResetCameraFeatures [▶ 1568]
- Reset [▶ 1567]

如果方法在非预期状态下被调用，则会返回错误代码 INVALIDSTATE（16#712）。

HRESULT

Hex	Dec	代码	描述
16#718	1816	NOTINIT	相机尚未初始化。使用 InitializeCamera [▶ 1565] 方法或将 TcCOM 参数 InitializationAutoMode [▶ 87] 设置为 AUTOINIT_SO 或 AUTOINIT_AFTER_SO，以对相机进行初始化。
16#71A	1818	NOINTERFACE	相机功能块未链接到相机 TcCOM 对象。在*符号初始化*下的 TwinCAT 项目实例中建立此链接。

连续图像采集的应用

例如，将相机置于图像采集模式、处理错误状态并连续显示成功采集图像的状态机如下所示：

```
VAR
    hr                  :   HRESULT;
    ipImageIn           :   ITcVnImage;
    ipImageInDisp       :   ITcVnDisplayableImage;
    fbCameraControl :   FB_VN_GevCameraControl;
    eCameraState        :   ETcVnCameraState;
    nNewImageCounter:   UINT;
END_VAR

eCameraState :=  fbCameraControl.GetState();

// CameraControl is in error state, so try to reset the camera connection
IF eCameraState = TCVN_CS_ERROR THEN
    hr :=  fbCameraControl.Reset();

// Camera not yet initialized
ELSIF eCameraState < TCVN_CS_INITIALIZED THEN
    hr :=  fbCameraControl.InitializeCamera();

// Camera not yet opened
ELSIF eCameraState < TCVN_CS_OPENED THEN
    hr := fbCameraControl.OpenCamera();

// Camera not yet streaming
ELSIF eCameraState < TCVN_CS_ACQUIRING THEN
    hr := fbCameraControl.StartAcquisition();

// Camera streaming
ELSIF eCameraState = TCVN_CS_ACQUIRING THEN
    hr := fbCameraControl.GetCurrentImage(ipImageIn);

    // Check if new Image was received
    IF SUCCEEDED(hr) AND ipImageIn <> 0 THEN
        nNewImageCounter := nNewImageCounter + 1;

        // Place to call vision algorithms
        hr := F_VN_TransformIntoDisplayableImage(ipImageIn, ipImageInDisp, hr);
    END_IF

END_IF
```

在这种情况下，可将相机配置为连续图像采集 [▶ 2730]或硬件触发器 [▶ 2731]。

软件触发器的应用

例如，将相机调入图像采集模式、处理错误状态以及通过软件触发器触发图像采集的状态机如下所示：

```
VAR
    hr               :   HRESULT;
    ipImageIn        :   ITcVnImage;
    ipImageInDisp    :   ITcVnDisplayableImage;
    fbCameraControl  :   FB_VN_GevCameraControl;
    eCameraState     :   ETcVnCameraState;
    nNewImageCounter:    UINT;
    bTrigger         :   BOOL := TRUE;
END_VAR

eCameraState :=  fbCameraControl.GetState();

// CameraControl is in error state, so try to reset the camera connection
IF eCameraState = TCVN_CS_ERROR THEN
    hr :=  fbCameraControl.Reset();

// Camera trigger image
ELSIF eCameraState = TCVN_CS_TRIGGERING THEN
    hr := fbCameraControl.TriggerImage();

// Camera not yet initialized
ELSIF eCameraState < TCVN_CS_INITIALIZED THEN
    hr :=  fbCameraControl.InitializeCamera();

// Camera not yet opened
ELSIF eCameraState < TCVN_CS_OPENED THEN
    hr := fbCameraControl.OpenCamera();

// Camera not yet streaming
ELSIF eCameraState < TCVN_CS_ACQUIRING THEN
    hr := fbCameraControl.StartAcquisition();

// Camera streaming
ELSIF eCameraState = TCVN_CS_ACQUIRING THEN

    IF bTrigger THEN
        hr := fbCameraControl.TriggerImage();
        IF SUCCEEDED(hr) THEN
            bTrigger := FALSE;
        END_IF
    ELSE
        hr := fbCameraControl.GetCurrentImage(ipImageIn);

        // Check if new Image was received
        IF SUCCEEDED(hr) AND ipImageIn <> 0 THEN
            nNewImageCounter := nNewImageCounter + 1;

            // Trigger next image
            bTrigger := TRUE;

            // Place to call vision algorithms
            hr := F_VN_TransformIntoDisplayableImage(ipImageIn, ipImageInDisp, hr);
        END_IF
    END_IF

END_IF
```

在这种情况下，必须将相机配置为软件触发器 [▸ 2731]（触发源 = 软件）。

相关功能块

- FB_VN_SimpleCameraControl [▸ 1575]用于相机和文件源
- FB_VN_GevCameraControl [▸ 1550]用于 GigE Vision 相机
- FB_VN_FileSourceControl [▸ 1541]用于文件源

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.4.2.1 ClearImageQueue

ClearImageQueue
HRESULT ClearImageQueue

Delete all images contained in the TcVnGevImageProvider TcCOM module receive queue and reset the corresponding omitted images counter.

Syntax

Definition:

```
METHOD ClearImageQueue : HRESULT
```

Return value

HRESULT [▶ 122]

6.1.5.4.2.2 CloseCamera

CloseCamera
HRESULT CloseCamera

Close the control channel to the camera.

Syntax

Definition:

```
METHOD CloseCamera : HRESULT
```

Return value

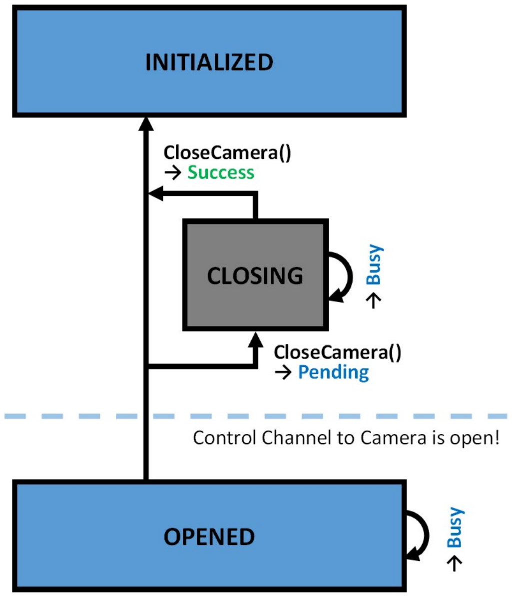
HRESULT [▶ 122]

更多信息

方法CloseCamera将关闭相机打开的通信通道。

状态转换

通过CloseCamera，会让相机从OPENED状态进入INITIALIZED状态。通常，这种转换通过中间状态CLOSING进行。

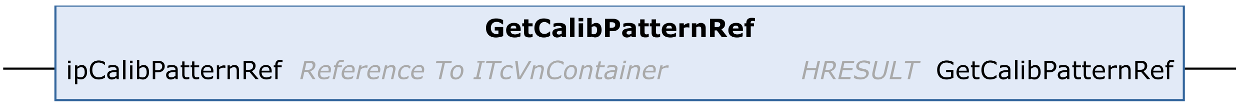


附图 19: 使用 FB_VN_GevCameraControl.CloseCamera 的状态转换

所有可能的状态转换和它们的返回代码都显示在以下的转换矩阵中:

<table>
<tr><th colspan="2" rowspan="2">返回代码 (HRESULT)</th><th colspan="4">结果状态</th></tr>
<tr><th>OPENED</th><th>CLOSING</th><th>INITIALIZED</th><th>ERROR</th></tr>
<tr><td rowspan="2">初始状态</td><td>OPENED</td><td>BUSY</td><td>PENDING</td><td rowspan="2">SUCCESS</td><td rowspan="2">FAILED</td></tr>
<tr><td>CLOSING</td><td>-</td><td>BUSY</td></tr>
</table>

6.1.5.4.2.3 GetCalibPatternRef



Gets the reference calibration pattern points from the TcVnGevImageProvider TcCOM module (Can be set from the calibration assistant)

Syntax

Definition:

```
METHOD GetCalibPatternRef : HRESULT
VAR_INPUT
    ipCalibPatternRef : Reference To ITcVnContainer;
END_VAR
```

Inputs

Name	Type	Description
ipCalibPatternRef	Reference To ITcVnContainer [▶ 349]	Returns the reference calibration pattern points (ContainerType_Vector_TcVnPoint3_REAL)

Return value

HRESULT [▶ 122]

6.1.5.4.2.4 GetCameraMatrix

Gets the camera matrix from the TcVnGevImageProvider TcCOM module (requires intrinsic calibration parameters, e.g. set by the calibration assistant)

Syntax

Definition:

```
METHOD GetCameraMatrix : HRESULT
VAR_IN_OUT
    aCameraMatrix : TcVnMatrix3x3_LREAL;
END_VAR
```

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▸ 141]	Returns the camera matrix

Return value

HRESULT [▸ 122]

6.1.5.4.2.5 GetCurrentImage

Gets the current available image (first in receive queue).

Syntax

Definition:

```
METHOD GetCurrentImage : HRESULT
VAR_INPUT
    ipImage : Reference To ITcVnImage;
END_VAR
```

Inputs

Name	Type	Description
ipImage	Reference To ITcVnImage [▸ 390]	Returns the image

Return value

HRESULT [▸ 122]

6.1.5.4.2.6 GetCurrentImageUndistorted

Gets the current available image (first in receive queue) with undistortion applied. Incompatible with pixel formats that contain an encoding, e.g. a Bayer pattern. Requires intrinsic calibration parameters in the TcVnGevImageProvider TcCOM module, e.g. set by the calibration assistant.

Syntax

Definition:

```
METHOD GetCurrentImageUndistorted : HRESULT
VAR_INPUT
    ipImage : Reference To ITcVnImage;
END_VAR
```

Inputs

Name	Type	Description
ipImage	Reference To ITcVnImage [▸ 390]	Returns the undistorted image

Return value

HRESULT [▸ 122]

6.1.5.4.2.7 GetCurrentImageWithGvspInfo

Gets the current available image (first in receive queue) and its corresponding GVSP info.

Syntax

Definition:

```
METHOD GetCurrentImageWithGvspInfo : HRESULT
VAR_INPUT
    ipImage    : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    stGvspInfo : GVSP_IMAGE_INFO;
END_VAR
```

Inputs

Name	Type	Description
ipImage	Reference To ITcVnImage [▸ 390]	Returns the image

In/Outputs

Name	Type	Description
stGvspInfo	GVSP_IMAGE_INFO [▸ 204]	Returns the GVSP info

Return value

HRESULT [▸ 122]

6.1.5.4.2.8 GetCurrentImageWithGvspInfoUndistorted

Gets the current available image (first in receive queue) with its corresponding GVSP info and undistortion applied. Incompatible with pixel formats that contain an encoding, e.g. a Bayer pattern. Requires intrinsic calibration parameters in the TcVnGevImageProvider TcCOM module, e.g. set by the calibration assistant.

Syntax

Definition:

```
METHOD GetCurrentImageWithGvspInfoUndistorted : HRESULT
VAR_INPUT
    ipImage    : Reference To ITcVnImage;
END_VAR
VAR_IN_OUT
    stGvspInfo : GVSP_IMAGE_INFO;
END_VAR
```

Inputs

Name	Type	Description
ipImage	Reference To ITcVnImage [▶ 390]	Returns the undistorted image

In/Outputs

Name	Type	Description
stGvspInfo	GVSP_IMAGE_INFO [▶ 204]	Returns the GVSP info

Return value

HRESULT [▶ 122]

6.1.5.4.2.9 GetCurrentImageWithTimestamps

Gets the current available image (first in receive queue) and corresponding timestamps.

Syntax

Definition:

```
METHOD GetCurrentImageWithTimestamps : HRESULT
VAR_INPUT
    ipImage     : Reference To ITcVnImage;
    nTimestamps : ULINT;
    pTimestamps : Pointer To LINT;
END_VAR
VAR_OUTPUT
    stGvspInfo  : GVSP_IMAGE_INFO;
END_VAR
```

Inputs

Name	Type	Description
ipImage	Reference To ITcVnImage [▶ 390]	Returns the image
nTimestamps	ULINT	Select timestamps to return (ETcVnTimestamp)
pTimestamps	Pointer To LINT	Returns the requested timestamps (make sure to provide an array of sufficient size).

Outputs

Name	Type	Description
stGvspInfo	GVSP_IMAGE_INFO [▶ 204]	Returns the GVSP info

Return value

HRESULT [▶ 122]

6.1.5.4.2.10 GetDistortionCoefficients

Gets the distortion coefficients from the TcVnGevImageProvider TcCOM module (requires intrinsic calibration parameters, e.g. set by the calibration assistant)

Syntax

Definition:

```
METHOD GetDistortionCoefficients : HRESULT
VAR_IN_OUT
    aDistortionCoefficients : TcVnArray8_LREAL;
END_VAR
```

In/Outputs

Name	Type	Description
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	Returns the distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]

Return value

HRESULT [▶ 122]

6.1.5.4.2.11 GetLastImageFromQueue

Gets the last received image from the queue.

Syntax

Definition:

```
METHOD GetLastImageFromQueue : HRESULT
VAR_INPUT
    ipImage : Reference To ITcVnImage;
END_VAR
```

Inputs

Name	Type	Description
ipImage	Reference To ITcVnImage [▶ 390]	Returns the image

Return value

HRESULT [▶ 122]

6.1.5.4.2.12 GetOmittedImagesNum

GetOmittedImagesNum
nOmitted ULINT HRESULT GetOmittedImagesNum

Gets the number of omitted images since the last call of ClearImageQueue(). If the image receive queue in the TcVnGevImageProvider TcCOM module is full and a new image arrives, the first one in the queue will be deleted and the omitted counter is increased.

Syntax

Definition:

```
METHOD GetOmittedImagesNum : HRESULT
VAR_IN_OUT
    nOmitted : ULINT;
END_VAR
```

In/Outputs

Name	Type	Description
nOmitted	ULINT	Returns the number of omitted images

Return value

HRESULT [▶ 122]

6.1.5.4.2.13 GetRotationMatrix

GetRotationMatrix
aRotationMatrix TcVnMatrix3x3_LREAL HRESULT GetRotationMatrix

Gets the rotation matrix from the TcVnGevImageProvider TcCOM module (requires extrinsic calibration parameters, e.g. set by the calibration assistant)

Syntax

Definition:

```
METHOD GetRotationMatrix : HRESULT
VAR_IN_OUT
    aRotationMatrix : TcVnMatrix3x3_LREAL;
END_VAR
```

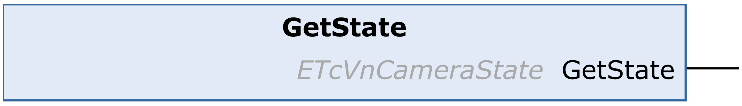
In/Outputs

Name	Type	Description
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	Returns the rotation matrix

Return value

HRESULT [▶ 122]

6.1.5.4.2.14 GetState



Gets the current state of the internal camera control state machine.

Syntax

Definition:

```
METHOD GetState : ETcVnCameraState
```

Return value

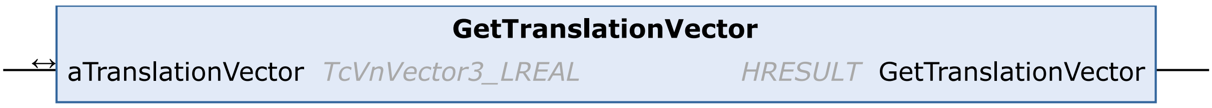
ETcVnCameraState [▶ 147]

更多信息

方法GetState将相机对象的状态作为枚举ETcVnCameraState [▶ 147]的值返回。可能出现以下状态：

- ERROR
- INITIAL
- INITIALIZING
- INITIALIZED
- OPENING
- OPENED
- CLOSING
- STARTACQUISITION
- STOPACQUISITION
- ACQUIRING
- TRIGGERING
- RESETTINGFEATURES

6.1.5.4.2.15 GetTranslationVector



Gets the translation vector from the TcVnGevImageProvider TcCOM module (requires extrinsic calibration parameters, e.g. set by the calibration assistant)

Syntax

Definition:

```
METHOD GetTranslationVector : HRESULT
VAR_IN_OUT
    aTranslationVector : TcVnVector3_LREAL;
END_VAR
```

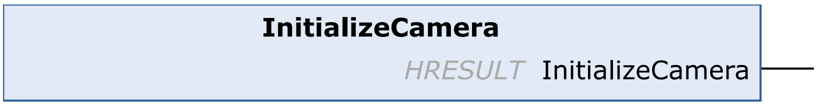
In/Outputs

Name	Type	Description
aTranslationVector	TcVnVector3_LREAL [▶ 141]	Returns the translation vector

Return value

HRESULT [▶ 122]

6.1.5.4.2.16 InitializeCamera



Initialize the camera to the intended state (includes sending the 'InitCameraCommands' defined in the GevImageAcquisition TcCOM module to the camera).

Syntax

Definition:

```
METHOD InitializeCamera : HRESULT
```

Return value

HRESULT [▶ 122]

更多信息

方法InitializeCamera初始化所连接的 GigE Vision 相机，正如存储的初始化命令中已经配置的参数。这个方法应用于明确控制相机的初始化时间。为此，图像采集对象 [▶ 87]的 TcCOM 参数中的InitializationAutoMode必须设置为NO_AUTOINIT。因此，未进行自动的相机初始化；相反，必须通过InitializeCamera手动完成。

如果使用InitializationAutoMode进行了不同的选择，将会发生自动初始化且不需要InitializeCamera方法。

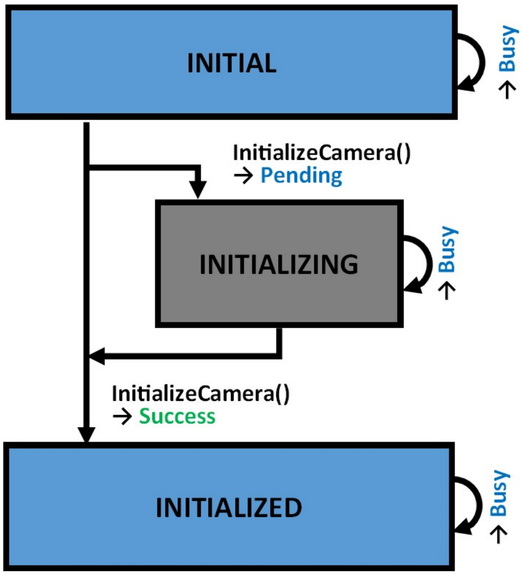
GigE Vision 相机的初始化包括以下步骤：

1. 打开相机的控制通道
2. 执行 Force-IP 命令（可选）
3. 设置 UserSet 或写入初始化命令 [▶ 77]。

在初始化过程中，该功能块处于 INITIALIZING 状态。一旦初始化完成，功能块就会进入 INITIALIZED 状态。如果通过调用CloseCamera [▶ 1557]方法关闭处于 OPENED 状态的相机连接，也会进入为这种状态。

状态转换

通过InitializeCamera，会让相机从INITIAL状态进入INITIALIZED状态。通常，这种转换通过中间状态INITIALIZING进行。

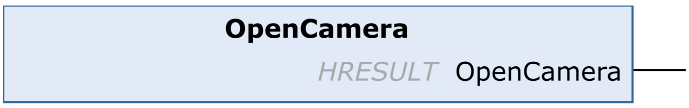


附图 20: 使用 FB_VN_GevCameraControl.InitializeCamera 的状态转换

所有可能的状态转换和它们的返回代码都显示在以下的转换矩阵中：

返回代码（HRESULT）		结果状态			
		INITIAL	INITIALIZING	INITIALIZED	ERROR
初始状态	INITIAL	BUSY	PENDING	SUCCESS	FAILED
	INITIALIZING	-	BUSY		

6.1.5.4.2.17 OpenCamera



Open a control channel to the camera.

Syntax

Definition:

```
METHOD OpenCamera : HRESULT
```

Return value

HRESULT [▸ 122]

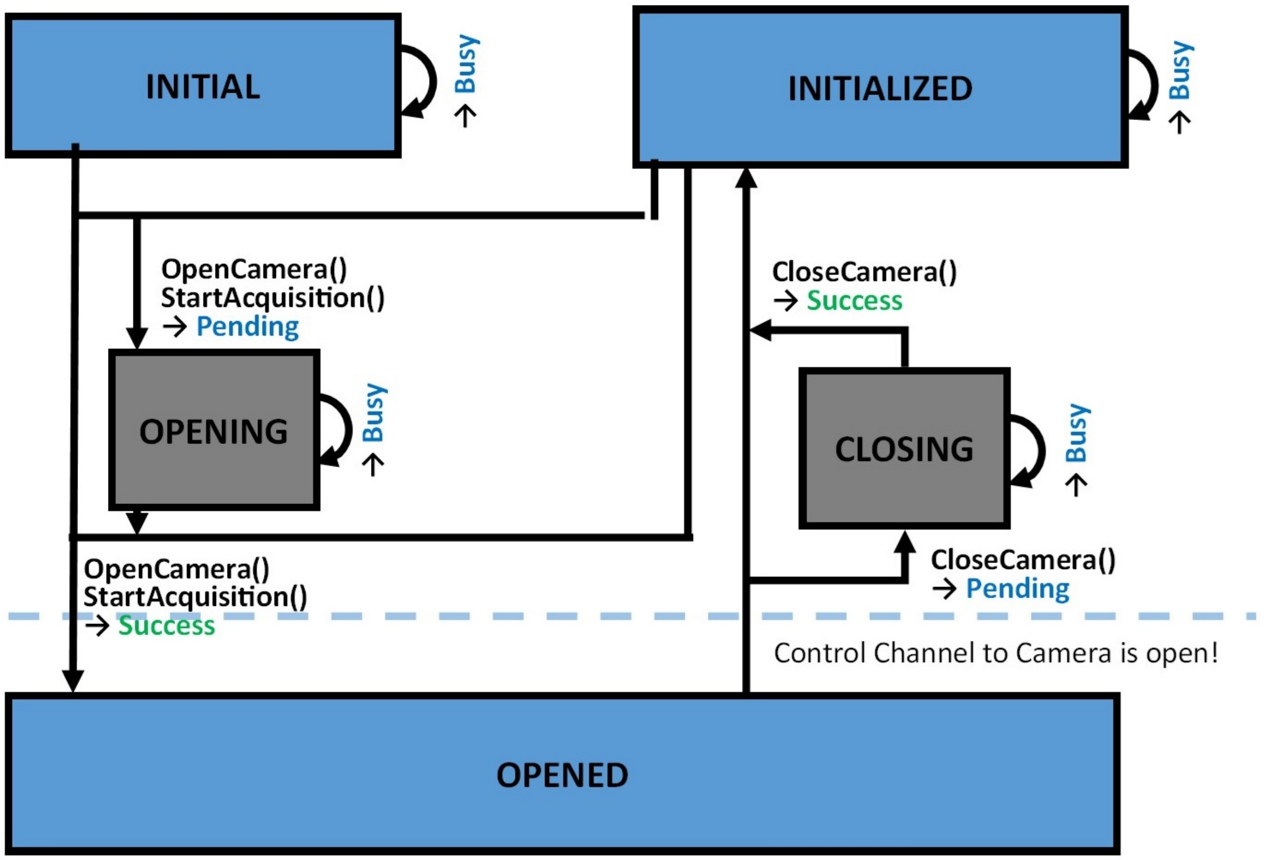
更多信息

方法OpenCamera打开了相机的通信通道。在相应的OPENED状态下，相机寄存器可以被读写，图像可以被采集，且连续的图像记录可以被启动。

如果在图像采集对象 [▸ 87]的 TcCOM 参数中InitializationAutoMode被设置为AUTOINIT_AFTER_SO，那么在第一次调用 OpenCamera 方法时 GigE Vision 相机就会被初始化。其中，这包括写入初始化命令 [▸ 77]或设置 UserSets。另外，也可以调用InitializeCamera [▸ 1565]方法，以便手动执行初始化。

状态转换

通过OpenCamera，相机将置于OPENED状态。这可在INITIAL和INITIALIZED状态下进行。通常，这种转换通过中间状态OPENING进行。



附图 21: 使用 FB_VN_GevCameraControl.OpenCamera 的状态转换

所有可能的状态转换和它们的返回代码都显示在以下的转换矩阵中：

返回代码（HRESULT）		结果状态				
		INITIAL	INITIALIZED	OPENING	OPENED	ERROR
初始状态	INITIAL	BUSY	-	PENDING	SUCCESS	FAILED
	INITIALIZED	-	BUSY			
	OPENING		-	BUSY		

6.1.5.4.2.18 Reset

Reset the camera controller to initial state (might require multiple calls depending on current state, until S_OK is returned)

Syntax

Definition:

```
METHOD Reset : HRESULT
```

Return value

HRESULT [▶ 122]

更多信息

如果相机处于错误状态，方法Reset将对其进行复位。

状态转换

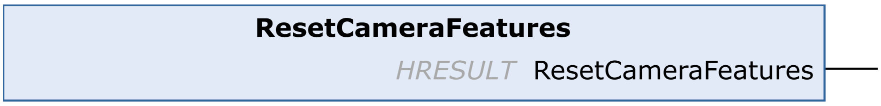
通过Reset，相机功能块应从ERROR状态进入INITIAL状态。如果复位成功，这个转换立即发生，否则功能块仍处于ERROR状态。

所有可能的状态转换和它们的返回代码都显示在以下的转换矩阵中：

返回代码（HRESULT）		结果状态	
		ERROR	INITIAL
初始状态	ERROR	FAILED	SUCCESS

在成功的情况下，S_OK 或 S_FALSE（如果没有错误需要重置）将作为 SUCCESS 代码返回。

6.1.5.4.2.19 ResetCameraFeatures



Reset the camera features to initial state (sends the 'InitCameraCommands' defined in the GevImageAcquisition TcCOM module to the camera).

Syntax

Definition:

```
METHOD ResetCameraFeatures : HRESULT
```

Return value

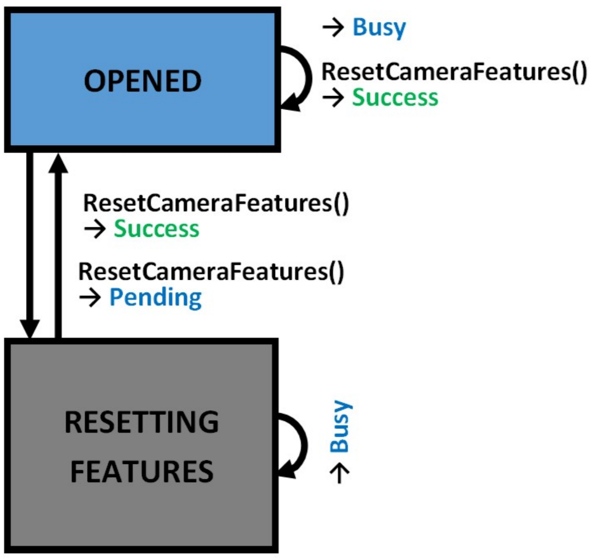
HRESULT [▸ 122]

更多信息

方法ResetCameraFeatures会将相机参数重置为原始状态。这意味着，初始化命令 [▸ 77]（不含可选的Force-IP 命令）将被重写到相机中。

状态转换

通过ResetCameraFeatures，相机只应重置其参数，同时保持在OPENED状态。然而，通常情况下，相机会短暂地进入中间状态RESETTINGFEATURES，因为参数重置需要一点时间。



附图 22：使用 FB_VN_GevCameraControl.ResetCameraFeatures 的状态转换

所有可能的状态转换和它们的返回代码都显示在以下的转换矩阵中：

返回代码（HRESULT）		结果状态		
		OPENED	RESETTINGFEATURES	ERROR
初始状态	OPENED	BUSY	PENDING	FAILED
	RESETTINGFEATURES	SUCCESS	BUSY	

6.1.5.4.2.20 SetCameraMatrix

Sets the camera matrix to the TcVnGevImageProvider TcCOM module

Syntax

Definition:

```
METHOD SetCameraMatrix : HRESULT
VAR_IN_OUT
    aCameraMatrix : TcVnMatrix3x3_LREAL;
END_VAR
```

In/Outputs

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 141]	The camera matrix to be copied to the TcCOM module

Return value

HRESULT [▶ 122]

6.1.5.4.2.21 SetDistortionCoefficients

Sets the distortion coefficients to the TcVnGevImageProvider TcCOM module

Syntax

Definition:

```
METHOD SetDistortionCoefficients : HRESULT
VAR_IN_OUT
    aDistortionCoefficients : TcVnArray8_LREAL;
END_VAR
```

In/Outputs

Name	Type	Description
aDistortionCoefficients	TcVnArray8_LREAL [▶ 141]	The distortion coefficients to be copied to the TcCOM module

Return value

HRESULT [▶ 122]

6.1.5.4.2.22 SetRotationMatrix

SetRotationMatrix	
↔ aRotationMatrix *TcVnMatrix3x3_LREAL*	*HRESULT* SetRotationMatrix

Sets the rotation matrix to the TcVnGevImageProvider TcCOM module

Syntax

Definition:

```
METHOD SetRotationMatrix : HRESULT
VAR_IN_OUT
    aRotationMatrix : TcVnMatrix3x3_LREAL;
END_VAR
```

In/Outputs

Name	Type	Description
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 141]	The rotation matrix to be copied to the TcCOM module

Return value

HRESULT [▶ 122]

6.1.5.4.2.23 SetTranslationVector

SetTranslationVector	
↔ aTranslationVector *TcVnVector3_LREAL*	*HRESULT* SetTranslationVector

Sets the translation vector to the TcVnGevImageProvider TcCOM module

Syntax

Definition:

```
METHOD SetTranslationVector : HRESULT
VAR_IN_OUT
    aTranslationVector : TcVnVector3_LREAL;
END_VAR
```

In/Outputs

Name	Type	Description
aTranslationVector	TcVnVector3_LREAL [▶ 141]	The translation vector to be copied to the TcCOM module

Return value

HRESULT [▶ 122]

6.1.5.4.2.24 StartAcquisition

StartAcquisition
HRESULT StartAcquisition

Send the 'StartAcquisitionCommands' defined in the TcVnGevImageProvider TcCOM module to the camera.

Syntax

Definition:

```
METHOD StartAcquisition : HRESULT
```

Return value

HRESULT [▶ 122]

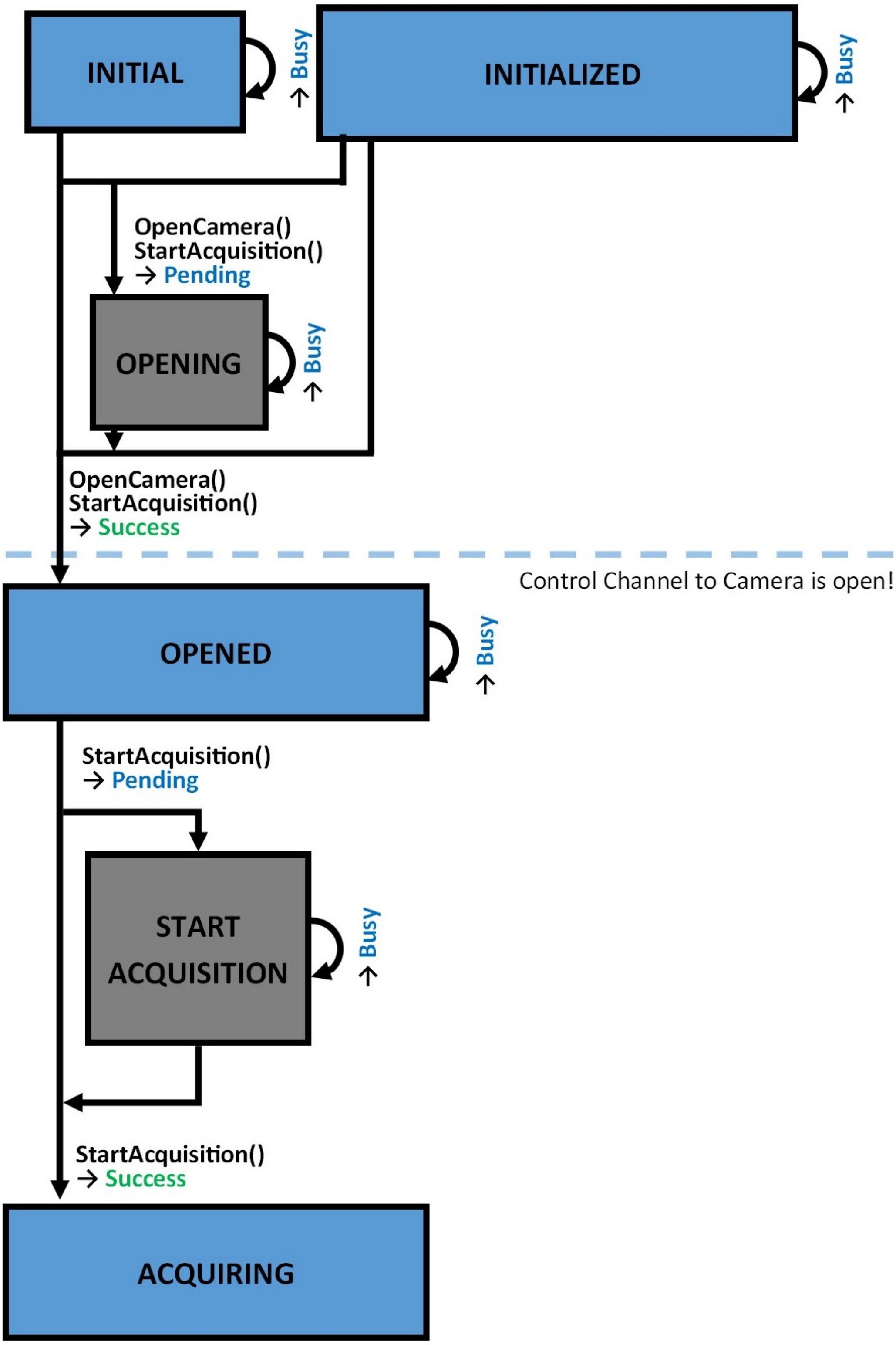
更多信息

方法StartAcquisition启动相机的图像采集。这在个别情况下意味着什么，取决于相机的参数化。关于更多信息，可查看样本图像采集和触发 [▶ 2729]。图像采集可以从INITIAL、INITIALIZED和OPENED状态开始。

如果相机仍在INITIAL或INITIALIZED状态，则方法StartAcquisition还执行方法OpenCamera [▶ 1566]的任务。通常情况下，不会进入OPENED状态。

状态转换

通过StartAcquisition，相机将处于ACQUIRING状态。这可以在INITIAL、INITIALIZED和OPENED状态下进行。如果相机处于 INITIAL 或 INITIALIZED 状态，则首先执行向 OPENED 的状态转换。通常情况下，这通过中间状态 OPENING 进行。通常，从 OPENED 到 ACQUIRING 的转变通过中间状态 STARTACQUISITION 进行。



附图 23: 使用 FB_VN_GevCameraControl.StartAcquisition 的状态转换

所有可能的状态转换和它们的返回代码都显示在以下的转换矩阵中：

<table>
<tr><th colspan="2" rowspan="2">返回代码（HRESULT）</th><th colspan="7">结果状态</th></tr>
<tr><th>INITIAL</th><th>INITIALIZED</th><th>OPENING</th><th>OPENED</th><th>STARTACQUISITION</th><th>ACQUIRING</th><th>ERROR</th></tr>
<tr><td rowspan="5">初始状态</td><td>INITIAL</td><td>BUSY</td><td>-</td><td>PENDING</td><td rowspan="3">SUCCESS</td><td rowspan="3">-</td><td rowspan="3">-</td><td rowspan="5">FAILED</td></tr>
<tr><td>INITIALIZED</td><td rowspan="4">-</td><td>BUSY</td><td>PENDING</td></tr>
<tr><td>OPENING</td><td rowspan="3">-</td><td>BUSY</td></tr>
<tr><td>OPENED</td><td rowspan="2">-</td><td>BUSY</td><td>PENDING</td><td rowspan="2">SUCCESS</td></tr>
<tr><td>STARTACQUISITION</td><td>-</td><td>BUSY</td></tr>
</table>

6.1.5.4.2.25 StopAcquisition

StopAcquisition	
HRESULT	StopAcquisition

Send the 'StopAcquisitionCommands' defined in the TcVnGevImageProvider TcCOM module to the camera.

Syntax

Definition:

```
METHOD StopAcquisition : HRESULT
```

Return value

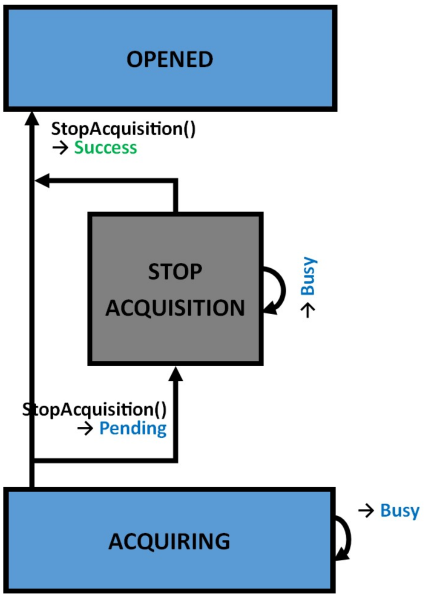
HRESULT [▸ 122]

更多信息

方法StopAcquisition停止相机的图像采集。

状态转换

通过StopAcquisition，将让相机从ACQUIRING状态进入OPENED状态。通常情况下，这不会立即发生，而是通过中间状态STOPACQUISITION进行。



附图 24: 使用 FB_VN_GevCameraControl.StopAcquisition 的状态转换

所有可能的状态转换和它们的返回代码都显示在以下的转换矩阵中：

返回代码（HRESULT）		结果状态			
		ACQUIRING	STOPACQUISITION	OPENED	ERROR
初始状态	ACQUIRING	BUSY	PENDING	SUCCESS	FAILED
	STOPACQUISITION	–	BUSY		

6.1.5.4.2.26 TriggerImage

Send the 'SoftwareTriggerCommands' defined in the TcVnGevImageProvider TcCOM module to the camera.

Syntax

Definition:

```
METHOD TriggerImage : HRESULT
```

Return value

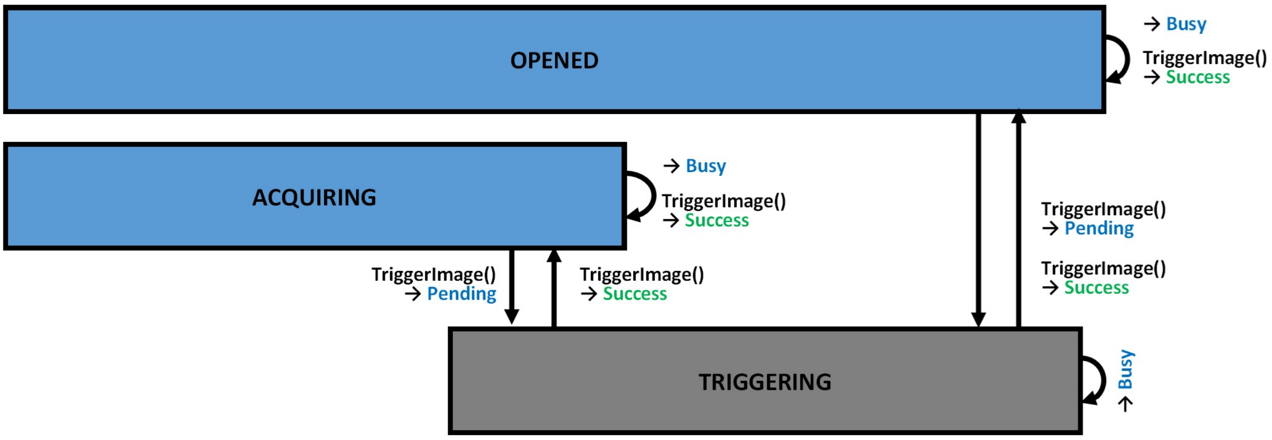
HRESULT [▶ 122]

更多信息

配置相机：图像采集和触发 [▶ 2729]

状态转换

通过TriggerImage，在保持当前的ACQUIRING或OPENED状态的情况下，相机将只采集一个图像。由于图像采集通常需要几个周期，相机通常会短暂地进入中间状态TRIGGERING。



附图 25: 使用 FB_VN_GevCameraControl.TriggerImage 的状态转换

所有可能的状态转换和它们的返回代码都显示在以下的转换矩阵中：

返回代码（HRESULT）		结果状态			
		OPENED	ACQUIRING	TRIGGERING	ERROR
初始状态	OPENED	BUSY	–	PENDING	FAILED
	ACQUIRING	–	BUSY		
	TRIGGERING	SUCCESS	SUCCESS	BUSY	

6.1.5.4.3 FB_VN_SimpleCameraControl

This FB provides the basic functionality to control a camera or a FileSource.

Do not call the main FB directly. Only use the available methods.

Methods

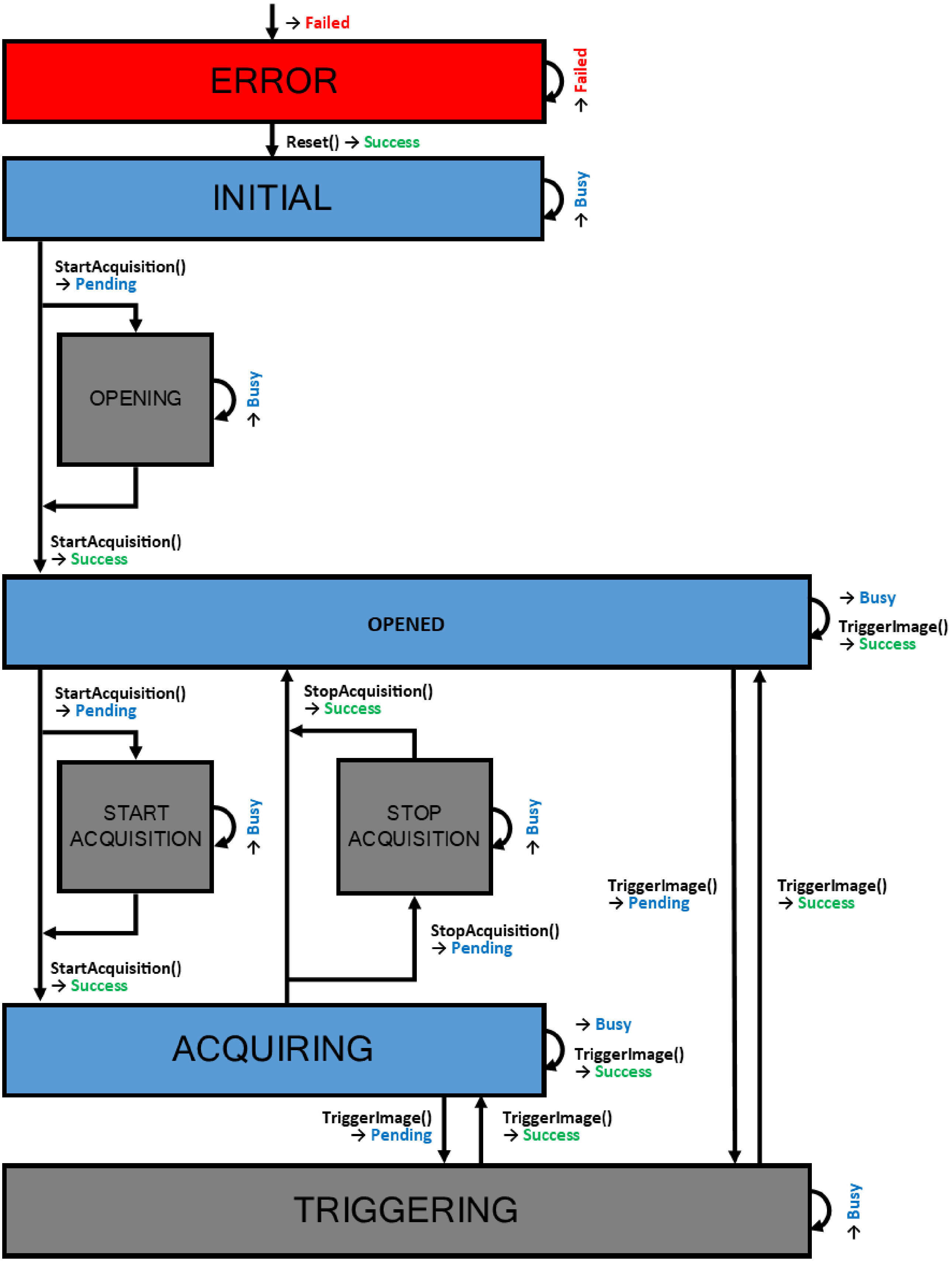
Name	Description
GetCurrentImage [▸ 1578]	Get the current available image (if any).
GetState [▸ 1579]	Get the current state of the internal camera control state machine.
Reset [▸ 1579]	Reset the camera to initial state (might require multiple calls depending on current state, until S_OK is returned)
StartAcquisition [▸ 1580]	Start the image acquisition.
StopAcquisition [▸ 1580]	Stop the image acquisition.
TriggerImage [▸ 1580]	Trigger the next image.

更多信息

功能块FB_VN_SimpleCameraControl结合了FB_VN_GevCameraControl [▸ 1550]和FB_VN_FileSourceControl [▸ 1541]的共同特征。这意味着，功能块的实例可以同时链接到相机的图像提供者和文件源控制的图像提供者，如第一步 [▸ 44]一样。

状态机

下图显示了状态机。主要状态以蓝色标记，过渡状态为灰色，而错误状态为红色。相比文件源控制，相机在从TCVN_CS_INITIAL到TCVN_CS_ACQUIRING的转换过程中也始终处于过渡状态TCVN_CS_OPENING，且必要时也处于主状态TCVN_CS_OPENED。如果存在初始化命令，并且选择了初始化自动模式AUTOINIT_AFTER_SO，那么也会短暂地进入发生自动初始化的INITIALIZING状态。如果转换发生得非常快，相应的状态就不会向外界报告，因为在一个周期内已经达到了后续的状态。在链接到文件源控制时，不会进入这三种状态。在TCVN_CS_OPENED状态下使用触发图像的方法仅可用于一台相机。



以下方法可用于控制状态机：

- GetState [▸ 1579]
- StartAcquisition [▸ 1580]
- StopAcquisition [▸ 1580]
- TriggerImage [▸ 1580]

- Reset [▶ 1579]

如果在非预期状态下调用方法，将返回错误代码INVALIDSTATE（16#712）。

带有连续图像采集的应用

例如，用于将相机置于图像采集模式、处理错误状态并持续显示成功采集的图像的状态机如下所示：

```
VAR
    hr              :   HRESULT;
    ipImageIn       :   ITcVnImage;
    ipImageInDisp   :   ITcVnDisplayableImage;
    fbCameraControl :   FB_VN_SimpleCameraControl;
    eCameraState    :   ETcVnCameraState;
    nNewImageCounter:   UINT;
END_VAR

eCameraState := fbCameraControl.GetState();

// CameraControl is in error state, so try to reset the camera connection
IF eCameraState = TCVN_CS_ERROR THEN
    hr := fbCameraControl.Reset();

// Camera not yet streaming
ELSIF eCameraState < TCVN_CS_ACQUIRING THEN
    hr := fbCameraControl.StartAcquisition();

// Camera streaming
ELSIF eCameraState = TCVN_CS_ACQUIRING THEN
    hr := fbCameraControl.GetCurrentImage(ipImageIn);

    // Check if new Image was received
    IF SUCCEEDED(hr) AND ipImageIn <> 0 THEN
        nNewImageCounter := nNewImageCounter + 1;

        // Place to call vision algorithms
        hr := F_VN_TransformIntoDisplayableImage(ipImageIn, ipImageInDisp, hr);
    END_IF

END_IF
```

在这种情况下，可以将相机配置为连续图像采集 [▶ 2730]或配置为硬件触发 [▶ 2731]。

带有软件触发器的应用

例如，将相机置于图像采集模式、处理错误状态并通过软件触发器触发图像采集的状态机如下所示：

```
VAR
    hr              :   HRESULT;
    ipImageIn       :   ITcVnImage;
    ipImageInDisp   :   ITcVnDisplayableImage;
    fbCameraControl :   FB_VN_SimpleCameraControl;
    eCameraState    :   ETcVnCameraState;
    nNewImageCounter:   UINT;
    bTrigger        :   BOOL := TRUE;
END_VAR

eCameraState := fbCameraControl.GetState();

// CameraControl is in error state, so try to reset the camera connection
IF eCameraState = TCVN_CS_ERROR THEN
    hr := fbCameraControl.Reset();

// Camera trigger image
ELSIF eCameraState = TCVN_CS_TRIGGERING THEN
    hr := fbCameraControl.TriggerImage();

// Camera not yet streaming
ELSIF eCameraState < TCVN_CS_ACQUIRING THEN
    hr := fbCameraControl.StartAcquisition();

// Camera streaming
ELSIF eCameraState = TCVN_CS_ACQUIRING THEN

    IF bTrigger THEN
        hr := fbCameraControl.TriggerImage();
        IF SUCCEEDED(hr) THEN
            bTrigger := FALSE;
        END_IF
```

```
    ELSE
        hr := fbCameraControl.GetCurrentImage(ipImageIn);

        // Check if new Image was received
        IF SUCCEEDED(hr) AND ipImageIn <> 0 THEN
            nNewImageCounter := nNewImageCounter + 1;

            // Trigger next image
            bTrigger := TRUE;

            // Place to call vision algorithms
            hr := F_VN_TransformIntoDisplayableImage(ipImageIn, ipImageInDisp, hr);
        END_IF
    END_IF

END_IF
```

在这种情况下，必须将相机配置为软件触发器 [▶ 2731]（TriggerSource = Software）。

样本

在许多函数样本 [▶ 2640]中，FB_VN_SimpleCameraControl功能块用于加载样本图像。

相关函数

- FB_VN_SimpleCameraControl [▶ 1575]用于相机和文件源
- FB_VN_GevCameraControl [▶ 1550]用于 GigE Vision 相机
- FB_VN_FileSourceControl [▶ 1541]用于文件源

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.4.3.1 GetCurrentImage

Get the current available image (if any).

Syntax

Definition:

```
METHOD GetCurrentImage : HRESULT
VAR_INPUT
    ipImage : Reference To ITcVnImage;
END_VAR
```

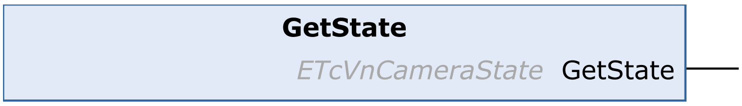
Inputs

Name	Type	Description
ipImage	Reference To ITcVnImage [▶ 390]	Returns the image

Return value

HRESULT [▶ 122]

6.1.5.4.3.2 GetState



Get the current state of the internal camera control state machine.

Syntax

Definition:

```
METHOD GetState : ETcVnCameraState
```

Return value

ETcVnCameraState [▶ 147]

更多信息

方法GetState将相机对象或文件源对象的状态作为枚举ETcVnCameraState [▶ 147]的值返回。可能出现以下状态：

- ERROR
- INITIAL
- OPENING
- OPENED
- STARTACQUISITION
- STOPACQUISITION
- ACQUIRING
- TRIGGERING

6.1.5.4.3.3 Reset

Reset the camera to initial state (might require multiple calls depending on current state, until S_OK is returned)

Syntax

Definition:

```
METHOD Reset : HRESULT
```

Return value

HRESULT [▶ 122]

更多信息

如果设备处于错误状态，方法Reset将对其进行复位。

状态转换

通过Reset，会让设备从ERROR状态进入INITIAL状态。如果复位成功，这个转换立即发生，否则功能块仍处于ERROR状态。

所有可能的状态转换和它们的返回代码都显示在以下的转换矩阵中：

返回代码（HRESULT）		结果状态	
		ERROR	INITIAL
初始状态	ERROR	FAILED	SUCCESS

在成功的情况下，S_OK 或 S_FALSE（如果没有错误需要重置）将作为 SUCCESS 代码返回。

6.1.5.4.3.4 StartAcquisition

StartAcquisition
HRESULT StartAcquisition

Start the image acquisition.

Syntax

Definition:

```
METHOD StartAcquisition : HRESULT
```

Return value

HRESULT [▸ 122]

更多信息

- 该方法与相机相关，对应于FB_VN_GevCameraControl [▸ 1550]的StartAcquisition [▸ 1570]方法。
- 该方法与文件源控制相关，对应于FB_VN_FileSourceControl [▸ 1541]的StartAcquisition [▸ 1546]方法。

6.1.5.4.3.5 StopAcquisition

StopAcquisition
HRESULT StopAcquisition

Stop the image acquisition.

Syntax

Definition:

```
METHOD StopAcquisition : HRESULT
```

Return value

HRESULT [▸ 122]

更多信息

- 该方法与相机相关，对应于FB_VN_GevCameraControl [▸ 1550]的StopAcquisition [▸ 1573]方法。
- 该方法与文件源控制相关，对应于FB_VN_FileSourceControl [▸ 1541]的StopAcquisition [▸ 1547]方法。

6.1.5.4.3.6 TriggerImage

TriggerImage
HRESULT TriggerImage

Trigger the next image.

Syntax

Definition:

```
METHOD TriggerImage : HRESULT
```

Return value

HRESULT [▶ 122]

更多信息

- 该方法与相机相关，对应于FB_VN_GevCameraControl [▶ 1550]的TriggerImage [▶ 1574]方法。
- 该方法与文件源控制相关，对应于FB_VN_FileSourceControl [▶ 1541]的TriggerImage [▶ 1574]方法。

6.1.5.5 Image Processing

该组包含用于图像处理的功能块。在这些综合功能块中，设定的值在执行后会被保留。此外，功能块也可以包含几个方法。

另一方面，Functions [▶ 408]章中包含的函数没有内部状态信息，且变量值也不保留。

功能块

- FB_VN_GeneralizedHoughBallard [▶ 1581]
- FB_VN_SSIM [▶ 1586]

6.1.5.5.1 FB_VN_GeneralizedHoughBallard

This FB provides the Generalized Hough Ballard functionality. First, set the parameters via the corresponding methods (optional, defaults are used otherwise). Then, set the template (required). Afterwards, the Detect method can be executed.

Do not call the main FB directly. Only use the available methods.

Methods

Name	Description
Detect [▶ 1582]	Detect the template in an image (SetTemplate must have been called before).
SetCannyThreshold [▶ 1583]	Sets the canny edge detection thresholds.
SetInvAccuRatio [▶ 1583]	Sets the inverted ratio of the accumulator size in relation to the source image's size (e.g. a value of 2 means that the size is halved in both directions.)
SetLevels [▶ 1584]	Sets the number of R-table levels.
SetMinDist [▶ 1584]	Sets the minimum distance between the centers of different objects.
SetTemplate [▶ 1585]	Sets the template image to search for in the Detect method.
SetVotesThreshold [▶ 1585]	Sets the accumulator threshold, i.e. the number of votes required to detect a match (too small values lead to false detections).

更多信息

功能块FB_VN_GeneralizedHoughBallard通过广义霍夫变换在灰度图像中找到任意形状。所识别模板形状的中心位置被确定，而模板的不同缩放和旋转未考虑在内。

参数

输入图像

模板和输入图像必须是元素类型为 USINT（8 位）的单通道图像。首先，必须使用SetTemplate方法传递模板图像。之后，可以通过调用Detect方法进行执行，然后传递输入图像。

检测的参数

以下参数可以通过方法调用设置。否则，将使用这些默认值。由于在传递模板图像时，这些参数已经用于内部计算，所以必须始终先设置这些值。

CannyThresholdLow = 50

CannyThresholdHigh = 100

投资回报率 = 1.0

Levels = 360

MinDist = 1.0

VotesThreshold = 100

找到的中心点列表（返回值）

所有在图像中找到的中心点位置都会在容器ipPositions中调用Detect方法后返回。

应用

执行过程包括 3 个步骤。首先设置参数，然后传递模板图像。之后，只需调用Detect方法，在图像上搜索模板即可。只有当需要更改参数或模板时，相应的步骤才需要重新执行。

```
// Set parameters first
hr := fbGenHoughBallard.SetCannyThreshold(50,150);
IF SUCCEEDED(hr) THEN hr := fbGenHoughBallard.SetInvAccuRatio(2); END_IF
IF SUCCEEDED(hr) THEN hr := fbGenHoughBallard.SetLevels(180); END_IF
IF SUCCEEDED(hr) THEN hr := fbGenHoughBallard.SetMinDist(20); END_IF
IF SUCCEEDED(hr) THEN hr := fbGenHoughBallard.SetVotesThreshold(80); END_IF

// Check if successful and set template image
IF SUCCEEDED(hr) THEN hr := fbGenHoughBallard.SetTemplate(ipTemplateImage); END_IF

// Check if successful and call detection
IF SUCCEEDED(hr) THEN hr := fbGenHoughBallard.Detect(ipSrcImage,ipPositions); END_IF
```

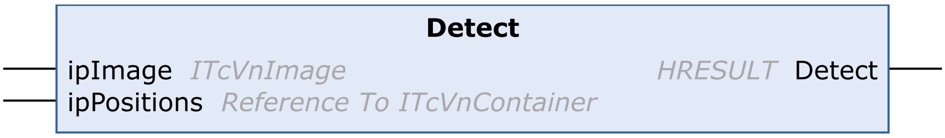
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.5.1.1 Detect



Detect the template in an image (SetTemplate must have been called before).

Syntax

Definition:

```
METHOD Detect : HRESULT
VAR_INPUT
    ipImage      : ITcVnImage;
    ipPositions : Reference To ITcVnContainer;
END_VAR
```

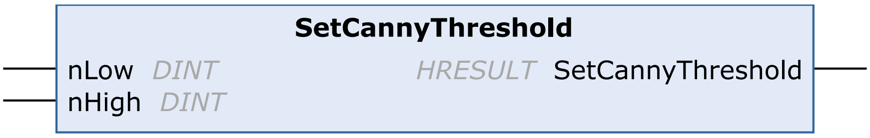
Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	Source image (1 channel, USINT)
ipPositions	Reference To ITcVnContainer [▶ 349]	Returns the centers of the detected template positions (ContainerType_Vector_TcVnPoint2_REAL; Non-zero interface pointers are reused.)

Return value

HRESULT [▶ 122]

6.1.5.5.1.2 SetCannyThreshold



Sets the canny edge detection thresholds.

Syntax

Definition:

```
METHOD SetCannyThreshold : HRESULT
VAR_INPUT
    nLow  : DINT;
    nHigh : DINT;
END_VAR
```

Inputs

Name	Type	Description
nLow	DINT	Low threshold (> 0)
nHigh	DINT	High threshold (> nLow, usually 2 to 3 * nLow)

Return value

HRESULT [▶ 122]

6.1.5.5.1.3 SetInvAccuRatio

Sets the inverted ratio of the accumulator size in relation to the source image's size (e.g. a value of 2 means that the size is halved in both directions.)

Syntax

Definition:

```
METHOD SetInvAccuRatio : HRESULT
VAR_INPUT
    fInvAccuRatio : LREAL;
END_VAR
```

Inputs

Name	Type	Description
fInvAccuRatio	LREAL	Inverted accumulator ratio (> 0)

Return value

HRESULT [▶ 122]

6.1.5.5.1.4 SetLevels

Sets the number of R-table levels.

Syntax

Definition:

```
METHOD SetLevels : HRESULT
VAR_INPUT
    nLevels : DINT;
END_VAR
```

Inputs

Name	Type	Description
nLevels	DINT	Number of R-table levels (> 0)

Return value

HRESULT [▶ 122]

6.1.5.5.1.5 SetMinDist

Sets the minimum distance between the centers of different objects.

Syntax

Definition:

```
METHOD SetMinDist : HRESULT
VAR_INPUT
    fMinDist : LREAL;
END_VAR
```

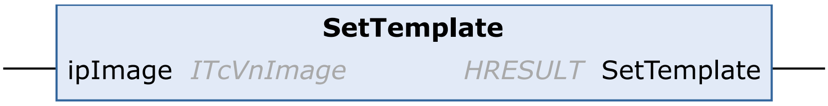
Inputs

Name	Type	Description
fMinDist	LREAL	Minimum distance (> 0)

Return value

HRESULT [▶ 122]

6.1.5.5.1.6 SetTemplate



Sets the template image to search for in the Detect method.

Syntax

Definition:

```
METHOD SetTemplate : HRESULT
VAR_INPUT
    ipImage : ITcVnImage;
END_VAR
```

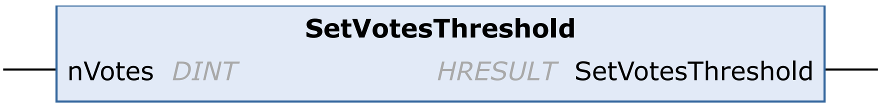
Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	Template image (1 channel, USINT)

Return value

HRESULT [▶ 122]

6.1.5.5.1.7 SetVotesThreshold



Sets the accumulator threshold, i.e. the number of votes required to detect a match (too small values lead to false detections).

Syntax

Definition:

```
METHOD SetVotesThreshold : HRESULT
VAR_INPUT
    nVotes : DINT;
END_VAR
```

Inputs

Name	Type	Description
nVotes	DINT	Number of votes required to detect a match (> 0)

Return value

HRESULT [▶ 122]

6.1.5.5.2 FB_VN_SSIM

This FB provides the structural similarity (SSIM) functionality. First, set the reference image. Afterwards, the Compute method can be executed.

Do not call the main FB directly. Only use the available methods.

Methods

Name	Description
Compute [▶ 1586]	Compute the structural similarity (SetReference must have been called before).
SetReference [▶ 1587]	Sets the reference image, used by the SSIM method.

更多信息

功能块FB_VN_SSIM计算两个灰度图像的结构相似性。作为参考，假设是一个无干扰的图像。

参数

输入图像

参考和输入图像必须是元素类型为USINT（8 位）的单通道图像。首先，必须使用SetReference 方法传递参考图像。之后，可以通过调用Compute方法进行执行，然后传递输入图像。

输出图像（返回值）

输出图像的类型是Real，并按 0-1 的数值范围进行缩放。1 表示同意，而 0 表示最大偏离。

应用

执行过程包括 2 个步骤。首先，必须设置参考图像。之后，可以调用Compute方法，该方法被传递给要比较的当前输入图像。

为了进一步分析结果图像，例如使用F_VN_DetectBlobs [▶ 1139]函数，需要将图像转换成USINT。函数F_VN_ConvertElementTypeExp [▶ 748]可用于此目的。通过函数F_VN_ImageAverage [▶ 1178]或F_VN_MinPixelValue [▶ 1188]，可以直接根据结果图像确定匹配值。

```
// Set reference image first
hr := fbSSIM.SetReference(ipImage := ipRefImage);

// Check if successful, set imput image and compute the result
IF SUCCEEDED(hr) and ipRefImage <> 0 THEN
    hr := fbSSIM.Compute(ipSrcImage:= ipImageIn, ipSSIM:= ipResultImage);
    // Convert and scale the result image to USINT
    hr := F_VN_ConvertElementTypeExp(ipResultImage, ipResult2Image, ETcVnElementType.TCVN_ET_USINT,
(255.0/1.0), 0, hr);
END_IF
```

大图像的高内存使用率

由于该函数在后台用真实值进行许多计算，因此根据图像大小需要更多的路由器内存和计算能力。

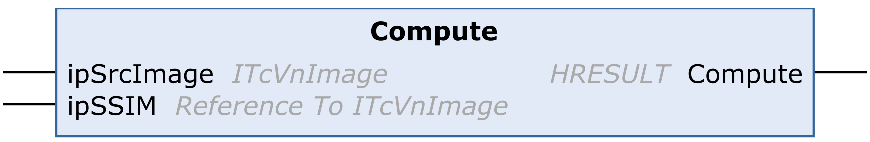
Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.1.5.5.2.1 Compute



Compute the structural similarity (SetReference must have been called before).

Syntax

Definition:

```
METHOD Compute : HRESULT
VAR_INPUT
    ipSrcImage : ITcVnImage;
    ipSSIM     : Reference To ITcVnImage;
END_VAR
```

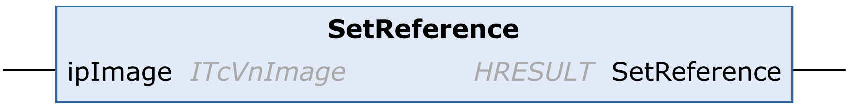
Inputs

Name	Type	Description
ipSrcImage	ITcVnImage [▶ 390]	Source image (USINT)
ipSSIM	Reference To ITcVnImage [▶ 390]	Returns the SSIM (REAL, values from 0 to 1)

Return value

HRESULT [▶ 122]

6.1.5.5.2.2 SetReference



Sets the reference image, used by the SSIM method.

Syntax

Definition:

```
METHOD SetReference : HRESULT
VAR_INPUT
    ipImage : ITcVnImage;
END_VAR
```

Inputs

Name	Type	Description
ipImage	ITcVnImage [▶ 390]	Reference image (USINT)

Return value

HRESULT [▶ 122]

6.2 C++

本章包含用于在 C++ 中编程使用库的 TwinCAT Vision API 说明。

- 数据类型 [▶ 1588]
  （别名、常量、数组、枚举、结构）
- 接口 [▶ 1659]
  （图像、容器等）
- 函数 [▶ 1812]
  （图像处理）

- 功能块 [▶ 2549]
  （与相机和文件系统通信、图像处理）

6.2.1 Data Types

共有四种数据类型，每种类型的前缀都是“TcVn”：

- 某些数组类型的别名 [▶ 1588]涉及图像处理
- 具有特定数据类型组合和定义数的数组 [▶ 1590]经常被图像处理函数使用。
- 枚举 [▶ 1590]有助于函数的参数设置，并包含附加前缀“E”
- 结构 [▶ 1637]整合多个相关信息片段（如几何描述）

此外，还提供用于描述容器类型的常量 [▶ 1588]。

6.2.1.1 Aliases

Name	Type
GVCP_REGISTER_ADDRESS	ULONG
TcVnPoint	TcVnPoint2_DINT [▶ 1588]
TcVnPoint2_DINT	TcVnVector2_DINT [▶ 1590]
TcVnPoint2_LREAL	TcVnVector2_LREAL [▶ 1590]
TcVnPoint2_REAL	TcVnVector2_REAL [▶ 1590]
TcVnPoint3_LREAL	TcVnVector3_LREAL [▶ 1590]
TcVnPoint3_REAL	TcVnVector3_REAL [▶ 1590]
TcVnRectangle	TcVnRectangle_DINT [▶ 1657]

6.2.1.2 Constants

以下常量将容器 [▶ 132]类型表示为 GUID。例如，在使用 CreateContainer [▶ 2064] 等函数时就需要这些。

Name	Type
ContainerType_Vector_SINT	GUID
ContainerType_Vector_USINT	GUID
ContainerType_Vector_INT	GUID
ContainerType_Vector_UINT	GUID
ContainerType_Vector_DINT	GUID
ContainerType_Vector_UDINT	GUID
ContainerType_Vector_ULINT	GUID
ContainerType_Vector_REAL	GUID
ContainerType_Vector_LREAL	GUID
ContainerType_Vector_ITcVnImage	GUID
ContainerType_Vector_TcVnPoint2_DINT	GUID
ContainerType_Vector_TcVnPoint2_REAL	GUID
ContainerType_Vector_TcVnPoint2_LREAL	GUID
ContainerType_Vector_TcVnPoint3_REAL	GUID
ContainerType_Vector_TcVnPoint3_LREAL	GUID
ContainerType_Vector_Vector_TcVnPoint2_DINT	GUID
ContainerType_Vector_Vector_TcVnPoint2_REAL	GUID
ContainerType_Vector_Vector_TcVnPoint3_REAL	GUID
ContainerType_Vector_Vector_UDINT	GUID
ContainerType_Vector_Vector_ULINT	GUID
ContainerType_Vector_Vector_REAL	GUID
ContainerType_Vector_Vector_LREAL	GUID
ContainerType_Vector_TcVnVector2_SINT	GUID
ContainerType_Vector_TcVnVector2_USINT	GUID
ContainerType_Vector_TcVnVector2_INT	GUID
ContainerType_Vector_TcVnVector2_UINT	GUID
ContainerType_Vector_TcVnVector2_DINT	GUID
ContainerType_Vector_TcVnVector2_REAL	GUID
ContainerType_Vector_TcVnVector3_SINT	GUID
ContainerType_Vector_TcVnVector3_USINT	GUID
ContainerType_Vector_TcVnVector3_INT	GUID
ContainerType_Vector_TcVnVector3_UINT	GUID
ContainerType_Vector_TcVnVector3_REAL	GUID
ContainerType_Vector_TcVnVector4_SINT	GUID
ContainerType_Vector_TcVnVector4_USINT	GUID
ContainerType_Vector_TcVnVector4_INT	GUID
ContainerType_Vector_TcVnVector4_UINT	GUID
ContainerType_Vector_TcVnVector4_DINT	GUID
ContainerType_Vector_TcVnVector4_LREAL	GUID
ContainerType_Vector_TcVnRectangle_DINT	GUID
ContainerType_Vector_TcVnKeyPoint	GUID
ContainerType_Vector_TcVnDMatch	GUID
ContainerType_Vector_Vector_TcVnDMatch	GUID
ContainerType_String_SINT	GUID
ContainerType_Vector_String_SINT	GUID

6.2.1.3 Arrays

Name	Type
TcVnArray10_ITcVnImage	PVOID[10]
TcVnArray3_Point2_REAL	TcVnPoint2_REAL [▶ 1588][3]
TcVnArray33_UDINT	ULONG[33]
TcVnArray4_LREAL	double[4]
TcVnArray4_Point2_REAL	TcVnPoint2_REAL [▶ 1588][4]
TcVnArray7_LREAL	double[7]
TcVnArray8_LREAL	double[8]
TcVnMatrix2x3_LREAL	double[2][3]
TcVnMatrix3x3_LREAL	double[3][3]
TcVnVector2_DINT	LONG[2]
TcVnVector2_INT	SHORT[2]
TcVnVector2_LREAL	double[2]
TcVnVector2_REAL	float[2]
TcVnVector2_SINT	char[2]
TcVnVector2_UINT	USHORT[2]
TcVnVector2_USINT	unsigned char[2]
TcVnVector3_INT	SHORT[3]
TcVnVector3_LREAL	double[3]
TcVnVector3_REAL	float[3]
TcVnVector3_SINT	char[3]
TcVnVector3_UINT	USHORT[3]
TcVnVector3_USINT	unsigned char[3]
TcVnVector4_DINT	LONG[4]
TcVnVector4_INT	SHORT[4]
TcVnVector4_LREAL	double[4]
TcVnVector4_SINT	char[4]
TcVnVector4_UINT	USHORT[4]
TcVnVector4_USINT	unsigned char[4]

6.2.1.4 Enums

6.2.1.4.1 ETcVn2dCodeSearchStrategy

Offers search strategies for 2d code reading (multiple TCVN_CSS_XXX_INVERTED or multiple TCVN_CSS_XXX_FLIPPED cannot be combined).

Syntax

Definition:

```
enum ETcVn2dCodeSearchStrategy : ULONG
{
    CSS_DEFAULT            = 1,
    CSS_ONLY_NOT_INVERTED  = 8,
    CSS_FIRST_NOT_INVERTED = 10,
    CSS_ONLY_INVERTED      = 12,
    CSS_FIRST_INVERTED     = 14,
    CSS_ONLY_NOT_FLIPPED   = 64,
    CSS_FIRST_NOT_FLIPPED  = 80,
    CSS_ONLY_FLIPPED       = 96,
    CSS_FIRST_FLIPPED      = 112
};
```

Values

Name	Description
CSS_DEFAULT	The algorithm searches for codes using the default searching strategy, all other flags are ignored (dependant on code type).
CSS_ONLY_NOT_INVERTED	The algorithm searches for codes only in the non-inverted image.
CSS_FIRST_NOT_INVERTED	The algorithm first searches for codes in the non-inverted and then in the inverted image.
CSS_ONLY_INVERTED	The algorithm searches for codes only in the inverted image.
CSS_FIRST_INVERTED	The algorithm first searches for codes in the inverted and then in the non-inverted image.
CSS_ONLY_NOT_FLIPPED	The algorithm searches for codes only in the non-mirrored image.
CSS_FIRST_NOT_FLIPPED	The algorithm first searches for codes in the non-mirrored and then in the mirrored image.
CSS_ONLY_FLIPPED	The algorithm searches for codes only in the mirrored image.
CSS_FIRST_FLIPPED	The algorithm first searches for codes in the mirrored and then in the non-mirrored image.

6.2.1.4.2 ETcVnAdaptiveThresholdMethod

Offers methods for adaptive threshold.

Syntax

Definition:

```
enum ETcVnAdaptiveThresholdMethod : LONG
{
    ATM_MEAN     = 0,
    ATM_GAUSSIAN = 1
};
```

Values

Name	Description
ATM_MEAN	The threshold value is calculated as the mean of the nBlockSize x nBlockSize neighborhood of (x, y) minus fConstant.
ATM_GAUSSIAN	The threshold value is the weighted sum (cross-correlation with a Gaussian window) of the nBlockSize x nBlockSize neighborhood of (x, y) minus fConstant.

6.2.1.4.3 ETcVnBarcodeSearchDirection

Offers search directions for linear barcodes.

Syntax

Definition:

```
enum ETcVnBarcodeSearchDirection : ULONG
{
    BSD_ANY        = 0,
    BSD_HORIZONTAL = 1,
    BSD_VERTICAL   = 2
};
```

Values

Name	Description
BSD_ANY	The algorithm searches for codes in any supported directions.
BSD_HORIZONTAL	The algorithm searches for codes in horizontal direction.
BSD_VERTICAL	The algorithm searches for codes in vertical direction.

6.2.1.4.4 ETcVnBarcodeType

Offers linear barcode types to search for.

Syntax

Definition:

```
enum ETcVnBarcodeType : ULONG
{
    BT_CODABAR         = 4,
    BT_CODE39          = 8,
    BT_CODE93          = 16,
    BT_CODE128         = 32,
    BT_EAN8            = 128,
    BT_EAN13           = 256,
    BT_ITF             = 512,
    BT_UPCA            = 32768,
    BT_UPCE            = 65536,
    BT_ANY             = 99260,
    BT_CODE39EXTENDED = 262144
};
```

Values

Name	Description
BT_CODABAR	The algorithm searches for Codabar codes.
BT_CODE39	The algorithm searches for Code-39 codes.
BT_CODE93	The algorithm searches for Code-93 codes.
BT_CODE128	The algorithm searches for Code-128 codes.
BT_EAN8	The algorithm searches for EAN-8 codes.
BT_EAN13	The algorithm searches for EAN-13 codes.
BT_ITF	The algorithm searches for ITF codes.
BT_UPCA	The algorithm searches for UPC-A codes.
BT_UPCE	The algorithm searches for UPC-E codes.
BT_ANY	The algorithm searches for any supported linear barcode. If the type of the code is known, it is recommended to select the specific type directly.
BT_CODE39EXTENDED	The algorithm searches for Code-93-Extended codes.

6.2.1.4.5 ETcVnBlobCombination

Offers multiple blob contours to choose from, which should be returned in a multi-threshold scenario (used in TcVnParamsBlobDetection, which in return is used in F_VN_DetectBlobs).

Syntax

Definition:

```
enum ETcVnBlobCombination : LONG
{
    BC_SMALLEST          = 0,
    BC_LARGEST           = 1,
    BC_MIN_THRESHOLD     = 2,
    BC_MAX_THRESHOLD     = 3,
    BC_MEDIAN_THRESHOLD = 4
};
```

Values

Name	Description
BC_SMALLEST	Returns the smallest blob contour.
BC_LARGEST	Returns the largest blob contour.
BC_MIN_THRESHOLD	Returns the blob contour for the minimum threshold.
BC_MAX_THRESHOLD	Returns the blob contour for the maximum threshold.
BC_MEDIAN_THRESHOLD	Returns the blob contour for the median threshold.

6.2.1.4.6 ETcVnBoostClassifierType

Offers different types of Boost classifiers.

Syntax

Definition:

```
enum ETcVnBoostClassifierType : LONG
{
    BCT_DISCRETE = 0,
    BCT_REAL     = 1,
    BCT_LOGIT    = 2,
    BCT_GENTLE   = 3
};
```

Values

Name	Description
BCT_DISCRETE	Discrete AdaBoost
BCT_REAL	Real AdaBoost
BCT_LOGIT	LogitBoost
BCT_GENTLE	Gentle AdaBoost

6.2.1.4.7 ETcVnBorderInterpolationMethod

Offers methods to extrapolate values of non-existing pixels. On the one hand, this is used for filtering functions to enable filtering at the image borders (where the filter mask reaches over the border). On the other hand, this is used to extrapolate the undefined pixels after a geometric image transformation.

Syntax

Definition:

```
enum ETcVnBorderInterpolationMethod : LONG
{
    BIM_CONSTANT             = 0,
    BIM_REPLICATE            = 1,
    BIM_REFLECT              = 2,
    BIM_WRAP                 = 3,
    BIM_REFLECT_101          = 4,
    BIM_DEFAULT              = 4,
    BIM_TRANSPARENT          = 5,
```

```
    BIM_ISOLATED_CONSTANT    = 16,
    BIM_ISOLATED_REPLICATE   = 17,
    BIM_ISOLATED_REFLECT     = 18,
    BIM_ISOLATED_WRAP        = 19,
    BIM_ISOLATED_REFLECT_101 = 20
};
```

Values

Name	Description
BIM_CONSTANT	iiiiiii | abcdefgh | iiiiiii (with some specified i)
BIM_REPLICATE	aaaaaaa | abcdefgh | hhhhhhh
BIM_REFLECT	gfedcba | abcdefgh | hgfedcb
BIM_WRAP	bcdefgh | abcdefgh | abcdefg
BIM_REFLECT_101	hgfedcb | abcdefgh | gfedcba
BIM_DEFAULT	Choose this if you don't know which method to select (same as REFLECT_101).
BIM_TRANSPARENT	Corresponding pixels in the destination image will not be modified. Only available for geometric image transformations.
BIM_ISOLATED_CONSTANT	Similar to CONSTANT, but ROIs are handled as isolated, which means surrounding image pixels outside the ROI are ignored.
BIM_ISOLATED_REPLICATE	Similar to REPLICATE, but ROIs are handled as isolated, which means surrounding image pixels outside the ROI are ignored.
BIM_ISOLATED_REFLECT	Similar to REFLECT, but ROIs are handled as isolated, which means surrounding image pixels outside the ROI are ignored.
BIM_ISOLATED_WRAP	Similar to WRAP, but ROIs are handled as isolated, which means surrounding image pixels outside the ROI are ignored.
BIM_ISOLATED_REFLECT_101	Similar to REFLECT_101, but ROIs are handled as isolated, which means surrounding image pixels outside the ROI are ignored.

6.2.1.4.8 ETcVnCalibrationPattern

Offers common calibration pattern types.

Syntax

Definition:

```
enum ETcVnCalibrationPattern : LONG
{
    CP_CIRCLES_SYM          = 0,
    CP_CIRCLES_ASYM         = 1,
    CP_CIRCLES_ASYM_INDENT = 2
};
```

Values

Name	Description
CP_CIRCLES_SYM	Symmetric circle pattern
CP_CIRCLES_ASYM	Asymmetric circle pattern, i.e. every 2nd row is indented, starting unindented
CP_CIRCLES_ASYM_INDENT	Asymmetric circle pattern, i.e. every 2nd row is indented, starting indented

6.2.1.4.9 ETcVnCalibrationPatternOrigin

Offers calibration pattern origins.

Syntax

Definition:

```
enum ETcVnCalibrationPatternOrigin : LONG
{
    CPO_TOPLEFT      = 0,
    CPO_TOPCENTER    = 1,
    CPO_TOPRIGHT     = 2,
    CPO_CENTERLEFT   = 3,
    CPO_CENTER       = 4,
    CPO_CENTERRIGHT  = 5,
    CPO_BOTTOMLEFT   = 6,
    CPO_BOTTOMCENTER = 7,
    CPO_BOTTOMRIGHT  = 8
};
```

Values

Name	Description
CPO_TOPLEFT	Origin is at top left
CPO_TOPCENTER	Origin is at top center
CPO_TOPRIGHT	Origin is at top right
CPO_CENTERLEFT	Origin is at center left
CPO_CENTER	Origin is at center
CPO_CENTERRIGHT	Origin is at center right
CPO_BOTTOMLEFT	Origin is at bottom left
CPO_BOTTOMCENTER	Origin is at bottom center
CPO_BOTTOMRIGHT	Origin is at bottom right

6.2.1.4.10 ETcVnCameraState

Specifies the state of a camera controller, which controls an attached camera.

Syntax

Definition:

```
enum ETcVnCameraState : LONG
{
    CS_ERROR             = -1,
    CS_INITIAL           = 0,
    CS_INITIALIZING      = 1,
    CS_INITIALIZED       = 2,
    CS_OPENING           = 3,
    CS_OPENED            = 4,
    CS_STARTACQUISITION  = 5,
    CS_ACQUIRING         = 6,
    CS_STOPACQUISITION   = 7,
    CS_RESETTINGFEATURES = 8,
    CS_TRIGGERING        = 9,
    CS_CLOSING           = 10
};
```

Values

Name	Description
CS_ERROR	The camera controller is in an error state (e.g. the connection to the camera was lost).
CS_INITIAL	The camera controller is in the initial state and ready to establish the connection to the camera.
CS_INITIALIZING	The camera controller is initializing the camera.
CS_INITIALIZED	The camera controller has initialized the camera.
CS_OPENING	The camera controller is establishing the connection to the camera.
CS_OPENED	The connection to the camera has been established and the camera controller is ready to start the image acquisition.
CS_STARTACQUISITION	The camera controller is starting the image acquisition.
CS_ACQUIRING	The camera is sending images (either streaming or manually triggered, depending on the configuration).
CS_STOPACQUISITION	The camera controller is stopping the image acquisition.
CS_RESETTINGFEATURES	The camera controller is resetting the features on the camera.
CS_TRIGGERING	The camera controller is processing a softwaretrigger.
CS_CLOSING	The camera controller is closing the connection to the camera.

6.2.1.4.11 ETcVnClusteringAlgorithm

Offers clustering algorithms

Syntax

Definition:

```
enum ETcVnClusteringAlgorithm : LONG
{
    CA_KMEANSPP = 0,
    CA_LBG      = 1
};
```

Values

Name	Description
CA_KMEANSPP	KMeans++ (fixed number of clusters)
CA_LBG	LBG variant (dynamic number of clusters)

6.2.1.4.12 ETcVnColorMap

Offers color maps (similar to GNU Octave/MATLAB types).

Syntax

Definition:

```
enum ETcVnColorMap : LONG
{
    CM_AUTUMN           = 0,
    CM_BONE             = 1,
    CM_JET              = 2,
    CM_WINTER           = 3,
    CM_RAINBOW          = 4,
    CM_OCEAN            = 5,
    CM_SUMMER           = 6,
    CM_SPRING           = 7,
    CM_COOL             = 8,
    CM_HSV              = 9,
    CM_PINK             = 10,
    CM_HOT              = 11,
    CM_PARULA           = 12,
    CM_MAGMA            = 13,
    CM_INFERNO          = 14,
    CM_PLASMA           = 15,
    CM_VIRIDIS          = 16,
    CM_CIVIDIS          = 17,
    CM_TWILIGHT         = 18,
    CM_TWILIGHT_SHIFTED = 19,
    CM_TURBO            = 20
};
```

Values

Name	Description
CM_AUTUMN	Red - orange - yellow
CM_BONE	Black - blueish gray - white
CM_JET	Dark blue - green - dark red
CM_WINTER	Blue - green
CM_RAINBOW	Red - green - purple
CM_OCEAN	Black - blue - white
CM_SUMMER	Green - yellow
CM_SPRING	Pink - yellow
CM_COOL	Cyan - magenta
CM_HSV	Red - green - blue - red
CM_PINK	Black - pink - light yellow - white
CM_HOT	Black - red - yellow - white
CM_PARULA	Blue - yellow
CM_MAGMA	Black - magenta - light yellow
CM_INFERNO	Black - magenta - yellow
CM_PLASMA	Blue - magenta - yellow
CM_VIRIDIS	Purple - yellow
CM_CIVIDIS	Blue - gray - yellow
CM_TWILIGHT	White - light blue - dark purple - light red - white
CM_TWILIGHT_SHIFTED	Dark purple - light blue - white - light red - dark purple
CM_TURBO	Dark blue - light green - dark red

6.2.1.4.13 ETcVnColorMapSize

Offers color map sizes.

Syntax

Definition:

```
enum ETcVnColorMapSize : ULONG
{
    CMS_256   = 256,
    CMS_65536 = 65536
};
```

Values

Name	Description
CMS_256	256 colors, used for 8 bit images (TCVN_ET_USINT)
CMS_65536	65536 colors, used for 16 bit images (TCVN_ET_UINT)

6.2.1.4.14 ETcVnColorSpaceTransform

Offers color space transformations.

Syntax

Definition:

```
enum ETcVnColorSpaceTransform : LONG
{
    CST_BGR_TO_BGRA                  = 0,
    CST_RGB_TO_RGBA                  = 0,
    CST_BGRA_TO_BGR                  = 1,
    CST_RGBA_TO_RGB                  = 1,
    CST_BGR_TO_RGBA                  = 2,
    CST_RGB_TO_BGRA                  = 2,
    CST_BGRA_TO_RGB                  = 3,
    CST_RGBA_TO_BGR                  = 3,
    CST_BGR_TO_RGB                   = 4,
    CST_RGB_TO_BGR                   = 4,
    CST_BGRA_TO_RGBA                 = 5,
    CST_RGBA_TO_BGRA                 = 5,
    CST_BGR_TO_GRAY                  = 6,
    CST_RGB_TO_GRAY                  = 7,
    CST_GRAY_TO_BGR                  = 8,
    CST_GRAY_TO_RGB                  = 8,
    CST_GRAY_TO_BGRA                 = 9,
    CST_GRAY_TO_RGBA                 = 9,
    CST_BGRA_TO_GRAY                 = 10,
    CST_RGBA_TO_GRAY                 = 11,
    CST_RGB_TO_BGR_565               = 12,
    CST_BGR_TO_BGR_565               = 13,
    CST_BGR_565_TO_RGB               = 14,
    CST_BGR_565_TO_BGR               = 15,
    CST_RGBA_TO_BGR_565              = 16,
    CST_BGRA_TO_BGR_565              = 17,
    CST_BGR_565_TO_RGBA              = 18,
    CST_BGR_565_TO_BGRA              = 19,
    CST_GRAY_TO_BGR_565              = 20,
    CST_BGR_565_TO_GRAY              = 21,
    CST_RGB_TO_BGR_555               = 22,
    CST_BGR_TO_BGR_555               = 23,
    CST_BGR_555_TO_RGB               = 24,
    CST_BGR_555_TO_BGR               = 25,
    CST_RGBA_TO_BGR_555              = 26,
    CST_BGRA_TO_BGR_555              = 27,
    CST_BGR_555_TO_RGBA              = 28,
    CST_BGR_555_TO_BGRA              = 29,
    CST_GRAY_TO_BGR_555              = 30,
    CST_BGR_555_TO_GRAY              = 31,
    CST_BGR_TO_XYZ                   = 32,
    CST_RGB_TO_XYZ                   = 33,
    CST_XYZ_TO_BGR                   = 34,
    CST_XYZ_TO_RGB                   = 35,
    CST_BGR_TO_YCRCB                 = 36,
    CST_RGB_TO_YCRCB                 = 37,
    CST_YCRCB_TO_BGR                 = 38,
    CST_YCRCB_TO_RGB                 = 39,
    CST_BGR_TO_HSV                   = 40,
    CST_RGB_TO_HSV                   = 41,
```

```
    CST_BGR_TO_LAB                    = 44,
    CST_RGB_TO_LAB                    = 45,
    CST_BAYER_RG_TO_BGR               = 46,
    CST_BAYER_BG_TO_RGB               = 46,
    CST_BAYER_GR_TO_BGR               = 47,
    CST_BAYER_GB_TO_RGB               = 47,
    CST_BAYER_BG_TO_BGR               = 48,
    CST_BAYER_RG_TO_RGB               = 48,
    CST_BAYER_GB_TO_BGR               = 49,
    CST_BAYER_GR_TO_RGB               = 49,
    CST_BGR_TO_LUV                    = 50,
    CST_RGB_TO_LUV                    = 51,
    CST_BGR_TO_HLS                    = 52,
    CST_RGB_TO_HLS                    = 53,
    CST_HSV_TO_BGR                    = 54,
    CST_HSV_TO_RGB                    = 55,
    CST_LAB_TO_BGR                    = 56,
    CST_LAB_TO_RGB                    = 57,
    CST_LUV_TO_BGR                    = 58,
    CST_LUV_TO_RGB                    = 59,
    CST_HLS_TO_BGR                    = 60,
    CST_HLS_TO_RGB                    = 61,
    CST_BAYER_RG_TO_BGR_VNG           = 62,
    CST_BAYER_GR_TO_BGR_VNG           = 63,
    CST_BAYER_BG_TO_BGR_VNG           = 64,
    CST_BAYER_GB_TO_BGR_VNG           = 65,
    CST_BAYER_RG_TO_RGB_VNG           = 64,
    CST_BAYER_GR_TO_RGB_VNG           = 65,
    CST_BAYER_BG_TO_RGB_VNG           = 62,
    CST_BAYER_GB_TO_RGB_VNG           = 63,
    CST_BGR_TO_HSV_FULL               = 66,
    CST_RGB_TO_HSV_FULL               = 67,
    CST_BGR_TO_HLS_FULL               = 68,
    CST_RGB_TO_HLS_FULL               = 69,
    CST_HSV_TO_BGR_FULL               = 70,
    CST_HSV_TO_RGB_FULL               = 71,
    CST_HLS_TO_BGR_FULL               = 72,
    CST_HLS_TO_RGB_FULL               = 73,
    CST_LBGR_TO_LAB                   = 74,
    CST_LRGB_TO_LAB                   = 75,
    CST_LBGR_TO_LUV                   = 76,
    CST_LRGB_TO_LUV                   = 77,
    CST_LAB_TO_LBGR                   = 78,
    CST_LAB_TO_LRGB                   = 79,
    CST_LUV_TO_LBGR                   = 80,
    CST_LUV_TO_LRGB                   = 81,
    CST_BGR_TO_YUV                    = 82,
    CST_RGB_TO_YUV                    = 83,
    CST_YUV_TO_BGR                    = 84,
    CST_YUV_TO_RGB                    = 85,
    CST_BAYER_RG_TO_GRAY              = 86,
    CST_BAYER_GR_TO_GRAY              = 87,
    CST_BAYER_BG_TO_GRAY              = 88,
    CST_BAYER_GB_TO_GRAY              = 89,
    CST_YUV_420_NV12_TO_RGB           = 90,
    CST_YUV_420_NV12_TO_BGR           = 91,
    CST_YUV_420_NV21_TO_RGB           = 92,
    CST_YUV_420_NV21_TO_BGR           = 93,
    CST_YUV_420_SP_TO_RGB             = 92,
    CST_YUV_420_SP_TO_BGR             = 93,
    CST_YUV_420_NV12_TO_RGBA          = 94,
    CST_YUV_420_NV12_TO_BGRA          = 95,
    CST_YUV_420_NV21_TO_RGBA          = 96,
    CST_YUV_420_NV21_TO_BGRA          = 97,
    CST_YUV_420_SP_TO_RGBA            = 96,
    CST_YUV_420_SP_TO_BGRA            = 97,
    CST_YUV_420_YV12_TO_RGB           = 98,
    CST_YUV_420_YV12_TO_BGR           = 99,
    CST_YUV_420_IYUV_TO_RGB           = 100,
    CST_YUV_420_IYUV_TO_BGR           = 101,
    CST_YUV_420_I420_TO_RGB           = 100,
    CST_YUV_420_I420_TO_BGR           = 101,
    CST_YUV_420_P_TO_RGB              = 98,
    CST_YUV_420_P_TO_BGR              = 99,
    CST_YUV_420_YV12_TO_RGBA          = 102,
    CST_YUV_420_YV12_TO_BGRA          = 103,
    CST_YUV_420_IYUV_TO_RGBA          = 104,
    CST_YUV_420_IYUV_TO_BGRA          = 105,
    CST_YUV_420_I420_TO_RGBA          = 104,
    CST_YUV_420_I420_TO_BGRA          = 105,
```

```
    CST_YUV_420_P_TO_RGBA              = 102,
    CST_YUV_420_P_TO_BGRA              = 103,
    CST_YUV_420_TO_GRAY                = 106,
    CST_YUV_420_NV21_TO_GRAY           = 106,
    CST_YUV_420_NV12_TO_GRAY           = 106,
    CST_YUV_420_YV12_TO_GRAY           = 106,
    CST_YUV_420_IYUV_TO_GRAY           = 106,
    CST_YUV_420_I420_TO_GRAY           = 106,
    CST_YUV_420_SP_TO_GRAY             = 106,
    CST_YUV_420_P_TO_GRAY              = 106,
    CST_YUV_422_UYVY_TO_RGB            = 107,
    CST_YUV_422_UYVY_TO_BGR            = 108,
    CST_YUV_422_Y422_TO_RGB            = 107,
    CST_YUV_422_Y422_TO_BGR            = 108,
    CST_YUV_422_UYNV_TO_RGB            = 107,
    CST_YUV_422_UYNV_TO_BGR            = 108,
    CST_YUV_422_UYVY_TO_RGBA           = 111,
    CST_YUV_422_UYVY_TO_BGRA           = 112,
    CST_YUV_422_Y422_TO_RGBA           = 111,
    CST_YUV_422_Y422_TO_BGRA           = 112,
    CST_YUV_422_UYNV_TO_RGBA           = 111,
    CST_YUV_422_UYNV_TO_BGRA           = 112,
    CST_YUV_422_YUY2_TO_RGB            = 115,
    CST_YUV_422_YUY2_TO_BGR            = 116,
    CST_YUV_422_YVYU_TO_RGB            = 117,
    CST_YUV_422_YVYU_TO_BGR            = 118,
    CST_YUV_422_YUYV_TO_RGB            = 115,
    CST_YUV_422_YUYV_TO_BGR            = 116,
    CST_YUV_422_YUNV_TO_RGB            = 115,
    CST_YUV_422_YUNV_TO_BGR            = 116,
    CST_YUV_422_YUY2_TO_RGBA           = 119,
    CST_YUV_422_YUY2_TO_BGRA           = 120,
    CST_YUV_422_YVYU_TO_RGBA           = 121,
    CST_YUV_422_YVYU_TO_BGRA           = 122,
    CST_YUV_422_YUYV_TO_RGBA           = 119,
    CST_YUV_422_YUYV_TO_BGRA           = 120,
    CST_YUV_422_YUNV_TO_RGBA           = 119,
    CST_YUV_422_YUNV_TO_BGRA           = 120,
    CST_YUV_422_UYVY_TO_GRAY           = 123,
    CST_YUV_422_YUY2_TO_GRAY           = 124,
    CST_YUV_422_Y422_TO_GRAY           = 123,
    CST_YUV_422_UYNV_TO_GRAY           = 123,
    CST_YUV_422_YVYU_TO_GRAY           = 124,
    CST_YUV_422_YUYV_TO_GRAY           = 124,
    CST_YUV_422_YUNV_TO_GRAY           = 124,
    CST_RGBA_TO_PREMULTIPLICATED_RGBA = 125,
    CST_PREMULTIPLICATED_RGBA_TO_RGBA = 126,
    CST_RGB_TO_YUV_420_I420            = 127,
    CST_BGR_TO_YUV_420_I420            = 128,
    CST_RGB_TO_YUV_420_IYUV            = 127,
    CST_BGR_TO_YUV_420_IYUV            = 128,
    CST_RGBA_TO_YUV_420_I420           = 129,
    CST_BGRA_TO_YUV_420_I420           = 130,
    CST_RGBA_TO_YUV_420_IYUV           = 129,
    CST_BGRA_TO_YUV_420_IYUV           = 130,
    CST_RGB_TO_YUV_420_YV12            = 131,
    CST_BGR_TO_YUV_420_YV12            = 132,
    CST_RGBA_TO_YUV_420_YV12           = 133,
    CST_BGRA_TO_YUV_420_YV12           = 134,
    CST_BAYER_RG_TO_BGR_EA             = 135,
    CST_BAYER_GR_TO_BGR_EA             = 136,
    CST_BAYER_BG_TO_BGR_EA             = 137,
    CST_BAYER_GB_TO_BGR_EA             = 138,
    CST_BAYER_RG_TO_RGB_EA             = 137,
    CST_BAYER_GR_TO_RGB_EA             = 138,
    CST_BAYER_BG_TO_RGB_EA             = 135,
    CST_BAYER_GB_TO_RGB_EA             = 136,
    CST_BAYER_RG_TO_BGRA               = 139,
    CST_BAYER_GR_TO_BGRA               = 140,
    CST_BAYER_BG_TO_BGRA               = 141,
    CST_BAYER_GB_TO_BGRA               = 142,
    CST_BAYER_RG_TO_RGBA               = 141,
    CST_BAYER_GR_TO_RGBA               = 142,
    CST_BAYER_BG_TO_RGBA               = 139,
    CST_BAYER_GB_TO_RGBA               = 140,
    CST_MAX                            = 143
};
```

Values

Name	Description
CST_BGR_TO_BGRA	Transform the color space from BGR to BGRA.
CST_RGB_TO_RGBA	Transform the color space from RGB to RGBA.
CST_BGRA_TO_BGR	Transform the color space from BGRA to BGR.
CST_RGBA_TO_RGB	Transform the color space from RGBA to RGB.
CST_BGR_TO_RGBA	Transform the color space from BGR to RGBA.
CST_RGB_TO_BGRA	Transform the color space from RGB to BGRA.
CST_BGRA_TO_RGB	Transform the color space from BGRA to RGB.
CST_RGBA_TO_BGR	Transform the color space from RGBA to BGR.
CST_BGR_TO_RGB	Transform the color space from BGR to RGB.
CST_RGB_TO_BGR	Transform the color space from RGB to BGR.
CST_BGRA_TO_RGBA	Transform the color space from BGRA to RGBA.
CST_RGBA_TO_BGRA	Transform the color space from RGBA to BGRA.
CST_BGR_TO_GRAY	Transform the color space from BGR to Gray.
CST_RGB_TO_GRAY	Transform the color space from RGB to Gray.
CST_GRAY_TO_BGR	Transform the color space from Gray to BGR.
CST_GRAY_TO_RGB	Transform the color space from Gray to RGB.
CST_GRAY_TO_BGRA	Transform the color space from Gray to BGRA.
CST_GRAY_TO_RGBA	Transform the color space from Gray to RGBA.
CST_BGRA_TO_GRAY	Transform the color space from BGRA to Gray.
CST_RGBA_TO_GRAY	Transform the color space from RGBA to Gray.
CST_RGB_TO_BGR_565	Transform the color space from RGB(24 bit 888) to BGR(16 bit 565).
CST_BGR_TO_BGR_565	Transform the color space from BGR(24 bit 888) to BGR(16 bit 565).
CST_BGR_565_TO_RGB	Transform the color space from BGR(16 bit 565) to RGB(24 bit 888).
CST_BGR_565_TO_BGR	Transform the color space from BGR(16 bit 565) to BGR(24 bit 888).
CST_RGBA_TO_BGR_565	Transform the color space from RGBA(32 bit 8888) to BGR(16 bit 565).
CST_BGRA_TO_BGR_565	Transform the color space from BGRA(32 bit 8888) to BGR(16 bit 565).
CST_BGR_565_TO_RGBA	Transform the color space from BGR(16 bit 565) to RGBA(32 bit 8888).
CST_BGR_565_TO_BGRA	Transform the color space from BGR(16 bit 565) to BGRA(32 bit 8888).
CST_GRAY_TO_BGR_565	Transform the color space from Gray to BGR(16 bit 565).
CST_BGR_565_TO_GRAY	Transform the color space from BGR(16 bit 565) to Gray.
CST_RGB_TO_BGR_555	Transform the color space from RGB(24 bit 888) to BGR(16 bit 555).
CST_BGR_TO_BGR_555	Transform the color space from BGR(24 bit 888) to BGR(16 bit 555).
CST_BGR_555_TO_RGB	Transform the color space from BGR(16 bit 555) to RGB(24 bit 888).
CST_BGR_555_TO_BGR	Transform the color space from BGR(16 bit 555) to BGR(24 bit 888).
CST_RGBA_TO_BGR_555	Transform the color space from RGBA(32 bit 8888) to BGR(16 bit 555).
CST_BGRA_TO_BGR_555	Transform the color space from BGRA(32 bit 8888) to BGR(16 bit 555).

Name	Description
CST_BGR_555_TO_RGBA	Transform the color space from BGR(16 bit 555) to RGBA(32 bit 8888).
CST_BGR_555_TO_BGRA	Transform the color space from BGR(16 bit 555) to BGRA(32 bit 8888).
CST_GRAY_TO_BGR_555	Transform the color space from Gray to BGR(16 bit 555).
CST_BGR_555_TO_GRAY	Transform the color space from BGR(16 bit 555) to Gray.
CST_BGR_TO_XYZ	Transform the color space from BGR to CIE XYZ (scaled to the full value range of the image).
CST_RGB_TO_XYZ	Transform the color space from RGB to CIE XYZ (scaled to the full value range of the image).
CST_XYZ_TO_BGR	Transform the color space from CIE XYZ (scaled to the full value range of the image) to BGR.
CST_XYZ_TO_RGB	Transform the color space from CIE XYZ (scaled to the full value range of the image) to RGB.
CST_BGR_TO_YCRCB	Transform the color space from BGR to YCrCb (scaled to the full value range of the image).
CST_RGB_TO_YCRCB	Transform the color space from RGB to YCrCb (scaled to the full value range of the image).
CST_YCRCB_TO_BGR	Transform the color space from YCrCb (scaled to the full value range of the image) to BGR.
CST_YCRCB_TO_RGB	Transform the color space from YCrCb (scaled to the full value range of the image) to RGB.
CST_BGR_TO_HSV	Transform the color space from BGR to HSV (for images of type USINT, H is scaled to a range from 0 to 180 and S,V from 0 to 255).
CST_RGB_TO_HSV	Transform the color space from RGB to HSV (for images of type USINT, H is scaled to a range from 0 to 180 and S,V from 0 to 255).
CST_BGR_TO_LAB	Transform the color space from BGR to CIE L*a*b* (for images of type USINT, all channels are scaled to a range from 0 to 255).
CST_RGB_TO_LAB	Transform the color space from RGB to CIE L*a*b* (for images of type USINT, all channels are scaled to a range from 0 to 255).
CST_BAYER_RG_TO_BGR	Transform the color space from BayerRG to BGR.
CST_BAYER_BG_TO_RGB	Transform the color space from BayerBG to RGB.
CST_BAYER_GR_TO_BGR	Transform the color space from BayerGR to BGR.
CST_BAYER_GB_TO_RGB	Transform the color space from BayerGB to RGB.
CST_BAYER_BG_TO_BGR	Transform the color space from BayerBG to BGR.
CST_BAYER_RG_TO_RGB	Transform the color space from BayerRG to RGB.
CST_BAYER_GB_TO_BGR	Transform the color space from BayerGB to BGR.
CST_BAYER_GR_TO_RGB	Transform the color space from BayerGR to RGB.
CST_BGR_TO_LUV	Transform the color space from BGR to CIE L*u*v* (for images of type USINT, all channels are scaled to a range from 0 to 255).
CST_RGB_TO_LUV	Transform the color space from RGB to CIE L*u*v* (for images of type USINT, all channels are scaled to a range from 0 to 255).
CST_BGR_TO_HLS	Transform the color space from BGR to HLS (for images of type USINT, H is scaled to a range from 0 to 180 and L,S from 0 to 255).

Name	Description
CST_RGB_TO_HLS	Transform the color space from RGB to HLS (for images of type USINT, H is scaled to a range from 0 to 180 and L,S from 0 to 255).
CST_HSV_TO_BGR	Transform the color space from HSV (for images of type USINT, H is scaled to a range from 0 to 180 and S,V from 0 to 255) to BGR.
CST_HSV_TO_RGB	Transform the color space from HSV (for images of type USINT, H is scaled to a range from 0 to 180 and S,V from 0 to 255) to RGB.
CST_LAB_TO_BGR	Transform the color space from CIE L*a*b* (for images of type USINT, all channels are scaled to a range from 0 to 255) to BGR.
CST_LAB_TO_RGB	Transform the color space from CIE L*a*b* (for images of type USINT, all channels are scaled to a range from 0 to 255) to RGB.
CST_LUV_TO_BGR	Transform the color space from CIE L*u*v* (for images of type USINT, all channels are scaled to a range from 0 to 255) to BGR.
CST_LUV_TO_RGB	Transform the color space from CIE L*u*v* (for images of type USINT, all channels are scaled to a range from 0 to 255) to RGB.
CST_HLS_TO_BGR	Transform the color space from HLS (for images of type USINT, H is scaled to a range from 0 to 180 and L,S from 0 to 255) to BGR.
CST_HLS_TO_RGB	Transform the color space from HLS (for images of type USINT, H is scaled to a range from 0 to 180 and L,S from 0 to 255) to RGB.
CST_BAYER_RG_TO_BGR_VNG	Transform the color space from BayerRG to BGR, using Variable Number of Gradients.
CST_BAYER_GR_TO_BGR_VNG	Transform the color space from BayerGR to BGR, using Variable Number of Gradients.
CST_BAYER_BG_TO_BGR_VNG	Transform the color space from BayerBG to BGR, using Variable Number of Gradients.
CST_BAYER_GB_TO_BGR_VNG	Transform the color space from BayerGB to BGR, using Variable Number of Gradients.
CST_BAYER_RG_TO_RGB_VNG	Transform the color space from BayerRG to RGB, using Variable Number of Gradients.
CST_BAYER_GR_TO_RGB_VNG	Transform the color space from BayerGR to RGB, using Variable Number of Gradients.
CST_BAYER_BG_TO_RGB_VNG	Transform the color space from BayerBG to RGB, using Variable Number of Gradients.
CST_BAYER_GB_TO_RGB_VNG	Transform the color space from BayerGB to RGB, using Variable Number of Gradients.
CST_BGR_TO_HSV_FULL	Transform the color space from BGR to HSV (for images of type USINT, all channels are scaled to a range from 0 to 255).
CST_RGB_TO_HSV_FULL	Transform the color space from RGB to HSV (for images of type USINT, all channels are scaled to a range from 0 to 255).
CST_BGR_TO_HLS_FULL	Transform the color space from BGR to HLS (for images of type USINT, all channels are scaled to a range from 0 to 255).
CST_RGB_TO_HLS_FULL	Transform the color space from RGB to HLS (for images of type USINT, all channels are scaled to a range from 0 to 255).

Name	Description
CST_HSV_TO_BGR_FULL	Transform the color space from HSV (for images of type USINT, all channels are scaled to a range from 0 to 255) to BGR.
CST_HSV_TO_RGB_FULL	Transform the color space from HSV (for images of type USINT, all channels are scaled to a range from 0 to 255) to RGB.
CST_HLS_TO_BGR_FULL	Transform the color space from HLS (for images of type USINT, all channels are scaled to a range from 0 to 255) to BGR.
CST_HLS_TO_RGB_FULL	Transform the color space from HLS (for images of type USINT, all channels are scaled to a range from 0 to 255) to RGB.
CST_LBGR_TO_LAB	Transform the color space from LBGR to CIE L*a*b* (for images of type USINT, all channels are scaled to a range from 0 to 255).
CST_LRGB_TO_LAB	Transform the color space from LRGB to CIE L*a*b* (for images of type USINT, all channels are scaled to a range from 0 to 255).
CST_LBGR_TO_LUV	Transform the color space from LBGR to CIE L*u*v* (for images of type USINT, all channels are scaled to a range from 0 to 255).
CST_LRGB_TO_LUV	Transform the color space from LRGB to CIE L*u*v* (for images of type USINT, all channels are scaled to a range from 0 to 255).
CST_LAB_TO_LBGR	Transform the color space from CIE L*a*b* (for images of type USINT, all channels are scaled to a range from 0 to 255) to LBGR.
CST_LAB_TO_LRGB	Transform the color space from CIE L*a*b* (for images of type USINT, all channels are scaled to a range from 0 to 255) to LRGB.
CST_LUV_TO_LBGR	Transform the color space from CIE L*u*v* (for images of type USINT, all channels are scaled to a range from 0 to 255) to LBGR.
CST_LUV_TO_LRGB	Transform the color space from CIE L*u*v* (for images of type USINT, all channels are scaled to a range from 0 to 255) to LRGB.
CST_BGR_TO_YUV	Transform the color space from BGR to YUV (scaled to the full value range of the image).
CST_RGB_TO_YUV	Transform the color space from RGB to YUV (scaled to the full value range of the image).
CST_YUV_TO_BGR	Transform the color space from YUV (scaled to the full value range of the image) to BGR.
CST_YUV_TO_RGB	Transform the color space from YUV (scaled to the full value range of the image) to RGB.
CST_BAYER_RG_TO_GRAY	Transform the color space from BayerRG to Gray.
CST_BAYER_GR_TO_GRAY	Transform the color space from BayerGR to Gray.
CST_BAYER_BG_TO_GRAY	Transform the color space from BayerBG to Gray.
CST_BAYER_GB_TO_GRAY	Transform the color space from BayerGB to Gray.
CST_YUV_420_NV12_TO_RGB	Transform the color space from YUV420 NV12 to RGB.
CST_YUV_420_NV12_TO_BGR	Transform the color space from YUV420 NV12 to BGR.
CST_YUV_420_NV21_TO_RGB	Transform the color space from YUV420 NV21 (SP) to RGB.
CST_YUV_420_NV21_TO_BGR	Transform the color space from YUV420 NV21 (SP) to BGR.

Name	Description
CST_YUV_420_SP_TO_RGB	Transform the color space from YUV420 NV21 (SP) to RGB.
CST_YUV_420_SP_TO_BGR	Transform the color space from YUV420 NV21 (SP) to BGR.
CST_YUV_420_NV12_TO_RGBA	Transform the color space from YUV420 NV12 to RGBA.
CST_YUV_420_NV12_TO_BGRA	Transform the color space from YUV420 NV12 to BGRA.
CST_YUV_420_NV21_TO_RGBA	Transform the color space from YUV420 NV21 (SP) to RGBA.
CST_YUV_420_NV21_TO_BGRA	Transform the color space from YUV420 NV21 (SP) to BGRA.
CST_YUV_420_SP_TO_RGBA	Transform the color space from YUV420 NV21 (SP) to RGBA.
CST_YUV_420_SP_TO_BGRA	Transform the color space from YUV420 NV21 (SP) to BGRA.
CST_YUV_420_YV12_TO_RGB	Transform the color space from YUV420 YV12 (P) to RGB.
CST_YUV_420_YV12_TO_BGR	Transform the color space from YUV420 YV12 (P) to BGR.
CST_YUV_420_IYUV_TO_RGB	Transform the color space from YUV420 IYUV (I420) to RGB.
CST_YUV_420_IYUV_TO_BGR	Transform the color space from YUV420 IYUV (I420) to BGR.
CST_YUV_420_I420_TO_RGB	Transform the color space from YUV420 IYUV (I420) to RGB.
CST_YUV_420_I420_TO_BGR	Transform the color space from YUV420 IYUV (I420) to BGR.
CST_YUV_420_P_TO_RGB	Transform the color space from YUV420 YV12 (P) to RGB.
CST_YUV_420_P_TO_BGR	Transform the color space from YUV420 YV12 (P) to BGR.
CST_YUV_420_YV12_TO_RGBA	Transform the color space from YUV420 YV12 (P) to RGBA.
CST_YUV_420_YV12_TO_BGRA	Transform the color space from YUV420 YV12 (P) to BGRA.
CST_YUV_420_IYUV_TO_RGBA	Transform the color space from YUV420 IYUV (I420) to RGBA.
CST_YUV_420_IYUV_TO_BGRA	Transform the color space from YUV420 IYUV (I420) to BGRA.
CST_YUV_420_I420_TO_RGBA	Transform the color space from YUV420 IYUV (I420) to RGBA.
CST_YUV_420_I420_TO_BGRA	Transform the color space from YUV420 IYUV (I420) to BGRA.
CST_YUV_420_P_TO_RGBA	Transform the color space from YUV420 YV12 (P) to RGBA.
CST_YUV_420_P_TO_BGRA	Transform the color space from YUV420 YV12 (P) to BGRA.
CST_YUV_420_TO_GRAY	Transform the color space from YUV420 to Gray.
CST_YUV_420_NV21_TO_GRAY	Transform the color space from YUV420 to Gray.
CST_YUV_420_NV12_TO_GRAY	Transform the color space from YUV420 to Gray.
CST_YUV_420_YV12_TO_GRAY	Transform the color space from YUV420 to Gray.
CST_YUV_420_IYUV_TO_GRAY	Transform the color space from YUV420 to Gray.
CST_YUV_420_I420_TO_GRAY	Transform the color space from YUV420 to Gray.

Name	Description
CST_YUV_420_SP_TO_GRAY	Transform the color space from YUV420 to Gray.
CST_YUV_420_P_TO_GRAY	Transform the color space from YUV420 to Gray.
CST_YUV_422_UYVY_TO_RGB	Transform the color space from YUV422 UYVY (Y422, UYNV) to RGB.
CST_YUV_422_UYVY_TO_BGR	Transform the color space from YUV422 UYVY (Y422, UYNV) to BGR.
CST_YUV_422_Y422_TO_RGB	Transform the color space from YUV422 UYVY (Y422, UYNV) to RGB.
CST_YUV_422_Y422_TO_BGR	Transform the color space from YUV422 UYVY (Y422, UYNV) to BGR.
CST_YUV_422_UYNV_TO_RGB	Transform the color space from YUV422 UYVY (Y422, UYNV) to RGB.
CST_YUV_422_UYNV_TO_BGR	Transform the color space from YUV422 UYVY (Y422, UYNV) to BGR.
CST_YUV_422_UYVY_TO_RGBA	Transform the color space from YUV422 UYVY (Y422, UYNV) to RGBA.
CST_YUV_422_UYVY_TO_BGRA	Transform the color space from YUV422 UYVY (Y422, UYNV) to BGRA.
CST_YUV_422_Y422_TO_RGBA	Transform the color space from YUV422 UYVY (Y422, UYNV) to RGB.
CST_YUV_422_Y422_TO_BGRA	Transform the color space from YUV422 UYVY (Y422, UYNV) to BGRA.
CST_YUV_422_UYNV_TO_RGBA	Transform the color space from YUV422 UYVY (Y422, UYNV) to RGB.
CST_YUV_422_UYNV_TO_BGRA	Transform the color space from YUV422 UYVY (Y422, UYNV) to BGRA.
CST_YUV_422_YUY2_TO_RGB	Transform the color space from YUV422 YUYV (YUY2, YUNV) to RGB.
CST_YUV_422_YUY2_TO_BGR	Transform the color space from YUV422 YUYV (YUY2, YUNV) to BGR.
CST_YUV_422_YVYU_TO_RGB	Transform the color space from YUV422 YVYU to RGB.
CST_YUV_422_YVYU_TO_BGR	Transform the color space from YUV422 YVYU to BGR.
CST_YUV_422_YUYV_TO_RGB	Transform the color space from YUV422 YUYV (YUY2, YUNV) to RGB.
CST_YUV_422_YUYV_TO_BGR	Transform the color space from YUV422 YUYV (YUY2, YUNV) to BGR.
CST_YUV_422_YUNV_TO_RGB	Transform the color space from YUV422 YUYV (YUY2, YUNV) to RGB.
CST_YUV_422_YUNV_TO_BGR	Transform the color space from YUV422 YUYV (YUY2, YUNV) to BGR.
CST_YUV_422_YUY2_TO_RGBA	Transform the color space from YUV422 YUYV (YUY2, YUNV) to RGBA.
CST_YUV_422_YUY2_TO_BGRA	Transform the color space from YUV422 YUYV (YUY2, YUNV) to BGRA.
CST_YUV_422_YVYU_TO_RGBA	Transform the color space from YUV422 YVYU to RGBA.
CST_YUV_422_YVYU_TO_BGRA	Transform the color space from YUV422 YVYU to BGRA.
CST_YUV_422_YUYV_TO_RGBA	Transform the color space from YUV422 YUYV (YUY2, YUNV) to RGBA.
CST_YUV_422_YUYV_TO_BGRA	Transform the color space from YUV422 YUYV (YUY2, YUNV) to BGRA.

Name	Description
CST_YUV_422_YUNV_TO_RGBA	Transform the color space from YUV422 YUYV (YUY2, YUNV) to RGBA.
CST_YUV_422_YUNV_TO_BGRA	Transform the color space from YUV422 YUYV (YUY2, YUNV) to BGRA.
CST_YUV_422_UYVY_TO_GRAY	Transform the color space from YUV422 UYVY (Y422, UYNV) to Gray.
CST_YUV_422_YUY2_TO_GRAY	Transform the color space from YUV422 YUYV (YUY2, YUNV) to Gray.
CST_YUV_422_Y422_TO_GRAY	Transform the color space from YUV422 UYVY (Y422, UYNV) to Gray.
CST_YUV_422_UYNV_TO_GRAY	Transform the color space from YUV422 UYVY (Y422, UYNV) to Gray.
CST_YUV_422_YVYU_TO_GRAY	Transform the color space from YUV422 YVYU to Gray.
CST_YUV_422_YUYV_TO_GRAY	Transform the color space from YUV422 YUYV (YUY2, YUNV) to Gray.
CST_YUV_422_YUNV_TO_GRAY	Transform the color space from YUV422 YUYV (YUY2, YUNV) to Gray.
CST_RGBA_TO_PREMULTIPLICATED_RGBA	Transform the color space from RGBA to premultiplicated RGBA.
CST_PREMULTIPLICATED_RGBA_TO_RGBA	Transform the color space from premultiplicated RGBA to RGBA.
CST_RGB_TO_YUV_420_I420	Transform the color space from RGB to YUV420 IYUV (I420).
CST_BGR_TO_YUV_420_I420	Transform the color space from BGR to YUV420 IYUV (I420).
CST_RGB_TO_YUV_420_IYUV	Transform the color space from RGB to YUV420 IYUV (I420).
CST_BGR_TO_YUV_420_IYUV	Transform the color space from BGR to YUV420 IYUV (I420).
CST_RGBA_TO_YUV_420_I420	Transform the color space from RGBA to YUV420 IYUV (I420).
CST_BGRA_TO_YUV_420_I420	Transform the color space from BGRA to YUV420 IYUV (I420).
CST_RGBA_TO_YUV_420_IYUV	Transform the color space from RGBA to YUV420 IYUV (I420).
CST_BGRA_TO_YUV_420_IYUV	Transform the color space from BGRA to YUV420 IYUV (I420).
CST_RGB_TO_YUV_420_YV12	Transform the color space from RGB to YUV420 YV12 (P).
CST_BGR_TO_YUV_420_YV12	Transform the color space from BGR to YUV420 YV12 (P).
CST_RGBA_TO_YUV_420_YV12	Transform the color space from RGBA to YUV420 YV12 (P).
CST_BGRA_TO_YUV_420_YV12	Transform the color space from BGRA to YUV420 YV12 (P).
CST_BAYER_RG_TO_BGR_EA	Transform the color space from BayerRG to BGR, using an Edge Aware algorithm.
CST_BAYER_GR_TO_BGR_EA	Transform the color space from BayerGR to BGR, using an Edge Aware algorithm.
CST_BAYER_BG_TO_BGR_EA	Transform the color space from BayerBG to BGR, using an Edge Aware algorithm.
CST_BAYER_GB_TO_BGR_EA	Transform the color space from BayerGB to BGR, using an Edge Aware algorithm.

Name	Description
CST_BAYER_RG_TO_RGB_EA	Transform the color space from BayerRG to RGB, using an Edge Aware algorithm.
CST_BAYER_GR_TO_RGB_EA	Transform the color space from BayerGR to RGB, using an Edge Aware algorithm.
CST_BAYER_BG_TO_RGB_EA	Transform the color space from BayerBG to RGB, using an Edge Aware algorithm.
CST_BAYER_GB_TO_RGB_EA	Transform the color space from BayerGB to RGB, using an Edge Aware algorithm.
CST_BAYER_RG_TO_BGRA	Transform the color space from BayerRG to BGRA.
CST_BAYER_GR_TO_BGRA	Transform the color space from BayerGR to BGRA.
CST_BAYER_BG_TO_BGRA	Transform the color space from BayerBG to BGRA.
CST_BAYER_GB_TO_BGRA	Transform the color space from BayerGB to BGRA.
CST_BAYER_RG_TO_RGBA	Transform the color space from BayerRG to RGBA.
CST_BAYER_GR_TO_RGBA	Transform the color space from BayerGR to RGBA.
CST_BAYER_BG_TO_RGBA	Transform the color space from BayerBG to RGBA.
CST_BAYER_GB_TO_RGBA	Transform the color space from BayerGB to RGBA.
CST_MAX	For internal use only, adapted when adding new values.

6.2.1.4.15 ETcVnColorTrainingMethod

Offers color training methods.

Syntax

Definition:

```
enum ETcVnColorTrainingMethod : LONG
{
    CTM_LAB = 0,
    CTM_RGB = 1
};
```

Values

Name	Description
CTM_LAB	Trains the color in CIE L*a*b* color space.
CTM_RGB	Trains the color in RGB color space.

6.2.1.4.16 ETcVnConnectedComponentsAlgorithm

Offers connected components algorithms.

Syntax

Definition:

```
enum ETcVnConnectedComponentsAlgorithm : LONG
{
    CCA_WU    = 0,
    CCA_GRANA = 1
};
```

Values

Name	Description
CCA_WU	SAUF algorithm
CCA_GRANA	BBDT algorithm for 8-way connectivity, SAUF algorithm for 4-way connectivity

6.2.1.4.17 ETcVnContainerExportFormat

Offers container export formats.

Syntax

Definition:

```
enum ETcVnContainerExportFormat : USHORT
{
    CEF_XML             = 0,
    CEF_XML_SERIALIZED = 1,
    CEF_CSV             = 2,
    CEF_BINARY          = 3
};
```

Values

Name	Description
CEF_XML	XML format with human readable data
CEF_XML_SERIALIZED	XML format with serialized data
CEF_CSV	CSV format (limited to two dimensional representation, so not applicable for all container types)
CEF_BINARY	Binary serialized data (recommended for large containers)

6.2.1.4.18 ETcVnContourApproximationMethod

Offers methods for contour approximation.

Syntax

Definition:

```
enum ETcVnContourApproximationMethod : LONG
{
    CAM_NONE      = 1,
    CAM_SIMPLE    = 2,
    CAM_TC89_L1   = 3,
    CAM_TC89_KCOS = 4
};
```

Values

Name	Description
CAM_NONE	No approximation, every single point is stored, i.e. two subsequent points are always direct neighbors (horizontal, vertical or diagonal).
CAM_SIMPLE	Lossless compression of segments that form straight lines in horizontal, vertical or diagonal direction leaving only their endpoints (For instance, an up-right rectangle is reduced to its four corner points.)
CAM_TC89_L1	Compression using a Teh-Chin chain approximation algorithm (IEEE-paper C. Teh, R. Chin, On the Detection of Dominant Points on Digital Curves, 1989).
CAM_TC89_KCOS	Compression using a Teh-Chin chain approximation algorithm (IEEE-paper C. Teh, R. Chin, On the Detection of Dominant Points on Digital Curves, 1989).

6.2.1.4.19 ETcVnContourRetrievalMode

Offers retrieval modes for a contour search.

Syntax

Definition:

```
enum ETcVnContourRetrievalMode : LONG
{
    CRM_EXTERNAL               = 0,
    CRM_LIST                   = 1,
    CRM_CONNECTED_COMPONENTS = 2,
    CRM_TREE                   = 3,
    CRM_FLOODFILL              = 4
};
```

Values

Name	Description
CRM_EXTERNAL	Returns only external contours.
CRM_LIST	Returns all found contours, not considering their hierarchy.
CRM_CONNECTED_COMPONENTS	Returns all contours with a 2-level hierarchy (External contours are assigned level 0, internal contours level 1).
CRM_TREE	Returns all contours and their full hierarchy.
CRM_FLOODFILL	Returns the found contours using a floodfill algorithm (only available for DINT images).

6.2.1.4.20 ETcVnContoursMatchComparisonMethod

Offers comparison methods for contour matching.

Syntax

Definition:

```
enum ETcVnContoursMatchComparisonMethod : LONG
{
    CMCM_CONTOURS_MATCH_I1 = 1,
    CMCM_CONTOURS_MATCH_I2 = 2,
    CMCM_CONTOURS_MATCH_I3 = 3
};
```

Values

Name	Description
CMCM_CONTOURS_MATCH_I1	Contours are compared using the sum over the differences between the reciprocal individual characteristics.
CMCM_CONTOURS_MATCH_I2	Contours are compared using the sum over the differences between the individual characteristics.
CMCM_CONTOURS_MATCH_I3	Contours are compared using only the maximum difference between the individual characteristics.

6.2.1.4.21 ETcVnDiffusivityTypeKAZE

Offers diffusivity types for feature detection methods KAZE and AKAZE.

Syntax

Definition:

```
enum ETcVnDiffusivityTypeKAZE : LONG
{
    DT1_KAZE_PM_G1        = 0,
    DT1_KAZE_PM_G2        = 1,
    DT1_KAZE_WEICKERT     = 2,
    DT1_KAZE_CHARBONNIER = 3
};
```

Values

Name	Description
DT1_KAZE_PM_G1	Uses the Perona-Malik diffusivity filter g1.
DT1_KAZE_PM_G2	Uses the Perona-Malik diffusivity filter g2.
DT1_KAZE_WEICKERT	Uses the Weickert diffusivity filter.
DT1_KAZE_CHARBONNIER	Uses the Charbonnier diffusivity filter.

6.2.1.4.22 ETcVnDistanceTransformationLabel

Offers types of the label array to build.

Syntax

Definition:

```
enum ETcVnDistanceTransformationLabel : LONG
{
    DTL_CCOMP = 0,
    DTL_PIXEL = 1
};
```

Values

Name	Description
DTL_CCOMP	Labels connected components (Each connected component of zeros in the source image and the pixels closest to the connected component will be assigned the same label).
DTL_PIXEL	Labels pixels (Each zero pixel and the non-zero pixels closest to it get their own label).

6.2.1.4.23 ETcVnDistanceTransformationMask

Offers different sizes of the distance transformation mask. Please note, that some sizes are not supported by certain distance types.

Syntax

Definition:

```
enum ETcVnDistanceTransformationMask : LONG
{
    DTM_PRECISE = 0,
    DTM_3       = 3,
    DTM_5       = 5
};
```

Values

Name	Description
DTM_PRECISE	Uses a precise mask.
DTM_3	Uses a 3x3 mask.
DTM_5	Uses a 5x5 mask.

6.2.1.4.24 ETcVnDistanceType

Offers distance types.

Syntax

Definition:

```
enum ETcVnDistanceType : LONG
{
    DT_USER   = -1,
    DT_L1     = 1,
    DT_L2     = 2,
    DT_C      = 3,
    DT_L12    = 4,
    DT_FAIR   = 5,
    DT_WELSCH = 6,
    DT_HUBER  = 7
};
```

Values

Name	Description
DT_USER	User defined distance
DT_L1	|x1 - x2| + |y1 - y2|
DT_L2	Euclidean distance
DT_C	max(|x1 - x2|, |y1 - y2|)
DT_L12	L1 - L2 : 2*(sqrt(1 + x*x/2) - 1))
DT_FAIR	c^2(|x|/c-log(1+|x|/c)), c = 1.3998
DT_WELSCH	c^2/2(1-exp(-(x/c)^2)), c = 2.9846
DT_HUBER	|x|<c ? x^2/2 : c(|x|-c/2), c=1.345

6.2.1.4.25 ETcVnDrawMatchesFlags

Offers a combination of flags to support overdrawing an existing destination image and/or skipping single (i.e. non-matched) keypoints and/or drawing additional (rich-)keypoint information (size and orientation). Used by F_VN_DrawKeypointsExp.

Syntax

Definition:

```
enum ETcVnDrawMatchesFlags : LONG
{
    DMF_DEFAULT                           = 0,
    DMF_OVERDRAW                          = 1,
    DMF_SKIPSINGLE                        = 2,
    DMF_OVERDRAW_SKIPSINGLE               = 3,
    DMF_RICHKEYPOINT                      = 4,
    DMF_OVERDRAW_RICHKEYPOINT             = 5,
    DMF_SKIPSINGLE_RICHKEYPOINT           = 6,
    DMF_OVERDRAW_SKIPSINGLE_RICHKEYPOINT  = 7
};
```

Values

Name	Description
DMF_DEFAULT	Draw all keypoints into a new image.
DMF_OVERDRAW	Draw all keypoints into the existing destination image.
DMF_SKIPSINGLE	Draw the keypoints into a new image but skip single keypoints.
DMF_OVERDRAW_SKIPSINGLE	Draw the keypoints into the existing destination image but skip single keypoints.
DMF_RICHKEYPOINT	Draw all keypoints with rich information into a new image.
DMF_OVERDRAW_RICHKEYPOINT	Draw all keypoints with rich information into the existing destination image.
DMF_SKIPSINGLE_RICHKEYPOINT	Draw the keypoints with rich information into a new image but skip single keypoints.
DMF_OVERDRAW_SKIPSINGLE_RICHKEYPOINT	Draw the keypoints with rich information into the existing destination image but skip single keypoints.

6.2.1.4.26 ETcVnDrawShape

Offers shapes to be drawn.

Syntax

Definition:

```
enum ETcVnDrawShape : LONG
{
    DS_RANDOM  = -1,
    DS_CIRCLE  = 0,
    DS_SQUARE  = 1,
    DS_PLUS    = 2,
    DS_X       = 3,
    DS_DIAMOND = 4,
    DS_MAX     = 5
};
```

Values

Name	Description
DS_RANDOM	Randomly select a circle, square, plus, X or diamond.
DS_CIRCLE	Circle
DS_SQUARE	Square
DS_PLUS	+
DS_X	x
DS_DIAMOND	Diamond
DS_MAX	For internal use only, adapted when adding new values.

6.2.1.4.27 ETcVnEdgeDetectionAlgorithm

Offers edge detection algorithms.

Syntax

Definition:

```
enum ETcVnEdgeDetectionAlgorithm : LONG
{
    EDA_INTERPOLATION   = 0,
    EDA_APPROX_ERF      = 1,
    EDA_APPROX_GAUSSIAN = 2
};
```

Values

Name	Description
EDA_INTERPOLATION	Interpolates pixels (bilinear) and then finds the maximum gradient. This approach is fast and stable, but usually less precise than the function approximation methods.
EDA_APPROX_ERF	Approximates the edge with an erf function. This approach is slower than the interpolation method, but usually more precise. However, it can be inaccurate if the edge does not suit the erf model.
EDA_APPROX_GAUSSIAN	Approximates the edge with a gaussian function. This method is intended to find the center of relatively thin lines, so it is likely to be inaccurate for other edges.

6.2.1.4.28 ETcVnEdgeDirection

Offers edge directions relative to the search direction.

Syntax

Definition:

```
enum ETcVnEdgeDirection : LONG
{
    ED_DARK_TO_LIGHT = 0,
    ED_LIGHT_TO_DARK = 1
};
```

Values

Name	Description
ED_DARK_TO_LIGHT	Dark to light
ED_LIGHT_TO_DARK	Light to dark

6.2.1.4.29 ETcVnElementType

Offers element types.

Syntax

Definition:

```
enum ETcVnElementType : LONG
{
    ET_SAME_AS_SOURCE = -1,
    ET_USINT          = 0,
    ET_SINT           = 1,
    ET_UINT           = 2,
    ET_INT            = 3,
    ET_DINT           = 4,
    ET_REAL           = 5,
    ET_LREAL          = 6
};
```

Values

Name	Description
ET_SAME_AS_SOURCE	Sets the element type of the destination image to the source image element type.
ET_USINT	USINT (depth: 8 bit, 0..255)
ET_SINT	SINT (depth: 8 bit, -128..127)
ET_UINT	UINT (depth: 16 bit, 0..65535)
ET_INT	INT (depth: 16 bit, -32768..32767)
ET_DINT	DINT (depth: 32 bit, -2147483648..2147483647)
ET_REAL	REAL (depth: 32 bit, ~ -3.402823E 10^38 .. ~ 3.402823E 38)
ET_LREAL	LREAL (depth: 64 bit, ~ -1.79769313486231E 308 .. ~ 1.79769313486232E 308)

6.2.1.4.30 ETcVnEstimationAlgorithm

Offers estimation algorithms for matching point sets.

Syntax

Definition:

```
enum ETcVnEstimationAlgorithm : LONG
{
    EA_DEFAULT = 0,
    EA_LMEDS   = 4,
    EA_RANSAC  = 8,
    EA_RHO     = 16
};
```

Values

Name	Description
EA_DEFAULT	Use all points.
EA_LMEDS	Least Median of Squares
EA_RANSAC	Random Sample Consensus
EA_RHO	Progressive Sample Consensus

6.2.1.4.31 ETcVnExtremePointDirection

Offers search directions for the extreme point.

Syntax

Definition:

```
enum ETcVnExtremePointDirection : LONG
{
    EPD_TOP_LEFT      = 0,
    EPD_TOP_MEDIAN    = 1,
    EPD_TOP_RIGHT     = 2,
    EPD_BOTTOM_LEFT   = 3,
    EPD_BOTTOM_MEDIAN = 4,
    EPD_BOTTOM_RIGHT  = 5,
    EPD_LEFT_TOP      = 6,
    EPD_LEFT_MEDIAN   = 7,
    EPD_LEFT_BOTTOM   = 8,
    EPD_RIGHT_TOP     = 9,
    EPD_RIGHT_MEDIAN  = 10,
    EPD_RIGHT_BOTTOM  = 11
};
```

Values

Name	Description
EPD_TOP_LEFT	Find the topmost point (min y, take left one if more than 1).
EPD_TOP_MEDIAN	Find the topmost point (min y, take median one if more than 1).
EPD_TOP_RIGHT	Find the topmost point (min y, take right one if more than 1).
EPD_BOTTOM_LEFT	Find the bottommost point (max y, take left one if more than 1).
EPD_BOTTOM_MEDIAN	Find the bottommost point (max y, take median one if more than 1).
EPD_BOTTOM_RIGHT	Find the bottommost point (max y, take right one if more than 1).
EPD_LEFT_TOP	Find the leftmost point (min x, take top one if more than 1).
EPD_LEFT_MEDIAN	Find the leftmost point (min x, take median one if more than 1).
EPD_LEFT_BOTTOM	Find the leftmost point (min x, take bottom one if more than 1).
EPD_RIGHT_TOP	Find the rightmost point (max x, take top one if more than 1).
EPD_RIGHT_MEDIAN	Find the rightmost point (max x, take median one if more than 1).
EPD_RIGHT_BOTTOM	Find the rightmost point (max x, take bottom one if more than 1).

6.2.1.4.32 ETcVnFeatureDescriptorTypeAKAZE

Offers descriptor types for AKAZE method.

Syntax

Definition:

```
enum ETcVnFeatureDescriptorTypeAKAZE : LONG
{
    FDT_AKAZE_KAZE_UPRIGHT = 2,
    FDT_AKAZE_KAZE         = 3,
    FDT_AKAZE_MLDB_UPRIGHT = 4,
    FDT_AKAZE_MLDB         = 5
};
```

Values

Name	Description
FDT_AKAZE_KAZE_UPRIGHT	KAZE descriptor (upright, i.e. rotation dependend).
FDT_AKAZE_KAZE	KAZE descriptor (rotation invariant).
FDT_AKAZE_MLDB_UPRIGHT	MLDB descriptor (upright, i.e. rotation dependend).
FDT_AKAZE_MLDB	MLDB descriptor (rotation invariant).

6.2.1.4.33 ETcVnFeatureScalingType

Offers feature scaling types

Syntax

Definition:

```
enum ETcVnFeatureScalingType : LONG
{
    FST1_MAXABS          = 0,
    FST1_MINMAX          = 1,
    FST1_STANDARDIZATION = 2
};
```

Values

Name	Description
FST1_MAXABS	Use the max of the absolute values of the feature vector. The normlized values are in the range [-1 to 1] or [0 to 1] if all the original values are positive.
FST1_MINMAX	Use the min and max values of the feature vector. The normlized values are in the range [0 to 1]
FST1_STANDARDIZATION	Use the mean and the standard deviation of the feature vector. The normalized vector has a mean value equal to zero and standard deviation equal to one

6.2.1.4.34 ETcVnFeatureScoreTypeORB

Offers algorithms used to rank features.

Syntax

Definition:

```
enum ETcVnFeatureScoreTypeORB : LONG
{
    FST_ORB_HARRIS = 0,
    FST_ORB_FAST   = 1
};
```

Values

Name	Description
FST_ORB_HARRIS	Harris algorithm (produces more stable keypoints, but computation takes longer).
FST_ORB_FAST	FAST algorithm (produces slightly less stable keypoints, but computation is faster).

6.2.1.4.35 ETcVnFilterDirection

Offers directions, in which to apply filter.

Syntax

Definition:

```
enum ETcVnFilterDirection : SHORT
{
    FD_X = 0,
    FD_Y = 1
};
```

Values

Name	Description
FD_X	X-direction
FD_Y	Y-direction

6.2.1.4.36 ETcVnFlipAxis

Defines the axis around which to flip (mirror) the image.

Syntax

Definition:

```
enum ETcVnFlipAxis : SHORT
{
    FA_XY = -1,
    FA_X  = 0,
    FA_Y  = 1
};
```

Values

Name	Description
FA_XY	Flip the image around both axes
FA_X	Flip the image around the X axis
FA_Y	Flip the image around the Y axis

6.2.1.4.37 ETcVnFontType

Offers font types.

Syntax

Definition:

```
enum ETcVnFontType : LONG
{
    FT_HERSHEY_SIMPLEX              = 0,
    FT_HERSHEY_PLAIN                = 1,
    FT_HERSHEY_DUPLEX               = 2,
    FT_HERSHEY_COMPLEX              = 3,
    FT_HERSHEY_TRIPLEX              = 4,
    FT_HERSHEY_COMPLEX_SMALL        = 5,
    FT_HERSHEY_SCRIPT_SIMPLEX       = 6,
    FT_HERSHEY_SCRIPT_COMPLEX       = 7,
    FT_HERSHEY_PLAIN_ITALIC         = 17,
    FT_HERSHEY_COMPLEX_ITALIC       = 19,
    FT_HERSHEY_TRIPLEX_ITALIC       = 20,
    FT_HERSHEY_COMPLEX_SMALL_ITALIC = 21
};
```

Values

Name	Description
FT_HERSHEY_SIMPLEX	Normal size sans-serif font.
FT_HERSHEY_PLAIN	Small size sans-serif font.
FT_HERSHEY_DUPLEX	More complex normal size sans-serif font.
FT_HERSHEY_COMPLEX	Normal size serif font.
FT_HERSHEY_TRIPLEX	More complex normal size serif font.
FT_HERSHEY_COMPLEX_SMALL	Small size serif font.
FT_HERSHEY_SCRIPT_SIMPLEX	Hand-writing style font.
FT_HERSHEY_SCRIPT_COMPLEX	More complex hand-writing style font.
FT_HERSHEY_PLAIN_ITALIC	Small size sans-serif font (italic).
FT_HERSHEY_COMPLEX_ITALIC	Normal size serif font (italic).
FT_HERSHEY_TRIPLEX_ITALIC	More complex normal size serif font (italic).
FT_HERSHEY_COMPLEX_SMALL_ITALIC	Small size serif font (italic).

6.2.1.4.38 ETcVnHoughMethod

Offers Hough methods.

Syntax

Definition:

```
enum ETcVnHoughMethod : LONG
{
    HM_STANDARD      = 0,
    HM_PROBABILISTIC = 1,
    HM_MULTI_SCALE   = 2,
    HM_GRADIENT      = 3,
    HM_GRADIENT_ALT  = 4
};
```

Values

Name	Description
HM_STANDARD	Classical or standard Hough transform. Every line is represented by two floating-point numbers (f1, f2), where f1 is a distance between (0,0) origin and the line, and f2 is the angle between x-axis and the normal to the line.
HM_PROBABILISTIC	Probabilistic Hough transform. More efficient for pictures containing long linear segments. Returns line segments rather than the whole line, while each segment is represented by start and end point.
HM_MULTI_SCALE	Multi scale variant of classical Hough transform.
HM_GRADIENT	Method designed to find circles. See paper: HK Yuen, John Princen, John Illingworth, and Josef Kittler. Comparative study of hough transform methods for circle finding. Image and Vision Computing, 8(1):71 - 77, 1990
HM_GRADIENT_ALT	Variation of the GRADIENT method.

6.2.1.4.39 ETcVnInitializableFunction

Offers initializable functions.

Syntax

Definition:

```
enum ETcVnInitializableFunction : LONG
{
    IF_OCR = 0,
};
```

Values

Name	Description
IF_OCR	F_VN_OCR

6.2.1.4.40 ETcVnInterpolationType

Offers interpolation types.

Syntax

Definition:

```
enum ETcVnInterpolationType : LONG
{
    IT_NEAREST_NEIGHBOR = 0,
    IT_BILINEAR         = 1,
    IT_BICUBIC          = 2,
    IT_AREA_BASED       = 3,
    IT_LANCZOS4         = 4
};
```

Values

Name	Description
IT_NEAREST_NEIGHBOR	Nearest neighbor interpolation
IT_BILINEAR	Bilinear interpolation
IT_BICUBIC	Bicubic interpolation
IT_AREA_BASED	Area based interpolation
IT_LANCZOS4	Lanczos4 interpolation

6.2.1.4.41 ETcVnKeypointDetectionTypeAGAST

Offers different neighborhood types for AGAST method (For details see paper: E. Mair et al.: Adaptive and Generic Corner Detection Based on the Accelerated Segment Test, 2010).

Syntax

Definition:

```
enum ETcVnKeypointDetectionTypeAGAST : LONG
{
    KDT_AGAST_5_8   = 0,
    KDT_AGAST_7_12d = 1,
    KDT_AGAST_7_12s = 2,
    KDT_AGAST_9_16  = 3
};
```

Values

Name	Description
KDT_AGAST_5_8	5 of 8
KDT_AGAST_7_12d	7 of 12 diamond
KDT_AGAST_7_12s	7 of 12 square
KDT_AGAST_9_16	9 of 16

6.2.1.4.42 ETcVnKeypointDetectionTypeFAST

Offers different neighborhood types for FAST method (For details see paper: E. Rosten: Machine Learning for High-speed Corner Detection, 2006).

Syntax

Definition:

```
enum ETcVnKeypointDetectionTypeFAST : LONG
{
    KDT_FAST_5_8  = 0,
    KDT_FAST_7_12 = 1,
    KDT_FAST_9_16 = 2
};
```

Values

Name	Description
KDT_FAST_5_8	5 of 8
KDT_FAST_7_12	7 of 12
KDT_FAST_9_16	9 of 16

6.2.1.4.43 ETcVnKnn

Offers kNN model types.

Syntax

Definition:

```
enum ETcVnKnn : ULONG
{
    KNN_CLASSIFIER       = 1,
    KNN_NOVELTY_DETECTOR = 2,
    KNN_REGRESSOR        = 4
};
```

Values

Name	Description
KNN_CLASSIFIER	Classifier
KNN_NOVELTY_DETECTOR	Novelty detector
KNN_REGRESSOR	Regressor

6.2.1.4.44 ETcVnLineType

Offers line types.

Syntax

Definition:

```
enum ETcVnLineType : LONG
{
    LT_4_CONNECTED = 4,
    LT_8_CONNECTED = 8,
    LT_ANTIALIASED = 16
};
```

Values

Name	Description
LT_4_CONNECTED	4-connected (pixels are connected horizontally and vertically).
LT_8_CONNECTED	8-connected (pixels are connected horizontally, vertically and diagonally).
LT_ANTIALIASED	Antialiased (drawn using Gaussian filtering, only implemented for 8-bit images).

6.2.1.4.45 ETcVnMorphologicalOperator

Offers morphological operators.

Syntax

Definition:

```
enum ETcVnMorphologicalOperator : LONG
{
    MO_EROSION                       = 0x0,
    MO_DILATION                      = 0x1,
    MO_OPENING                       = 0x2,
    MO_CLOSING                       = 0x3,
    MO_GRADIENT                      = 0x4,
    MO_WHITE_TOPHAT                  = 0x5,
    MO_BLACK_TOPHAT                  = 0x6,
    MO_OPENING_BY_RECONSTRUCTION     = 0x40000002,
    MO_CLOSING_BY_RECONSTRUCTION     = 0x40000003,
    MO_WHITE_TOPHAT_BY_RECONSTRUCTION = 0x40000005,
    MO_BLACK_TOPHAT_BY_RECONSTRUCTION = 0x40000006
};
```

Values

Name	Description
MO_EROSION	Shrinks objects (regions of foreground (i.e. white) pixels), removing regions smaller than the structuring element completely.
MO_DILATION	Expands objects and thereby closes small holes inside objects.
MO_OPENING	Applies an erosion first, a dilation second. Objects smaller than the structuring element are removed while outer shapes remain largely the same.
MO_CLOSING	Applies a dilation first, an erosion second. Holes inside objects that fit into the structuring element are closed completely while outer shapes remain largely the same.
MO_GRADIENT	Difference between the dilation and the erosion of an image.
MO_WHITE_TOPHAT	Difference between an input image and its opening.
MO_BLACK_TOPHAT	Difference between an input image and its closing.
MO_OPENING_BY_RECONSTRUCTION	Opening with subsequent reconstruction of objects that were not removed by the opening.
MO_CLOSING_BY_RECONSTRUCTION	Closing with subsequent reconstruction of objects that were not removed by the closing.
MO_WHITE_TOPHAT_BY_RECONSTRUCTION	White tophat with subsequent reconstruction of objects that were not removed by the white tophat.
MO_BLACK_TOPHAT_BY_RECONSTRUCTION	Black tophat with subsequent reconstruction of objects that were not removed by the black tophat.

6.2.1.4.46 ETcVnNbc

Offers normal Bayes classifier types.

Syntax

Definition:

```
enum ETcVnNbc : ULONG
{
    NBC_CLASSIFIER       = 1,
    NBC_NOVELTY_DETECTOR = 2
};
```

Values

Name	Description
NBC_CLASSIFIER	Classifier
NBC_NOVELTY_DETECTOR	Novelty detector

6.2.1.4.47 ETcVnNormalizationType

Offers normalization types.

Syntax

Definition:

```
enum ETcVnNormalizationType : LONG
{
    NT_INF                  = 1,
    NT_L1                   = 2,
    NT_L2                   = 4,
    NT_L2SQR                = 5,
    NT_HAMMING              = 6,
    NT_HAMMING2             = 7,
    NT_RELATIVE_INF         = 9,
    NT_RELATIVE_L1          = 10,
    NT_RELATIVE_L2          = 12,
    NT_RELATIVE_L2SQR       = 13,
    NT_RELATIVE_HAMMING     = 14,
    NT_RELATIVE_HAMMING2    = 15,
    NT_MINMAX               = 32
};
```

Values

Name	Description
NT_INF	Infinity norm
NT_L1	L1 norm
NT_L2	L2 (euclidean) norm
NT_L2SQR	Squared L2 norm
NT_HAMMING	Hamming distance (bitwise)
NT_HAMMING2	Hamming distance (each 2 bits combined to one)
NT_RELATIVE_INF	Infinity norm (when comparing 2 arrays, the norm of their difference is divided by the norm of the second array)
NT_RELATIVE_L1	L1 norm (when comparing 2 arrays, the norm of their difference is divided by the norm of the second array)
NT_RELATIVE_L2	L2 (euclidean) norm (when comparing 2 arrays, the norm of their difference is divided by the norm of the second array)
NT_RELATIVE_L2SQR	Squared L2 norm (when comparing 2 arrays, the norm of their difference is divided by the norm of the second array)
NT_RELATIVE_HAMMING	Hamming distance (bitwise; when comparing 2 arrays, the norm of their difference is divided by the norm of the second array)
NT_RELATIVE_HAMMING2	Hamming distance (each 2 bits combined to one; when comparing 2 arrays, the norm of their difference is divided by the norm of the second array)
NT_MINMAX	Normalize the values to a range given by a minimum and a maximum value.

6.2.1.4.48 ETcVnOcrModelType

Offers different OCR model types

Syntax

Definition:

```
enum ETcVnOcrModelType : ULONGLONG
{
    OMT_NUMBERS                = 580542139465730,
    OMT_NUMBERS_SC             = 580542139465732,
    OMT_UCLETTERS              = 580542139465736,
    OMT_NUMBERS_SC_UCLETTERS = 580542139465744
};
```

Values

Name	Description
OMT_NUMBERS	Classify numbers
OMT_NUMBERS_SC	Classify numbers and special characters
OMT_UCLETTERS	Classify uppercase letters
OMT_NUMBERS_SC_UCLETTERS	Classify numbers, special characters and uppercase letters

6.2.1.4.49 ETcVnOcrOptions

Provides different options to indicate whether certain actions must be performed to achieve the intended result.

Syntax

Definition:

```
enum ETcVnOcrOptions : ULONG
{
    OO_NONE        = 0,
    OO_WITHBLANKS = 1
};
```

Values

Name	Description
OO_NONE	No options
OO_WITHBLANKS	Indicates that if blanks were found, they should be included in the result. By default, all blanks are omitted from the result.

6.2.1.4.50 ETcVnOrientationMethod

Offers methods to calculate the orientation of a set of points.

Syntax

Definition:

```
enum ETcVnOrientationMethod : LONG
{
    OM_PCA           = 0,
    OM_FITELLIPSE    = 1,
    OM_MOMENTS       = 2,
    OM_ENCLOSINGRECT = 3
};
```

Values

Name	Description
OM_PCA	Apply the PCA on the points and calculate the mean point as the center point and the rotation angle of the main principal axis
OM_FITELLIPSE	Fit an ellipse around the points and calculate the center point and the rotation angle of the main axis
OM_MOMENTS	Calculate the moments of the points and estimate the center and the rotation angle
OM_ENCLOSINGRECT	Calculate a minimum area rectangle enclosing the points and calculate the center point and the rotation angle of the rectangle

6.2.1.4.51 ETcVnPaddingMode

Offers padding modes.

Syntax

Definition:

```
enum ETcVnPaddingMode : LONG
{
    PM_NONE        = 0,
    PM_CROP_CENTER = 1,
    PM_LETTERBOX   = 2
};
```

Values

Name	Description
PM_NONE	Resize the image to the desired size without extra processing
PM_CROP_CENTER	Resize and crop the image
PM_LETTERBOX	Resize the image to the desired size while preserving the aspect ratio of the original image

6.2.1.4.52 ETcVnPixelConnectivity

Offers pixel connectivities.

Syntax

Definition:

```
enum ETcVnPixelConnectivity : LONG
{
    PC_4 = 4,
    PC_8 = 8
};
```

Values

Name	Description
PC_4	4-way
PC_8	8-way

6.2.1.4.53 ETcVnPixelEncoding

Offers pixel encodings.

Syntax

Definition:

```
enum ETcVnPixelEncoding : BYTE
{
    PE_NONE                 = 0,
    PE_BAYER_GR             = 1,
    PE_BAYER_RG             = 2,
    PE_BAYER_GB             = 3,
    PE_BAYER_BG             = 4,
    PE_YUV_411_UYYVYY       = 5,
    PE_YUV_422_UYVY         = 6,
    PE_YUV_422_YUYV         = 7,
    PE_YCBCR_411_CBYYCRYY = 8,
    PE_YCBCR_422_CBYCRY     = 9,
    PE_YCBCR_422_YCBYCR     = 10
};
```

Values

Name	Description
PE_NONE	No encoding available, i.e. every pixel value is independent of other pixels.
PE_BAYER_GR	Pixels are encoded as a BayerGR pattern.
PE_BAYER_RG	Pixels are encoded as a BayerRG pattern.
PE_BAYER_GB	Pixels are encoded as a BayerGB pattern.
PE_BAYER_BG	Pixels are encoded as a BayerBG pattern.
PE_YUV_411_UYYVYY	Pixels are encoded as YUV411 (UYYVYY).
PE_YUV_422_UYVY	Pixels are encoded as YUV422 (UYVY).
PE_YUV_422_YUYV	Pixels are encoded as YUV422 (YUYV).
PE_YCBCR_411_CBYYCRYY	Pixels are encoded as YCbCr411 (CbYYCrYY).
PE_YCBCR_422_CBYCRY	Pixels are encoded as YCbCr422 (CbYCrY).
PE_YCBCR_422_YCBYCR	Pixels are encoded as YCbCr422 (YCbYCr).

6.2.1.4.54 ETcVnPixelPackMode

Offers pixel packing modes.

Syntax

Definition:

```
enum ETcVnPixelPackMode : BYTE
{
    PPM_NONE           = 0,
    PPM_MONO1P         = 1,
    PPM_MONO2P         = 2,
    PPM_MONO4P         = 3,
    PPM_MONO10PACKED   = 4,
    PPM_MONO12PACKED   = 5,
    PPM_RGB10V1PACKED = 6,
    PPM_RGB10P32       = 7,
    PPM_RGB12V1PACKED = 8,
    PPM_RGB565P        = 9,
    PPM_BGR565P        = 10,
    PPM_MONO10P        = 11,
    PPM_MONO12P        = 12,
    PPM_MONO14P        = 13
};
```

Values

Name	Description
PPM_NONE	No packing
PPM_MONO1P	Mono1p
PPM_MONO2P	Mono2p
PPM_MONO4P	Mono4p
PPM_MONO10PACKED	Mono10Packed or BayerXX10Packed
PPM_MONO12PACKED	Mono12Packed or BayerXX12Packed
PPM_RGB10V1PACKED	RGB10V1Packed
PPM_RGB10P32	RGB10V2Packed (RGB10p32)
PPM_RGB12V1PACKED	RGB12V1Packed
PPM_RGB565P	RGB565p
PPM_BGR565P	BGR565p
PPM_MONO10P	Mono10p or BayerXX10p
PPM_MONO12P	Mono12p or BayerXX12p
PPM_MONO14P	Mono14p or BayerXX14p

6.2.1.4.55 ETcVnPrototypeClusterer

Offers prototype clusterer model types.

Syntax

Definition:

```
enum ETcVnPrototypeClusterer : ULONG
{
    PC_NOVELTY_DETECTOR = 2,
    PC_CLUSTERER        = 8
};
```

Values

Name	Description
PC_NOVELTY_DETECTOR	Novelty detector
PC_CLUSTERER	Clusterer

6.2.1.4.56 ETcVnRectangleIntersection

Offers rectangle intersection types.

Syntax

Definition:

```
enum ETcVnRectangleIntersection : LONG
{
    RI_NONE    = 0,
    RI_PARTIAL = 1,
    RI_FULL    = 2
};
```

Values

Name	Description
RI_NONE	No intersection
RI_PARTIAL	Partial intersection
RI_FULL	Full intersection

6.2.1.4.57 ETcVnRotationAngle

Offers rotation angles.

Syntax

Definition:

```
enum ETcVnRotationAngle : LONG
{
    RA_90_DEG  = 0,
    RA_180_DEG = 1,
    RA_270_DEG = 2
};
```

Values

Name	Description
RA_90_DEG	90 degrees
RA_180_DEG	180 degrees
RA_270_DEG	270 degrees

6.2.1.4.58 ETcVnRTrees

Offers RTrees model types.

Syntax

Definition:

```
enum ETcVnRTrees : ULONG
{
    RT_CLASSIFIER = 1,
    RT_REGRESSOR  = 4
};
```

Values

Name	Description
RT_CLASSIFIER	Classifier
RT_REGRESSOR	Regressor

6.2.1.4.59 ETcVnSignedNormalization

Offers options for normalizing images with signed data types.

Syntax

Definition:

```
enum ETcVnSignedNormalization : SHORT
{
    SN_FIX_ZERO   = 0,
    SN_FULL_SCALE = 1
};
```

Values

Name	Description
SN_FIX_ZERO	The value 0 is fixed, i.e. the normalized image might only reach either the minimum or maximum value.
SN_FULL_SCALE	The image is normalized to the full value range, i.e. a prior value of 0 might be != 0 after normalization.

6.2.1.4.60 ETcVnSolvePnPMethod

Offers SolvePnP methods.

Syntax

Definition:

```
enum ETcVnSolvePnPMethod : LONG
{
    SPM_ITERATIVE    = 0,
    SPM_EPNP         = 1,
    SPM_P3P          = 2,
    SPM_AP3P         = 5,
    SPM_IPPE         = 6,
    SPM_IPPE_SQUARE = 7,
    SPM_SQPNP        = 8
};
```

Values

Name	Description
SPM_ITERATIVE	Levenberg-Marquardt optimization. Requires at least 4 planar or 6 non-planar reference points.
SPM_EPNP	Efficient perspective-n-point camera pose estimation. Requires at least 4 reference points.
SPM_P3P	Complete solution classification for the perspective-three-point problem. Requires exactly 4 reference points to find a unique solution.
SPM_AP3P	An efficient algebraic solution to the perspective-three-point problem. Requires exactly 4 reference points to find a unique solution.
SPM_IPPE	Infinitesimal plane-based pose estimation. Requires at least 4 coplanar reference points.
SPM_IPPE_SQUARE	IPPE specialization for squares. Requires exactly 4 reference points in the following order (hsl: half square length): (-hsl, hsl, 0), (hsl, hsl, 0), (hsl, -hsl, 0), (-hsl, -hsl, 0)
SPM_SQPNP	A consistently fast and globally optimal solution to the perspective-n-point problem. Requires at least 4 reference points.

6.2.1.4.61 ETcVnSta

Offers Simplified TopoART model types.

Syntax

Definition:

```
enum ETcVnSta : ULONG
{
    STA_CLASSIFIER       = 1,
    STA_NOVELTY_DETECTOR = 2,
    STA_REGRESSOR        = 4,
    STA_CLUSTERER        = 8
};
```

Values

Name	Description
STA_CLASSIFIER	Classifier
STA_NOVELTY_DETECTOR	Novelty detector
STA_REGRESSOR	Regressor
STA_CLUSTERER	Clusterer

6.2.1.4.62 ETcVnStructuringElementShape

Offers shapes for a structuring element.

Syntax

Definition:

```
enum ETcVnStructuringElementShape : LONG
{
    SES_RECTANGLE = 0,
    SES_CROSS     = 1,
    SES_ELLIPSE   = 2
};
```

Values

Name	Description
SES_RECTANGLE	Rectangle
SES_CROSS	Cross
SES_ELLIPSE	Ellipse

6.2.1.4.63 ETcVnSvm

Offers SVM model types.

Syntax

Definition:

```
enum ETcVnSvm : ULONG
{
    SVM_C_CLASSIFIER     = 1,
    SVM_NU_CLASSIFIER    = 65537,
    SVM_NOVELTY_DETECTOR = 2,
    SVM_EPS_REGRESSOR    = 4,
    SVM_NU_REGRESSOR     = 65540
};
```

Values

Name	Description
SVM_C_CLASSIFIER	Classifier
SVM_NU_CLASSIFIER	Classifier
SVM_NOVELTY_DETECTOR	Novelty detector
SVM_EPS_REGRESSOR	Regressor
SVM_NU_REGRESSOR	Regressor

6.2.1.4.64 ETcVnSvmKernelType

Offers SVM kernel types.

Syntax

Definition:

```
enum ETcVnSvmKernelType : LONG
{
    SKT_LINEAR  = 0,
    SKT_POLY    = 1,
    SKT_RBF     = 2,
    SKT_SIGMOID = 3,
    SKT_CHI2    = 4,
    SKT_INTER   = 5
};
```

Values

Name	Description
SKT_LINEAR	Linear kernel (fast)
SKT_POLY	Polynomial kernel
SKT_RBF	Radial basis function (RBF) kernel (good default)
SKT_SIGMOID	Sigmoid kernel
SKT_CHI2	Chi-squared kernel
SKT_INTER	Histogram intersection kernel

6.2.1.4.65 ETcVnSvmSgdClassifierMarginType

Offers different margin types for SVMSGD classifiers.

Syntax

Definition:

```
enum ETcVnSvmSgdClassifierMarginType : LONG
{
    SSCMT_SOFT_MARGIN = 0,
    SSCMT_HARD_MARGIN = 1
};
```

Values

Name	Description
SSCMT_SOFT_MARGIN	Soft margin allowing outliers. (beneficial for classes that cannot be separated linearly)
SSCMT_HARD_MARGIN	Hard margin. (best-suited for linearly separable classes)

6.2.1.4.66 ETcVnSvmSgdClassifierType

Offers different types of SVMSGD classifiers.

Syntax

Definition:

```
enum ETcVnSvmSgdClassifierType : LONG
{
    SSCT_SGD  = 0,
    SSCT_ASGD = 1
};
```

Values

Name	Description
SSCT_SGD	Stochastic Gradient Descent
SSCT_ASGD	Average Stochastic Gradient Descent

6.2.1.4.67 ETcVnTemplateMatchMethod

Offers methods for template matching.

Syntax

Definition:

```
enum ETcVnTemplateMatchMethod : LONG
{
    TMM_SQDIFF        = 0,
    TMM_SQDIFF_NORMED = 1,
    TMM_CCORR         = 2,
    TMM_CCORR_NORMED  = 3,
    TMM_CCOEFF        = 4,
    TMM_CCOEFF_NORMED = 5
};
```

Values

Name	Description
TMM_SQDIFF	Squared difference (supports template mask)
TMM_SQDIFF_NORMED	Normalized squared difference
TMM_CCORR	Cross-correlation
TMM_CCORR_NORMED	Normalized cross-correlation (supports template mask)
TMM_CCOEFF	Correlation coefficient
TMM_CCOEFF_NORMED	Normalized correlation coefficient

6.2.1.4.68 ETcVnThresholdType

Offers threshold types.

Syntax

Definition:

```
enum ETcVnThresholdType : LONG
{
    TT_BINARY              = 0,
    TT_BINARY_INV          = 1,
    TT_TRUNC               = 2,
    TT_TOZERO              = 3,
    TT_TOZERO_INV          = 4,
    TT_OTSU_BINARY         = 8,
    TT_OTSU_BINARY_INV     = 9,
    TT_OTSU_TRUNC          = 10,
    TT_OTSU_TOZERO         = 11,
    TT_OTSU_TOZERO_INV     = 12,
    TT_TRIANGLE_BINARY     = 16,
    TT_TRIANGLE_BINARY_INV = 17,
    TT_TRIANGLE_TRUNC      = 18,
    TT_TRIANGLE_TOZERO     = 19,
    TT_TRIANGLE_TOZERO_INV = 20
};
```

Values

Name	Description
TT_BINARY	Binary threshold
TT_BINARY_INV	Inverted binary threshold
TT_TRUNC	Truncated threshold (pixels > thresh are set to thresh, others keep their value)
TT_TOZERO	To zero threshold (pixels < thresh are set to zero, others keep their value)
TT_TOZERO_INV	Inverted to zero threshold (pixels > thresh are set to zero, others keep their value)
TT_OTSU_BINARY	Binary threshold with the threshold value selected according to the Otsu algorithm
TT_OTSU_BINARY_INV	Inverted binary threshold with the threshold value selected according to the Otsu algorithm
TT_OTSU_TRUNC	Truncated threshold with the threshold value selected according to the Otsu algorithm
TT_OTSU_TOZERO	To zero threshold with the threshold value selected according to the Otsu algorithm
TT_OTSU_TOZERO_INV	Inverted to zero threshold with the threshold value selected according to the Otsu algorithm
TT_TRIANGLE_BINARY	Binary threshold with the threshold value selected according to the Triangle algorithm
TT_TRIANGLE_BINARY_INV	Inverted binary threshold with the threshold value selected according to the Triangle algorithm
TT_TRIANGLE_TRUNC	Truncated threshold with the threshold value selected according to the Triangle algorithm
TT_TRIANGLE_TOZERO	To zero threshold with the threshold value selected according to the Triangle algorithm
TT_TRIANGLE_TOZERO_INV	Inverted to zero threshold with the threshold value selected according to the Triangle algorithm

6.2.1.4.69 ETcVnTimestamp

Offers image acquisition timestamps.

Syntax

Definition:

```
enum ETcVnTimestamp : ULONGLONG
{
    TS_IMAGE_COMPLETED = 1
};
```

Values

Name	Description
TS_IMAGE_COMPLETED	Timestamp when the image was completed, i.e. all packets were received by the GigEVision driver.

6.2.1.4.70 ETcVnVectorCompareMethod

Offers methods for vector comparison.

Syntax

Definition:

```
enum ETcVnVectorCompareMethod : LONG
{
    VCM_EUCLIDEAN    = 0,
    VCM_ELEMENTWISE = 1
};
```

Values

Name	Description
VCM_EUCLIDEAN	Euclidean
VCM_ELEMENTWISE	Elementwise

6.2.1.4.71 ETcVnWorldCoordinateSystem

Offers world coordinate system orientations.

Syntax

Definition:

```
enum ETcVnWorldCoordinateSystem : LONG
{
    WCS_X_RIGHT_Y_DOWN = 0,
    WCS_X_DOWN_Y_RIGHT = 1,
    WCS_X_LEFT_Y_DOWN  = 2,
    WCS_X_UP_Y_RIGHT   = 3,
    WCS_X_RIGHT_Y_UP   = 4,
    WCS_X_DOWN_Y_LEFT  = 5,
    WCS_X_LEFT_Y_UP    = 6,
    WCS_X_UP_Y_LEFT    = 7
};
```

Values

Name	Description
WCS_X_RIGHT_Y_DOWN	X axis points right, Y axis down (same orientation as image coordinate system)
WCS_X_DOWN_Y_RIGHT	X axis points down, Y axis right
WCS_X_LEFT_Y_DOWN	X axis points left, Y axis down
WCS_X_UP_Y_RIGHT	X axis points up, Y axis right
WCS_X_RIGHT_Y_UP	X axis points right, Y axis up
WCS_X_DOWN_Y_LEFT	X axis points down, Y axis left
WCS_X_LEFT_Y_UP	X axis points left, Y axis up
WCS_X_UP_Y_LEFT	X axis points up, Y axis left

6.2.1.4.72 ETcWatchdogAccumulationType

Offers watchdog accumulation types to compute the fraction processed.

Syntax

Definition:

```
enum ETcWatchdogAccumulationType : USHORT
{
    WD_ACC_TYPE_MEAN    = 1,
    WD_ACC_TYPE_PRODUCT = 2
};
```

Values

Name	Description
WD_ACC_TYPE_MEAN	Calculates the mean of the individual fractions processed over all monitored functions (recommended for independent functions).
WD_ACC_TYPE_PRODUCT	Calculates the product of the individual fractions processed over all monitored functions (recommended for dependent functions).

6.2.1.4.73 GEV_CAMERA_STATE

Describes the GigEVision camera connection state.

Syntax

Definition:

```
enum GEV_CAMERA_STATE : BYTE
{
    GEV_CAMERA_IDLE               = 0,
    GEV_CONTROL_CHANNEL_OPEN_MASK = 1,
    GEV_STREAM_CHANNELS_OPEN_MASK = 2,
    GEV_CONTROL_CHANNEL_OPEN      = 1,
    GEV_STREAM_CHANNELS_OPEN      = 3
};
```

Values

Name	Description
GEV_CAMERA_IDLE	Camera is idle (no control or stream channel open).
GEV_CONTROL_CHANNEL_OPEN_MASK	The control channel is open.
GEV_STREAM_CHANNELS_OPEN_MASK	At least 1 stream channel is open.
GEV_CONTROL_CHANNEL_OPEN	The control channel is open, but no stream channel.
GEV_STREAM_CHANNELS_OPEN	At least 1 stream channel and the control channel are open.

6.2.1.5 Structs

6.2.1.5.1 GVSP_IMAGE_INFO

Shows GVSP (GigE Vision Streaming Protocol) meta information.

Syntax

Definition:

```
struct GVSP_IMAGE_INFO
{
    ULONG                     CameraIpAddress;
    ULONG                     LocalIpAddress;
    USHORT                    CameraUdpPort;
    USHORT                    LocalUdpPort;
    USHORT                    GvspChannelId;
    USHORT                    GevStatus;
    ULONGLONG                 BlockId;
    GVSP_LEADER_PAYLOAD_IMAGE LeaderInfo;
};
```

Parameters

Name	Type	Description
CameraIpAddress	ULONG	Camera IP address
LocalIpAddress	ULONG	Local IP address
CameraUdpPort	USHORT	Camera UDP port
LocalUdpPort	USHORT	Local UDP port
GvspChannelId	USHORT	ID of GVSP channel
GevStatus	USHORT	GigE Vision status code
BlockId	ULONGLONG	Block Id (incremented for each acquired image, but reset to 0 on overflow)
LeaderInfo	GVSP_LEADER_PAYLOAD_IMAGE [▶ 1638]	Contains information about timestamp, pixel format and size.

6.2.1.5.2 GVSP_LEADER_PAYLOAD_IMAGE

Shows information over the GVSP (GigE Vision Streaming Protocol) leader payload image.

Syntax

Definition:

```
struct GVSP_LEADER_PAYLOAD_IMAGE
{
    ULONGLONG         Timestamp;
    GVSP_PIXEL_FORMAT PixelFormat;
    ULONG             SizeX;
    ULONG             SizeY;
    ULONG             OffsetX;
    ULONG             OffsetY;
    USHORT            PaddingX;
    USHORT            PaddingY;
};
```

Parameters

Name	Type	Description
Timestamp	ULONGLONG	Image timestamp
PixelFormat	GVSP_PIXEL_FORMAT [▶ 1638]	Image pixel format
SizeX	ULONG	Image size in x direction
SizeY	ULONG	Image size in y direction
OffsetX	ULONG	Image x-offset from (0,0) origin
OffsetY	ULONG	Image y-offset from (0,0) origin
PaddingX	USHORT	Image padding in x direction
PaddingY	USHORT	Image padding in y direction

6.2.1.5.3 GVSP_PIXEL_FORMAT

Shows the GVSP (GigE Vision Streaming Protocol) pixel format.

Syntax

Definition:

```
struct GVSP_PIXEL_FORMAT
{
    BYTE   Color;
```

```
    BYTE   EffectivePixelSize;
    USHORT Id;
};
```

Parameters

Name	Type	Description
Color	BYTE	Indicates if the pixel format is mono or color.
EffectivePixelSize	BYTE	Effective pixel size in bit
Id	USHORT	GVSP pixel format ID

6.2.1.5.4 TcVnCameraCalibrationOptions

Offers camera calibration options.

Syntax

Definition:

```
struct TcVnCameraCalibrationOptions
{
    UCHAR bUseIntrinsicGuess : 1;
    UCHAR bFixAspectRatio : 1;
    UCHAR bFixPrincipalPoint : 1;
    UCHAR bZeroTangentDist : 1;
    UCHAR bFixFocalLength : 1;
    UCHAR bFixK1 : 1;
    UCHAR bFixK2 : 1;
    UCHAR bFixK3 : 1;
    UCHAR bFixK4 : 1;
    UCHAR bFixK5 : 1;
    UCHAR bFixK6 : 1;
    UCHAR bRationalModel : 1;
};
```

Parameters

Name	Type	Description
bUseIntrinsicGuess	UCHAR : 1	The camera matrix and distortion coefficients already contain a valid initial guess, which is optimized further.
bFixAspectRatio	UCHAR : 1	The ratio fx/fy stays the same as in the input camera matrix.
bFixPrincipalPoint	UCHAR : 1	The principal point is fixed to the image center (or provided cx, cy if bUseIntrinsicGuess).
bZeroTangentDist	UCHAR : 1	The tangential distortion coefficients (p1, p2) are forced to 0.
bFixFocalLength	UCHAR : 1	The parameters fx and fy stay the same as in the input camera matrix.
bFixK1	UCHAR : 1	The radial distortion coefficient k1 is fixed to 0 (or provided input if bUseIntrinsicGuess)
bFixK2	UCHAR : 1	The radial distortion coefficient k2 is fixed to 0 (or provided input if bUseIntrinsicGuess)
bFixK3	UCHAR : 1	The radial distortion coefficient k3 is fixed to 0 (or provided input if bUseIntrinsicGuess)
bFixK4	UCHAR : 1	The radial distortion coefficient k4 is fixed to 0 (or provided input if bUseIntrinsicGuess)
bFixK5	UCHAR : 1	The radial distortion coefficient k5 is fixed to 0 (or provided input if bUseIntrinsicGuess)
bFixK6	UCHAR : 1	The radial distortion coefficient k6 is fixed to 0 (or provided input if bUseIntrinsicGuess)
bRationalModel	UCHAR : 1	The radial distortion coefficients k4, k5, k6 are enabled.

6.2.1.5.5 TcVnCircularArc

Describes a circular arc.

Syntax

Definition:

```
struct TcVnCircularArc
{
    TcVnPoint2_REAL aCenter;
    float           fRadius;
    float           fStartAngle;
    float           fEndAngle;
};
```

Parameters

Name	Type	Description
aCenter	TcVnPoint2_REAL [▶ 1588]	Center of the circular arc
fRadius	float	Radius of the circular arc
fStartAngle	float	Start angle of the circular arc
fEndAngle	float	End angle of the circular arc

6.2.1.5.6 TcVnCodeGrades1D

Describes code quality grades for 1D barcodes according to ISO / IEC 15416:2016. The grades range from 0 (very bad) to 4 (very good) and are averaged over several scan lines.

Syntax

Definition:

```
struct TcVnCodeGrades1D
{
    float fDecode;
    float fSymbolContrast;
    float fMinReflectance;
    float fMinEdgeContrast;
    float fModulation;
    float fDefects;
    float fDecodability;
    float fOverall;
};
```

Parameters

Name	Type	Description
fDecode	float	Decode grade
fSymbolContrast	float	Symbol contrast grade
fMinReflectance	float	Minimum reflectance grade
fMinEdgeContrast	float	Minimum edge contrast grade
fModulation	float	Modulation grade
fDefects	float	Defects grade
fDecodability	float	Decodability grade
fOverall	float	Overall code grade, i.e. the minimum achieved individual grade averaged over all scan lines. If different scan lines lead to different decoded data, the overall grade is 0.

6.2.1.5.7 TcVnCodeGradesDM

Describes code quality grades for Data Matrix codes according to ISO / IEC 15415:2011. The grades range from 0 (very bad) to 4 (very good).

Syntax

Definition:

```
struct TcVnCodeGradesDM
{
    unsigned char nDecode;
    unsigned char nSymbolContrast;
    unsigned char nModulation;
    unsigned char nReflectanceMargin;
    unsigned char nFixedPatternDamage;
    unsigned char nAxialNonuniformity;
    unsigned char nGridNonuniformity;
```

```
    unsigned char nUnusedErrorCorrection;
    unsigned char nOverall;
};
```

Parameters

Name	Type	Description
nDecode	unsigned char	Decode grade
nSymbolContrast	unsigned char	Symbol contrast grade
nModulation	unsigned char	Modulation grade
nReflectanceMargin	unsigned char	Reflectance margin grade
nFixedPatternDamage	unsigned char	Fixed pattern damage grade
nAxialNonuniformity	unsigned char	Axial nonuniformity grade
nGridNonuniformity	unsigned char	Grid nonuniformity grade
nUnusedErrorCorrection	unsigned char	Unused error correction grade
nOverall	unsigned char	Overall code grade, i.e. the minimum achieved individual grade

6.2.1.5.8 TcVnCodeGradesQR

Describes code quality grades for QR codes according to ISO / IEC 15415:2011. The grades range from 0 (very bad) to 4 (very good).

Syntax

Definition:

```
struct TcVnCodeGradesQR
{
    unsigned char nDecode;
    unsigned char nSymbolContrast;
    unsigned char nModulation;
    unsigned char nReflectanceMargin;
    unsigned char nFixedPatternDamage;
    unsigned char nAxialNonuniformity;
    unsigned char nGridNonuniformity;
    unsigned char nUnusedErrorCorrection;
    unsigned char nFormatInfo;
    unsigned char nVersionInfo;
    unsigned char nOverall;
};
```

Parameters

Name	Type	Description
nDecode	unsigned char	Decode grade
nSymbolContrast	unsigned char	Symbol contrast grade
nModulation	unsigned char	Modulation grade
nReflectanceMargin	unsigned char	Reflectance margin grade
nFixedPatternDamage	unsigned char	Fixed pattern damage grade
nAxialNonuniformity	unsigned char	Axial nonuniformity grade
nGridNonuniformity	unsigned char	Grid nonuniformity grade
nUnusedErrorCorrection	unsigned char	Unused error correction grade
nFormatInfo	unsigned char	Format information grade
nVersionInfo	unsigned char	Version information grade
nOverall	unsigned char	Overall code grade, i.e. the minimum achieved individual grade

6.2.1.5.9 TcVnDMatch

Describes a descriptor match

Syntax

Definition:

```
struct TcVnDMatch
{
    LONG  nQueryIdx;
    LONG  nTrainIdx;
    LONG  nImageIdx;
    float fDistance;
};
```

Parameters

Name	Type	Description
nQueryIdx	LONG	Query descriptor index
nTrainIdx	LONG	Train descriptor index
nImageIdx	LONG	Train image index
fDistance	float	Distance between the descriptors (smaller distance means better match)

6.2.1.5.10 TcVnImageInfo

Shows image information.

Syntax

Definition:

```
struct TcVnImageInfo
{
    ULONGLONG       nImageSize;
    ULONG           nWidth;
    ULONG           nHeight;
    USHORT          nXPadding;
    USHORT          nYPadding;
    TcVnPixelFormat stPixelFormat;
};
```

Parameters

Name	Type	Description
nImageSize	ULONGLONG	Image size (number of pixels)
nWidth	ULONG	Image width
nHeight	ULONG	Image height
nXPadding	USHORT	Image x-padding
nYPadding	USHORT	Image y-padding
stPixelFormat	TcVnPixelFormat [▶ 1657]	Pixel format

6.2.1.5.11 TcVnKeyPoint

Describes a key point.

Syntax

Definition:

```
struct TcVnKeyPoint
{
    TcVnPoint2_REAL aPoint;
```

```
    float           fDiameter;
    float           fAngle;
    float           fResponse;
    LONG            nOctave;
    LONG            nClassId;
};
```

Parameters

Name	Type	Description
aPoint	TcVnPoint2_REAL [▸ 1588]	Position
fDiameter	float	Diameter
fAngle	float	Angle
fResponse	float	Response
nOctave	LONG	Octave
nClassId	LONG	Class ID

6.2.1.5.12 TcVnMatrix

Offers a user-defined matrix with variable rows, columns and element-type.

Syntax

Definition:

```
struct TcVnMatrix
{
    ULONG            nRows;
    ULONG            nCols;
    ETcVnElementType eType;
    PVOID            pData;
};
```

Parameters

Name	Type	Description
nRows	ULONG	Number of rows
nCols	ULONG	Number of columns
eType	ETcVnElementType [▸ 1615]	Element type
pData	PVOID	Pointer to the data

6.2.1.5.13 TcVnMoments

Offers image or contour moments.

Syntax

Definition:

```
struct TcVnMoments
{
    double fM00;
    double fM10;
    double fM01;
    double fM20;
    double fM11;
    double fM02;
    double fM30;
    double fM21;
    double fM12;
    double fM03;
    double fMu20;
    double fMu11;
    double fMu02;
    double fMu30;
    double fMu21;
```

```
    double fMu12;
    double fMu03;
    double fNu20;
    double fNu11;
    double fNu02;
    double fNu30;
    double fNu21;
    double fNu12;
    double fNu03;
};
```

Parameters

Name	Type	Description
fM00	double	Spatial moment 00
fM10	double	Spatial moment 10
fM01	double	Spatial moment 01
fM20	double	Spatial moment 20
fM11	double	Spatial moment 11
fM02	double	Spatial moment 02
fM30	double	Spatial moment 30
fM21	double	Spatial moment 21
fM12	double	Spatial moment 12
fM03	double	Spatial moment 03
fMu20	double	Central moment 20
fMu11	double	Central moment 11
fMu02	double	Central moment 02
fMu30	double	Central moment 30
fMu21	double	Central moment 21
fMu12	double	Central moment 12
fMu03	double	Central moment 03
fNu20	double	Normalized central moment 20
fNu11	double	Normalized central moment 11
fNu02	double	Normalized central moment 02
fNu30	double	Normalized central moment 30
fNu21	double	Normalized central moment 21
fNu12	double	Normalized central moment 12
fNu03	double	Normalized central moment 03

6.2.1.5.14 TcVnParamsAGAST

Offers parameters for AGAST method.

Syntax

Definition:

```
struct TcVnParamsAGAST
{
    LONG                            nThreshold;
    bool                            bNonMaxSuppression;
    ETcVnKeypointDetectionTypeAGAST eType;
};
```

Parameters

Name	Type	Default	Description
nThreshold	LONG	10	Threshold for the intensity difference between the center pixel and its surrounding circle
bNonMaxSuppression	bool	true	If true, non-maximum suppression is applied.
eType	ETcVnKeypointDetectionTypeAGAST [▶ 1621]	KDT_AGAST_9_16	Neighborhood type

6.2.1.5.15 TcVnParamsAKAZE

Offers parameters for AKAZE method.

Syntax

Definition:

```
struct TcVnParamsAKAZE
{
    ETcVnFeatureDescriptorTypeAKAZE eDescrType;
    ULONG                           nDescrSize;
    ULONG                           nDescrChannels;
    float                           fThreshold;
    ULONG                           nOctaves;
    ULONG                           nOctaveLayers;
    ETcVnDiffusivityTypeKAZE        eDiffusivity;
};
```

Parameters

Name	Type	Default	Description
eDescrType	ETcVnFeatureDescriptorTypeAKAZE [▶ 1617]	FDT_AKAZE_MLDB	Type of the descriptor
nDescrSize	ULONG	0	Size of the descriptor in bits (only for MLDB; 0 = full size)
nDescrChannels	ULONG	3	Number of descriptor channels (currently only 3 supported for MLDB)
fThreshold	float	0.001f	Detector response threshold
nOctaves	ULONG	2	Maximum octave evolution
nOctaveLayers	ULONG	1	Number of sublevels per scale level
eDiffusivity	ETcVnDiffusivityTypeKAZE [▶ 1611]	DT1_KAZE_PM_G2	Diffusivity type

6.2.1.5.16 TcVnParamsBlobDetection

Offers parameters for blob detection

Syntax

Definition:

```
struct TcVnParamsBlobDetection
{
    bool                    bFilterByArea;
    bool                    bFilterByCircularity;
    bool                    bFilterByConvexity;
    bool                    bFilterByEccentricity;
    bool                    bFilterByInertiaRatio;
    float                   fMinArea;
    float                   fMaxArea;
    float                   fMinCircularity;
    float                   fMaxCircularity;
    float                   fMinConvexity;
    float                   fMaxConvexity;
    float                   fMinEccentricity;
    float                   fMaxEccentricity;
    float                   fMinInertiaRatio;
    float                   fMaxInertiaRatio;
    ETcVnThresholdType      eThresholdType;
    float                   fMinThreshold;
    float                   fMaxThreshold;
    float                   fThresholdStep;
    float                   fMinBlobDistance;
    ULONG                   nMinRepeatability;
    ETcVnBlobCombination eBlobCombination;
};
```

Parameters

Name	Type	Default	Description
bFilterByArea	bool	true	Enable filtering by area (fMinArea <= area <= fMaxArea); strongly recommended to activate for filtering noise with fMinArea.
bFilterByCircularity	bool	false	Enable filtering by circularity (fMinCircularity <= circularity(4pi * area / perimeter^2) <= fMaxCircularity).
bFilterByConvexity	bool	false	Enable filtering by convexity (fMinConvexity <= convexity(area / hullArea) <= fMaxConvexity).
bFilterByEccentricity	bool	false	Enable filtering by eccentricity (fMinEccentricity <= eccentricity <= fMaxEccentricity).
bFilterByInertiaRatio	bool	false	Enable filtering by inertia ratio (fMinInertiaRatio <= inertia ratio <= fMaxInertiaRatio).
fMinArea	float	10	Min estimated blob area in pixel
fMaxArea	float	100000000	Max estimated blob area in pixel
fMinCircularity	float	0	Min circularity (1.0: ideal circle, < 1: less circular, 0: not circular at all)
fMaxCircularity	float	1	Max circularity (1.0: ideal circle, < 1: less circular, 0: not circular at all)
fMinConvexity	float	0	Min convexity (1.0: blob fully convex, < 1: less convex)
fMaxConvexity	float	1	Max convexity (1.0: blob fully convex, < 1: less convex)
fMinEccentricity	float	0	Min eccentricity (0.0: circular, 1.0: linear)
fMaxEccentricity	float	1	Max eccentricity (0.0: circular, 1.0: linear)
fMinInertiaRatio	float	0	Min inertia ratio (1.0: equal width and height, 0.0: linear)
fMaxInertiaRatio	float	1	Max inertia ratio (1.0: equal width and height, 0.0: linear)

Name	Type	Default	Description
eThresholdType	ETcVnThresholdType [▶ 1634]	TT_BINARY	Threshold type for internally applied threshold method (OTSU_XXX only supported for USINT images).
fMinThreshold	float	30	Threshold to start with (if fThresholdStep > 0, otherwise this is the only threshold used).
fMaxThreshold	float	225	Threshold to end with (if fThresholdStep > 0, otherwise this value is not used).
fThresholdStep	float	0	Sets to 0 if only 1 threshold should be used (much faster than multiple thresholds and combining the results).
fMinBlobDistance	float	5	Minimum distance between the center points of two different blobs (only used if fThresholdStep > 0; if distance < fMinBlobDistance, the blobs are treated as the same).
nMinRepeatability	ULONG	2	Minimum number of threshold steps, for which the same contour has to be detected (only used if fThresholdStep > 0; same means center point distance < fMinBlobDistance).
eBlobCombination	ETcVnBlobCombination [▶ 1592]	BC_MEDIAN_THRESHOLD	Selects, which of the multi-threshold blob contours should be returned.

6.2.1.5.17 TcVnParamsBRISK

Offers parameters for BRISK method.

Syntax

Definition:

```
struct TcVnParamsBRISK
{
    LONG  nThreshold;
    ULONG nOctaves;
    float fPatternScale;
};
```

Parameters

Name	Type	Default	Description
nThreshold	LONG	30	Detection threshold
nOctaves	ULONG	3	Detection octaves (0 for single scale)
fPatternScale	float	1	Scale factor for the neighborhood pattern

6.2.1.5.18 TcVnParamsFAST

Offers parameters for FAST method.

Syntax

Definition:

```
struct TcVnParamsFAST
{
    LONG                          nThreshold;
    bool                          bNonMaxSuppression;
    ETcVnKeypointDetectionTypeFAST eType;
};
```

Parameters

Name	Type	Default	Description
nThreshold	LONG	10	Threshold for the intensity difference between the center pixel and its surrounding circle
bNonMaxSuppression	bool	true	If true, non-maximum suppression is applied.
eType	ETcVnKeypointDetectionTypeFAST [▶ 1622]	KDT_FAST_9_16	Neighborhood type

6.2.1.5.19 TcVnParamsGFTT

Offers parameters for GFTT method.

Syntax

Definition:

```
struct TcVnParamsGFTT
{
    ULONG  nMaxCorners;
    double fQualityLevel;
    double fMinDistance;
    ULONG  nBlockSize;
    bool   bUseHarrisDetector;
    double fHarrisK;
};
```

Parameters

Name	Type	Default	Description
nMaxCorners	ULONG	1000	Maximum number of corners to return (strongest ones)
fQualityLevel	double	0.01	Minimum accepted corner quality, relative to the strongest one
fMinDistance	double	1	Minimum euclidean distance between returned keypoints
nBlockSize	ULONG	3	Neighborhood block size
bUseHarrisDetector	bool	false	If true, a Harris detector is used instead of the default method.
fHarrisK	double	0.04	Free parameter of the Harris detector

6.2.1.5.20 TcVnParamsKAZE

Offers parameters for KAZE method

Syntax

Definition:

```
struct TcVnParamsKAZE
{
    bool                     bExtended;
    bool                     bUpright;
    float                    fThreshold;
    ULONG                    nOctaves;
    ULONG                    nOctaveLayers;
    ETcVnDiffusivityTypeKAZE eDiffusivity;
};
```

Parameters

Name	Type	Default	Description
bExtended	bool	false	If true, the descriptor size is extended from the default 64 byte to 128 byte.
bUpright	bool	false	If true, upright (rotation dependent) descriptors are computed.
fThreshold	float	0.001f	Detector response threshold
nOctaves	ULONG	4	Maximum octave evolution of the image
nOctaveLayers	ULONG	2	Number of sublevels per scale level
eDiffusivity	ETcVnDiffusivityTypeKAZE [▸ 1611]	DT1_KAZE_PM_G2	Diffusivity type

6.2.1.5.21 TcVnParamsMSER

Offers parameters for MSER method

Syntax

Definition:

```
struct TcVnParamsMSER
{
    LONG   nDelta;
    ULONG  nMinArea;
    ULONG  nMaxArea;
    double fMaxVariation;
    double fMinDiversity;
    ULONG  nMaxEvolution;
    double fAreaThreshold;
    double fMinMargin;
    ULONG  nEdgeBlurSize;
};
```

Parameters

Name	Type	Default	Description
nDelta	LONG	5	Delta for size comparison
nMinArea	ULONG	60	Minimum blob size
nMaxArea	ULONG	14400	Maximum blob size
fMaxVariation	double	0.25	Maximum blob size variation
fMinDiversity	double	0.2	Minimum MSER diversity
nMaxEvolution	ULONG	200	Maximum evolution steps (only used for color images)
fAreaThreshold	double	1.01	Area threshold to cause re-initialization (only used for color images)
fMinMargin	double	0.003	Minimum margin (only used for color images)
nEdgeBlurSize	ULONG	5	Aperture size for edge bluring (only used for color images)

6.2.1.5.22 TcVnParamsORB

Offers parameters for ORB method.

Syntax

Definition:

```
struct TcVnParamsORB
{
    ULONG                     nMaxPoints;
    float                     fPyramidScale;
    ULONG                     nPyramidLevels;
    ULONG                     nEdgeThreshold;
    ULONG                     nFirstLevel;
    ULONG                     nBriefPoints;
    ETcVnFeatureScoreTypeORB  eScoreType;
    ULONG                     nPatchSize;
    LONG                      nFastThreshold;
};
```

Parameters

Name	Type	Default	Description
nMaxPoints	ULONG	500	Maximum number of returned keypoints
fPyramidScale	float	1.2f	Pyramid decimation ratio (must be greater than 1, should be smaller than 2)
nPyramidLevels	ULONG	8	Number of pyramid levels
nEdgeThreshold	ULONG	31	Size of the border, where no features are detected (should match nPatchSize)
nFirstLevel	ULONG	0	First pyramid level (currently, only 0 is supported)
nBriefPoints	ULONG	2	Number of points to produce each BRIEF descriptor element (2, 3, 4)
eScoreType	ETcVnFeatureScoreTypeORB [▶ 1618]	FST_ORB_HARRIS	Score type (HARRIS is more stable but slightly slower than FAST)
nPatchSize	ULONG	31	Patch size of the BRIEF descriptor
nFastThreshold	LONG	20	Threshold for the FAST keypoint detection

6.2.1.5.23 TcVnParamsSB

Offers parameters for SB method (a simple blob detector with multiple thresholds).

Syntax

Definition:

```
struct TcVnParamsSB
{
    bool          bFilterByArea;
    bool          bFilterByCircularity;
    bool          bFilterByColor;
    bool          bFilterByConvexity;
    bool          bFilterByInertia;
    float         fMinArea;
    float         fMaxArea;
    float         fMinCircularity;
    float         fMaxCircularity;
    unsigned char nBlobColor;
    float         fMinConvexity;
    float         fMaxConvexity;
    float         fMinInertiaRatio;
    float         fMaxInertiaRatio;
    float         fMinBlobDist;
    ULONG         nMinRepeatability;
    float         fMinThreshold;
    float         fMaxThreshold;
    float         fThresholdStep;
};
```

Parameters

Name	Type	Default	Description
bFilterByArea	bool	true	Enable filtering by area (fMinArea <= area < fMaxArea).
bFilterByCircularity	bool	false	Enable filtering by circularity (fMinCircularity <= circularity(4pi * area / perimeter^2) < fMaxCircularity).
bFilterByColor	bool	false	Enable filtering by color (thresholdedColor(0 or 255) = nBlobColor).
bFilterByConvexity	bool	false	Enable filtering by convexity (fMinConvexity <= convexity(area / hullArea) < fMaxConvexity).
bFilterByInertia	bool	false	Enable filtering by inertia ratio (fMinInertiaRatio <= inertia ratio < fMaxInertiaRatio).
fMinArea	float	25	Min estimated blob area in pixel
fMaxArea	float	15000	Max estimated blob area in pixel
fMinCircularity	float	0	Min circularity (1.0: ideal circle, < 1: less circular, 0: not circular at all)
fMaxCircularity	float	1	Max circularity (1.0: ideal circle, < 1: less circular, 0: not circular at all)
nBlobColor	unsigned char	255	0 or 255
fMinConvexity	float	0	Min convexity (1.0: blob fully convex, < 1: less convex)
fMaxConvexity	float	1	Max convexity (1.0: blob fully convex, < 1: less convex)
fMinInertiaRatio	float	0	Min inertia ratio (0.0 .. 1.0)
fMaxInertiaRatio	float	1	Max inertia ratio (0.0 .. 1.0)
fMinBlobDist	float	5	Min distance between different blobs
nMinRepeatability	ULONG	2	Min number of same detected blobs (dist < fMinBlobDist) in different thresholds
fMinThreshold	float	30	Min threshold (start)
fMaxThreshold	float	225	Max threshold (stop)
fThresholdStep	float	10	Threshold step between min and max threshold

6.2.1.5.24 TcVnPixelFormat

Contains detailed information about the pixel format.

Syntax

Definition:

```
struct TcVnPixelFormat
{
    UCHAR                bSupported : 1;
    UCHAR                bSigned : 1;
    UCHAR                bPlanar : 1;
    UCHAR                bFloat : 1;
    BYTE                 nChannels;
    ETcVnPixelEncoding ePixelEncoding;
    ETcVnPixelPackMode ePixelPackMode;
    USHORT               nElementSize;
    USHORT               nTotalSize;
};
```

Parameters

Name	Type	Description
bSupported	UCHAR : 1	If false, the pixel format is not supported.
bSigned	UCHAR : 1	If true, pixel intensities are signed values.
bPlanar	UCHAR : 1	If true, the image channels are stored planar instead of interleaved (e.g. RRRRRGGGGGBBBBB instead of RGBRGBRGBRGBRGB).
bFloat	UCHAR : 1	If true, the pixel format is floating point.
nChannels	BYTE	Number of channels
ePixelEncoding	ETcVnPixelEncoding [▶ 1627]	Pixel encoding
ePixelPackMode	ETcVnPixelPackMode [▶ 1628]	Pixel pack mode
nElementSize	USHORT	Size (bit) of a single pixel channel
nTotalSize	USHORT	Size (bit) of all pixel channels

6.2.1.5.25 TcVnRectangle_DINT

Contains origin and size of a rectangle.

Syntax

Definition:

```
struct TcVnRectangle_DINT
{
    LONG nX;
    LONG nY;
    LONG nWidth;
    LONG nHeight;
};
```

Parameters

Name	Type	Description
nX	LONG	X coordinate of the top-left corner
nY	LONG	Y coordinate of the top-left corner
nWidth	LONG	Width
nHeight	LONG	Height

6.2.1.5.26 TcVnRectangle_UDINT

Contains origin and size of a rectangle.

Syntax

Definition:

```
struct TcVnRectangle_UDINT
{
    ULONG nX;
    ULONG nY;
    ULONG nWidth;
    ULONG nHeight;
};
```

Parameters

Name	Type	Description
nX	ULONG	X coordinate of the top-left corner
nY	ULONG	Y coordinate of the top-left corner
nWidth	ULONG	Width
nHeight	ULONG	Height

6.2.1.5.27 TcVnRotatedRectangle

Contains center, size and angle of a rotated rectangle.

Syntax

Definition:

```
struct TcVnRotatedRectangle
{
    TcVnPoint2_REAL aCenter;
    TcVnSize2_REAL  stSize;
    float           fAngle;
};
```

Parameters

Name	Type	Description
aCenter	TcVnPoint2_REAL [▶ 1588]	Center point
stSize	TcVnSize2_REAL [▶ 1658]	Size composed of fWidth and fHeight
fAngle	float	Angle in degree

6.2.1.5.28 TcVnSize2_REAL

Contains width and height.

Syntax

Definition:

```
struct TcVnSize2_REAL
{
    float fWidth;
    float fHeight;
};
```

Parameters

Name	Type	Description
fWidth	float	Width
fHeight	float	Height

6.2.2 Interfaces

接口不需要许可证。

接口分组

TwinCAT Vision 库的所有接口按主题分类为以下几组：

- 图像 [▶ 1765] ()
- 容器 [▶ 1659] ()
- 机器学习 [▶ 1802] ()
- 其他 [▶ 1805] ()

6.2.2.1 Containers

该组包含用于处理容器 [▶ 132]的接口。

常规信息

容器基本上是几个元素组成的矢量。元素的数量可以动态地改变。路由器内存 [▶ 49]中的相应内存为此自动分配和释放。

容器类型

容器的所有元素都为相同的类型。根据容器包含的元素类型，容器的特点为独特的 GUID。为了便于使用，这些 GUID 由常量 [▶ 139]定义。容器的所有类型名称以“ContainerType_Vector_”开始（“ContainerType_String_SINT”除外）。对于所有使用容器的 API 元素，类型要求都有记录。

容器类型可按以下方式确定：

```
ipContainer.GetElementTypeGuid(nTypeGuid);
```

6.2.2.1.1 Function Blocks

该组包含用于实现某些功能块的接口。

6.2.2.1.1.1 ITcVnCustomContainerOperation_ITcVnContainer

Offers an interface for custom, elementwise container operations (for containers of containers) with up to 3 different containers.

Inheritance Hierarchy

ITcVnCustomContainerOperation_ITcVnContainer

Methods

Name	Origin	Description
Execute [▸ 1660]	ITcVnCustomContainerOperation_ITcVnContainer	Executes the custom operation on the container elements.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.1.1.1 Execute

Executes the custom operation on the container elements.

Syntax

Definition:

```
HRESULT Execute(
    ITcVnContainer* ipElement1,
    ITcVnContainer* ipElement2,
    ITcVnContainer* ipElement3
)
```

Parameters

Name	Type	Description
ipElement1	ITcVnContainer* [▸ 1760]	Current element of the 1st container.
ipElement2	ITcVnContainer* [▸ 1760]	Current element of the 2nd container.
ipElement3	ITcVnContainer* [▸ 1760]	Current element of the 3rd container.

Return value

HRESULT [▸ 122]

6.2.2.1.1.2 ITcVnCustomContainerOperation_ITcVnForwardIterator

Offers an interface for custom, elementwise container operations with up to 3 different containers.

Inheritance Hierarchy

ITcVnCustomContainerOperation_ITcVnForwardIterator

Methods

Name	Origin	Description
Execute [▸ 1661]	ITcVnCustomContainerOperation_ITcVnForwardIterator	Executes the custom operation on the container elements.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.1.2.1 Execute

Executes the custom operation on the container elements.

Syntax

Definition:

```
HRESULT Execute(
    ITcVnForwardIterator* ipElement1,
    ITcVnForwardIterator* ipElement2,
    ITcVnForwardIterator* ipElement3
)
```

Parameters

Name	Type	Description
ipElement1	ITcVnForwardIterator* [▶ 1752]	Current element of the 1st container.
ipElement2	ITcVnForwardIterator* [▶ 1752]	Current element of the 2nd container.
ipElement3	ITcVnForwardIterator* [▶ 1752]	Current element of the 3rd container.

Return value

HRESULT [▶ 122]

6.2.2.1.1.3 ITcVnCustomElementCondition_ITcVnContainer

Offers an interface for a custom condition computed for a container.

Inheritance Hierarchy

ITcVnCustomElementCondition_ITcVnContainer

Methods

Name	Origin	Description
Condition [▶ 1661]	ITcVnCustomElementCondition_ITcVnContainer	Evaluates the condition for the container.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.1.3.1 Condition

Evaluates the condition for the container.

Syntax

Definition:

```
bool Condition(
    ITcVnContainer* ipElement
)
```

Parameters

Name	Type	Description
ipElement	ITcVnContainer* [▸ 1760]	The container for which the condition is evaluated.

Return value

bool

6.2.2.1.1.4 ITcVnCustomElementCondition_ITcVnForwardIterator

Offers an interface for a custom condition computed for an element represented by an ITcVnForwardIterator.

Inheritance Hierarchy

ITcVnCustomElementCondition_ITcVnForwardIterator

Methods

Name	Origin	Description
Condition [▸ 1662]	ITcVnCustomElementCondition_ITcVnForwardIterator	Evaluates the condition for the provided element.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.1.4.1 Condition

Evaluates the condition for the provided element.

Syntax

Definition:

```
bool Condition(
    ITcVnForwardIterator* ipElement
)
```

Parameters

Name	Type	Description
ipElement	ITcVnForwardIterator* [▸ 1752]	Element for which the condition is evaluated.

Return value

bool

6.2.2.1.2 ITcVnAccess

接口ITcVnAccess用于访问容器元素 [▸ 133]，并由迭代器 [▸ 1750]提供。与ITcVnRandomAccess [▸ 1704]不同，只有相关迭代器在那一刻指向的元素可以被寻址。

6.2.2.1.2.1 ITcVnAccess_DINT

Offers an access interface for DINT values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
    ITcVnAccess_DINT

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1663]	ITcVnAccess_DINT	Gets the value
Set [▸ 1663]	ITcVnAccess_DINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.1.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    LONG& nValue
)
```

Parameters

Name	Type	Description
nValue	LONG&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.2.1.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    LONG nValue
)
```

Parameters

Name	Type	Description
nValue	LONG	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.2.2 ITcVnAccess_INT

Offers an access interface for INT values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
ITcVnAccess_INT

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1664]	ITcVnAccess_INT	Gets the value
Set [▸ 1665]	ITcVnAccess_INT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.2.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    SHORT& nValue
)
```

Parameters

Name	Type	Description
nValue	SHORT&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.2.2.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    SHORT nValue
)
```

Parameters

Name	Type	Description
nValue	SHORT	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.2.3 ITcVnAccess_ITcVnImage

Offers an access interface for images.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
ITcVnAccess_ITcVnImage

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1665]	ITcVnAccess_ITcVnImage	Gets the image.
Set [▸ 1666]	ITcVnAccess_ITcVnImage	Sets the image.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.3.1 Get

Gets the image.

Syntax

Definition:

```
HRESULT Get(
    ITcVnImage*& ipImage
)
```

Parameters

Name	Type	Description
ipImage	ITcVnImage* [▸ 1797]&	Returns the image.

Return value

HRESULT [▸ 122]

6.2.2.1.2.3.2 Set

Sets the image.

Syntax

Definition:

```
HRESULT Set(
    ITcVnImage* ipImage
)
```

Parameters

Name	Type	Description
ipImage	ITcVnImage* [▸ 1797]	The image to set.

Return value

HRESULT [▸ 122]

6.2.2.1.2.4 ITcVnAccess_LREAL

Offers an access interface for LREAL values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
ITcVnAccess_LREAL

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1666]	ITcVnAccess_LREAL	Gets the value
Set [▸ 1667]	ITcVnAccess_LREAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.4.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    double& fValue
)
```

Parameters

Name	Type	Description
fValue	double&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.2.4.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    double fValue
)
```

Parameters

Name	Type	Description
fValue	double	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.2.5 ITcVnAccess_REAL

Offers an access interface for REAL values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
        ITcVnAccess_REAL

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1668]	ITcVnAccess_REAL	Gets the value
Set [▸ 1668]	ITcVnAccess_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.5.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    float& fValue
)
```

Parameters

Name	Type	Description
fValue	float&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.5.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    float fValue
)
```

Parameters

Name	Type	Description
fValue	float	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.6 ITcVnAccess_SINT

Offers an access interface for SINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcVnAccess_SINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1669]	ITcVnAccess_SINT	Gets the value
Set [▶ 1669]	ITcVnAccess_SINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.6.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    char& nValue
)
```

Parameters

Name	Type	Description
nValue	char&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.6.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    char nValue
)
```

Parameters

Name	Type	Description
nValue	char	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.7 ITcVnAccess_TcVnDMatch

Offers an access interface for TcVnDMatch values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcVnAccess_TcVnDMatch

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1670]	ITcVnAccess_TcVnDMatch	Gets the value
Set [▶ 1670]	ITcVnAccess_TcVnDMatch	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.7.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnDMatch& stDMatch
)
```

Parameters

Name	Type	Description
stDMatch	TcVnDMatch [▶ 1643]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.7.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    TcVnDMatch& stDMatch
)
```

Parameters

Name	Type	Description
stDMatch	TcVnDMatch [▶ 1643]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.8 ITcVnAccess_TcVnKeyPoint

Offers an access interface for TcVnKeyPoint values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcVnAccess_TcVnKeyPoint

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1671]	ITcVnAccess_TcVnKeyPoint	Gets the value.
Set [▶ 1672]	ITcVnAccess_TcVnKeyPoint	Sets the value.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.8.1 Get

Gets the value.

Syntax

Definition:

```
HRESULT Get(
    TcVnKeyPoint& stKeyPoint
)
```

Parameters

Name	Type	Description
stKeyPoint	TcVnKeyPoint [▶ 1643]&	Returns the value.

Return value

HRESULT [▶ 122]

6.2.2.1.2.8.2 Set

Sets the value.

Syntax

Definition:

```
HRESULT Set(
    TcVnKeyPoint& stKeyPoint
)
```

Parameters

Name	Type	Description
stKeyPoint	TcVnKeyPoint [▸ 1643]&	The value to set.

Return value

HRESULT [▸ 122]

6.2.2.1.2.9 ITcVnAccess_TcVnPoint2_DINT

Offers an access interface for TcVnPoint2_DINT values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
        ITcVnAccess_TcVnPoint2_DINT

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1672]	ITcVnAccess_TcVnPoint2_DINT	Gets the value.
Set [▸ 1673]	ITcVnAccess_TcVnPoint2_DINT	Sets the value.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.9.1 Get

Gets the value.

Syntax

Definition:

```
HRESULT Get(
    TcVnPoint2_DINT& aPoint
)
```

Parameters

Name	Type	Description
aPoint	TcVnPoint2_DINT [▶ 1588]&	Returns the value.

Return value

HRESULT [▶ 122]

6.2.2.1.2.9.2 Set

Sets the value.

Syntax

Definition:

```
HRESULT Set(
    TcVnPoint2_DINT& aPoint
)
```

Parameters

Name	Type	Description
aPoint	TcVnPoint2_DINT [▶ 1588]&	The value to set.

Return value

HRESULT [▶ 122]

6.2.2.1.2.10 ITcVnAccess_TcVnPoint2_LREAL

Offers an access interface for TcVnPoint2_LREAL values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcVnAccess_TcVnPoint2_LREAL

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1674]	ITcVnAccess_TcVnPoint2_LREAL	Gets the value
Set [▶ 1674]	ITcVnAccess_TcVnPoint2_LREAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.10.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnPoint2_LREAL& aPoint
)
```

Parameters

Name	Type	Description
aPoint	TcVnPoint2_LREAL [▶ 1588]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.10.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    TcVnPoint2_LREAL& aPoint
)
```

Parameters

Name	Type	Description
aPoint	TcVnPoint2_LREAL [▶ 1588]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.11 ITcVnAccess_TcVnPoint2_REAL

Offers an access interface for TcVnPoint2_REAL values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
        ITcVnAccess_TcVnPoint2_REAL

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1675]	ITcVnAccess_TcVnPoint2_REAL	Gets the value
Set [▶ 1675]	ITcVnAccess_TcVnPoint2_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.11.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnPoint2_REAL& aPoint
)
```

Parameters

Name	Type	Description
aPoint	TcVnPoint2_REAL [▶ 1588]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.11.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    TcVnPoint2_REAL& aPoint
)
```

Parameters

Name	Type	Description
aPoint	TcVnPoint2_REAL [▶ 1588]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.12 ITcVnAccess_TcVnPoint3_LREAL

Offers an access interface for TcVnPoint3_LREAL values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
ITcVnAccess_TcVnPoint3_LREAL

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1676]	ITcVnAccess_TcVnPoint3_LREAL	Gets the value
Set [▸ 1676]	ITcVnAccess_TcVnPoint3_LREAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.12.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnPoint3_LREAL& aPoint
)
```

Parameters

Name	Type	Description
aPoint	TcVnPoint3_LREAL [▸ 1588]&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.2.12.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    TcVnPoint3_LREAL& aPoint
)
```

Parameters

Name	Type	Description
aPoint	TcVnPoint3_LREAL [▶ 1588]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.13 ITcVnAccess_TcVnPoint3_REAL

Offers an access interface for TcVnPoint3_REAL values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcVnAccess_TcVnPoint3_REAL

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1677]	ITcVnAccess_TcVnPoint3_REAL	Gets the value
Set [▶ 1678]	ITcVnAccess_TcVnPoint3_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.13.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnPoint3_REAL& aPoint
)
```

Parameters

Name	Type	Description
aPoint	TcVnPoint3_REAL [▶ 1588]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.13.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    TcVnPoint3_REAL& aPoint
)
```

Parameters

Name	Type	Description
aPoint	TcVnPoint3_REAL [▶ 1588]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.14 ITcVnAccess_TcVnRectangle_DINT

Offers an access interface for TcVnRectangle_DINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
        ITcVnAccess_TcVnRectangle_DINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1678]	ITcVnAccess_TcVnRectangle_DINT	Gets the value
Set [▶ 1679]	ITcVnAccess_TcVnRectangle_DINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.14.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnRectangle_DINT& stRectangle
)
```

Parameters

Name	Type	Description
stRectangle	TcVnRectangle_DINT [▸ 1657]&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.2.14.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    TcVnRectangle_DINT& stRectangle
)
```

Parameters

Name	Type	Description
stRectangle	TcVnRectangle_DINT [▸ 1657]&	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.2.15 ITcVnAccess_TcVnVector2_DINT

Offers an access interface for TcVnVector2_DINT values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
    ITcVnAccess_TcVnVector2_DINT

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1680]	ITcVnAccess_TcVnVector2_DINT	Gets the value
Set [▸ 1680]	ITcVnAccess_TcVnVector2_DINT	Sets the value.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.15.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnVector2_DINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector2_DINT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.15.2 Set

Sets the value.

Syntax

Definition:

```
HRESULT Set(
    TcVnVector2_DINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector2_DINT [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.16 ITcVnAccess_TcVnVector2_INT

Offers an access interface for TcVnVector2_INT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
　　ITcVnAccess_TcVnVector2_INT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1681]	ITcVnAccess_TcVnVector2_INT	Gets the value
Set [▶ 1681]	ITcVnAccess_TcVnVector2_INT	Sets the value.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.16.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnVector2_INT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector2_INT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.16.2 Set

Sets the value.

Syntax

Definition:

```
HRESULT Set(
    TcVnVector2_INT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector2_INT [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.17 ITcVnAccess_TcVnVector2_REAL

Offers an access interface for TcVnVector2_REAL values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
ITcVnAccess_TcVnVector2_REAL

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1682]	ITcVnAccess_TcVnVector2_REAL	Gets the value
Set [▸ 1682]	ITcVnAccess_TcVnVector2_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.17.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnVector2_REAL& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector2_REAL [▸ 1590]&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.2.17.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    TcVnVector2_REAL& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector2_REAL [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.18 ITcVnAccess_TcVnVector2_SINT

Offers an access interface for TcVnVector2_SINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_TcVnVector2_SINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1683]	ITcVnAccess_TcVnVector2_SINT	Gets the value
Set [▶ 1684]	ITcVnAccess_TcVnVector2_SINT	Sets the value.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.18.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnVector2_SINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector2_SINT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.18.2 Set

Sets the value.

Syntax

Definition:

```
HRESULT Set(
    TcVnVector2_SINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector2_SINT [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.19 ITcVnAccess_TcVnVector2_UINT

Offers an access interface for TcVnVector2_UINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
        ITcVnAccess_TcVnVector2_UINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1684]	ITcVnAccess_TcVnVector2_UINT	Gets the value
Set [▶ 1685]	ITcVnAccess_TcVnVector2_UINT	Sets the value.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.19.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnVector2_UINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector2_UINT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.19.2 Set

Sets the value.

Syntax

Definition:

```
HRESULT Set(
    TcVnVector2_UINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector2_UINT [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.20 ITcVnAccess_TcVnVector2_USINT

Offers an access interface for TcVnVector2_USINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcVnAccess_TcVnVector2_USINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1686]	ITcVnAccess_TcVnVector2_USINT	Gets the value
Set [▶ 1686]	ITcVnAccess_TcVnVector2_USINT	Sets the value.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.20.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnVector2_USINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector2_USINT [▸ 1590]&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.2.20.2 Set

Sets the value.

Syntax

Definition:

```
HRESULT Set(
    TcVnVector2_USINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector2_USINT [▸ 1590]&	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.2.21 ITcVnAccess_TcVnVector3_INT

Offers an access interface for TcVnVector3_INT values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
    ITcVnAccess_TcVnVector3_INT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1687]	ITcVnAccess_TcVnVector3_INT	Gets the value
Set [▶ 1687]	ITcVnAccess_TcVnVector3_INT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.21.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnVector3_INT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector3_INT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.21.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    TcVnVector3_INT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector3_INT [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.22 ITcVnAccess_TcVnVector3_REAL

Offers an access interface for TcVnVector3_REAL values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
    ITcVnAccess_TcVnVector3_REAL

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1688]	ITcVnAccess_TcVnVector3_REAL	Gets the value
Set [▸ 1688]	ITcVnAccess_TcVnVector3_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.22.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnVector3_REAL& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector3_REAL [▸ 1590]&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.2.22.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    TcVnVector3_REAL& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector3_REAL [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.23 ITcVnAccess_TcVnVector3_SINT

Offers an access interface for TcVnVector3_SINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_TcVnVector3_SINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1689]	ITcVnAccess_TcVnVector3_SINT	Gets the value
Set [▶ 1690]	ITcVnAccess_TcVnVector3_SINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.23.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnVector3_SINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector3_SINT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.23.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    TcVnVector3_SINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector3_SINT [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.24 ITcVnAccess_TcVnVector3_UINT

Offers an access interface for TcVnVector3_UINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
        ITcVnAccess_TcVnVector3_UINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1690]	ITcVnAccess_TcVnVector3_UINT	Gets the value
Set [▶ 1691]	ITcVnAccess_TcVnVector3_UINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.24.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnVector3_UINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector3_UINT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.24.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    TcVnVector3_UINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector3_UINT [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.25 ITcVnAccess_TcVnVector3_USINT

Offers an access interface for TcVnVector3_USINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcVnAccess_TcVnVector3_USINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1692]	ITcVnAccess_TcVnVector3_USINT	Gets the value
Set [▶ 1692]	ITcVnAccess_TcVnVector3_USINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.25.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnVector3_USINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector3_USINT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.25.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    TcVnVector3_USINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector3_USINT [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.26 ITcVnAccess_TcVnVector4_DINT

Offers an access interface for TcVnVector4_DINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_TcVnVector4_DINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1693]	ITcVnAccess_TcVnVector4_DINT	Gets the value
Set [▶ 1693]	ITcVnAccess_TcVnVector4_DINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.26.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnVector4_DINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector4_DINT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.26.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    TcVnVector4_DINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector4_DINT [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.27 ITcVnAccess_TcVnVector4_INT

Offers an access interface for TcVnVector4_INT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_TcVnVector4_INT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1694]	ITcVnAccess_TcVnVector4_INT	Gets the value
Set [▶ 1694]	ITcVnAccess_TcVnVector4_INT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.27.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnVector4_INT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector4_INT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.27.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    TcVnVector4_INT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector4_INT [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.28 ITcVnAccess_TcVnVector4_LREAL

Offers an access interface for TcVnVector4_LREAL values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_TcVnVector4_LREAL

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1695]	ITcVnAccess_TcVnVector4_LREAL	Gets the value
Set [▶ 1696]	ITcVnAccess_TcVnVector4_LREAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.28.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnVector4_LREAL& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector4_LREAL [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.28.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    TcVnVector4_LREAL& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector4_LREAL [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.29 ITcVnAccess_TcVnVector4_SINT

Offers an access interface for TcVnVector4_SINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
        ITcVnAccess_TcVnVector4_SINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1696]	ITcVnAccess_TcVnVector4_SINT	Gets the value
Set [▶ 1697]	ITcVnAccess_TcVnVector4_SINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.29.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnVector4_SINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector4_SINT [▸ 1590]&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.2.29.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    TcVnVector4_SINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector4_SINT [▸ 1590]&	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.2.30 ITcVnAccess_TcVnVector4_UINT

Offers an access interface for TcVnVector4_UINT values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
    ITcVnAccess_TcVnVector4_UINT

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1698]	ITcVnAccess_TcVnVector4_UINT	Gets the value
Set [▸ 1698]	ITcVnAccess_TcVnVector4_UINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.30.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnVector4_UINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector4_UINT [▸ 1590]&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.2.30.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    TcVnVector4_UINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector4_UINT [▸ 1590]&	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.2.31 ITcVnAccess_TcVnVector4_USINT

Offers an access interface for TcVnVector4_USINT values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
ITcVnAccess_TcVnVector4_USINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1699]	ITcVnAccess_TcVnVector4_USINT	Gets the value
Set [▶ 1699]	ITcVnAccess_TcVnVector4_USINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.31.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    TcVnVector4_USINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector4_USINT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.31.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    TcVnVector4_USINT& aVector
)
```

Parameters

Name	Type	Description
aVector	TcVnVector4_USINT [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.32 ITcVnAccess_UDINT

Offers an access interface for UDINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
        ITcVnAccess_UDINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1700]	ITcVnAccess_UDINT	Gets the value
Set [▶ 1700]	ITcVnAccess_UDINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.32.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    ULONG& nValue
)
```

Parameters

Name	Type	Description
nValue	ULONG&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.32.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    ULONG nValue
)
```

Parameters

Name	Type	Description
nValue	ULONG	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.33 ITcVnAccess_UINT

Offers an access interface for UINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcVnAccess_UINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1701]	ITcVnAccess_UINT	Gets the value
Set [▶ 1701]	ITcVnAccess_UINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.33.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    USHORT& nValue
)
```

Parameters

Name	Type	Description
nValue	USHORT&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.33.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    USHORT nValue
)
```

Parameters

Name	Type	Description
nValue	USHORT	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.2.34 ITcVnAccess_ULINT

Offers an access interface for ULINT values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
    ITcVnAccess_ULINT

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1702]	ITcVnAccess_ULINT	Gets the value
Set [▸ 1703]	ITcVnAccess_ULINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.34.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    ULONGLONG& nValue
)
```

Parameters

Name	Type	Description
nValue	ULONGLONG&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.34.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    ULONGLONG nValue
)
```

Parameters

Name	Type	Description
nValue	ULONGLONG	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.2.35 ITcVnAccess_USINT

Offers an access interface for USINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_USINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1703]	ITcVnAccess_USINT	Gets the value
Set [▶ 1704]	ITcVnAccess_USINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.2.35.1 Get

Gets the value

Syntax

Definition:

```
HRESULT Get(
    unsigned char& nValue
)
```

Parameters

Name	Type	Description
nValue	unsigned char&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.2.35.2 Set

Sets the value

Syntax

Definition:

```
HRESULT Set(
    unsigned char nValue
)
```

Parameters

Name	Type	Description
nValue	unsigned char	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3 ITcVnRandomAccess

接口ITcVnRandomAccess用于访问容器的任何元素 [▶ 133]，并由ITcVnRandomAccessIterator [▶ 1757]提供。与ITcVnAccess [▶ 1663]不同，容器中的任何元素都可以被寻址。

6.2.2.1.3.1 ITcVnRandomAccess_DINT

Offers a random access interface for DINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
        ITcVnAccess_DINT [▶ 1663]
                ITcVnRandomAccess_DINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1663]	ITcVnAccess_DINT	Gets the value
Set [▶ 1663]	ITcVnAccess_DINT	Sets the value
GetAt [▶ 1705]	ITcVnRandomAccess_DINT	Gets the value
SetAt [▶ 1705]	ITcVnRandomAccess_DINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.1.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG nOffset,
    LONG&    nValue
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
nValue	LONG&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.1.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG nOffset,
    LONG     nValue
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
nValue	LONG	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.2 ITcVnRandomAccess_INT

Offers a random access interface for INT values

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_INT [▶ 1664]
        ITcVnRandomAccess_INT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1664]	ITcVnAccess_INT	Gets the value
Set [▶ 1665]	ITcVnAccess_INT	Sets the value
GetAt [▶ 1706]	ITcVnRandomAccess_INT	Gets the value
SetAt [▶ 1706]	ITcVnRandomAccess_INT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.2.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG nOffset,
    SHORT&   nValue
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
nValue	SHORT&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.2.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG nOffset,
    SHORT    nValue
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
nValue	SHORT	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.3 ITcVnRandomAccess_ITcVnImage

Offers an random access interface for images.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_ITcVnImage [▶ 1665]
        ITcVnRandomAccess_ITcVnImage

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1665]	ITcVnAccess_ITcVnImage	Gets the image.
Set [▶ 1666]	ITcVnAccess_ITcVnImage	Sets the image.
GetAt [▶ 1707]	ITcVnRandomAccess_ITcVnImage	Gets the image.
SetAt [▶ 1708]	ITcVnRandomAccess_ITcVnImage	Sets the image.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.3.1 GetAt

Gets the image.

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG      nOffset,
    ITcVnImage*& ipImage
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position.
ipImage	ITcVnImage* [▶ 1797]&	Returns the image.

Return value

HRESULT [▶ 122]

6.2.2.1.3.3.2 SetAt

Sets the image.

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG     nOffset,
    ITcVnImage* ipImage
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
ipImage	ITcVnImage* [▶ 1797]	The image to set.

Return value

HRESULT [▶ 122]

6.2.2.1.3.4 ITcVnRandomAccess_LREAL

Offers a random access interface for LREAL values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
        ITcVnAccess_LREAL [▶ 1666]
                ITcVnRandomAccess_LREAL

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1666]	ITcVnAccess_LREAL	Gets the value
Set [▶ 1667]	ITcVnAccess_LREAL	Sets the value
GetAt [▶ 1709]	ITcVnRandomAccess_LREAL	Gets the value
SetAt [▶ 1709]	ITcVnRandomAccess_LREAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.4.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG nOffset,
    double&  fValue
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
fValue	double&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.3.4.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG nOffset,
    double   fValue
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position.
fValue	double	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.3.5 ITcVnRandomAccess_REAL

Offers a random access interface for REAL values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]

    ITcVnAccess_REAL [▸ 1667]

        ITcVnRandomAccess_REAL

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1668]	ITcVnAccess_REAL	Gets the value
Set [▶ 1668]	ITcVnAccess_REAL	Sets the value
GetAt [▶ 1710]	ITcVnRandomAccess_REAL	Gets the value
SetAt [▶ 1710]	ITcVnRandomAccess_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.5.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG nOffset,
    float&   fValue
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
fValue	float&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.5.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG nOffset,
    float    fValue
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
fValue	float	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.6 ITcVnRandomAccess_SINT

Offers a random access interface for SINT values

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_SINT [▶ 1668]
        ITcVnRandomAccess_SINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1669]	ITcVnAccess_SINT	Gets the value
Set [▶ 1669]	ITcVnAccess_SINT	Sets the value
GetAt [▶ 1711]	ITcVnRandomAccess_SINT	Gets the value
SetAt [▶ 1711]	ITcVnRandomAccess_SINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.6.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG nOffset,
    char&    nValue
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
nValue	char&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.6.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG nOffset,
    char     nValue
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
nValue	char	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.7 ITcVnRandomAccess_TcVnDMatch

Offers an random access interface for TcVnDMatch values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_TcVnDMatch [▶ 1670]
        ITcVnRandomAccess_TcVnDMatch

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1670]	ITcVnAccess_TcVnDMatch	Gets the value
Set [▶ 1670]	ITcVnAccess_TcVnDMatch	Sets the value
GetAt [▶ 1712]	ITcVnRandomAccess_TcVnDMatch	Gets the value
SetAt [▶ 1713]	ITcVnRandomAccess_TcVnDMatch	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.7.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG     nOffset,
    TcVnDMatch&  stDMatch
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position.
stDMatch	TcVnDMatch [▶ 1643]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.7.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG     nOffset,
    TcVnDMatch&  stDMatch
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
stDMatch	TcVnDMatch [▶ 1643]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.8 ITcVnRandomAccess_TcVnKeyPoint

Offers a random access interface for TcVnKeyPoint values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_TcVnKeyPoint [▶ 1671]
        ITcVnRandomAccess_TcVnKeyPoint

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1671]	ITcVnAccess_TcVnKeyPoint	Gets the value.
Set [▶ 1672]	ITcVnAccess_TcVnKeyPoint	Sets the value.
GetAt [▶ 1714]	ITcVnRandomAccess_TcVnKeyPoint	Gets the value
SetAt [▶ 1714]	ITcVnRandomAccess_TcVnKeyPoint	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.8.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG       nOffset,
    TcVnKeyPoint& stKeyPoint
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
stKeyPoint	TcVnKeyPoint [▶ 1643]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.8.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG       nOffset,
    TcVnKeyPoint& stKeyPoint
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
stKeyPoint	TcVnKeyPoint [▶ 1643]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.9 ITcVnRandomAccess_TcVnPoint2_DINT

Offers a random access interface for TcVnPoint2_DINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_TcVnPoint2_DINT [▶ 1672]
        ITcVnRandomAccess_TcVnPoint2_DINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1672]	ITcVnAccess_TcVnPoint2_DINT	Gets the value.
Set [▶ 1673]	ITcVnAccess_TcVnPoint2_DINT	Sets the value.
GetAt [▶ 1715]	ITcVnRandomAccess_TcVnPoint2_DINT	Gets the value
SetAt [▶ 1715]	ITcVnRandomAccess_TcVnPoint2_DINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.9.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG         nOffset,
    TcVnPoint2_DINT& aPoint
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aPoint	TcVnPoint2_DINT [▶ 1588]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.9.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG         nOffset,
    TcVnPoint2_DINT& aPoint
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aPoint	TcVnPoint2_DINT [▶ 1588]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.10 ITcVnRandomAccess_TcVnPoint2_LREAL

Offers a random access interface for TcVnPoint2_LREAL values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
  ITcVnAccess_TcVnPoint2_LREAL [▶ 1673]
    ITcVnRandomAccess_TcVnPoint2_LREAL

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1674]	ITcVnAccess_TcVnPoint2_LREAL	Gets the value
Set [▶ 1674]	ITcVnAccess_TcVnPoint2_LREAL	Sets the value
GetAt [▶ 1716]	ITcVnRandomAccess_TcVnPoint2_LREAL	Gets the value
SetAt [▶ 1717]	ITcVnRandomAccess_TcVnPoint2_LREAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.10.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG          nOffset,
    TcVnPoint2_LREAL& aPoint
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aPoint	TcVnPoint2_LREAL [▶ 1588]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.10.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG          nOffset,
    TcVnPoint2_LREAL& aPoint
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aPoint	TcVnPoint2_LREAL [▶ 1588]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.11 ITcVnRandomAccess_TcVnPoint2_REAL

Offers a random access interface for TcVnPoint2_REAL values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcVnAccess_TcVnPoint2_REAL [▶ 1674]
ITcVnRandomAccess_TcVnPoint2_REAL

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1675]	ITcVnAccess_TcVnPoint2_REAL	Gets the value
Set [▶ 1675]	ITcVnAccess_TcVnPoint2_REAL	Sets the value
GetAt [▶ 1718]	ITcVnRandomAccess_TcVnPoint2_REAL	Gets the value
SetAt [▶ 1718]	ITcVnRandomAccess_TcVnPoint2_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.11.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG          nOffset,
    TcVnPoint2_REAL&  aPoint
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aPoint	TcVnPoint2_REAL [▶ 1588]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.11.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG          nOffset,
    TcVnPoint2_REAL&  aPoint
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aPoint	TcVnPoint2_REAL [▶ 1588]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.12 ITcVnRandomAccess_TcVnPoint3_LREAL

Offers a random access interface for TcVnPoint3_LREAL values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_TcVnPoint3_LREAL [▶ 1676]
        ITcVnRandomAccess_TcVnPoint3_LREAL

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1676]	ITcVnAccess_TcVnPoint3_LREAL	Gets the value
Set [▶ 1676]	ITcVnAccess_TcVnPoint3_LREAL	Sets the value
GetAt [▶ 1719]	ITcVnRandomAccess_TcVnPoint3_LREAL	Gets the value
SetAt [▶ 1719]	ITcVnRandomAccess_TcVnPoint3_LREAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.12.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG           nOffset,
    TcVnPoint3_LREAL&  aPoint
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aPoint	TcVnPoint3_LREAL [▶ 1588]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.12.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG           nOffset,
    TcVnPoint3_LREAL&  aPoint
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aPoint	TcVnPoint3_LREAL [▸ 1588]&	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.3.13 ITcVnRandomAccess_TcVnPoint3_REAL

Offers a random access interface for TcVnPoint3_REAL values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
    ITcVnAccess_TcVnPoint3_REAL [▸ 1677]
        ITcVnRandomAccess_TcVnPoint3_REAL

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1677]	ITcVnAccess_TcVnPoint3_REAL	Gets the value
Set [▸ 1678]	ITcVnAccess_TcVnPoint3_REAL	Sets the value
GetAt [▸ 1720]	ITcVnRandomAccess_TcVnPoint3_REAL	Gets the value
SetAt [▸ 1721]	ITcVnRandomAccess_TcVnPoint3_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.13.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG          nOffset,
    TcVnPoint3_REAL& aPoint
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aPoint	TcVnPoint3_REAL [▶ 1588]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.13.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG         nOffset,
    TcVnPoint3_REAL& aPoint
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aPoint	TcVnPoint3_REAL [▶ 1588]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.14 ITcVnRandomAccess_TcVnRectangle_DINT

Offers a random access interface for TcVnRectangle_DINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_TcVnRectangle_DINT [▶ 1678]
        ITcVnRandomAccess_TcVnRectangle_DINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1678]	ITcVnAccess_TcVnRectangle_DINT	Gets the value
Set [▶ 1679]	ITcVnAccess_TcVnRectangle_DINT	Sets the value
GetAt [▶ 1722]	ITcVnRandomAccess_TcVnRectangle_DINT	Gets the value
SetAt [▶ 1722]	ITcVnRandomAccess_TcVnRectangle_DINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.14.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG             nOffset,
    TcVnRectangle_DINT&  stRectangle
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
stRectangle	TcVnRectangle_DINT [▶ 1657]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.14.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG             nOffset,
    TcVnRectangle_DINT&  stRectangle
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
stRectangle	TcVnRectangle_DINT [▶ 1657]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.15 ITcVnRandomAccess_TcVnVector2_DINT

Offers a random access interface for TcVnVector2_DINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_TcVnVector2_DINT [▶ 1679]
        ITcVnRandomAccess_TcVnVector2_DINT

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1680]	ITcVnAccess_TcVnVector2_DINT	Gets the value
Set [▸ 1680]	ITcVnAccess_TcVnVector2_DINT	Sets the value.
GetAt [▸ 1723]	ITcVnRandomAccess_TcVnVector2_DINT	Gets the value
SetAt [▸ 1723]	ITcVnRandomAccess_TcVnVector2_DINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.15.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG           nOffset,
    TcVnVector2_DINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector2_DINT [▸ 1590]&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.3.15.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG           nOffset,
    TcVnVector2_DINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector2_DINT [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.16 ITcVnRandomAccess_TcVnVector2_INT

Offers a random access interface for TcVnVector2_INT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_TcVnVector2_INT [▶ 1680]
        ITcVnRandomAccess_TcVnVector2_INT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1681]	ITcVnAccess_TcVnVector2_INT	Gets the value
Set [▶ 1681]	ITcVnAccess_TcVnVector2_INT	Sets the value.
GetAt [▶ 1724]	ITcVnRandomAccess_TcVnVector2_INT	Gets the value
SetAt [▶ 1725]	ITcVnRandomAccess_TcVnVector2_INT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.16.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG         nOffset,
    TcVnVector2_INT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector2_INT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.16.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG          nOffset,
    TcVnVector2_INT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector2_INT [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.17 ITcVnRandomAccess_TcVnVector2_REAL

Offers a random access interface for TcVnVector2_REAL values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
        ITcVnAccess_TcVnVector2_REAL [▶ 1682]
                ITcVnRandomAccess_TcVnVector2_REAL

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1682]	ITcVnAccess_TcVnVector2_REAL	Gets the value
Set [▶ 1682]	ITcVnAccess_TcVnVector2_REAL	Sets the value
GetAt [▶ 1726]	ITcVnRandomAccess_TcVnVector2_REAL	Gets the value
SetAt [▶ 1726]	ITcVnRandomAccess_TcVnVector2_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.17.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG          nOffset,
    TcVnVector2_REAL& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector2_REAL [▸ 1590]&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.3.17.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG          nOffset,
    TcVnVector2_REAL& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector2_REAL [▸ 1590]&	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.3.18 ITcVnRandomAccess_TcVnVector2_SINT

Offers a random access interface for TcVnVector2_SINT values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
        ITcVnAccess_TcVnVector2_SINT [▸ 1683]
                ITcVnRandomAccess_TcVnVector2_SINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1683]	ITcVnAccess_TcVnVector2_SINT	Gets the value
Set [▶ 1684]	ITcVnAccess_TcVnVector2_SINT	Sets the value.
GetAt [▶ 1727]	ITcVnRandomAccess_TcVnVector2_SINT	Gets the value
SetAt [▶ 1727]	ITcVnRandomAccess_TcVnVector2_SINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.18.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG          nOffset,
    TcVnVector2_SINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector2_SINT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.18.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG          nOffset,
    TcVnVector2_SINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector2_SINT [▸ 1590]&	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.3.19 ITcVnRandomAccess_TcVnVector2_UINT

Offers a random access interface for TcVnVector2_UINT values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
  ITcVnAccess_TcVnVector2_UINT [▸ 1684]
    ITcVnRandomAccess_TcVnVector2_UINT

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1684]	ITcVnAccess_TcVnVector2_UINT	Gets the value
Set [▸ 1685]	ITcVnAccess_TcVnVector2_UINT	Sets the value.
GetAt [▸ 1728]	ITcVnRandomAccess_TcVnVector2_UINT	Gets the value
SetAt [▸ 1729]	ITcVnRandomAccess_TcVnVector2_UINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.19.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG          nOffset,
    TcVnVector2_UINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector2_UINT [▸ 1590]&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.3.19.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG           nOffset,
    TcVnVector2_UINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector2_UINT [▸ 1590]&	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.3.20 ITcVnRandomAccess_TcVnVector2_USINT

Offers a random access interface for TcVnVector2_USINT values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
    ITcVnAccess_TcVnVector2_USINT [▸ 1685]
        ITcVnRandomAccess_TcVnVector2_USINT

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1686]	ITcVnAccess_TcVnVector2_USINT	Gets the value
Set [▸ 1686]	ITcVnAccess_TcVnVector2_USINT	Sets the value.
GetAt [▸ 1730]	ITcVnRandomAccess_TcVnVector2_USINT	Gets the value
SetAt [▸ 1730]	ITcVnRandomAccess_TcVnVector2_USINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.20.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG             nOffset,
    TcVnVector2_USINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector2_USINT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.20.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG             nOffset,
    TcVnVector2_USINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector2_USINT [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.21 ITcVnRandomAccess_TcVnVector3_INT

Offers a random access interface for TcVnVector3_INT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_TcVnVector3_INT [▶ 1686]
        ITcVnRandomAccess_TcVnVector3_INT

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1687]	ITcVnAccess_TcVnVector3_INT	Gets the value
Set [▸ 1687]	ITcVnAccess_TcVnVector3_INT	Sets the value
GetAt [▸ 1731]	ITcVnRandomAccess_TcVnVector3_INT	Gets the value
SetAt [▸ 1731]	ITcVnRandomAccess_TcVnVector3_INT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.21.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG          nOffset,
    TcVnVector3_INT&  aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector3_INT [▸ 1590]&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.3.21.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG          nOffset,
    TcVnVector3_INT&  aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector3_INT [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.22 ITcVnRandomAccess_TcVnVector3_REAL

Offers a random access interface for TcVnVector3_REAL values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_TcVnVector3_REAL [▶ 1688]
        ITcVnRandomAccess_TcVnVector3_REAL

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1688]	ITcVnAccess_TcVnVector3_REAL	Gets the value
Set [▶ 1688]	ITcVnAccess_TcVnVector3_REAL	Sets the value
GetAt [▶ 1732]	ITcVnRandomAccess_TcVnVector3_REAL	Gets the value
SetAt [▶ 1733]	ITcVnRandomAccess_TcVnVector3_REAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.22.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG          nOffset,
    TcVnVector3_REAL& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector3_REAL [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.22.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG          nOffset,
    TcVnVector3_REAL& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector3_REAL [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.23 ITcVnRandomAccess_TcVnVector3_SINT

Offers a random access interface for TcVnVector3_SINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_TcVnVector3_SINT [▶ 1689]
        ITcVnRandomAccess_TcVnVector3_SINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1689]	ITcVnAccess_TcVnVector3_SINT	Gets the value
Set [▶ 1690]	ITcVnAccess_TcVnVector3_SINT	Sets the value
GetAt [▶ 1734]	ITcVnRandomAccess_TcVnVector3_SINT	Gets the value
SetAt [▶ 1734]	ITcVnRandomAccess_TcVnVector3_SINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.23.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG          nOffset,
    TcVnVector3_SINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector3_SINT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.23.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG          nOffset,
    TcVnVector3_SINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector3_SINT [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.24 ITcVnRandomAccess_TcVnVector3_UINT

Offers a random access interface for TcVnVector3_UINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_TcVnVector3_UINT [▶ 1690]
        ITcVnRandomAccess_TcVnVector3_UINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1690]	ITcVnAccess_TcVnVector3_UINT	Gets the value
Set [▶ 1691]	ITcVnAccess_TcVnVector3_UINT	Sets the value
GetAt [▶ 1735]	ITcVnRandomAccess_TcVnVector3_UINT	Gets the value
SetAt [▶ 1735]	ITcVnRandomAccess_TcVnVector3_UINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.24.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG           nOffset,
    TcVnVector3_UINT&  aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector3_UINT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.24.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG           nOffset,
    TcVnVector3_UINT&  aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector3_UINT [▸ 1590]&	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.3.25 ITcVnRandomAccess_TcVnVector3_USINT

Offers a random access interface for TcVnVector3_USINT values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
    ITcVnAccess_TcVnVector3_USINT [▸ 1691]
        ITcVnRandomAccess_TcVnVector3_USINT

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1692]	ITcVnAccess_TcVnVector3_USINT	Gets the value
Set [▸ 1692]	ITcVnAccess_TcVnVector3_USINT	Sets the value
GetAt [▸ 1736]	ITcVnRandomAccess_TcVnVector3_USINT	Gets the value
SetAt [▸ 1737]	ITcVnRandomAccess_TcVnVector3_USINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.25.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG           nOffset,
    TcVnVector3_USINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector3_USINT [▸ 1590]&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.3.25.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG            nOffset,
    TcVnVector3_USINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector3_USINT [▸ 1590]&	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.3.26 ITcVnRandomAccess_TcVnVector4_DINT

Offers a random access interface for TcVnVector4_DINT values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
        ITcVnAccess_TcVnVector4_DINT [▸ 1692]
                ITcVnRandomAccess_TcVnVector4_DINT

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1693]	ITcVnAccess_TcVnVector4_DINT	Gets the value
Set [▸ 1693]	ITcVnAccess_TcVnVector4_DINT	Sets the value
GetAt [▸ 1738]	ITcVnRandomAccess_TcVnVector4_DINT	Gets the value
SetAt [▸ 1738]	ITcVnRandomAccess_TcVnVector4_DINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.26.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG          nOffset,
    TcVnVector4_DINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector4_DINT [▸ 1590]&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.3.26.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG          nOffset,
    TcVnVector4_DINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector4_DINT [▸ 1590]&	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.3.27 ITcVnRandomAccess_TcVnVector4_INT

Offers a random access interface for TcVnVector4_INT values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
        ITcVnAccess_TcVnVector4_INT [▸ 1694]
                ITcVnRandomAccess_TcVnVector4_INT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1694]	ITcVnAccess_TcVnVector4_INT	Gets the value
Set [▶ 1694]	ITcVnAccess_TcVnVector4_INT	Sets the value
GetAt [▶ 1739]	ITcVnRandomAccess_TcVnVector4_INT	Gets the value
SetAt [▶ 1739]	ITcVnRandomAccess_TcVnVector4_INT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.27.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG         nOffset,
    TcVnVector4_INT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector4_INT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.27.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG         nOffset,
    TcVnVector4_INT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector4_INT [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.28 ITcVnRandomAccess_TcVnVector4_LREAL

Offers a random access interface for TcVnVector4_LREAL values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_TcVnVector4_LREAL [▶ 1695]
        ITcVnRandomAccess_TcVnVector4_LREAL

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1695]	ITcVnAccess_TcVnVector4_LREAL	Gets the value
Set [▶ 1696]	ITcVnAccess_TcVnVector4_LREAL	Sets the value
GetAt [▶ 1740]	ITcVnRandomAccess_TcVnVector4_LREAL	Gets the value
SetAt [▶ 1741]	ITcVnRandomAccess_TcVnVector4_LREAL	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.28.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG           nOffset,
    TcVnVector4_LREAL& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector4_LREAL [▸ 1590]&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.3.28.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG          nOffset,
    TcVnVector4_LREAL& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector4_LREAL [▸ 1590]&	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.3.29 ITcVnRandomAccess_TcVnVector4_SINT

Offers a random access interface for TcVnVector4_SINT values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
    ITcVnAccess_TcVnVector4_SINT [▸ 1696]
        ITcVnRandomAccess_TcVnVector4_SINT

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1696]	ITcVnAccess_TcVnVector4_SINT	Gets the value
Set [▸ 1697]	ITcVnAccess_TcVnVector4_SINT	Sets the value
GetAt [▸ 1742]	ITcVnRandomAccess_TcVnVector4_SINT	Gets the value
SetAt [▸ 1742]	ITcVnRandomAccess_TcVnVector4_SINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.29.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG           nOffset,
    TcVnVector4_SINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector4_SINT [▸ 1590]&	Returns the value

Return value

HRESULT [▸ 122]

6.2.2.1.3.29.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG           nOffset,
    TcVnVector4_SINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector4_SINT [▸ 1590]&	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.3.30 ITcVnRandomAccess_TcVnVector4_UINT

Offers a random access interface for TcVnVector4_UINT values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
    ITcVnAccess_TcVnVector4_UINT [▸ 1697]
        ITcVnRandomAccess_TcVnVector4_UINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1698]	ITcVnAccess_TcVnVector4_UINT	Gets the value
Set [▶ 1698]	ITcVnAccess_TcVnVector4_UINT	Sets the value
GetAt [▶ 1743]	ITcVnRandomAccess_TcVnVector4_UINT	Gets the value
SetAt [▶ 1743]	ITcVnRandomAccess_TcVnVector4_UINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.30.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG          nOffset,
    TcVnVector4_UINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector4_UINT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.30.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG          nOffset,
    TcVnVector4_UINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector4_UINT [▸ 1590]&	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.3.31 ITcVnRandomAccess_TcVnVector4_USINT

Offers a random access interface for TcVnVector4_USINT values.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
ITcVnAccess_TcVnVector4_USINT [▸ 1698]
ITcVnRandomAccess_TcVnVector4_USINT

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1699]	ITcVnAccess_TcVnVector4_USINT	Gets the value
Set [▸ 1699]	ITcVnAccess_TcVnVector4_USINT	Sets the value
GetAt [▸ 1744]	ITcVnRandomAccess_TcVnVector4_USINT	Gets the value
SetAt [▸ 1745]	ITcVnRandomAccess_TcVnVector4_USINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.31.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG           nOffset,
    TcVnVector4_USINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector4_USINT [▶ 1590]&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.31.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG           nOffset,
    TcVnVector4_USINT& aVector
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
aVector	TcVnVector4_USINT [▶ 1590]&	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.32 ITcVnRandomAccess_UDINT

Offers a random access interface for UDINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnAccess_UDINT [▶ 1700]
        ITcVnRandomAccess_UDINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1700]	ITcVnAccess_UDINT	Gets the value
Set [▶ 1700]	ITcVnAccess_UDINT	Sets the value
GetAt [▶ 1746]	ITcVnRandomAccess_UDINT	Gets the value
SetAt [▶ 1746]	ITcVnRandomAccess_UDINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.32.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG nOffset,
    ULONG&   nValue
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
nValue	ULONG&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.32.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG nOffset,
    ULONG    nValue
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
nValue	ULONG	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.33 ITcVnRandomAccess_UINT

Offers a random access interface for UINT values

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcVnAccess_UINT [▶ 1701]
ITcVnRandomAccess_UINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1701]	ITcVnAccess_UINT	Gets the value
Set [▶ 1701]	ITcVnAccess_UINT	Sets the value
GetAt [▶ 1747]	ITcVnRandomAccess_UINT	Gets the value
SetAt [▶ 1747]	ITcVnRandomAccess_UINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.33.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG nOffset,
    USHORT&  nValue
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
nValue	USHORT&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.33.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG nOffset,
    USHORT   nValue
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
nValue	USHORT	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.3.34 ITcVnRandomAccess_ULINT

Offers a randowm access interface for ULINT values.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcVnAccess_ULINT [▶ 1702]
ITcVnRandomAccess_ULINT

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Get [▶ 1702]	ITcVnAccess_ULINT	Gets the value
Set [▶ 1703]	ITcVnAccess_ULINT	Sets the value
GetAt [▶ 1748]	ITcVnRandomAccess_ULINT	Gets the value
SetAt [▶ 1748]	ITcVnRandomAccess_ULINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.34.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG    nOffset,
    ULONGLONG&  nValue
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
nValue	ULONGLONG&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.34.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG  nOffset,
    ULONGLONG nValue
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
nValue	ULONGLONG	The value to set

Return value

HRESULT [▸ 122]

6.2.2.1.3.35 ITcVnRandomAccess_USINT

Offers a random access interface for USINT values

Inheritance Hierarchy

ITcUnknown [▸ 1810]
    ITcVnAccess_USINT [▸ 1703]
        ITcVnRandomAccess_USINT

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
Get [▸ 1703]	ITcVnAccess_USINT	Gets the value
Set [▸ 1704]	ITcVnAccess_USINT	Sets the value
GetAt [▸ 1749]	ITcVnRandomAccess_USINT	Gets the value
SetAt [▸ 1750]	ITcVnRandomAccess_USINT	Sets the value

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.3.35.1 GetAt

Gets the value

Syntax

Definition:

```
HRESULT GetAt(
    LONGLONG       nOffset,
    unsigned char& nValue
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
nValue	unsigned char&	Returns the value

Return value

HRESULT [▶ 122]

6.2.2.1.3.35.2 SetAt

Sets the value

Syntax

Definition:

```
HRESULT SetAt(
    LONGLONG      nOffset,
    unsigned char nValue
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset to the current position
nValue	unsigned char	The value to set

Return value

HRESULT [▶ 122]

6.2.2.1.4 Iterators

该组包含迭代器处理的接口，以便访问容器 [▶ 132]的元素。

6.2.2.1.4.1 ITcVnBidirectionalIterator

Offers an interface for a bidirectional iterator.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnIteratorBase [▶ 1754]
        ITcVnForwardIterator [▶ 1752]
            ITcVnBidirectionalIterator

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
CheckIfBegin [▶ 1754]	ITcVnIteratorBase	Checks if the iterator points to the first element.
CheckIfEnd [▶ 1755]	ITcVnIteratorBase	Checks if the iterator points to the past-the-end element (alternatively use F_VN_CheckIfIteratorIsAtEnd).
GetValueSize [▶ 1755]	ITcVnIteratorBase	Gets the memory size (in byte) required by the value the iterator points to.
GetValueTypeGuid [▶ 1755]	ITcVnIteratorBase	Gets the type GUID of the value the iterator points to.
SetToBegin [▶ 1756]	ITcVnIteratorBase	Sets the iterator to the first element (alternatively use F_VN_SetIteratorToBegin).
SetToEnd [▶ 1756]	ITcVnIteratorBase	Sets the iterator to the past-the-end element.
CheckIfEqualTo [▶ 1752]	ITcVnForwardIterator	Checks if iterator is equal to another iterator.
GetContainer [▶ 1753]	ITcVnForwardIterator	Gets a pointer to the current element converted into an ITcVnContainer interface and increment its reference counter (only possible for container types). (Alternatively use F_VN_GetContainer.)
Increment [▶ 1753]	ITcVnForwardIterator	Increments the iterator. (Alternatively use F_VN_IncrementIterator.)
SetContainer [▶ 1753]	ITcVnForwardIterator	Sets the current element using an ITcVnContainer interface (only possible for container types). (Alternatively use F_VN_SetContainer.)
Decrement [▶ 1751]	ITcVnBidirectionalIterator	Decrements the iterator.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.4.1.1 Decrement

Decrements the iterator.

Syntax

Definition:

```
HRESULT Decrement()
```

Return value

HRESULT [▶ 122]

6.2.2.1.4.2 ITcVnForwardIterator

Offers an interface for a forward iterator.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcVnIteratorBase [▶ 1754]
ITcVnForwardIterator

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
CheckIfBegin [▶ 1754]	ITcVnIteratorBase	Checks if the iterator points to the first element.
CheckIfEnd [▶ 1755]	ITcVnIteratorBase	Checks if the iterator points to the past-the-end element (alternatively use F_VN_CheckIfIteratorIsAtEnd).
GetValueSize [▶ 1755]	ITcVnIteratorBase	Gets the memory size (in byte) required by the value the iterator points to.
GetValueTypeGuid [▶ 1755]	ITcVnIteratorBase	Gets the type GUID of the value the iterator points to.
SetToBegin [▶ 1756]	ITcVnIteratorBase	Sets the iterator to the first element (alternatively use F_VN_SetIteratorToBegin).
SetToEnd [▶ 1756]	ITcVnIteratorBase	Sets the iterator to the past-the-end element.
CheckIfEqualTo [▶ 1752]	ITcVnForwardIterator	Checks if iterator is equal to another iterator.
GetContainer [▶ 1753]	ITcVnForwardIterator	Gets a pointer to the current element converted into an ITcVnContainer interface and increment its reference counter (only possible for container types). (Alternatively use F_VN_GetContainer.)
Increment [▶ 1753]	ITcVnForwardIterator	Increments the iterator. (Alternatively use F_VN_IncrementIterator.)
SetContainer [▶ 1753]	ITcVnForwardIterator	Sets the current element using an ITcVnContainer interface (only possible for container types). (Alternatively use F_VN_SetContainer.)

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.4.2.1 CheckIfEqualTo

Checks if iterator is equal to another iterator.

Syntax

Definition:

```
HRESULT CheckIfEqualTo(
    ITcVnForwardIterator* ipIterator
)
```

Parameters

Name	Type	Description
ipIterator	ITcVnForwardIterator* [▶ 1752]	Iterator interface to compare with

Return value

HRESULT [▶ 122]

6.2.2.1.4.2.2 GetContainer

Gets a pointer to the current element converted into an ITcVnContainer interface and increment its reference counter (only possible for container types). (Alternatively use F_VN_GetContainer.)

Syntax

Definition:

```
HRESULT GetContainer(
    ITcVnContainer** pipContainer
)
```

Parameters

Name	Type	Description
pipContainer	ITcVnContainer* [▶ 1760]*	Returns the container interface.

Return value

HRESULT [▶ 122]

6.2.2.1.4.2.3 Increment

Increments the iterator. (Alternatively use F_VN_IncrementIterator.)

Syntax

Definition:

```
HRESULT Increment()
```

Return value

HRESULT [▶ 122]

6.2.2.1.4.2.4 SetContainer

Sets the current element using an ITcVnContainer interface (only possible for container types). (Alternatively use F_VN_SetContainer.)

Syntax

Definition:

```
HRESULT SetContainer(
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
ipContainer	ITcVnContainer* [▶ 1760]	Container interface of which the content is to be assigned to the current element.

Return value

HRESULT [▶ 122]

6.2.2.1.4.3 ITcVnIteratorBase

Offers a base interface for iterators

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcVnIteratorBase

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
CheckIfBegin [▶ 1754]	ITcVnIteratorBase	Checks if the iterator points to the first element.
CheckIfEnd [▶ 1755]	ITcVnIteratorBase	Checks if the iterator points to the past-the-end element (alternatively use F_VN_CheckIfIteratorIsAtEnd).
GetValueSize [▶ 1755]	ITcVnIteratorBase	Gets the memory size (in byte) required by the value the iterator points to.
GetValueTypeGuid [▶ 1755]	ITcVnIteratorBase	Gets the type GUID of the value the iterator points to.
SetToBegin [▶ 1756]	ITcVnIteratorBase	Sets the iterator to the first element (alternatively use F_VN_SetIteratorToBegin).
SetToEnd [▶ 1756]	ITcVnIteratorBase	Sets the iterator to the past-the-end element.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.4.3.1 CheckIfBegin

Checks if the iterator points to the first element.

Syntax

Definition:

```
HRESULT CheckIfBegin()
```

Return value

HRESULT [▸ 122]

6.2.2.1.4.3.2 CheckIfEnd

Checks if the iterator points to the past-the-end element (alternatively use F_VN_CheckIfIteratorIsAtEnd).

Syntax

Definition:

```
HRESULT CheckIfEnd()
```

Return value

HRESULT [▸ 122]

6.2.2.1.4.3.3 GetValueSize

Gets the memory size (in byte) required by the value the iterator points to.

Syntax

Definition:

```
HRESULT GetValueSize(
    ULONGLONG& nSize
)
```

Parameters

Name	Type	Description
nSize	ULONGLONG&	Returns the element size in bytes.

Return value

HRESULT [▸ 122]

6.2.2.1.4.3.4 GetValueTypeGuid

Gets the type GUID of the value the iterator points to.

Syntax

Definition:

```
HRESULT GetValueTypeGuid(
    GUID& nTypeGuid
)
```

Parameters

Name	Type	Description
nTypeGuid	GUID&	Returns the type GUID.

Return value

HRESULT [▸ 122]

6.2.2.1.4.3.5 SetToBegin

Sets the iterator to the first element (alternatively use F_VN_SetIteratorToBegin).

Syntax

Definition:

```
HRESULT SetToBegin()
```

Return value

HRESULT [▶ 122]

6.2.2.1.4.3.6 SetToEnd

Sets the iterator to the past-the-end element.

Syntax

Definition:

```
HRESULT SetToEnd()
```

Return value

HRESULT [▶ 122]

6.2.2.1.4.4 ITcVnIteratorCopyCreator

Offers an interface providing a method for creating a new iterator to the same position.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcVnIteratorCopyCreator

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
Create [▶ 1756]	ITcVnIteratorCopyCreator	Create a new iterator pointing to the position of the calling iterator.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.4.4.1 Create

Create a new iterator pointing to the position of the calling iterator.

Syntax

Definition:

```
HRESULT Create(
    ITcVnForwardIterator** pipIterator
)
```

Parameters

Name	Type	Description
pipIterator	ITcVnForwardIterator* [▶ 1752]*	Returns the created iterator.

Return value

HRESULT [▶ 122]

6.2.2.1.4.5 ITcVnRandomAccessIterator

Offers an interface for a random access iterator.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
  ITcVnIteratorBase [▶ 1754]
    ITcVnForwardIterator [▶ 1752]
      ITcVnBidirectionalIterator [▶ 1750]
        ITcVnRandomAccessIterator

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
CheckIfBegin [▶ 1754]	ITcVnIteratorBase	Checks if the iterator points to the first element.
CheckIfEnd [▶ 1755]	ITcVnIteratorBase	Checks if the iterator points to the past-the-end element (alternatively use F_VN_CheckIfIteratorIsAtEnd).
GetValueSize [▶ 1755]	ITcVnIteratorBase	Gets the memory size (in byte) required by the value the iterator points to.
GetValueTypeGuid [▶ 1755]	ITcVnIteratorBase	Gets the type GUID of the value the iterator points to.
SetToBegin [▶ 1756]	ITcVnIteratorBase	Sets the iterator to the first element (alternatively use F_VN_SetIteratorToBegin).
SetToEnd [▶ 1756]	ITcVnIteratorBase	Sets the iterator to the past-the-end element.
CheckIfEqualTo [▶ 1752]	ITcVnForwardIterator	Checks if iterator is equal to another iterator.
GetContainer [▶ 1753]	ITcVnForwardIterator	Gets a pointer to the current element converted into an ITcVnContainer interface and increment its reference counter (only possible for container types). (Alternatively use F_VN_GetContainer.)
Increment [▶ 1753]	ITcVnForwardIterator	Increments the iterator. (Alternatively use F_VN_IncrementIterator.)
SetContainer [▶ 1753]	ITcVnForwardIterator	Sets the current element using an ITcVnContainer interface (only possible for container types). (Alternatively use F_VN_SetContainer.)
Decrement [▶ 1751]	ITcVnBidirectionalIterator	Decrements the iterator.
CheckIfGreaterThan [▶ 1759]	ITcVnRandomAccessIterator	Checks if the iterator is greater than another iterator.
CheckIfLessThan [▶ 1759]	ITcVnRandomAccessIterator	Checks if the iterator is less than another iterator.
GetContainerAt [▶ 1759]	ITcVnRandomAccessIterator	Gets a pointer to the element at a specific offset from the current element converted into an ITcVnContainer interface and increment its reference counter (only possible for container types). (Alternatively use F_VN_GetAt_ITcVnContainer.)
SetContainerAt [▶ 1760]	ITcVnRandomAccessIterator	Sets the element at a specific offset from the current element using an ITcVnContainer interface (only possible for container types). (Alternatively use F_VN_SetAt_ITcVnContainer.)

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.4.5.1 CheckIfGreaterThan

Checks if the iterator is greater than another iterator.

Syntax

Definition:

```
HRESULT CheckIfGreaterThan(
    ITcVnRandomAccessIterator* ipIterator
)
```

Parameters

Name	Type	Description
ipIterator	ITcVnRandomAccessIterator* [▶ 1757]	Iterator interface to compare with.

Return value

HRESULT [▶ 122]

6.2.2.1.4.5.2 CheckIfLessThan

Checks if the iterator is less than another iterator.

Syntax

Definition:

```
HRESULT CheckIfLessThan(
    ITcVnRandomAccessIterator* ipIterator
)
```

Parameters

Name	Type	Description
ipIterator	ITcVnRandomAccessIterator* [▶ 1757]	Iterator interface to compare with.

Return value

HRESULT [▶ 122]

6.2.2.1.4.5.3 GetContainerAt

Gets a pointer to the element at a specific offset from the current element converted into an ITcVnContainer interface and increment its reference counter (only possible for container types). (Alternatively use F_VN_GetAt_ITcVnContainer.)

Syntax

Definition:

```
HRESULT GetContainerAt(
    LONGLONG          nOffset,
    ITcVnContainer** pipContainer
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset from the current element.
pipContainer	ITcVnContainer* [▶ 1760]*	Returns the pointer to the container interface.

Return value

HRESULT [▶ 122]

6.2.2.1.4.5.4 SetContainerAt

Sets the element at a specific offset from the current element using an ITcVnContainer interface (only possible for container types). (Alternatively use F_VN_SetAt_ITcVnContainer.)

Syntax

Definition:

```
HRESULT SetContainerAt(
    LONGLONG        nOffset,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
nOffset	LONGLONG	Offset from the current element.
ipContainer	ITcVnContainer* [▶ 1760]	Container interface of which the content is to be assigned to the current element.

Return value

HRESULT [▶ 122]

6.2.2.1.5 ITcVnContainer

Offers an interface for an object container.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnContainer

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
CheckIfBasicContainer [▶ 1761]	ITcVnContainer	Checks if the container contains only basic (non-container) elements.
CheckIfEmpty [▶ 1762]	ITcVnContainer	Checks if the container is empty. (Alternatively use F_VN_CheckIfEmpty.)
GetBidirectionalIterator [▶ 1762]	ITcVnContainer	Gets an interface pointer to a bidirectional iterator (if this iterator type is supported by the container) and increment its reference counter.
GetElementNum [▶ 1762]	ITcVnContainer	Gets the size (number of elements) of the container. (Alternatively use F_VN_GetNumberOfElements.)
GetElementSize [▶ 1763]	ITcVnContainer	Gets the size (in byte) of each element in the container.
GetElementTypeGuid [▶ 1763]	ITcVnContainer	Gets the GUID of the container elements.
GetExportSize [▶ 1763]	ITcVnContainer	Gets combined size (in byte) of all elements in the container.
GetForwardIterator [▶ 1764]	ITcVnContainer	Gets an interface pointer to a forward iterator (if this iterator type is supported by the container) and increment its reference counter. (Alternatively use F_VN_GetForwardIterator.)
GetRandomAccessIterator [▶ 1764]	ITcVnContainer	Gets an interface pointer to a random access iterator (if this iterator type is supported by the container) and increment its reference counter. (Alternatively use F_VN_GetRandomAccessIterator.)
GetTypeGuid [▶ 1764]	ITcVnContainer	Gets GUID of the container.
GetTypeName [▶ 1765]	ITcVnContainer	Gets the container type name as a string.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.1.5.1 CheckIfBasicContainer

Checks if the container contains only basic (non-container) elements.

Syntax

Definition:

```
HRESULT CheckIfBasicContainer()
```

Return value

HRESULT [▶ 122]

6.2.2.1.5.2 CheckIfEmpty

Checks if the container is empty. (Alternatively use F_VN_CheckIfEmpty.)

Syntax

Definition:

```
HRESULT CheckIfEmpty()
```

Return value

HRESULT [▶ 122]

6.2.2.1.5.3 GetBidirectionalIterator

Gets an interface pointer to a bidirectional iterator (if this iterator type is supported by the container) and increment its reference counter.

Syntax

Definition:

```
HRESULT GetBidirectionalIterator(
    ITcVnBidirectionalIterator** pipIterator
)
```

Parameters

Name	Type	Description
pipIterator	ITcVnBidirectionalIterator* [▶ 1750]*	Returns the iterator interface.

Return value

HRESULT [▶ 122]

6.2.2.1.5.4 GetElementNum

Gets the size (number of elements) of the container. (Alternatively use F_VN_GetNumberOfElements.)

Syntax

Definition:

```
HRESULT GetElementNum(
    ULONGLONG& nElementNum
)
```

Parameters

Name	Type	Description
nElementNum	ULONGLONG&	Returns the number of elements in the container.

Return value

HRESULT [▶ 122]

6.2.2.1.5.5 GetElementSize

Gets the size (in byte) of each element in the container.

Syntax

Definition:

```
HRESULT GetElementSize(
    ULONGLONG& nSize
)
```

Parameters

Name	Type	Description
nSize	ULONGLONG&	Returns the size in byte of a single element in the container.

Return value

HRESULT [▶ 122]

6.2.2.1.5.6 GetElementTypeGuid

Gets the GUID of the container elements.

Syntax

Definition:

```
HRESULT GetElementTypeGuid(
    GUID& nTypeGuid
)
```

Parameters

Name	Type	Description
nTypeGuid	GUID&	Returns the GUID of the container elements.

Return value

HRESULT [▶ 122]

6.2.2.1.5.7 GetExportSize

Gets combined size (in byte) of all elements in the container.

Syntax

Definition:

```
HRESULT GetExportSize(
    ULONGLONG& nExportSize
)
```

Parameters

Name	Type	Description
nExportSize	ULONGLONG&	Returns the combined size (in byte) of all elements in the container.

Return value

HRESULT [▶ 122]

6.2.2.1.5.8 GetForwardIterator

Gets an interface pointer to a forward iterator (if this iterator type is supported by the container) and increment its reference counter. (Alternatively use F_VN_GetForwardIterator.)

Syntax

Definition:

```
HRESULT GetForwardIterator(
    ITcVnForwardIterator** pipIterator
)
```

Parameters

Name	Type	Description
pipIterator	ITcVnForwardIterator* [▶ 1752]*	Returns the iterator interface.

Return value

HRESULT [▶ 122]

6.2.2.1.5.9 GetRandomAccessIterator

Gets an interface pointer to a random access iterator (if this iterator type is supported by the container) and increment its reference counter. (Alternatively use F_VN_GetRandomAccessIterator.)

Syntax

Definition:

```
HRESULT GetRandomAccessIterator(
    ITcVnRandomAccessIterator** pipIterator
)
```

Parameters

Name	Type	Description
pipIterator	ITcVnRandomAccessIterator* [▶ 1757]*	Returns the iterator interface.

Return value

HRESULT [▶ 122]

6.2.2.1.5.10 GetTypeGuid

Gets GUID of the container.

Syntax

Definition:

```
HRESULT GetTypeGuid(
    GUID& nTypeGuid
)
```

Parameters

Name	Type	Description
nTypeGuid	GUID&	Returns the GUID of the container.

Return value

HRESULT [▸ 122]

6.2.2.1.5.11 GetTypeName

Gets the container type name as a string.

Syntax

Definition:

```
HRESULT GetTypeName(
    PCHAR  sTypeName,
    USHORT nMaxLen
)
```

Parameters

Name	Type	Description
sTypeName	PCHAR	Returns the container type name as a string.
nMaxLen	USHORT	Maximum string length allowed to be written in sTypeName

Return value

HRESULT [▸ 122]

6.2.2.2 Images

该组包含用于记录和处理图像 [▸ 130]的接口。

6.2.2.2.1 Acquisition

该组包含用于处理图像采集的接口。

6.2.2.2.1.1 ITcIoFileImageAcquisition

Interface for file image acquisition.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
    ITcIoImageAcquisition [▸ 1770]
        ITcIoFileImageAcquisition

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
RegisterReceiver [▶ 1770]	ITcIoImageAcquisition	Register callback interface.
UnregisterReceiver [▶ 1771]	ITcIoImageAcquisition	Unregister callback interface.
OpenCamera [▶ 1771]	ITcIoImageAcquisition	Open camera (open the control channel, open a stream channel, and activate the GVSP receiver).
CloseCamera [▶ 1771]	ITcIoImageAcquisition	Close camera (deactivate the GVSP receiver, close the stream channel, and close the control channel).
StartAcquisition [▶ 1771]	ITcIoImageAcquisition	Start the image acquisition.
StopAcquisition [▶ 1772]	ITcIoImageAcquisition	Stop the image acquisition.
SoftwareTrigger [▶ 1772]	ITcIoImageAcquisition	Initialize the software trigger and trigger it. The initialization step is skipped if it was executed before.
InitializeCamera [▶ 1772]	ITcIoImageAcquisition	Send initialization commands to the camera.
CheckConnection [▶ 1766]	ITcIoFileImageAcquisition	Checks the connection
TriggerImage [▶ 1767]	ITcIoFileImageAcquisition	Initialize the software trigger and trigger a single image. Skips nSkipImages and triggers the capturing of image nSkipImages+1.
TriggerImageByName [▶ 1767]	ITcIoFileImageAcquisition	Initialize the software trigger and trigger a single image specified by its name in the client assistant.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.1.1.1 *CheckConnection*

Checks the connection

Syntax

Definition:

```
HRESULT CheckConnection()
```

Return value

HRESULT [▶ 122]

6.2.2.2.1.1.2 TriggerImage

Initialize the software trigger and trigger a single image. Skips nSkipImages and triggers the capturing of image nSkipImages+1.

Syntax

Definition:

```
HRESULT TriggerImage(
    LONG nSkipImages
)
```

Parameters

Name	Type	Description
nSkipImages	LONG	Number of images to skip

Return value

HRESULT [▶ 122]

6.2.2.2.1.1.3 TriggerImageByName

Initialize the software trigger and trigger a single image specified by its name in the client assistant.

Syntax

Definition:

```
HRESULT TriggerImageByName(
    PCCH sImageName
)
```

Parameters

Name	Type	Description
sImageName	PCCH	Image name to trigger

Return value

HRESULT [▶ 122]

6.2.2.2.1.2 ITcIoFileImageRecv

Interface for a image receiver.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcIoImageRecv [▶ 1773]
ITcIoFileImageRecv

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
ReceiveImage [▸ 1773]	ITcIoImageRecv	Receive an image from an instance of ITcIoGevImageAcquisition.
ReceiveOpResult [▸ 1773]	ITcIoImageRecv	Receive an operation result from an instance of ITcIoGevImageAcquisition.
ReceiveImage [▸ 1768]	ITcIoFileImageRecv	Receive an image from an instance of ITcIoFileImageAcquisition.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.1.2.1 ReceiveImage

Receive an image from an instance of ITcIoFileImageAcquisition.

Syntax

Definition:

```
HRESULT ReceiveImage(
    ITcVnImageBase* ipImage,
    PCHAR           sFileName,
    HRESULT         hrAcquisitionResult
)
```

Parameters

Name	Type	Description
ipImage	ITcVnImageBase* [▸ 1799]	Interface pointer to the image.
sFileName	PCHAR	Returns the filename to the image.
hrAcquisitionResult	HRESULT [▸ 122]	HRESULT indicating the status of the image acquisition.

Return value

HRESULT [▸ 122]

6.2.2.2.1.3 ITcIoGevImageAcquisition

Interface for GigE Vision image acquisition.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
    ITcIoImageAcquisition [▸ 1770]
        ITcIoGevImageAcquisition

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
RegisterReceiver [▶ 1770]	ITcIoImageAcquisition	Register callback interface.
UnregisterReceiver [▶ 1771]	ITcIoImageAcquisition	Unregister callback interface.
OpenCamera [▶ 1771]	ITcIoImageAcquisition	Open camera (open the control channel, open a stream channel, and activate the GVSP receiver).
CloseCamera [▶ 1771]	ITcIoImageAcquisition	Close camera (deactivate the GVSP receiver, close the stream channel, and close the control channel).
StartAcquisition [▶ 1771]	ITcIoImageAcquisition	Start the image acquisition.
StopAcquisition [▶ 1772]	ITcIoImageAcquisition	Stop the image acquisition.
SoftwareTrigger [▶ 1772]	ITcIoImageAcquisition	Initialize the software trigger and trigger it. The initialization step is skipped if it was executed before.
InitializeCamera [▶ 1772]	ITcIoImageAcquisition	Send initialization commands to the camera.
CheckConnection [▶ 1769]	ITcIoGevImageAcquisition	Checks the camera connection

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.1.3.1 CheckConnection

Checks the camera connection

Syntax

Definition:

```
HRESULT CheckConnection(
    GEV_CAMERA_STATE& eAssumedState,
    GEV_CAMERA_STATE& eActualState
)
```

Parameters

Name	Type	Description
eAssumedState	GEV_CAMERA_STATE [▶ 1637]&	The internally assumed state of the camera.
eActualState	GEV_CAMERA_STATE [▶ 1637]&	The actually observable state of the camera.

Return value

HRESULT [▶ 122]

6.2.2.2.1.4 ITcIoImageAcquisition

Interface for image acquisition.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
        ITcIoImageAcquisition

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
RegisterReceiver [▶ 1770]	ITcIoImageAcquisition	Register callback interface.
UnregisterReceiver [▶ 1771]	ITcIoImageAcquisition	Unregister callback interface.
OpenCamera [▶ 1771]	ITcIoImageAcquisition	Open camera (open the control channel, open a stream channel, and activate the GVSP receiver).
CloseCamera [▶ 1771]	ITcIoImageAcquisition	Close camera (deactivate the GVSP receiver, close the stream channel, and close the control channel).
StartAcquisition [▶ 1771]	ITcIoImageAcquisition	Start the image acquisition.
StopAcquisition [▶ 1772]	ITcIoImageAcquisition	Stop the image acquisition.
SoftwareTrigger [▶ 1772]	ITcIoImageAcquisition	Initialize the software trigger and trigger it. The initialization step is skipped if it was executed before.
InitializeCamera [▶ 1772]	ITcIoImageAcquisition	Send initialization commands to the camera.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.1.4.1 RegisterReceiver

Register callback interface.

Syntax

Definition:

```
HRESULT RegisterReceiver(
    ITcIoImageRecv* ipRecv
)
```

Parameters

Name	Type	Description
ipRecv	ITcIoImageRecv* [▶ 1773]	Pointer to an interface containing the callback function.

Return value

HRESULT [▶ 122]

6.2.2.2.1.4.2 UnregisterReceiver

Unregister callback interface.

Syntax

Definition:

```
HRESULT UnregisterReceiver()
```

Return value

HRESULT [▶ 122]

6.2.2.2.1.4.3 OpenCamera

Open camera (open the control channel, open a stream channel, and activate the GVSP receiver).

Syntax

Definition:

```
HRESULT OpenCamera()
```

Return value

HRESULT [▶ 122]

6.2.2.2.1.4.4 CloseCamera

Close camera (deactivate the GVSP receiver, close the stream channel, and close the control channel).

Syntax

Definition:

```
HRESULT CloseCamera()
```

Return value

HRESULT [▶ 122]

6.2.2.2.1.4.5 StartAcquisition

Start the image acquisition.

Syntax

Definition:

```
HRESULT StartAcquisition()
```

Return value

HRESULT [▶ 122]

6.2.2.2.1.4.6 StopAcquisition

Stop the image acquisition.

Syntax

Definition:

```
HRESULT StopAcquisition()
```

Return value

HRESULT [▶ 122]

6.2.2.2.1.4.7 SoftwareTrigger

Initialize the software trigger and trigger it. The initialization step is skipped if it was executed before.

Syntax

Definition:

```
HRESULT SoftwareTrigger(
    bool bSplitConcatenatedCommands,
    bool bOmitAcknowledgement
)
```

Parameters

Name	Type	Description
bSplitConcatenatedCommands	bool	If true, multi-read and multi-write commands are split into sequences of single read and single write commands, respectively.
bOmitAcknowledgement	bool	Indicates that no acknowledge packet should be requested.

Return value

HRESULT [▶ 122]

6.2.2.2.1.4.8 InitializeCamera

Send initialization commands to the camera.

Syntax

Definition:

```
HRESULT InitializeCamera()
```

Return value

HRESULT [▶ 122]

6.2.2.2.1.5 ITcIoImageRecv

Interface for a image receiver.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
        ITcIoImageRecv

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
ReceiveImage [▶ 1773]	ITcIoImageRecv	Receive an image from an instance of ITcIoGevImageAcquisition.
ReceiveOpResult [▶ 1773]	ITcIoImageRecv	Receive an operation result from an instance of ITcIoGevImageAcquisition.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.1.5.1 ReceiveImage

Receive an image from an instance of ITcIoGevImageAcquisition.

Syntax

Definition:

```
HRESULT ReceiveImage(
    ITcVnImageBase* ipImage,
    HRESULT         hrAcquisitionResult
)
```

Parameters

Name	Type	Description
ipImage	ITcVnImageBase* [▶ 1799]	Interface pointer to the image.
hrAcquisitionResult	HRESULT [▶ 122]	HRESULT indicating the status of the image acquisition.

Return value

HRESULT [▶ 122]

6.2.2.2.1.5.2 ReceiveOpResult

Receive an operation result from an instance of ITcIoGevImageAcquisition.

Syntax

Definition:

```
HRESULT ReceiveOpResult(
    HRESULT hrOperationResult
)
```

Parameters

Name	Type	Description
hrOperationResult	HRESULT [▸ 122]	HRESULT indicating the status of the performed operation.

Return value

HRESULT [▸ 122]

6.2.2.2.1.6 ITcVnFileImageProvider

Interface for an image provider for images from outside realtime environment.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
    ITcVnImageProvider [▸ 1779]
        ITcVnFileImageProvider

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
CheckIfBusy [▶ 1779]	ITcVnImageProvider	Check whether the module is busy; i.e, an operation is in progress.
CheckIfCameraConnected [▶ 1780]	ITcVnImageProvider	Check if the camera is properly connected (Returns S_OK if connected and S_FALSE if not. Can only check software connection, i.e. the check is unable to determine if the cable is connected or not in case that S_FALSE is returned).
GetCurrentImage [▶ 1780]	ITcVnImageProvider	Gets the current image by detaching its internal reference.
GetError [▶ 1780]	ITcVnImageProvider	Gets result of the last operation performed.
OpenCamera [▶ 1780]	ITcVnImageProvider	Open camera (open the control channel, open a stream channel, and activate the GVSP receiver).
CloseCamera [▶ 1781]	ITcVnImageProvider	Close camera (deactivate the GVSP receiver, close the stream channel, and close the control channel).
StartAcquisition [▶ 1781]	ITcVnImageProvider	Start image acquisition.
StopAcquisition [▶ 1781]	ITcVnImageProvider	Stop image acquisition.
SoftwareTrigger [▶ 1781]	ITcVnImageProvider	Initialize the software trigger and trigger it. The initialization step is skipped if it was executed before.
InitializeCamera [▶ 1782]	ITcVnImageProvider	Set camera to initial state.
GetCurrentImageAndFileName [▶ 1775]	ITcVnFileImageProvider	Get the current image by detaching its internal reference.
TriggerImage [▶ 1776]	ITcVnFileImageProvider	Initialize the software trigger and trigger a single image. Skips nSkipImages and triggers the capturing of image nSkipImages+1.
TriggerImageByName [▶ 1776]	ITcVnFileImageProvider	Initialize the software trigger and trigger a single image specified by its name in the client assistant.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.1.6.1 GetCurrentImageAndFileName

Get the current image by detaching its internal reference.

Syntax

Definition:

```
HRESULT GetCurrentImageAndFileName(
    ITcVnImage** pipImage,
    PCHAR        sFileName,
    USHORT       nMaxLen
)
```

Parameters

Name	Type	Description
pipImage	ITcVnImage* [▸ 1797]*	pointer to the interface pointer to be returned, might be 0 if the image acquisition failed
sFileName	PCHAR	returns the filename as a string
nMaxLen	USHORT	maximum string length allowed to be written in sFileName

Return value

HRESULT [▸ 122]

6.2.2.2.1.6.2 TriggerImage

Initialize the software trigger and trigger a single image. Skips nSkipImages and triggers the capturing of image nSkipImages+1.

Syntax

Definition:

```
HRESULT TriggerImage(
    LONG nSkipImages
)
```

Parameters

Name	Type	Description
nSkipImages	LONG	Number of images to skip

Return value

HRESULT [▸ 122]

6.2.2.2.1.6.3 TriggerImageByName

Initialize the software trigger and trigger a single image specified by its name in the client assistant.

Syntax

Definition:

```
HRESULT TriggerImageByName(
    PCCH sImageName
)
```

Parameters

Name	Type	Description
sImageName	PCCH	Image name to trigger

Return value

HRESULT [▸ 122]

6.2.2.2.1.7 ITcVnGevImageProvider

Interface for a GigE Vision image provider.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
  ITcVnImageProvider [▸ 1779]
    ITcVnGevImageProvider

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
CheckIfBusy [▸ 1779]	ITcVnImageProvider	Check whether the module is busy; i.e, an operation is in progress.
CheckIfCameraConnected [▸ 1780]	ITcVnImageProvider	Check if the camera is properly connected (Returns S_OK if connected and S_FALSE if not. Can only check software connection, i.e. the check is unable to determine if the cable is connected or not in case that S_FALSE is returned).
GetCurrentImage [▸ 1780]	ITcVnImageProvider	Gets the current image by detaching its internal reference.
GetError [▸ 1780]	ITcVnImageProvider	Gets result of the last operation performed.
OpenCamera [▸ 1780]	ITcVnImageProvider	Open camera (open the control channel, open a stream channel, and activate the GVSP receiver).
CloseCamera [▸ 1781]	ITcVnImageProvider	Close camera (deactivate the GVSP receiver, close the stream channel, and close the control channel).
StartAcquisition [▸ 1781]	ITcVnImageProvider	Start image acquisition.
StopAcquisition [▸ 1781]	ITcVnImageProvider	Stop image acquisition.
SoftwareTrigger [▸ 1781]	ITcVnImageProvider	Initialize the software trigger and trigger it. The initialization step is skipped if it was executed before.
InitializeCamera [▸ 1782]	ITcVnImageProvider	Set camera to initial state.
GetCurrentGevImage [▸ 1778]	ITcVnGevImageProvider	Gets the current GigE Vision image by detaching its internal reference.
GetCurrentImageWithGvspInfo [▸ 1778]	ITcVnGevImageProvider	Gets the current image by detaching its internal reference and additionally provide the GVSP info.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.1.7.1 GetCurrentGevImage

Gets the current GigE Vision image by detaching its internal reference.

Syntax

Definition:

```
HRESULT GetCurrentGevImage(
    ITcIoGevImage** pipGevImage
)
```

Parameters

Name	Type	Description
pipGevImage	ITcIoGevImage* [▶ 1793]*	Pointer to the interface pointer to be returned, might be 0 if the image acquisition failed.

Return value

HRESULT [▶ 122]

6.2.2.2.1.7.2 GetCurrentImageWithGvspInfo

Gets the current image by detaching its internal reference and additionally provide the GVSP info.

Syntax

Definition:

```
HRESULT GetCurrentImageWithGvspInfo(
    ITcVnImage**     pipImage,
    GVSP_IMAGE_INFO& stGvspInfo
)
```

Parameters

Name	Type	Description
pipImage	ITcVnImage* [▶ 1797]*	Pointer to the interface pointer to be returned, might be 0 if the image acquisition failed.
stGvspInfo	GVSP_IMAGE_INFO [▶ 1637]&	Contains useful meta information like image id, time stamp etc.

Return value

HRESULT [▶ 122]

6.2.2.2.1.8 ITcVnImageProvider

Interface for an image provider.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnImageProvider

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
CheckIfBusy [▶ 1779]	ITcVnImageProvider	Check whether the module is busy; i.e, an operation is in progress.
CheckIfCameraConnected [▶ 1780]	ITcVnImageProvider	Check if the camera is properly connected (Returns S_OK if connected and S_FALSE if not. Can only check software connection, i.e. the check is unable to determine if the cable is connected or not in case that S_FALSE is returned).
GetCurrentImage [▶ 1780]	ITcVnImageProvider	Gets the current image by detaching its internal reference.
GetError [▶ 1780]	ITcVnImageProvider	Gets result of the last operation performed.
OpenCamera [▶ 1780]	ITcVnImageProvider	Open camera (open the control channel, open a stream channel, and activate the GVSP receiver).
CloseCamera [▶ 1781]	ITcVnImageProvider	Close camera (deactivate the GVSP receiver, close the stream channel, and close the control channel).
StartAcquisition [▶ 1781]	ITcVnImageProvider	Start image acquisition.
StopAcquisition [▶ 1781]	ITcVnImageProvider	Stop image acquisition.
SoftwareTrigger [▶ 1781]	ITcVnImageProvider	Initialize the software trigger and trigger it. The initialization step is skipped if it was executed before.
InitializeCamera [▶ 1782]	ITcVnImageProvider	Set camera to initial state.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.1.8.1 CheckIfBusy

Check whether the module is busy; i.e, an operation is in progress.

Syntax

Definition:

```
HRESULT CheckIfBusy()
```

Return value

HRESULT [▶ 122]

6.2.2.2.1.8.2 CheckIfCameraConnected

Check if the camera is properly connected (Returns S_OK if connected and S_FALSE if not. Can only check software connection, i.e. the check is unable to determine if the cable is connected or not in case that S_FALSE is returned).

Syntax

Definition:

```
HRESULT CheckIfCameraConnected()
```

Return value

HRESULT [▶ 122]

6.2.2.2.1.8.3 GetCurrentImage

Gets the current image by detaching its internal reference.

Syntax

Definition:

```
HRESULT GetCurrentImage(
    ITcVnImage** pipImage
)
```

Parameters

Name	Type	Description
pipImage	ITcVnImage* [▶ 1797]*	Pointer to the interface pointer to be returned, might be 0 if the image acquisition failed.

Return value

HRESULT [▶ 122]

6.2.2.2.1.8.4 GetError

Gets result of the last operation performed.

Syntax

Definition:

```
HRESULT GetError()
```

Return value

HRESULT [▶ 122]

6.2.2.2.1.8.5 OpenCamera

Open camera (open the control channel, open a stream channel, and activate the GVSP receiver).

Syntax

Definition:

```
HRESULT OpenCamera()
```

Return value

HRESULT [▶ 122]

6.2.2.2.1.8.6 CloseCamera

Close camera (deactivate the GVSP receiver, close the stream channel, and close the control channel).

Syntax

Definition:

```
HRESULT CloseCamera()
```

Return value

HRESULT [▶ 122]

6.2.2.2.1.8.7 StartAcquisition

Start image acquisition.

Syntax

Definition:

```
HRESULT StartAcquisition()
```

Return value

HRESULT [▶ 122]

6.2.2.2.1.8.8 StopAcquisition

Stop image acquisition.

Syntax

Definition:

```
HRESULT StopAcquisition()
```

Return value

HRESULT [▶ 122]

6.2.2.2.1.8.9 SoftwareTrigger

Initialize the software trigger and trigger it. The initialization step is skipped if it was executed before.

Syntax

Definition:

```
HRESULT SoftwareTrigger()
```

Return value

HRESULT [▶ 122]

6.2.2.2.1.8.10 InitializeCamera

Set camera to initial state.

Syntax

Definition:

```
HRESULT InitializeCamera()
```

Return value

HRESULT [▶ 122]

6.2.2.2.2 Export

该组包含处理与图像有关的数据输出的接口。

6.2.2.2.2.1 ITcVnBitmapExport

Interface for exporting an image as a Windows Bitmap (BMP).

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnBitmapExport

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
GetBitmapSize [▶ 1782]	ITcVnBitmapExport	Gets size of the image if it is converted to a Windows Bitmap (BMP).
GetBitmapImage [▶ 1783]	ITcVnBitmapExport	Export the image as a Windows Bitmap (BMP) into a given buffer.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.2.1.1 GetBitmapSize

Gets size of the image if it is converted to a Windows Bitmap (BMP).

Syntax

Definition:

```
HRESULT GetBitmapSize(
    ULONGLONG& nBitmapSize,
    ULONG&     nBitmapWidth,
    ULONG&     nBitmapHeight
)
```

Parameters

Name	Type	Description
nBitmapSize	ULONGLONG&	Output parameter containing the required buffer size.
nBitmapWidth	ULONG&	Desired width or 0 to keep the original width (in) and actual width (out).
nBitmapHeight	ULONG&	Desired height or 0 to keep the original height (in) and actual height (out).

Return value

HRESULT [▶ 122]

6.2.2.2.2.1.2 *GetBitmapImage*

Export the image as a Windows Bitmap (BMP) into a given buffer.

Syntax

Definition:

```
HRESULT GetBitmapImage(
    ULONGLONG& nBitmapSize,
    PVOID      pDestBuffer,
    ULONG&     nBitmapWidth,
    ULONG&     nBitmapHeight
)
```

Parameters

Name	Type	Description
nBitmapSize	ULONGLONG&	Maximum buffer size (in) and actual buffer size (out).
pDestBuffer	PVOID	Pointer to the destination buffer.
nBitmapWidth	ULONG&	Desired width or 0 to keep the original width (in) and actual width (out).
nBitmapHeight	ULONG&	Desired height or 0 to keep the original height (in) and actual height (out).

Return value

HRESULT [▶ 122]

6.2.2.2.2.2 ITcVnBitmapExportNotification

Interface required for sending displayable images as ADS notifications

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcVnBitmapExportNotification

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
GetTimestamp [▶ 1784]	ITcVnBitmapExportNotification	Gets the timestamp of the latest image change.
GetBitmapImageRpcUnlocked [▶ 1784]	ITcVnBitmapExportNotification	Export the image as a Windows Bitmap (BMP) into a given buffer by means of an unlocked remote procedure call. It lies within the responsibility of the user to ensure that no conflicting accesses can occur.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.2.2.1 GetTimestamp

Gets the timestamp of the latest image change.

Syntax

Definition:

```
HRESULT GetTimestamp(
    LONGLONG& nTimestamp
)
```

Parameters

Name	Type	Description
nTimestamp	LONGLONG&	Returns the timestamp.

Return value

HRESULT [▶ 122]

6.2.2.2.2.2.2 GetBitmapImageRpcUnlocked

Export the image as a Windows Bitmap (BMP) into a given buffer by means of an unlocked remote procedure call. It lies within the responsibility of the user to ensure that no conflicting accesses can occur.

Syntax

Definition:

```
HRESULT GetBitmapImageRpcUnlocked(
    ULONGLONG& nBitmapSize,
    PVOID      pDestBuffer,
```

```
    ULONG&      nBitmapWidth,
    ULONG&      nBitmapHeight
)
```

Parameters

Name	Type	Description
nBitmapSize	ULONGLONG&	Maximum buffer size (in) and actual buffer size (out).
pDestBuffer	PVOID	Pointer to the destination buffer.
nBitmapWidth	ULONG&	Desired width or 0 to keep the original width (in) and actual width (out).
nBitmapHeight	ULONG&	Desired height or 0 to keep the original height (in) and actual height (out).

Return value

HRESULT [▶ 122]

6.2.2.2.2.3 ITcVnBitmapExportRpcUnlocked

Interface to export an image as a Windows Bitmap (BMP) into a given buffer by means of an unlocked remote procedure call.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnBitmapExport [▶ 1782]
        ITcVnBitmapExportRpcUnlocked

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
GetBitmapSize [▶ 1782]	ITcVnBitmapExport	Gets size of the image if it is converted to a Windows Bitmap (BMP).
GetBitmapImage [▶ 1783]	ITcVnBitmapExport	Export the image as a Windows Bitmap (BMP) into a given buffer.
GetBitmapImageRpcUnlocked [▶ 1786]	ITcVnBitmapExportRpcUnlocked	Export the image as a Windows Bitmap (BMP) into a given buffer by means of an unlocked remote procedure call. It lies within the responsibility of the user to ensure that no conflicting accesses can occur.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.2.3.1 GetBitmapImageRpcUnlocked

Export the image as a Windows Bitmap (BMP) into a given buffer by means of an unlocked remote procedure call. It lies within the responsibility of the user to ensure that no conflicting accesses can occur.

Syntax

Definition:

```
HRESULT GetBitmapImageRpcUnlocked(
    ULONGLONG& nBitmapSize,
    PVOID      pDestBuffer,
    ULONG&     nBitmapWidth,
    ULONG&     nBitmapHeight
)
```

Parameters

Name	Type	Description
nBitmapSize	ULONGLONG&	Maximum buffer size (in) and actual buffer size (out).
pDestBuffer	PVOID	Pointer to the destination buffer.
nBitmapWidth	ULONG&	Desired width or 0 to keep the original width (in) and actual width (out).
nBitmapHeight	ULONG&	Desired height or 0 to keep the original height (in) and actual height (out).

Return value

HRESULT [▸ 122]

6.2.2.2.2.4 ITcVnHistogramExport

Interface for exporting a histogram for an image as an array.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
        ITcVnHistogramExport

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
GetHistogramSize [▸ 1787]	ITcVnHistogramExport	Gets the size of the histogram array.
GetHistogramArray [▸ 1787]	ITcVnHistogramExport	Gets the histogram array.
GetCustomHistogramSize [▸ 1787]	ITcVnHistogramExport	Gets the size of a custom histogram array.
GetCustomHistogramArray [▸ 1788]	ITcVnHistogramExport	Gets a custom histogram array.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.2.4.1 GetHistogramSize

Gets the size of the histogram array.

Syntax

Definition:

```
HRESULT GetHistogramSize(
    ULONGLONG& nHistogramSize
)
```

Parameters

Name	Type	Description
nHistogramSize	ULONGLONG&	Output parameter containing the required buffer size.

Return value

HRESULT [▶ 122]

6.2.2.2.2.4.2 GetHistogramArray

Gets the histogram array.

Syntax

Definition:

```
HRESULT GetHistogramArray(
    ULONGLONG& nHistogramSize,
    PVOID      pDestBuffer
)
```

Parameters

Name	Type	Description
nHistogramSize	ULONGLONG&	Maximum buffer size (in) and actual buffer size (out).
pDestBuffer	PVOID	Pointer to the destination buffer.

Return value

HRESULT [▶ 122]

6.2.2.2.2.4.3 GetCustomHistogramSize

Gets the size of a custom histogram array.

Syntax

Definition:

```
HRESULT GetCustomHistogramSize(
    ULONGLONG& nHistogramSize,
    ULONG&     nBins
)
```

Parameters

Name	Type	Description
nHistogramSize	ULONGLONG&	Output parameter containing the required buffer size.
nBins	ULONG&	Desired number of bins or 0 to keep the default for the corresponding image format (in) and default number of bins (out).

Return value

HRESULT [▸ 122]

6.2.2.2.2.4.4 GetCustomHistogramArray

Gets a custom histogram array.

Syntax

Definition:

```
HRESULT GetCustomHistogramArray(
    ULONGLONG& nHistogramSize,
    PVOID      pDestBuffer,
    ULONG&     nBins,
    double&    fLowerBound,
    double&    fUpperBound
)
```

Parameters

Name	Type	Description
nHistogramSize	ULONGLONG&	Maximum buffer size (in) and actual buffer size (out).
pDestBuffer	PVOID	Pointer to the destination buffer.
nBins	ULONG&	Desired number of bins or 0 to keep the default for the corresponding image format (in) and default number of bins (out).
fLowerBound	double&	Lower (inclusive) boundary of the 0-th histogram bin (in), or receive the default if fLowerBound AND fUpperBound are set to 0 (out).
fUpperBound	double&	Upper (exclusive) boundary of the last histogram bin nBins-1 (in), or receive the default if fLowerBound AND fUpperBound are set to 0 (out).

Return value

HRESULT [▸ 122]

6.2.2.2.2.5 ITcVnTiffExport

Interface for exporting an image as tiff.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnTiffExport

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
GetTiffSize [▶ 1789]	ITcVnTiffExport	Gets size of the image if it is converted to tiff.
GetTiffImage [▶ 1789]	ITcVnTiffExport	Export the image as tiff into a given buffer.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.2.5.1 GetTiffSize

Gets size of the image if it is converted to tiff.

Syntax

Definition:

```
HRESULT GetTiffSize(
    ULONGLONG& nTiffSize
)
```

Parameters

Name	Type	Description
nTiffSize	ULONGLONG&	Output parameter containing the required buffer size.

Return value

HRESULT [▶ 122]

6.2.2.2.2.5.2 GetTiffImage

Export the image as tiff into a given buffer.

Syntax

Definition:

```
HRESULT GetTiffImage(
    ULONGLONG& nTiffSize,
    PVOID      pDestBuffer
)
```

Parameters

Name	Type	Description
nTiffSize	ULONGLONG&	Maximum buffer size (in) and actual buffer size (out).
pDestBuffer	PVOID	Pointer to the destination buffer.

Return value

HRESULT [▶ 122]

6.2.2.2.2.6 ITcVnTiffExportNotification

Interface for sending displayable images as tiff via ADS notifications.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcVnTiffExportNotification

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
GetTimestamp [▶ 1790]	ITcVnTiffExportNotification	Gets the timestamp of the latest image change.
GetTiffImageRpcUnlocked [▶ 1791]	ITcVnTiffExportNotification	Export the image as tiff into a given buffer by means of an unlocked remote procedure call. It lies within the responsibility of the user to ensure that no conflicting accesses can occur.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.2.6.1 GetTimestamp

Gets the timestamp of the latest image change.

Syntax

Definition:

```
HRESULT GetTimestamp(
    LONGLONG& nTimestamp
)
```

Parameters

Name	Type	Description
nTimestamp	LONGLONG&	Returns the timestamp.

Return value

HRESULT [▸ 122]

6.2.2.2.2.6.2 *GetTiffImageRpcUnlocked*

Export the image as tiff into a given buffer by means of an unlocked remote procedure call. It lies within the responsibility of the user to ensure that no conflicting accesses can occur.

Syntax

Definition:

```
HRESULT GetTiffImageRpcUnlocked(
    ULONGLONG& nTiffSize,
    PVOID      pDestBuffer
)
```

Parameters

Name	Type	Description
nTiffSize	ULONGLONG&	Maximum buffer size (in) and actual buffer size (out).
pDestBuffer	PVOID	Pointer to the destination buffer.

Return value

HRESULT [▸ 122]

6.2.2.2.2.7 ITcVnTiffExportRpcUnlocked

Interface for exporting an image as tiff by means of an unlocked remote procedure call.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
ITcVnTiffExport [▸ 1789]
ITcVnTiffExportRpcUnlocked

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
GetTiffSize [▸ 1789]	ITcVnTiffExport	Gets size of the image if it is converted to tiff.
GetTiffImage [▸ 1789]	ITcVnTiffExport	Export the image as tiff into a given buffer.
GetTiffImageRpcUnlocked [▸ 1792]	ITcVnTiffExportRpcUnlocked	Export the image as tiff into a given buffer by means of an unlocked remote procedure call. It lies within the responsibility of the user to ensure that no conflicting accesses can occur.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.2.7.1 GetTiffImageRpcUnlocked

Export the image as tiff into a given buffer by means of an unlocked remote procedure call. It lies within the responsibility of the user to ensure that no conflicting accesses can occur.

Syntax

Definition:

```
HRESULT GetTiffImageRpcUnlocked(
    ULONGLONG& nTiffSize,
    PVOID      pDestBuffer
)
```

Parameters

Name	Type	Description
nTiffSize	ULONGLONG&	Maximum buffer size (in) and actual buffer size (out).
pDestBuffer	PVOID	Pointer to the destination buffer.

Return value

HRESULT [▶ 122]

6.2.2.2.3 Import

该组包含用于处理与图像相关的数据导入的接口。

6.2.2.2.3.1 ITcVnTiffImport

Interface for importing a tiff image.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
        ITcVnTiffImport

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
ImportTiffImage [▶ 1793]	ITcVnTiffImport	Import a tiff image.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.3.1.1 ImportTiffImage

Import a tiff image.

Syntax

Definition:

```
HRESULT ImportTiffImage(
    PVOID pTiffImage
)
```

Parameters

Name	Type	Description
pTiffImage	PVOID	Pointer to the tiff file.

Return value

HRESULT [▶ 122]

6.2.2.2.4 ITcIoGevImage

Interface for GigE Vision images.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcVnImageBase [▶ 1799]
ITcIoGevImage

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
GetImageData [▸ 1799]	ITcVnImageBase	Gets the data interface (Do not explicitly free the pointer in there!).
GetImageSize [▸ 1800]	ITcVnImageBase	Gets the image size.
GetWidth [▸ 1800]	ITcVnImageBase	Gets the image width (alternatively use F_VN_GetImageWidth).
GetHeight [▸ 1800]	ITcVnImageBase	Gets the image height (alternatively use F_VN_GetImageHeight).
GetXPadding [▸ 1801]	ITcVnImageBase	Gets the horizontal padding.
GetYPadding [▸ 1801]	ITcVnImageBase	Gets the vertical padding.
GetPixelFormat [▸ 1801]	ITcVnImageBase	Gets the pixel format (alternatively use F_VN_GetPixelFormat).
GetImageInfo [▸ 1802]	ITcVnImageBase	Gets a struct containing all common meta infos of the image. This basically encompasses all meta information accessible via this interface. (Alternatively use F_VN_GetImageInfo.)
CheckIfCopy [▸ 1794]	ITcIoGevImage	Check if image data is a copy.
DecreaseHeight [▸ 1794]	ITcIoGevImage	Sets image height to newHeight if it is smaller than the current value.
GetBlockId [▸ 1795]	ITcIoGevImage	Gets the GVSP block ID.
GetGevStatus [▸ 1795]	ITcIoGevImage	Gets the block status.
GetGvspImageInfo [▸ 1796]	ITcIoGevImage	Gets a pointer to the GVSP leader payload.
SetGevStatus [▸ 1796]	ITcIoGevImage	Sets the block status.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.4.1 CheckIfCopy

Check if image data is a copy.

Syntax

Definition:

```
HRESULT CheckIfCopy()
```

Return value

HRESULT [▸ 122]

6.2.2.2.4.2 DecreaseHeight

Sets image height to newHeight if it is smaller than the current value.

Syntax

Definition:

```
HRESULT DecreaseHeight(
    ULONG nNewHeight
)
```

Parameters

Name	Type	Description
nNewHeight	ULONG	Height value to set.

Return value

HRESULT [▶ 122]

6.2.2.2.4.3 GetBlockId

Gets the GVSP block ID.

Syntax

Definition:

```
HRESULT GetBlockId(
    ULONGLONG& nBlockId
)
```

Parameters

Name	Type	Description
nBlockId	ULONGLONG&	Returns the block ID.

Return value

HRESULT [▶ 122]

6.2.2.2.4.4 GetGevStatus

Gets the block status.

Syntax

Definition:

```
HRESULT GetGevStatus(
    USHORT& nGevStatus
)
```

Parameters

Name	Type	Description
nGevStatus	USHORT&	Returns the GigE Vision status code.

Return value

HRESULT [▶ 122]

6.2.2.2.4.5 GetGvspImageInfo

Gets a pointer to the GVSP leader payload.

Syntax

Definition:

```
HRESULT GetGvspImageInfo(
    GVSP_IMAGE_INFO** ppGvspImageInfo
)
```

Parameters

Name	Type	Description
ppGvspImageInfo	GVSP_IMAGE_INFO [▶ 1637]**	Pointer to the GVSP meta information.

Return value

HRESULT [▶ 122]

6.2.2.2.4.6 SetGevStatus

Sets the block status.

Syntax

Definition:

```
HRESULT SetGevStatus(
    USHORT nGevStatus
)
```

Parameters

Name	Type	Description
nGevStatus	USHORT	GigE Vision status code.

Return value

HRESULT [▶ 122]

6.2.2.2.5 ITcVnDisplayableImage

Interface for displayable images.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnDisplayableImage

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.6 ITcVnImage

Basic interface for images.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
ITcVnImageBase [▸ 1799]
ITcVnImage

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
GetImageData [▸ 1799]	ITcVnImageBase	Gets the data interface (Do not explicitly free the pointer in there!).
GetImageSize [▸ 1800]	ITcVnImageBase	Gets the image size.
GetWidth [▸ 1800]	ITcVnImageBase	Gets the image width (alternatively use F_VN_GetImageWidth).
GetHeight [▸ 1800]	ITcVnImageBase	Gets the image height (alternatively use F_VN_GetImageHeight).
GetXPadding [▸ 1801]	ITcVnImageBase	Gets the horizontal padding.
GetYPadding [▸ 1801]	ITcVnImageBase	Gets the vertical padding.
GetPixelFormat [▸ 1801]	ITcVnImageBase	Gets the pixel format (alternatively use F_VN_GetPixelFormat).
GetImageInfo [▸ 1802]	ITcVnImageBase	Gets a struct containing all common meta infos of the image. This basically encompasses all meta information accessible via this interface. (Alternatively use F_VN_GetImageInfo.)
Init [▸ 1797]	ITcVnImage	Initialize an image with an ITcVnImageBase interface.
GetRowPointer [▸ 1798]	ITcVnImage	Gets a pointer to a specific row of an image.
ReleaseRowPointer [▸ 1798]	ITcVnImage	Release the pointer to a specific row of an image.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.6.1 Init

Initialize an image with an ITcVnImageBase interface.

Syntax

Definition:

```
HRESULT Init(
    ITcVnImageBase* ipImageBase
)
```

Parameters

Name	Type	Description
ipImageBase	ITcVnImageBase* [▶ 1799]	Image from which to obtain the data.

Return value

HRESULT [▶ 122]

6.2.2.2.6.2 GetRowPointer

Gets a pointer to a specific row of an image.

Syntax

Definition:

```
HRESULT GetRowPointer(
    ULONG  nRowIndex,
    PVOID* ppRow
)
```

Parameters

Name	Type	Description
nRowIndex	ULONG	Row index.
ppRow	PVOID*	Returns a pointer to the requested image row.

Return value

HRESULT [▶ 122]

6.2.2.2.6.3 ReleaseRowPointer

Release the pointer to a specific row of an image.

Syntax

Definition:

```
HRESULT ReleaseRowPointer(
    PVOID* ppRow
)
```

Parameters

Name	Type	Description
ppRow	PVOID*	Pointer to the row pointer to be released.

Return value

HRESULT [▶ 122]

6.2.2.2.7 ITcVnImageBase

Base interface for all image types.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
    ITcVnImageBase

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
GetImageData [▶ 1799]	ITcVnImageBase	Gets the data interface (Do not explicitly free the pointer in there!).
GetImageSize [▶ 1800]	ITcVnImageBase	Gets the image size.
GetWidth [▶ 1800]	ITcVnImageBase	Gets the image width (alternatively use F_VN_GetImageWidth).
GetHeight [▶ 1800]	ITcVnImageBase	Gets the image height (alternatively use F_VN_GetImageHeight).
GetXPadding [▶ 1801]	ITcVnImageBase	Gets the horizontal padding.
GetYPadding [▶ 1801]	ITcVnImageBase	Gets the vertical padding.
GetPixelFormat [▶ 1801]	ITcVnImageBase	Gets the pixel format (alternatively use F_VN_GetPixelFormat).
GetImageInfo [▶ 1802]	ITcVnImageBase	Gets a struct containing all common meta infos of the image. This basically encompasses all meta information accessible via this interface. (Alternatively use F_VN_GetImageInfo.)

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.2.7.1 GetImageData

Gets the data interface (Do not explicitly free the pointer in there!).

Syntax

Definition:

```
HRESULT GetImageData(
    ITcVnData** pipImageData
)
```

Parameters

Name	Type	Description
pipImageData	ITcVnData* [▶ 1806]*	Returns a pointer to the image data interface.

Return value

HRESULT [▶ 122]

6.2.2.2.7.2 GetImageSize

Gets the image size.

Syntax

Definition:

```
HRESULT GetImageSize(
    ULONGLONG& nSize
)
```

Parameters

Name	Type	Description
nSize	ULONGLONG&	Returns the image size in bytes.

Return value

HRESULT [▶ 122]

6.2.2.2.7.3 GetWidth

Gets the image width (alternatively use F_VN_GetImageWidth).

Syntax

Definition:

```
HRESULT GetWidth(
    ULONG& nWidth
)
```

Parameters

Name	Type	Description
nWidth	ULONG&	Returns the image width in pixels.

Return value

HRESULT [▶ 122]

6.2.2.2.7.4 GetHeight

Gets the image height (alternatively use F_VN_GetImageHeight).

Syntax

Definition:

```
HRESULT GetHeight(
    ULONG& nHeight
)
```

Parameters

Name	Type	Description
nHeight	ULONG&	Returns the image height in pixels.

Return value

HRESULT [▶ 122]

6.2.2.2.7.5 GetXPadding

Gets the horizontal padding.

Syntax

Definition:

```
HRESULT GetXPadding(
    USHORT& nXPadding
)
```

Parameters

Name	Type	Description
nXPadding	USHORT&	Returns the horizontal padding in bytes.

Return value

HRESULT [▶ 122]

6.2.2.2.7.6 GetYPadding

Gets the vertical padding.

Syntax

Definition:

```
HRESULT GetYPadding(
    USHORT& nYPadding
)
```

Parameters

Name	Type	Description
nYPadding	USHORT&	Returns the vertical padding in bytes.

Return value

HRESULT [▶ 122]

6.2.2.2.7.7 GetPixelFormat

Gets the pixel format (alternatively use F_VN_GetPixelFormat).

Syntax

Definition:

```
HRESULT GetPixelFormat(
    TcVnPixelFormat& stPixelFormat
)
```

Parameters

Name	Type	Description
stPixelFormat	TcVnPixelFormat [▸ 1657]&	Returns a struct describing the pixel format.

Return value

HRESULT [▸ 122]

6.2.2.2.7.8 GetImageInfo

Gets a struct containing all common meta infos of the image. This basically encompasses all meta information accessible via this interface. (Alternatively use F_VN_GetImageInfo.)

Syntax

Definition:

```
HRESULT GetImageInfo(
    TcVnImageInfo& stImageInfo
)
```

Parameters

Name	Type	Description
stImageInfo	TcVnImageInfo [▸ 1643]&	Returns a struct describing the image.

Return value

HRESULT [▸ 122]

6.2.2.3 Machine Learning

该组包含机器学习的接口。

6.2.2.3.1 ITcVnColorModel

Interface to train and classify an image color.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
        ITcVnMlModel [▸ 1804]
                ITcVnColorModel

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
GetTypeGuid [▸ 1804]	ITcVnMlModel	Gets the GUID of the model.
GetTypeName [▸ 1805]	ITcVnMlModel	Gets the model type name as a string.
TrainImageColor [▸ 1803]	ITcVnColorModel	Train the color of the provided image.
ClassifyImageColor [▸ 1803]	ITcVnColorModel	Compute the similarity of each image pixel to the trained reference color.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.3.1.1 TrainImageColor

Train the color of the provided image.

Syntax

Definition:

```
HRESULT TrainImageColor(
    ITcVnImage* ipSrcImage,
    ULONG       nDifferentColors,
    ITcVnImage* ipMask,
    ULONG       nSkipPixels
)
```

Parameters

Name	Type	Description
ipSrcImage	ITcVnImage* [▸ 1797]	Source image.
nDifferentColors	ULONG	Number of different colors to differentiate.
ipMask	ITcVnImage* [▸ 1797]	Optional image mask.
nSkipPixels	ULONG	Number of pixels to skip between each evaluated color sample (to achieve a better performance). 0 takes every pixel into account and tends to be more accurate.

Return value

HRESULT [▸ 122]

6.2.2.3.1.2 ClassifyImageColor

Compute the similarity of each image pixel to the trained reference color.

Syntax

Definition:

```
HRESULT ClassifyImageColor(
    ITcVnImage*  ipSrcImage,
    ITcVnImage** pipDestImage,
    float        fVariance,
    float        fLuminanceWeight
)
```

Parameters

Name	Type	Description
ipSrcImage	ITcVnImage* [▸ 1797]	Source image.
pipDestImage	ITcVnImage* [▸ 1797]*	Returns the color similarity.
fVariance	float	Allowed color variance.
fLuminanceWeight	float	Weight the impact of the luminance.

Return value

HRESULT [▸ 122]

6.2.2.3.2 ITcVnMlModel

Interface for a machine learning model.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
ITcVnMlModel

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
GetTypeGuid [▸ 1804]	ITcVnMlModel	Gets the GUID of the model.
GetTypeName [▸ 1805]	ITcVnMlModel	Gets the model type name as a string.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.3.2.1 GetTypeGuid

Gets the GUID of the model.

Syntax

Definition:

```
HRESULT GetTypeGuid(
    GUID& nTypeGuid
)
```

Parameters

Name	Type	Description
nTypeGuid	GUID&	Returns the GUID of the model.

Return value

HRESULT [▶ 122]

6.2.2.3.2.2 GetTypeName

Gets the model type name as a string.

Syntax

Definition:

```
HRESULT GetTypeName(
    PCHAR  sTypeName,
    USHORT nMaxLen
)
```

Parameters

Name	Type	Description
sTypeName	PCHAR	Returns the model type name as a string.
nMaxLen	USHORT	Maximum string length allowed to be written in sTypeName.

Return value

HRESULT [▶ 122]

6.2.2.4 Miscellaneous

该组包含其他接口。

6.2.2.4.1 ITcSerializableNotification

Interface required for sending serialized objects as ADS notifications.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
        ITcSerializableNotification

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
GetComparisonValue [▶ 1806]	ITcSerializableNotification	Gets a value that reflects changes of the underlying object (e.g., the timestamp of the latest object change).
SerializeRpcUnlocked [▶ 1806]	ITcSerializableNotification	Serialize an object into a given buffer.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.4.1.1 GetComparisonValue

Gets a value that reflects changes of the underlying object (e.g., the timestamp of the latest object change).

Syntax

Definition:

```
HRESULT GetComparisonValue(
    LONGLONG& nComparisonValue
)
```

Parameters

Name	Type	Description
nComparisonValue	LONGLONG&	Returns the comparison value.

Return value

HRESULT [▶ 122]

6.2.2.4.1.2 SerializeRpcUnlocked

Serialize an object into a given buffer.

Syntax

Definition:

```
HRESULT SerializeRpcUnlocked(
    ULONGLONG& nBufferSize,
    PVOID      pDestBuffer
)
```

Parameters

Name	Type	Description
nBufferSize	ULONGLONG&	Maximum buffer size (in) and actually used buffer size (out).
pDestBuffer	PVOID	Pointer to the destination buffer.

Return value

HRESULT [▶ 122]

6.2.2.4.2 ITcVnData

Interface for accessing data.

Inheritance Hierarchy

ITcUnknown [▶ 1810]
ITcVnData

Methods

Name	Origin	Description
TcAddRef [▶ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▶ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▶ 1811]	ITcUnknown	Decrements the reference counter.
CheckIfCopy [▶ 1807]	ITcVnData	Check if the data are a copy (returns S_OK for copies and S_FALSE otherwise).
CheckIfImage [▶ 1807]	ITcVnData	Check if the data belongs to an image (returns S_OK for images and S_FALSE otherwise).
GetSize [▶ 1808]	ITcVnData	Gets the size of the data.
CheckDataPointer [▶ 1808]	ITcVnData	Check if the data pointer is different from 0 (returns S_OK for non-zero data pointers and S_FALSE otherwise).
GetDataPointer [▶ 1808]	ITcVnData	Gets the data pointer (the obtained data pointer must be released by calling ReleaseDataPointer).
ReleaseDataPointer [▶ 1808]	ITcVnData	Release the data pointer.

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.4.2.1 CheckIfCopy

Check if the data are a copy (returns S_OK for copies and S_FALSE otherwise).

Syntax

Definition:

```
HRESULT CheckIfCopy()
```

Return value

HRESULT [▶ 122]

6.2.2.4.2.2 CheckIfImage

Check if the data belongs to an image (returns S_OK for images and S_FALSE otherwise).

Syntax

Definition:

```
HRESULT CheckIfImage()
```

Return value

HRESULT [▶ 122]

6.2.2.4.2.3 GetSize

Gets the size of the data.

Syntax

Definition:

```
HRESULT GetSize(
    ULONGLONG& nSize
)
```

Parameters

Name	Type	Description
nSize	ULONGLONG&	Size of the stored data in bytes.

Return value

HRESULT [▶ 122]

6.2.2.4.2.4 CheckDataPointer

Check if the data pointer is different from 0 (returns S_OK for non-zero data pointers and S_FALSE otherwise).

Syntax

Definition:

```
HRESULT CheckDataPointer()
```

Return value

HRESULT [▶ 122]

6.2.2.4.2.5 GetDataPointer

Gets the data pointer (the obtained data pointer must be released by calling ReleaseDataPointer).

Syntax

Definition:

```
HRESULT GetDataPointer(
    PVOID* ppData
)
```

Parameters

Name	Type	Description
ppData	PVOID*	Returns the data pointer.

Return value

HRESULT [▶ 122]

6.2.2.4.2.6 ReleaseDataPointer

Release the data pointer.

Syntax

Definition:

```
HRESULT ReleaseDataPointer(
    PVOID* ppData
)
```

Parameters

Name	Type	Description
ppData	PVOID*	Pointer to the data pointer to be released.

Return value

HRESULT [▸ 122]

6.2.2.4.3 ITcVnTimestamp

Offers an interface for timestamps.

Inheritance Hierarchy

ITcUnknown [▸ 1810]
ITcVnTimestamp

Methods

Name	Origin	Description
TcAddRef [▸ 1810]	ITcUnknown	Increments the reference counter.
TcQueryInterface [▸ 1810]	ITcUnknown	Get a reference to an implemented interface.
TcRelease [▸ 1811]	ITcUnknown	Decrements the reference counter.
UpdateTimestamp [▸ 1809]	ITcVnTimestamp	Updates the timestamp to the current time
GetTimestamp [▸ 1810]	ITcVnTimestamp	Gets the timestamp

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.2.4.3.1 UpdateTimestamp

Updates the timestamp to the current time

Syntax

Definition:

```
HRESULT UpdateTimestamp()
```

Return value

HRESULT [▸ 122]

6.2.2.4.3.2 GetTimestamp

Gets the timestamp

Syntax

Definition:

```
HRESULT GetTimestamp(
    LONGLONG& nTimestamp
)
```

Parameters

Name	Type	Description
nTimestamp	LONGLONG&	Returns the timestamp

Return value

HRESULT [▶ 122]

6.2.2.4.4 接口 ITcUnknown

ITcUnknown 定义引用计数以及对更具体接口的引用查询。

语法

```
TCOM_DECL_INTERFACE("00000001-0000-0000-e000-000000000064", ITcUnknown)
```

声明形式：TcInterfaces.h

要求包括：

方法

名称	描述
TcAddRef [▶ 1810]	增加引用计数器。
TcQueryInterface [▶ 1810]	通过 IID 查询已实施接口的引用。
TcRelease [▶ 1811]	减少引用计数器。

每个 TcCOM 界面都直接或间接从 ITcUnknown 派生 因此，每个 TcCOM 模块类均执行 ITcUnknown，因为它是从 ITComObject 派生。

ITcUnknown 的标准执行可确保在释放最后一个引用后删除对象。因此，在调用 TcRelease() 后，不得取消引用接口指针。

6.2.2.4.4.1 方法 ITcUnknown:TcAddRef

增加引用计数器并返回新值。

语法

```
ULONG TcAddRef( )
```

返回值

产生引用计数值。

6.2.2.4.4.2 方法 ITcUnknown:TcQueryInterface

根据接口 ID (IID) 给出的接口查询接口指针。

语法

```
HRESULT TcQueryInterface(RITCID iid, PPVOID pipItf )
```

返回值

如果成功，将返回 S_OK（″0″）或其它正值，请参见返回值。关于扩展信息，请特别参见 ADS 返回代码 [▶ 2753]中的 HRESULT 栏。

如果所要求的接口不可用，该方法将返回 ADSERR_DEVICE_NOINTERFACE。

参数

名称	类型	描述
iid	RITCID	接口 IID。
pipItf	PPVOID	指向接口指针。当所请求的接口类型在相应实例中可用时设置。

描述

通过 IID 查询已实施接口的引用。建议使用智能指针来初始化和保存接口指针。

变量 1：

```
HRESULT GetTraceLevel(ITcUnkown* ip, TcTraceLevel& tl)
{
HRESULT hr = S_OK;
if (ip != NULL)
{
ITComObjectPtr spObj;
hr = ip->TcQueryInterface(spObj.GetIID(), &spObj);
if (SUCCEEDED(hr))
{
hr = spObj->TcGetObjPara(PID_TcTraceLevel, &tl, sizeof(tl));
}
return hr;
}
}
```

与智能指针相关的接口 ID 可用作 TcQueryInterface 中的参数。“&”操作符将返回指向智能指针内部接口指针成员变量的指针。变量 1 假定当 TcQueryInterface 指示成功时，接口指针被初始化。如果范围仍然存在，则智能指针 spObj 的析构函数会释放引用。

变量 2：

```
HRESULT GetTraceLevel(ITcUnkown* ip, TcTraceLevel& tl)
{
HRESULT hr = S_OK;
ITComObjectPtr spObj = ip;
if (spObj != NULL)
{
spObj->TcGetObjParam(PID_TcTraceLevel, &tl);
}
else
{
hr = ADS_E_NOINTERFACE;
}
return hr;
}
```

如果接口指针 ip 被分配给智能指针 spObj，则将在 ip 引用的实例上使用 IID_ITComObject 隐式调用 TcQueryInterface 方法。这样会缩短代码，但会丢失 TcQueryInterface 的原始返回代码。

6.2.2.4.4.3 方法 ITcUnknown:TcRelease

该方法减少引用计数器。

语法

```
ULONG TcRelease( )
```

返回值

产生引用计数值。

描述

减少引用计数器并返回新值。

如果引用计数器为 0，则对象会自行删除。

6.2.3 Functions

图像处理算法被封装在函数中。更复杂的算法已作为功能块实现，可在 ImageProcessing [▶ 2594] 中找到。

函数调用的结构

函数结构如下：

```
hr := ProcessImage (hr, <…>, <…>);

Status variable of type HRESULT
Parameters and return values as references
Optional parameters with default values
```

hr	类型 HRESULT 的状态变量 [▶ 122]。借此可跟踪每个函数的执行状态。
参数 <...>	函数参数主要实现以下目的： • 待处理的输入参数（如图像或容器）。 • 用于设置算法的参数 • 作为返回结果引用的参数
可选参数 <...>	这些参数初始化为默认值，因此规格可选。

函数分组

TwinCAT Vision 库的所有函数按主题分为以下几组：

图像

- 基本图像运算 [▶ 2079]
- 代数图像运算 [▶ 1867]
- 图像分割 [▶ 2412]
- 图像颜色和对比处理 [▶ 2371]
- 图像分析 [▶ 2342]
- 图像滤波 [▶ 2393]

容器

- 基本容器运算 [▶ 1903]
- 代数容器运算 [▶ 1835]
- 容器统计 [▶ 2122]

轮廓

- 轮廓分析 [▶ 2192]

测量

- 测量 [▶ 2496]

• 几何和坐标转换 [▶ 2269]

代码读取

• 代码读取 [▶ 2114]

光学字符识别 (OCR)

• 光学字符识别 [▶ 2544]

绘图和文本

• 绘图 [▶ 2224]

机器学习

• 机器学习 [▶ 2452]

高级函数

• 傅立叶分析 [▶ 2256]
• 关键点特征 [▶ 2418]

其它

• 控制函数 [▶ 2222]
• 其它 [▶ 2528]

专家函数

许多函数都有专家版本。它们包含附加参数，在 PLC 库中以名称后缀“Exp”标识。在 C++ 中，这些附加参数被指定为默认值，如果不需要，可以在调用时省略。

运行时行为

某些图像处理算法的执行时间取决于其参数设置和图像内容。因此，它并不确定。尽管如此，为了保证实时行为，可对这些函数或函数组应用 Watchdog（看门狗） [▶ 127]，其可在指定时间后终止函数。

⚠ 警告

函数使用不当会导致 TwinCAT 崩溃

如果某些函数使用不当，可能会导致错误（并导致 TwinCAT 崩溃）。这些错误不会在函数中被拦截，因为相应的检查会大大延长执行时间。因此，您必须注意正确使用这些函数！例如，我们建议对输入参数进行验证。

还请参阅有关此

Code Quality [▶ 2107]

6.2.3.1 License Overview

以下列表显示了 TwinCAT Vision 功能与可用许可证的分配情况（见授权模式 [▶ 11]）。它显示了应用某些功能需要哪些许可证。

TF7100

始终需要基本许可证 TC3 Vision Base（TF7100）。

TC3 Vision Base

AdaptiveThreshold [▶ 2413]

AddContainers [▶ 1857]

AddImages [▶ 1869]

AddImagesWeighted [▶ 1869]

TC3 Vision Code Reading

TC3 Vision Code Quality

TC3 Vision Metrology 2D

TC3 Vision Matching

TC3 Vision Machine Learning

TC3 Vision OCR

OCR [▶ 2546]

6.2.3.2 Algebraic Container Operations

该组包含将代数算子按元素应用于容器 [▶ 132]的函数。

函数

代数

- AddContainers [▶ 1857]
- AddToContainerElements [▶ 1835]
- DivideContainers [▶ 1860]
- MultiplyContainers [▶ 1863]
- MultiplyWithContainerElements [▶ 1863]
- NegateContainer [▶ 1866]
- SubtractContainers [▶ 1867]

逐位算子

- BitwiseAndContainers [▶ 1858]
- BitwiseNotContainer [▶ 1858]
- BitwiseOrContainers [▶ 1859]
- BitwiseXorContainers [▶ 1860]

统计学特征

- MaxContainer [▶ 1861]
- MinContainer [▶ 1862]

6.2.3.2.1 AddToContainerElements

本章包含根据数据类型向每个容器元素添加数值的函数。

6.2.3.2.1.1 AddToContainerElements (LONG)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT         hrPrev,
    LONG            nValue,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nValue	LONG	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with DINT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.2 AddToContainerElements (SHORT)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT          hrPrev,
    SHORT            nValue,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nValue	SHORT	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with INT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.3 AddToContainerElements (double)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT         hrPrev,
    double          fValue,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fValue	double	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with LREAL elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.4 AddToContainerElements (float)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT         hrPrev,
    float           fValue,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fValue	float	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with REAL elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.5 AddToContainerElements (char)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT          hrPrev,
    char             nValue,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nValue	char	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with SINT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.6 AddToContainerElements (TcVnKeyPoint)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT          hrPrev,
    TcVnKeyPoint&    stValue,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stValue	TcVnKeyPoint [▶ 1643]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnKeyPoint elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.7 AddToContainerElements (TcVnPoint2_DINT)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT          hrPrev,
    TcVnPoint2_DINT& aValue,
    ITcVnContainer*  ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnPoint2_DINT [▶ 1588]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnPoint2_DINT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.8 AddToContainerElements (TcVnPoint2_LREAL)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT           hrPrev,
    TcVnPoint2_LREAL& aValue,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnPoint2_LREAL [▶ 1588]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnPoint2_LREAL elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.9 AddToContainerElements (TcVnPoint2_REAL)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT          hrPrev,
    TcVnPoint2_REAL& aValue,
    ITcVnContainer*  ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnPoint2_REAL [▶ 1588]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnPoint2_REAL elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.10 AddToContainerElements (TcVnPoint3_LREAL)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT           hrPrev,
    TcVnPoint3_LREAL& aValue,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnPoint3_LREAL [▶ 1588]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnPoint3_LREAL elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.11 AddToContainerElements (TcVnPoint3_REAL)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT          hrPrev,
    TcVnPoint3_REAL& aValue,
    ITcVnContainer*  ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnPoint3_REAL [▶ 1588]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnPoint3_REAL elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.12 AddToContainerElements (TcVnRectangle_DINT)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT              hrPrev,
    TcVnRectangle_DINT& stValue,
    ITcVnContainer*     ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stValue	TcVnRectangle_DINT [▶ 1657]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnRectangle_DINT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.13 AddToContainerElements (TcVnVector2_DINT)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT           hrPrev,
    TcVnVector2_DINT& aValue,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector2_DINT [▶ 1590]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector2_DINT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.14 AddToContainerElements (TcVnVector2_INT)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT          hrPrev,
    TcVnVector2_INT& aValue,
    ITcVnContainer*  ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector2_INT [▶ 1590]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector2_INT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.15 AddToContainerElements (TcVnVector2_REAL)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT           hrPrev,
    TcVnVector2_REAL& aValue,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector2_REAL [▶ 1590]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector2_REAL elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.16 AddToContainerElements (TcVnVector2_SINT)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT          hrPrev,
    TcVnVector2_SINT& aValue,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector2_SINT [▶ 1590]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector2_SINT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.17 AddToContainerElements (TcVnVector2_UINT)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT           hrPrev,
    TcVnVector2_UINT& aValue,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector2_UINT [▸ 1590]&	Value
ipContainer	ITcVnContainer* [▸ 1760]	Container with TcVnVector2_UINT elements

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.18 AddToContainerElements (TcVnVector2_USINT)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT            hrPrev,
    TcVnVector2_USINT& aValue,
    ITcVnContainer*    ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector2_USINT [▶ 1590]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector2_USINT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.19 AddToContainerElements (TcVnVector3_INT)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT          hrPrev,
    TcVnVector3_INT& aValue,
    ITcVnContainer*  ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector3_INT [▶ 1590]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector3_INT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.20 AddToContainerElements (TcVnVector3_REAL)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT            hrPrev,
    TcVnVector3_REAL&  aValue,
    ITcVnContainer*    ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector3_REAL [▸ 1590]&	Value
ipContainer	ITcVnContainer* [▸ 1760]	Container with TcVnVector3_REAL elements

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.21 AddToContainerElements (TcVnVector3_SINT)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT            hrPrev,
    TcVnVector3_SINT&  aValue,
    ITcVnContainer*    ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector3_SINT [▶ 1590]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector3_SINT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.22 AddToContainerElements (TcVnVector3_UINT)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT           hrPrev,
    TcVnVector3_UINT& aValue,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector3_UINT [▶ 1590]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector3_UINT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.23 AddToContainerElements (TcVnVector3_USINT)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT            hrPrev,
    TcVnVector3_USINT& aValue,
    ITcVnContainer*    ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector3_USINT [▶ 1590]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector3_USINT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.24 AddToContainerElements (TcVnVector4_DINT)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT           hrPrev,
    TcVnVector4_DINT& aValue,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector4_DINT [▶ 1590]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector4_DINT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.25 AddToContainerElements (TcVnVector4_INT)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT          hrPrev,
    TcVnVector4_INT& aValue,
    ITcVnContainer*  ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector4_INT [▶ 1590]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector4_INT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.26 AddToContainerElements (TcVnVector4_LREAL)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT            hrPrev,
    TcVnVector4_LREAL& aValue,
    ITcVnContainer*    ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector4_LREAL [▸ 1590]&	Value
ipContainer	ITcVnContainer* [▸ 1760]	Container with TcVnVector4_LREAL elements

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.27 AddToContainerElements (TcVnVector4_SINT)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT           hrPrev,
    TcVnVector4_SINT& aValue,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector4_SINT [▶ 1590]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector4_SINT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.28 AddToContainerElements (TcVnVector4_UINT)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT           hrPrev,
    TcVnVector4_UINT& aValue,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector4_UINT [▶ 1590]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector4_UINT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.29 AddToContainerElements (TcVnVector4_USINT)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT            hrPrev,
    TcVnVector4_USINT& aValue,
    ITcVnContainer*    ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector4_USINT [▸ 1590]&	Value
ipContainer	ITcVnContainer* [▸ 1760]	Container with TcVnVector4_USINT elements

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.30 AddToContainerElements (ULONG)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT         hrPrev,
    ULONG           nValue,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nValue	ULONG	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with UDINT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.31 AddToContainerElements (USHORT)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT         hrPrev,
    USHORT          nValue,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nValue	USHORT	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with UINT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.32 AddToContainerElements (ULONGLONG)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT          hrPrev,
    ULONGLONG        nValue,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nValue	ULONGLONG	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with ULINT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.1.33 AddToContainerElements (unsigned char)

Add a value to each container element.

Syntax

Definition:

```
HRESULT AddToContainerElements(
    HRESULT          hrPrev,
    unsigned char    nValue,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nValue	unsigned char	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with USINT elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.2 AddContainers

Element-wise addition of two containers (same length and type).

Syntax

Definition:

```
HRESULT AddContainers(
    HRESULT          hrPrev,
    ITcVnContainer*  ipSrcContainer1,
    ITcVnContainer*  ipSrcContainer2,
    ITcVnContainer*& ipDestContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer1	ITcVnContainer* [▶ 1760]	Source container 1
ipSrcContainer2	ITcVnContainer* [▶ 1760]	Source container 2
ipDestContainer	ITcVnContainer* [▶ 1760]&	Returns the resulting container

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.3 BitwiseAndContainers

Element-wise application of a bit-wise AND operator to two containers (same length and type, integer only).

Syntax

Definition:

```
HRESULT BitwiseAndContainers(
    HRESULT          hrPrev,
    ITcVnContainer*  ipSrcContainer1,
    ITcVnContainer*  ipSrcContainer2,
    ITcVnContainer*& ipDestContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer1	ITcVnContainer* [▶ 1760]	Source container 1
ipSrcContainer2	ITcVnContainer* [▶ 1760]	Source container 2
ipDestContainer	ITcVnContainer* [▶ 1760]&	Returns the resulting container

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.4 BitwiseNotContainer

Element-wise application of a bit-wise NOT operator to a container (integer only).

Syntax

Definition:

```
HRESULT BitwiseNotContainer(
    HRESULT          hrPrev,
    ITcVnContainer*  ipSrcContainer,
    ITcVnContainer*& ipDestContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▶ 1760]	Source container
ipDestContainer	ITcVnContainer* [▶ 1760]&	Returns the resulting container

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.5 BitwiseOrContainers

Element-wise application of a bit-wise OR operator to two containers (same length and type, integer only).

Syntax

Definition:

```
HRESULT BitwiseOrContainers(
    HRESULT          hrPrev,
    ITcVnContainer*  ipSrcContainer1,
    ITcVnContainer*  ipSrcContainer2,
    ITcVnContainer*& ipDestContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer1	ITcVnContainer* [▶ 1760]	Source container 1
ipSrcContainer2	ITcVnContainer* [▶ 1760]	Source container 2
ipDestContainer	ITcVnContainer* [▶ 1760]&	Returns the resulting container

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.6 BitwiseXorContainers

Element-wise application of a bit-wise XOR operator to two containers (same length and type, integer only).

Syntax

Definition:

```
HRESULT BitwiseXorContainers(
    HRESULT          hrPrev,
    ITcVnContainer*  ipSrcContainer1,
    ITcVnContainer*  ipSrcContainer2,
    ITcVnContainer*& ipDestContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer1	ITcVnContainer* [▶ 1760]	Source container 1
ipSrcContainer2	ITcVnContainer* [▶ 1760]	Source container 2
ipDestContainer	ITcVnContainer* [▶ 1760]&	Returns the resulting container

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.7 DivideContainers

Element-wise division of two containers (same length and type). In case of integer division by zero, the resulting element is set to nIntDivideByZeroResult and S_DIVISION_BY_ZERO is returned instead of S_OK.

Syntax

Definition:

```
HRESULT DivideContainers(
    HRESULT          hrPrev,
    ITcVnContainer*  ipSrcContainer1,
    ITcVnContainer*  ipSrcContainer2,
```

```
    ITcVnContainer*& ipDestContainer,
    LONGLONG         nIntDivideByZeroResult = 0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer1	ITcVnContainer* [▶ 1760]		Source container 1
ipSrcContainer2	ITcVnContainer* [▶ 1760]		Source container 2
ipDestContainer	ITcVnContainer* [▶ 1760]&		Returns the resulting container
nIntDivideByZeroResult	LONGLONG	0	Value set as result in case of integer division by zero

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.8 MaxContainer

Element-wise maximum of two containers (same length and type).

Syntax

Definition:

```
HRESULT MaxContainer(
    HRESULT          hrPrev,
    ITcVnContainer*  ipSrcContainer1,
    ITcVnContainer*  ipSrcContainer2,
    ITcVnContainer*& ipDestContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer1	ITcVnContainer* [▶ 1760]	Source container 1
ipSrcContainer2	ITcVnContainer* [▶ 1760]	Source container 2
ipDestContainer	ITcVnContainer* [▶ 1760]&	Returns the resulting container

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.9 MinContainer

Element-wise minimum of two containers (same length and type).

Syntax

Definition:

```
HRESULT MinContainer(
    HRESULT           hrPrev,
    ITcVnContainer*   ipSrcContainer1,
    ITcVnContainer*   ipSrcContainer2,
    ITcVnContainer*&  ipDestContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer1	ITcVnContainer* [▶ 1760]	Source container 1
ipSrcContainer2	ITcVnContainer* [▶ 1760]	Source container 2
ipDestContainer	ITcVnContainer* [▶ 1760]&	Returns the resulting container

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.10 MultiplyContainers

Element-wise multiplication of two containers (same length and type).

Syntax

Definition:

```
HRESULT MultiplyContainers(
    HRESULT          hrPrev,
    ITcVnContainer*  ipSrcContainer1,
    ITcVnContainer*  ipSrcContainer2,
    ITcVnContainer*& ipDestContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer1	ITcVnContainer* [▶ 1760]	Source container 1
ipSrcContainer2	ITcVnContainer* [▶ 1760]	Source container 2
ipDestContainer	ITcVnContainer* [▶ 1760]&	Returns the resulting container

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.11 MultiplyWithContainerElements (1)

Multiply each container element with a value.

Syntax

Definition:

```
HRESULT MultiplyWithContainerElements(
    HRESULT         hrPrev,
    double          fValue,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fValue	double	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with 1-dimensional elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.12 MultiplyWithContainerElements (2)

Multiply each container element with a value.

Syntax

Definition:

```
HRESULT MultiplyWithContainerElements(
    HRESULT             hrPrev,
    TcVnVector2_LREAL& aValue,
    ITcVnContainer*     ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector2_LREAL [▶ 1590]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with 2-dimensional elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.13 MultiplyWithContainerElements (3)

Multiply each container element with a value.

Syntax

Definition:

```
HRESULT MultiplyWithContainerElements(
    HRESULT            hrPrev,
    TcVnVector3_LREAL& aValue,
    ITcVnContainer*    ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector3_LREAL [▶ 1590]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with 3-dimensional elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.14 MultiplyWithContainerElements (4)

Multiply each container element with a value.

Syntax

Definition:

```
HRESULT MultiplyWithContainerElements(
    HRESULT            hrPrev,
    TcVnVector4_LREAL& aValue,
    ITcVnContainer*    ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aValue	TcVnVector4_LREAL [▶ 1590]&	Value
ipContainer	ITcVnContainer* [▶ 1760]	Container with 4-dimensional elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.15 NegateContainer

Element-wise negation of a container (two's complement).

Syntax

Definition:

```
HRESULT NegateContainer(
    HRESULT           hrPrev,
    ITcVnContainer*   ipSrcContainer,
    ITcVnContainer*&  ipDestContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▶ 1760]	Source container
ipDestContainer	ITcVnContainer* [▶ 1760]&	Returns the resulting container (same type as ipSrcContainer)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.2.16 SubtractContainers

Element-wise subtraction of two containers (same length and type).

Syntax

Definition:

```
HRESULT SubtractContainers(
    HRESULT          hrPrev,
    ITcVnContainer*  ipSrcContainer1,
    ITcVnContainer*  ipSrcContainer2,
    ITcVnContainer*& ipDestContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer1	ITcVnContainer* [▸ 1760]	Source container 1
ipSrcContainer2	ITcVnContainer* [▸ 1760]	Source container 2
ipDestContainer	ITcVnContainer* [▸ 1760]&	Returns the resulting container

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3 Algebraic Image Operations

该组包含对图像 [▸ 130]逐像素应用代数运算的函数。

加法

- AddImages [▸ 1869]
- AddImagesWeighted [▸ 1869]
- AddScalarToImage [▸ 1870]
- AddVectorToImage [▸ 1871]

减法

- SubtractImages [▸ 1901]
- SubtractImageFromScalar [▸ 1900]

- SubtractImageFromVector [▸ 1900]
- SubtractScalarFromImage [▸ 1902]
- SubtractVectorFromImage [▸ 1903]

乘法

- MultiplyImages [▸ 1898]
- MultiplyImageWithScalar [▸ 1898]
- MultiplyImageWithVector [▸ 1899]

除法

- DivideImages [▸ 1885]
- DivideImageByScalar [▸ 1883]
- DivideImageByVector [▸ 1884]
- DivideScalarByImage [▸ 1885]
- DivideVectorByImage [▸ 1886]

乘方和对数化

- ElementwiseExp [▸ 1887]
- ElementwiseLog [▸ 1887]

逐位 AND

- BitwiseAndImages(Exp) [▸ 1872]
- BitwiseAndScalarWithImage [▸ 1873]
- BitwiseAndVectorWithImage [▸ 1874]

逐位 NOT

- BitwiseNotImage(Exp) [▸ 1875]

逐位 OR

- BitwiseOrImages(Exp) [▸ 1876]
- BitwiseOrScalarWithImage [▸ 1878]
- BitwiseOrVectorWithImage [▸ 1878]

逐位 XOR

- BitwiseXorImages(Exp) [▸ 1879]
- BitwiseXorScalarWithImage [▸ 1881]
- BitwiseXorVectorWithImage [▸ 1881]

交叉混合

- BlendImages [▸ 1882]

最大

- MaxImage(Exp) [▸ 1888]
- MaxImageWithScalar(Exp) [▸ 1890]
- MaxImageWithVector(Exp) [▸ 1891]

最小

- MinImage(Exp) [▸ 1893]
- MinImageWithScalar(Exp) [▸ 1895]
- MinImageWithVector(Exp) [▸ 1896]

6.2.3.3.1 AddImages

Element-wise addition of two images using saturation arithmetics.

Syntax

Definition:

```
HRESULT AddImages(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage1,
    ITcVnImage*   ipSrcImage2,
    ITcVnImage*&  ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage1	ITcVnImage* [▶ 1797]	First source image
ipSrcImage2	ITcVnImage* [▶ 1797]	Second source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

样本

- 复制图像区域 [▶ 2643]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.2 AddImagesWeighted

Weighted, element-wise addition of two images using saturation arithmetics.

Syntax

Definition:

```
HRESULT AddImagesWeighted(
    HRESULT           hrPrev,
    ITcVnImage*       ipSrcImage1,
    ITcVnImage*       ipSrcImage2,
    ITcVnImage*&      ipDestImage,
    double            fWeight1,
    double            fWeight2,
    double            fDelta,
    ETcVnElementType eDestType = TCVN_ET_SAME_AS_SOURCE
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage1	ITcVnImage* [▶ 1797]		First source image
ipSrcImage2	ITcVnImage* [▶ 1797]		Second source image
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (An appropriate destination image will be created if required.)
fWeight1	double		Weight factor applied to ipSrcImage1
fWeight2	double		Weight factor applied to ipSrcImage2
fDelta	double		Value added to the weighted sum of both images
eDestType	ETcVnElementType [▶ 1615]	TCVN_ET_SAME_AS_SOURCE	Destination image depth

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.3 AddScalarToImage

Add a scalar value to each image pixel using saturation arithmetics.

Syntax

Definition:

```
HRESULT AddScalarToImage(
    HRESULT      hrPrev,
    double       fScalar,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fScalar	double	Scalar value
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.4 AddVectorToImage

Add a vector (1 element for each image channel) to each image pixel using saturation arithmetics.

Syntax

Definition:

```
HRESULT AddVectorToImage(
    HRESULT            hrPrev,
    TcVnVector4_LREAL& aVector,
    ITcVnImage*        ipSrcImage,
    ITcVnImage*&       ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aVector	TcVnVector4_LREAL [▶ 1590]&	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.5 BitwiseAndImages

Element-wise application of a bit-wise AND operator to two images.

Syntax

Definition:

```
HRESULT BitwiseAndImages(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage1,
    ITcVnImage*   ipSrcImage2,
    ITcVnImage*&  ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage1	ITcVnImage* [▶ 1797]	First source image
ipSrcImage2	ITcVnImage* [▶ 1797]	Second source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.6 BitwiseAndImages (Exp)

Element-wise application of a bit-wise AND operator to two images.

Syntax

Definition:

```
HRESULT BitwiseAndImages(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage1,
    ITcVnImage*  ipSrcImage2,
    ITcVnImage*& ipDestImage,
    ITcVnImage*  ipMask
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage1	ITcVnImage* [▶ 1797]	First source image
ipSrcImage2	ITcVnImage* [▶ 1797]	Second source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage* [▶ 1797]	Mask to limit the operation to specific pixel positions (mask value 0: skip pixel, mask value > 0: apply operation to pixel)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.7 BitwiseAndScalarWithImage

Bitwise and a scalar value with each image pixel.

Syntax

Definition:

```
HRESULT BitwiseAndScalarWithImage(
    HRESULT      hrPrev,
    double       fScalar,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fScalar	double	Scalar value
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.8 BitwiseAndVectorWithImage

Bitwise and a vector (1 element for each image channel) with each image pixel.

Syntax

Definition:

```
HRESULT BitwiseAndVectorWithImage(
    HRESULT             hrPrev,
    TcVnVector4_LREAL&  aVector,
    ITcVnImage*         ipSrcImage,
    ITcVnImage*&        ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aVector	TcVnVector4_LREAL [▶ 1590]&	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.9 BitwiseNotImage

Element-wise application of a bit-wise NOT operator to an image.

Syntax

Definition:

```
HRESULT BitwiseNotImage(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage,
    ITcVnImage*&  ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.10 BitwiseNotImage (Exp)

Element-wise application of a bit-wise NOT operator to an image.

Syntax

Definition:

```
HRESULT BitwiseNotImage(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage,
    ITcVnImage*  ipMask
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage* [▶ 1797]	Mask to limit the operation to specific pixel positions (mask value 0: skip pixel, mask value > 0: apply operation to pixel)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.11 BitwiseOrImages

Element-wise application of a bit-wise OR operator to two images.

Syntax

Definition:

```
HRESULT BitwiseOrImages(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage1,
    ITcVnImage*  ipSrcImage2,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage1	ITcVnImage* [▶ 1797]	First source image
ipSrcImage2	ITcVnImage* [▶ 1797]	Second source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.12 BitwiseOrImages (Exp)

Element-wise application of a bit-wise OR operator to two images.

Syntax

Definition:

```
HRESULT BitwiseOrImages(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage1,
    ITcVnImage*   ipSrcImage2,
    ITcVnImage*&  ipDestImage,
    ITcVnImage*   ipMask
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage1	ITcVnImage* [▶ 1797]	First source image
ipSrcImage2	ITcVnImage* [▶ 1797]	Second source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage* [▶ 1797]	Mask to limit the operation to specific pixel positions (mask value 0: skip pixel, mask value > 0: apply operation to pixel)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.13 BitwiseOrScalarWithImage

Bitwise or a scalar value with each image pixel.

Syntax

Definition:

```
HRESULT BitwiseOrScalarWithImage(
    HRESULT      hrPrev,
    double       fScalar,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fScalar	double	Scalar value
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.14 BitwiseOrVectorWithImage

Bitwise or a vector (1 element for each image channel) with each image pixel.

Syntax

Definition:

```
HRESULT BitwiseOrVectorWithImage(
    HRESULT             hrPrev,
    TcVnVector4_LREAL&  aVector,
    ITcVnImage*         ipSrcImage,
    ITcVnImage*&        ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aVector	TcVnVector4_LREAL [▸ 1590]&	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)
ipSrcImage	ITcVnImage* [▸ 1797]	Source image
ipDestImage	ITcVnImage* [▸ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.15 BitwiseXorImages

Element-wise application of a bit-wise XOR operator to two images.

Syntax

Definition:

```
HRESULT BitwiseXorImages(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage1,
    ITcVnImage*  ipSrcImage2,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage1	ITcVnImage* [▶ 1797]	First source image
ipSrcImage2	ITcVnImage* [▶ 1797]	Second source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.16 BitwiseXorImages (Exp)

Element-wise application of a bit-wise XOR operator to two images.

Syntax

Definition:

```
HRESULT BitwiseXorImages(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage1,
    ITcVnImage*   ipSrcImage2,
    ITcVnImage*&  ipDestImage,
    ITcVnImage*   ipMask
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage1	ITcVnImage* [▶ 1797]	First source image
ipSrcImage2	ITcVnImage* [▶ 1797]	Second source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage* [▶ 1797]	Mask to limit the operation to specific pixel positions (mask value 0: skip pixel, mask value > 0: apply operation to pixel)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.17 BitwiseXorScalarWithImage

Bitwise xor a scalar value with each image pixel.

Syntax

Definition:

```
HRESULT BitwiseXorScalarWithImage(
    HRESULT      hrPrev,
    double       fScalar,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fScalar	double	Scalar value
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.18 BitwiseXorVectorWithImage

Bitwise xor a vector (1 element for each image channel) with each image pixel.

Syntax

Definition:

```
HRESULT BitwiseXorVectorWithImage(
    HRESULT             hrPrev,
    TcVnVector4_LREAL& aVector,
    ITcVnImage*         ipSrcImage,
    ITcVnImage*&        ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aVector	TcVnVector4_LREAL [▶ 1590]&	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.19 BlendImages

Blends two images, i.e. dest = (src1 * weight1 + src2 * weight2) / (weight1 + weight2).
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT BlendImages(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage1,
    ITcVnImage*  ipSrcImage2,
    ITcVnImage*  ipWeights1,
    ITcVnImage*  ipWeights2,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage1	ITcVnImage* [▸ 1797]	First source image (ET_USINT or ET_REAL)
ipSrcImage2	ITcVnImage* [▸ 1797]	Second source image (same type and size as ipSrcImage1)
ipWeights1	ITcVnImage* [▸ 1797]	Weights for ipSrcImage1 (1 channel, ET_REAL)
ipWeights2	ITcVnImage* [▸ 1797]	Weights for ipSrcImage2 (1 channel, ET_REAL)
ipDestImage	ITcVnImage* [▸ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.20 DivideImageByScalar

Divide each image pixel by a scalar value using saturation arithmetics.

Syntax

Definition:

```
HRESULT DivideImageByScalar(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage,
    double        fScalar,
    ITcVnImage*&  ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]	Source image
fScalar	double	Scalar value
ipDestImage	ITcVnImage* [▸ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.21 DivideImageByVector

Divides each image pixel by a vector (1 element for each image channel) using saturation arithmetics.

Syntax

Definition:

```
HRESULT DivideImageByVector(
    HRESULT            hrPrev,
    ITcVnImage*        ipSrcImage,
    TcVnVector4_LREAL& aVector,
    ITcVnImage*&       ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
aVector	TcVnVector4_LREAL [▶ 1590]&	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.22 DivideImages

Element-wise division of two images using saturation arithmetics. (A division by zero equals zero.)

Syntax

Definition:

```
HRESULT DivideImages(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage1,
    ITcVnImage*   ipSrcImage2,
    ITcVnImage*&  ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage1	ITcVnImage* [▸ 1797]	First source image (dividend)
ipSrcImage2	ITcVnImage* [▸ 1797]	Second source image (divisor)
ipDestImage	ITcVnImage* [▸ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.23 DivideScalarByImage

Divides a scalar value by each image pixel using saturation arithmetics.

Syntax

Definition:

```
HRESULT DivideScalarByImage(
    HRESULT       hrPrev,
    double        fScalar,
    ITcVnImage*   ipSrcImage,
    ITcVnImage*&  ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fScalar	double	Scalar value
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.24 DivideVectorByImage

Divides a vector (1 element for each image channel) by each image pixel using saturation arithmetics.

Syntax

Definition:

```
HRESULT DivideVectorByImage(
    HRESULT             hrPrev,
    TcVnVector4_LREAL&  aVector,
    ITcVnImage*         ipSrcImage,
    ITcVnImage*&        ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aVector	TcVnVector4_LREAL [▶ 1590]&	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.25 ElementwiseExp

Computes the natural exponent of each pixel value.

Syntax

Definition:

```
HRESULT ElementwiseExp(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (REAL or LREAL)
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (Same type and size as ipSrcImage, an appropriate image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.26 ElementwiseLog

Computes the natural logarithm of each pixel value.

Syntax

Definition:

```
HRESULT ElementwiseLog(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (REAL or LREAL)
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (Same type and size as ipSrcImage, an appropriate image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.27 MaxImage

Element-wise maximum of two images.

Syntax

Definition:

```
HRESULT MaxImage(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage1,
    ITcVnImage*  ipSrcImage2,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage1	ITcVnImage* [▶ 1797]	First source image
ipSrcImage2	ITcVnImage* [▶ 1797]	Second source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.28 MaxImage (Exp)

Element-wise maximum of two images.

Syntax

Definition:

```
HRESULT MaxImage(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage1,
    ITcVnImage*   ipSrcImage2,
    ITcVnImage*&  ipDestImage,
    ITcVnImage*   ipMask
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage1	ITcVnImage* [▶ 1797]	First source image
ipSrcImage2	ITcVnImage* [▶ 1797]	Second source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage* [▶ 1797]	Mask of type USINT

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.29 MaxImageWithScalar

Element-wise maximum of image and scalar value.

Syntax

Definition:

```
HRESULT MaxImageWithScalar(
    HRESULT       hrPrev,
    double        fScalar,
    ITcVnImage*   ipSrcImage,
    ITcVnImage*&  ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fScalar	double	Scalar value
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.30 MaxImageWithScalar (Exp)

Element-wise maximum of image and scalar value.

Syntax

Definition:

```
HRESULT MaxImageWithScalar(
    HRESULT      hrPrev,
    double       fScalar,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage,
    ITcVnImage*  ipMask
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fScalar	double	Scalar value
ipSrcImage	ITcVnImage* [▸ 1797]	Source image
ipDestImage	ITcVnImage* [▸ 1797]&	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage* [▸ 1797]	Mask of type USINT

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.31 MaxImageWithVector

Element-wise maximum of image and vector (1 element for each image channel).

Syntax

Definition:

```
HRESULT MaxImageWithVector(
    HRESULT             hrPrev,
    TcVnVector4_LREAL&  aVector,
    ITcVnImage*         ipSrcImage,
    ITcVnImage*&        ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aVector	TcVnVector4_LREAL [▶ 1590]&	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.32 MaxImageWithVector (Exp)

Element-wise maximum of image and vector (1 element for each image channel).

Syntax

Definition:

```
HRESULT MaxImageWithVector(
    HRESULT            hrPrev,
    TcVnVector4_LREAL& aVector,
    ITcVnImage*        ipSrcImage,
    ITcVnImage*&       ipDestImage,
    ITcVnImage*        ipMask
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aVector	TcVnVector4_LREAL [▶ 1590]&	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage* [▶ 1797]	Mask of type USINT

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.33 MinImage

Element-wise minimum of two images.

Syntax

Definition:

```
HRESULT MinImage(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage1,
    ITcVnImage*   ipSrcImage2,
    ITcVnImage*&  ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage1	ITcVnImage* [▶ 1797]	First source image
ipSrcImage2	ITcVnImage* [▶ 1797]	Second source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.34 MinImage (Exp)

Element-wise minimum of two images.

Syntax

Definition:

```
HRESULT MinImage(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage1,
    ITcVnImage*   ipSrcImage2,
    ITcVnImage*&  ipDestImage,
    ITcVnImage*   ipMask
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage1	ITcVnImage* [▶ 1797]	First source image
ipSrcImage2	ITcVnImage* [▶ 1797]	Second source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage* [▶ 1797]	Mask of type USINT

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.35 MinImageWithScalar

Element-wise minimum of image and scalar value.

Syntax

Definition:

```
HRESULT MinImageWithScalar(
    HRESULT      hrPrev,
    double       fScalar,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fScalar	double	Scalar value
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.36 MinImageWithScalar (Exp)

Element-wise minimum of image and scalar value.

Syntax

Definition:

```
HRESULT MinImageWithScalar(
    HRESULT      hrPrev,
    double       fScalar,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage,
    ITcVnImage*  ipMask
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fScalar	double	Scalar value
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage* [▶ 1797]	Mask of type USINT

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.37 MinImageWithVector

Element-wise minimum of image and vector (1 element for each image channel).

Syntax

Definition:

```
HRESULT MinImageWithVector(
    HRESULT              hrPrev,
    TcVnVector4_LREAL&   aVector,
    ITcVnImage*          ipSrcImage,
    ITcVnImage*&         ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aVector	TcVnVector4_LREAL [▶ 1590]&	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.38 MinImageWithVector (Exp)

Element-wise minimum of image and vector (1 element for each image channel).

Syntax

Definition:

```
HRESULT MinImageWithVector(
    HRESULT             hrPrev,
    TcVnVector4_LREAL&  aVector,
    ITcVnImage*         ipSrcImage,
    ITcVnImage*&        ipDestImage,
    ITcVnImage*         ipMask
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aVector	TcVnVector4_LREAL [▶ 1590]&	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
ipMask	ITcVnImage* [▶ 1797]	Mask of type USINT

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.39 MultiplyImages

Element-wise multiplication of two images using saturation arithmetics.

Syntax

Definition:

```
HRESULT MultiplyImages(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage1,
    ITcVnImage*   ipSrcImage2,
    ITcVnImage*&  ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage1	ITcVnImage* [▶ 1797]	First source image
ipSrcImage2	ITcVnImage* [▶ 1797]	Second source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.40 MultiplyImageWithScalar

Multiply each image pixel by a scalar value using saturation arithmetics.

Syntax

Definition:

```
HRESULT MultiplyImageWithScalar(
    HRESULT       hrPrev,
    double        fScalar,
    ITcVnImage*   ipSrcImage,
    ITcVnImage*&  ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fScalar	double	Scalar value
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.41 MultiplyImageWithVector

Multiply each image pixel by a vector (1 element for each image channel) using saturation arithmetics.

Syntax

Definition:

```
HRESULT MultiplyImageWithVector(
    HRESULT            hrPrev,
    TcVnVector4_LREAL& aVector,
    ITcVnImage*        ipSrcImage,
    ITcVnImage*&       ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aVector	TcVnVector4_LREAL [▶ 1590]&	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.42 SubtractImageFromScalar

Subtract each image pixel from a scalar value using saturation arithmetics.

Syntax

Definition:

```
HRESULT SubtractImageFromScalar(
    HRESULT       hrPrev,
    double        fScalar,
    ITcVnImage*   ipSrcImage,
    ITcVnImage*&  ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fScalar	double	Scalar value
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.43 SubtractImageFromVector

Subtract each image pixel from a vector (1 element for each image channel) using saturation arithmetics.

Syntax

Definition:

```
HRESULT SubtractImageFromVector(
    HRESULT             hrPrev,
    TcVnVector4_LREAL&  aVector,
    ITcVnImage*         ipSrcImage,
    ITcVnImage*&        ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aVector	TcVnVector4_LREAL [▶ 1590]&	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.44 SubtractImages

Element-wise subtraction of two images using saturation arithmetics.

Syntax

Definition:

```
HRESULT SubtractImages(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage1,
    ITcVnImage*  ipSrcImage2,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage1	ITcVnImage* [▶ 1797]	First source image (minuend)
ipSrcImage2	ITcVnImage* [▶ 1797]	Second source image (subtrahend)
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.45 SubtractScalarFromImage

Subtract a scalar value from each image pixel using saturation arithmetics.

Syntax

Definition:

```
HRESULT SubtractScalarFromImage(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage,
    double       fScalar,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
fScalar	double	Scalar value
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.3.46 SubtractVectorFromImage

Subtract a vector (1 element for each image channel) from each image pixel using saturation arithmetics.

Syntax

Definition:

```
HRESULT SubtractVectorFromImage(
    HRESULT             hrPrev,
    ITcVnImage*         ipSrcImage,
    TcVnVector4_LREAL&  aVector,
    ITcVnImage*&        ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
aVector	TcVnVector4_LREAL [▶ 1590]&	4-element vector (1 element for each image channel. If the image has less channels, the further vector elements are ignored.)
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4 Basic Container Operations

该组包含处理容器 [▶ 132]的函数。

函数

创建容器

- CopyContainer [▶ 2063]
- CopyContainerElementsConditional_ITcVnContainer [▶ 2063]
- CopyContainerElementsConditional_ITcVnForwardIterator [▶ 2063]
- CreateContainer [▶ 2064]
- CreateContainerFromArray [▶ 2065]
- ExtractContainerRange [▶ 2071]
- ReserveContainerMemory [▶ 2077]

容器元素的处理

- AppendToContainer [▶ 1904]
- CheckIfEmpty [▶ 2060]
- ConvertContainerType [▶ 2061]
- EraseFromContainer [▶ 2066]
- FillContainer(Exp) [▶ 1952]
- GetAt [▶ 1978]
- GetContainer(Exp) [▶ 2072]
- GetNumberOfElements [▶ 2074]
- InsertIntoContainer [▶ 2004]
- ReverseContainer [▶ 2077]
- SetAt [▶ 2032]
- SetContainer [▶ 2078]

将容器数据导出到内存区域

- ExportContainer [▶ 2066]
- ExportSubContainer [▶ 2069]
- ExportContainer_String [▶ 2067]
- ExportSubContainer_String [▶ 2070]
- ExportContainerSize [▶ 2068]
- ExportSubContainerSize [▶ 2071]

迭代器的处理

- AdvanceIterator [▶ 2059]
- CheckIfIteratorIsAtEnd [▶ 2061]
- GetForwardIterator [▶ 2073]
- GetRandomAccessIterator [▶ 2075]
- IncrementIterator [▶ 2075]
- IteratorDistance [▶ 2076]
- SetIteratorToBegin [▶ 2079]

6.2.3.4.1 AppendToContainer

本章包含根据数据类型向容器中添加元素的功能。

6.2.3.4.1.1 AppendToContainer (LONG)

Append a single element of type DINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT          hrPrev,
    LONG             nElement,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	LONG	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.2 AppendToContainer (SHORT)

Append a single element of type INT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT          hrPrev,
    SHORT            nElement,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	SHORT	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.3 AppendToContainer (ITcVnContainer*)

Append a single element to a container or concatenate containers (if ipElement has the same typeId as ipContainer).

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT         hrPrev,
    ITcVnContainer* ipElement,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipElement	ITcVnContainer* [▶ 1760]	Single element to append to ipContainer or container with several elements to be concatenated
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element(s) will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.4 AppendToContainer (ITcVnForwardIterator*)

Append a single element (represented by an iterator)to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT                hrPrev,
    ITcVnForwardIterator* ipElement,
    ITcVnContainer*        ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipElement	ITcVnForwardIterator* [▶ 1752]	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element(s) will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.5 AppendToContainer (ITcVnImage*)

Append a single element of type ITcVnImage to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT          hrPrev,
    ITcVnImage*      ipElement,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipElement	ITcVnImage* [▶ 1797]	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.6 AppendToContainer (double)

Append a single element of type LREAL to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT          hrPrev,
    double           fElement,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fElement	double	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▸ 1760]	Container to which the element will be appended

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.7 AppendToContainer (float)

Append a single element of type REAL to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT         hrPrev,
    float           fElement,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fElement	float	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.8 AppendToContainer (char)

Append a single element of type SINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT         hrPrev,
    char            nElement,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	char	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.9 AppendToContainer (TcVnDMatch)

Append a single element of type TcVnDMatch to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT          hrPrev,
    TcVnDMatch&      stElement,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stElement	TcVnDMatch [▶ 1643]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.10 AppendToContainer (TcVnKeyPoint)

Append a single element of type TcVnKeyPoint to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT          hrPrev,
    TcVnKeyPoint&    stElement,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stElement	TcVnKeyPoint [▶ 1643]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.11 AppendToContainer (TcVnPoint2_DINT)

Append a single element of type TcVnPoint2_DINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT          hrPrev,
    TcVnPoint2_DINT& aElement,
    ITcVnContainer*  ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnPoint2_DINT [▶ 1588]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.12 AppendToContainer (TcVnPoint2_LREAL)

Append a single element of type TcVnPoint2_LREAL to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT           hrPrev,
    TcVnPoint2_LREAL& aElement,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnPoint2_LREAL [▶ 1588]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.13 AppendToContainer (TcVnPoint2_REAL)

Append a single element of type TcVnPoint2_REAL to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT          hrPrev,
    TcVnPoint2_REAL& aElement,
    ITcVnContainer*  ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnPoint2_REAL [▶ 1588]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.14 AppendToContainer (TcVnPoint3_LREAL)

Append a single element of type TcVnPoint3_LREAL to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT           hrPrev,
    TcVnPoint3_LREAL& aElement,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnPoint3_LREAL [▶ 1588]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.15 AppendToContainer (TcVnPoint3_REAL)

Append a single element of type TcVnPoint3_REAL to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT           hrPrev,
    TcVnPoint3_REAL&  aElement,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnPoint3_REAL [▶ 1588]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.16 AppendToContainer (TcVnRectangle_DINT)

Append a single element of type TcVnRectangle_DINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT              hrPrev,
    TcVnRectangle_DINT&  stElement,
    ITcVnContainer*      ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stElement	TcVnRectangle_DINT [▶ 1657]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.17 AppendToContainer (TcVnVector2_DINT)

Append a single element of type TcVnVector2_DINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT           hrPrev,
    TcVnVector2_DINT& aElement,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector2_DINT [▶ 1590]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.18 AppendToContainer (TcVnVector2_INT)

Append a single element of type TcVnVector2_INT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT          hrPrev,
    TcVnVector2_INT& aElement,
    ITcVnContainer*  ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector2_INT [▶ 1590]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.19 AppendToContainer (TcVnVector2_REAL)

Append a single element of type TcVnVector2_REAL to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT           hrPrev,
    TcVnVector2_REAL& aElement,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector2_REAL [▶ 1590]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.20 AppendToContainer (TcVnVector2_SINT)

Append a single element of type TcVnVector2_SINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT           hrPrev,
    TcVnVector2_SINT& aElement,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector2_SINT [▶ 1590]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.21 AppendToContainer (TcVnVector2_UINT)

Append a single element of type TcVnVector2_UINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT            hrPrev,
    TcVnVector2_UINT&  aElement,
    ITcVnContainer*    ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector2_UINT [▸ 1590]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▸ 1760]	Container to which the element will be appended

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.22 AppendToContainer (TcVnVector2_USINT)

Append a single element of type TcVnVector2_USINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT             hrPrev,
    TcVnVector2_USINT&  aElement,
    ITcVnContainer*     ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector2_USINT [▶ 1590]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.23 AppendToContainer (TcVnVector3_INT)

Append a single element of type TcVnVector3_INT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT           hrPrev,
    TcVnVector3_INT& aElement,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector3_INT [▶ 1590]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.24 AppendToContainer (TcVnVector3_REAL)

Append a single element of type TcVnVector3_REAL to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT            hrPrev,
    TcVnVector3_REAL&  aElement,
    ITcVnContainer*    ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector3_REAL [▸ 1590]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▸ 1760]	Container to which the element will be appended

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.25 AppendToContainer (TcVnVector3_SINT)

Append a single element of type TcVnVector3_SINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT            hrPrev,
    TcVnVector3_SINT&  aElement,
    ITcVnContainer*    ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector3_SINT [▶ 1590]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.26 AppendToContainer (TcVnVector3_UINT)

Append a single element of type TcVnVector3_UINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT           hrPrev,
    TcVnVector3_UINT& aElement,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector3_UINT [▶ 1590]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.27 AppendToContainer (TcVnVector3_USINT)

Append a single element of type TcVnVector3_USINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT             hrPrev,
    TcVnVector3_USINT& aElement,
    ITcVnContainer*     ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector3_USINT [▶ 1590]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.28 AppendToContainer (TcVnVector4_DINT)

Append a single element of type TcVnVector4_DINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT            hrPrev,
    TcVnVector4_DINT& aElement,
    ITcVnContainer*    ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector4_DINT [▸ 1590]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▸ 1760]	Container to which the element will be appended

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.29 AppendToContainer (TcVnVector4_INT)

Append a single element of type TcVnVector4_INT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT          hrPrev,
    TcVnVector4_INT& aElement,
    ITcVnContainer*  ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector4_INT [▸ 1590]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▸ 1760]	Container to which the element will be appended

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.30 AppendToContainer (TcVnVector4_LREAL)

Append a single element of type TcVnVector4_LREAL to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT            hrPrev,
    TcVnVector4_LREAL& aElement,
    ITcVnContainer*    ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector4_LREAL [▶ 1590]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.31 AppendToContainer (TcVnVector4_SINT)

Append a single element of type TcVnVector4_SINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT           hrPrev,
    TcVnVector4_SINT& aElement,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector4_SINT [▶ 1590]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.32 AppendToContainer (TcVnVector4_UINT)

Append a single element of type TcVnVector4_UINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT           hrPrev,
    TcVnVector4_UINT& aElement,
    ITcVnContainer*   ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector4_UINT [▶ 1590]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.33 AppendToContainer (TcVnVector4_USINT)

Append a single element of type TcVnVector4_USINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT             hrPrev,
    TcVnVector4_USINT&  aElement,
    ITcVnContainer*     ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector4_USINT [▶ 1590]&	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.34 AppendToContainer (ULONG)

Append a single element of type UDINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT         hrPrev,
    ULONG           nElement,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	ULONG	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.35 AppendToContainer (USHORT)

Append a single element of type UINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT         hrPrev,
    USHORT          nElement,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	USHORT	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.36 AppendToContainer (ULONGLONG)

Append a single element of type ULINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT          hrPrev,
    ULONGLONG        nElement,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	ULONGLONG	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.1.37 AppendToContainer (unsigned char)

Append a single element of type USINT to a container.

Syntax

Definition:

```
HRESULT AppendToContainer(
    HRESULT          hrPrev,
    unsigned char    nElement,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	unsigned char	Single element to append to ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container to which the element will be appended

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2 FillContainer

本章包含根据数据类型用特殊值填充容器的功能。这个函数适用于包括基本元素的容器。

6.2.3.4.2.1 FillContainer (LONG)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    LONG            nValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
nValue	LONG	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.2 FillContainer (SHORT)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    SHORT            nValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
nValue	SHORT	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.3 FillContainer (ITcVnImage*)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    ITcVnImage*      ipValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
ipValue	ITcVnImage* [▶ 1797]	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.4 FillContainer (double)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    double          fValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
fValue	double	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.5 FillContainer (float)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    float            fValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Container with basic elements
fValue	float	Value to set the container elements

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.6 FillContainer (char)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    char             nValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Container with basic elements
nValue	char	Value to set the container elements

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.7 FillContainer (TcVnDMatch)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    TcVnDMatch&      stValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Container with basic elements
stValue	TcVnDMatch [▸ 1643]&	Value to set the container elements

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.8 FillContainer (TcVnKeyPoint)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    TcVnKeyPoint&   stValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
stValue	TcVnKeyPoint [▶ 1643]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.9 FillContainer (TcVnPoint2_DINT)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnPoint2_DINT& aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnPoint2_DINT [▶ 1588]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.10 FillContainer (TcVnPoint2_LREAL)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnPoint2_LREAL&  aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnPoint2_LREAL [▶ 1588]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.11 FillContainer (TcVnPoint2_REAL)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnPoint2_REAL& aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnPoint2_REAL [▶ 1588]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.12 FillContainer (TcVnPoint3_LREAL)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnPoint3_LREAL& aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnPoint3_LREAL [▶ 1588]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.13 FillContainer (TcVnPoint3_REAL)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnPoint3_REAL& aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnPoint3_REAL [▶ 1588]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.14 FillContainer (TcVnRectangle_DINT)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT              hrPrev,
    ITcVnContainer*      ipContainer,
    TcVnRectangle_DINT& stValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
stValue	TcVnRectangle_DINT [▶ 1657]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.15 FillContainer (TcVnVector2_DINT)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector2_DINT& aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector2_DINT [▶ 1590]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.16 FillContainer (TcVnVector2_INT)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnVector2_INT& aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector2_INT [▶ 1590]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.17 FillContainer (TcVnVector2_REAL)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector2_REAL&  aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Container with basic elements
aValue	TcVnVector2_REAL [▸ 1590]&	Value to set the container elements

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.18 FillContainer (TcVnVector2_SINT)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector2_SINT&  aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector2_SINT [▶ 1590]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.19 FillContainer (TcVnVector2_UINT)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector2_UINT&  aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector2_UINT [▶ 1590]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.20 FillContainer (TcVnVector2_USINT)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector2_USINT& aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector2_USINT [▶ 1590]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.21 FillContainer (TcVnVector3_INT)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT           hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnVector3_INT& aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector3_INT [▶ 1590]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.22 FillContainer (TcVnVector3_REAL)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector3_REAL&  aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector3_REAL [▶ 1590]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.23 FillContainer (TcVnVector3_SINT)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector3_SINT&  aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Container with basic elements
aValue	TcVnVector3_SINT [▸ 1590]&	Value to set the container elements

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.24 FillContainer (TcVnVector3_UINT)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector3_UINT&  aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector3_UINT [▶ 1590]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.25 FillContainer (TcVnVector3_USINT)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector3_USINT&  aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector3_USINT [▶ 1590]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.26 FillContainer (TcVnVector4_DINT)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector4_DINT& aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Container with basic elements
aValue	TcVnVector4_DINT [▸ 1590]&	Value to set the container elements

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.27 FillContainer (TcVnVector4_INT)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnVector4_INT& aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector4_INT [▶ 1590]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.28 FillContainer (TcVnVector4_LREAL)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector4_LREAL& aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector4_LREAL [▶ 1590]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.29 FillContainer (TcVnVector4_SINT)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector4_SINT&   aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector4_SINT [▶ 1590]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.30 FillContainer (TcVnVector4_UINT)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector4_UINT&   aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector4_UINT [▶ 1590]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.31 FillContainer (TcVnVector4_USINT)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector4_USINT&  aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector4_USINT [▶ 1590]&	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.32 FillContainer (ULONG)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    ULONG            nValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
nValue	ULONG	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.33 FillContainer (USHORT)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    USHORT           nValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
nValue	USHORT	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.34 FillContainer (ULONGLONG)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    ULONGLONG       nValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
nValue	ULONGLONG	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.2.35 FillContainer (unsigned char)

Fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    unsigned char   nValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
nValue	unsigned char	Value to set the container elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3 FillContainerExp

本章包含专家函数，根据数据类型，可用于用特定的值填充容器，即使只是部分填充。这个函数适用于包括基本元素的容器。

6.2.3.4.3.1 FillContainer (LONG)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
```

```
    LONG            nValue,
    ULONGLONG       nStartIdx,
    ULONGLONG       nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
nValue	LONG	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.2 FillContainer (SHORT)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    SHORT           nValue,
    ULONGLONG       nStartIdx,
    ULONGLONG       nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
nValue	SHORT	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.3 FillContainer (ITcVnImage*)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    ITcVnImage*      ipValue,
    ULONGLONG        nStartIdx,
    ULONGLONG        nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
ipValue	ITcVnImage* [▶ 1797]	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.4 FillContainer (double)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    double           fValue,
    ULONGLONG        nStartIdx,
    ULONGLONG        nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
fValue	double	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.5 FillContainer (float)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    float            fValue,
    ULONGLONG        nStartIdx,
    ULONGLONG        nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
fValue	float	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.6 FillContainer (char)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    char             nValue,
    ULONGLONG        nStartIdx,
    ULONGLONG        nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
nValue	char	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.7 FillContainer (TcVnDMatch)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnDMatch&      stValue,
    ULONGLONG        nStartIdx,
    ULONGLONG        nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
stValue	TcVnDMatch [▶ 1643]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.8 FillContainer (TcVnKeyPoint)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnKeyPoint&     stValue,
    ULONGLONG         nStartIdx,
    ULONGLONG         nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
stValue	TcVnKeyPoint [▶ 1643]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.9 FillContainer (TcVnPoint2_DINT)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnPoint2_DINT&  aValue,
    ULONGLONG         nStartIdx,
    ULONGLONG         nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnPoint2_DINT [▶ 1588]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.10 FillContainer (TcVnPoint2_LREAL)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnPoint2_LREAL&   aValue,
    ULONGLONG           nStartIdx,
    ULONGLONG           nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnPoint2_LREAL [▶ 1588]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.11 FillContainer (TcVnPoint2_REAL)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnPoint2_REAL&  aValue,
    ULONGLONG         nStartIdx,
    ULONGLONG         nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnPoint2_REAL [▶ 1588]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.12 FillContainer (TcVnPoint3_LREAL)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnPoint3_LREAL& aValue,
    ULONGLONG         nStartIdx,
    ULONGLONG         nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Container with basic elements
aValue	TcVnPoint3_LREAL [▸ 1588]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.13 FillContainer (TcVnPoint3_REAL)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnPoint3_REAL& aValue,
    ULONGLONG        nStartIdx,
    ULONGLONG        nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnPoint3_REAL [▶ 1588]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.14 FillContainer (TcVnRectangle_DINT)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnRectangle_DINT& stValue,
    ULONGLONG           nStartIdx,
    ULONGLONG           nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
stValue	TcVnRectangle_DINT [▶ 1657]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.15 FillContainer (TcVnVector2_DINT)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector2_DINT& aValue,
    ULONGLONG         nStartIdx,
    ULONGLONG         nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector2_DINT [▶ 1590]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.16 FillContainer (TcVnVector2_INT)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector2_INT&  aValue,
    ULONGLONG         nStartIdx,
    ULONGLONG         nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector2_INT [▶ 1590]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.17 FillContainer (TcVnVector2_REAL)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector2_REAL&  aValue,
    ULONGLONG          nStartIdx,
    ULONGLONG          nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector2_REAL [▶ 1590]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.18 FillContainer (TcVnVector2_SINT)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector2_SINT&   aValue,
    ULONGLONG           nStartIdx,
    ULONGLONG           nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector2_SINT [▶ 1590]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.19 FillContainer (TcVnVector2_UINT)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector2_UINT& aValue,
    ULONGLONG         nStartIdx,
    ULONGLONG         nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector2_UINT [▶ 1590]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.20 FillContainer (TcVnVector2_USINT)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector2_USINT&  aValue,
    ULONGLONG           nStartIdx,
    ULONGLONG           nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector2_USINT [▶ 1590]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.21 FillContainer (TcVnVector3_INT)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector3_INT&  aValue,
    ULONGLONG         nStartIdx,
    ULONGLONG         nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector3_INT [▶ 1590]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.22 FillContainer (TcVnVector3_REAL)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector3_REAL&   aValue,
    ULONGLONG           nStartIdx,
    ULONGLONG           nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector3_REAL [▶ 1590]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.23 FillContainer (TcVnVector3_SINT)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector3_SINT& aValue,
    ULONGLONG          nStartIdx,
    ULONGLONG          nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector3_SINT [▶ 1590]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.24 FillContainer (TcVnVector3_UINT)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT              hrPrev,
    ITcVnContainer*      ipContainer,
    TcVnVector3_UINT&    aValue,
    ULONGLONG            nStartIdx,
    ULONGLONG            nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector3_UINT [▶ 1590]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.25 FillContainer (TcVnVector3_USINT)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT              hrPrev,
    ITcVnContainer*      ipContainer,
    TcVnVector3_USINT&   aValue,
    ULONGLONG            nStartIdx,
    ULONGLONG            nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector3_USINT [▶ 1590]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.26 FillContainer (TcVnVector4_DINT)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector4_DINT& aValue,
    ULONGLONG         nStartIdx,
    ULONGLONG         nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector4_DINT [▶ 1590]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.27 FillContainer (TcVnVector4_INT)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnVector4_INT& aValue,
    ULONGLONG        nStartIdx,
    ULONGLONG        nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Container with basic elements
aValue	TcVnVector4_INT [▸ 1590]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.28 FillContainer (TcVnVector4_LREAL)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector4_LREAL&  aValue,
    ULONGLONG           nStartIdx,
    ULONGLONG           nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector4_LREAL [▶ 1590]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.29 FillContainer (TcVnVector4_SINT)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector4_SINT&  aValue,
    ULONGLONG          nStartIdx,
    ULONGLONG          nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector4_SINT [▶ 1590]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.30 FillContainer (TcVnVector4_UINT)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector4_UINT&   aValue,
    ULONGLONG           nStartIdx,
    ULONGLONG           nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector4_UINT [▶ 1590]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.31 FillContainer (TcVnVector4_USINT)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector4_USINT& aValue,
    ULONGLONG         nStartIdx,
    ULONGLONG         nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
aValue	TcVnVector4_USINT [▶ 1590]&	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.32 FillContainer (ULONG)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    ULONG           nValue,
    ULONGLONG       nStartIdx,
    ULONGLONG       nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
nValue	ULONG	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.33 FillContainer (USHORT)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    USHORT          nValue,
    ULONGLONG       nStartIdx,
    ULONGLONG       nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
nValue	USHORT	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.34 FillContainer (ULONGLONG)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nValue,
    ULONGLONG        nStartIdx,
    ULONGLONG        nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
nValue	ULONGLONG	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.3.35 FillContainer (unsigned char)

Partially fill the container with the specified value. Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT FillContainer(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    unsigned char    nValue,
    ULONGLONG        nStartIdx,
    ULONGLONG        nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
nValue	unsigned char	Value to set the container elements
nStartIdx	ULONGLONG	Start index
nCount	ULONGLONG	Number of elements to set

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4 GetAt

如果需要访问一个容器中的单个元素，应该使用F_VN_GetAt函数。如果需要连续处理容器中的所有元素，使用迭代器 [▶ 2073]更好。

6.2.3.4.4.1 GetAt (LONG)

Gets the DINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    LONG&            nElement,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with DINT elements
nElement	LONG&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.2 GetAt (SHORT)

Gets the INT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    SHORT&           nElement,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with INT elements
nElement	SHORT&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.3 GetAt (ITcVnContainer*)

Gets the ITcVnContainer element at the specified index of the source container. (Alternatively use interface method .GetAt.)

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT           hrPrev,
    ITcVnContainer*   ipSrcContainer,
    ITcVnContainer*&  ipDestContainer,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▶ 1760]	Container with ITcVnContainer elements
ipDestContainer	ITcVnContainer* [▶ 1760]&	Returns the container at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.4 GetAt (ITcVnImage*)

Gets the ITcVnImage element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT            hrPrev,
    ITcVnContainer*  ipContainer,
    ITcVnImage*&     ipElement,
    ULONGLONG          nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with ITcVnImage elements
ipElement	ITcVnImage* [▶ 1797]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.5 GetAt (double)

Gets the LREAL element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    double&          fElement,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with LREAL elements
fElement	double&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.6 GetAt (float)

Gets the REAL element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    float&           fElement,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with REAL elements
fElement	float&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.7 GetAt (char)

Gets the SINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT           hrPrev,
    ITcVnContainer* ipContainer,
    char&             nElement,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with SINT elements
nElement	char&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.8 GetAt (TcVnDMatch)

Gets the TcVnDMatch element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT           hrPrev,
    ITcVnContainer* ipContainer,
    TcVnDMatch&       stElement,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnDMatch elements
stElement	TcVnDMatch [▶ 1643]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.9 GetAt (TcVnKeyPoint)

Gets the TcVnKeyPoint element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT           hrPrev,
    ITcVnContainer* ipContainer,
    TcVnKeyPoint&     stElement,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnKeyPoint elements
stElement	TcVnKeyPoint [▶ 1643]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.10 GetAt (TcVnPoint2_DINT)

Gets the TcVnPoint2_DINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnPoint2_DINT& aElement,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnPoint2_DINT elements
aElement	TcVnPoint2_DINT [▶ 1588]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.11 GetAt (TcVnPoint2_LREAL)

Gets the TcVnPoint2_LREAL element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnPoint2_LREAL&  aElement,
    ULONGLONG          nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnPoint2_LREAL elements
aElement	TcVnPoint2_LREAL [▶ 1588]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.12 GetAt (TcVnPoint2_REAL)

Gets the TcVnPoint2_REAL element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnPoint2_REAL& aElement,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnPoint2_REAL elements
aElement	TcVnPoint2_REAL [▶ 1588]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.13 GetAt (TcVnPoint3_LREAL)

Gets the TcVnPoint3_LREAL element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnPoint3_LREAL& aElement,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnPoint3_LREAL elements
aElement	TcVnPoint3_LREAL [▶ 1588]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.14 GetAt (TcVnPoint3_REAL)

Gets the TcVnPoint3_REAL element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnPoint3_REAL& aElement,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnPoint3_REAL elements
aElement	TcVnPoint3_REAL [▶ 1588]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.15 GetAt (TcVnRectangle_DINT)

Gets the TcVnRectangle_DINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT              hrPrev,
    ITcVnContainer*      ipContainer,
    TcVnRectangle_DINT&  stElement,
    ULONGLONG            nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnRectangle_DINT elements
stElement	TcVnRectangle_DINT [▶ 1657]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.16 GetAt (TcVnVector2_DINT)

Gets the TcVnVector2_DINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector2_DINT&  aElement,
    ULONGLONG          nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector2_DINT elements
aElement	TcVnVector2_DINT [▶ 1590]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.17 GetAt (TcVnVector2_INT)

Gets the TcVnVector2_INT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector2_INT&  aElement,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector2_INT elements
aElement	TcVnVector2_INT [▶ 1590]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.18 GetAt (TcVnVector2_REAL)

Gets the TcVnVector2_REAL element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector2_REAL&   aElement,
    ULONGLONG           nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector2_REAL elements
aElement	TcVnVector2_REAL [▶ 1590]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.19 GetAt (TcVnVector2_SINT)

Gets the TcVnVector2_SINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector2_SINT&  aElement,
    ULONGLONG          nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector2_SINT elements
aElement	TcVnVector2_SINT [▶ 1590]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.20 GetAt (TcVnVector2_UINT)

Gets the TcVnVector2_UINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector2_UINT&  aElement,
    ULONGLONG          nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector2_UINT elements
aElement	TcVnVector2_UINT [▶ 1590]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.21 GetAt (TcVnVector2_USINT)

Gets the TcVnVector2_USINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector2_USINT&  aElement,
    ULONGLONG           nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector2_USINT elements
aElement	TcVnVector2_USINT [▶ 1590]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.22 GetAt (TcVnVector3_INT)

Gets the TcVnVector3_INT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnVector3_INT& aElement,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Container with TcVnVector3_INT elements
aElement	TcVnVector3_INT [▸ 1590]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.23 GetAt (TcVnVector3_REAL)

Gets the TcVnVector3_REAL element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector3_REAL& aElement,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector3_REAL elements
aElement	TcVnVector3_REAL [▶ 1590]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.24 GetAt (TcVnVector3_SINT)

Gets the TcVnVector3_SINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector3_SINT&  aElement,
    ULONGLONG          nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector3_SINT elements
aElement	TcVnVector3_SINT [▶ 1590]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.25 GetAt (TcVnVector3_UINT)

Gets the TcVnVector3_UINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector3_UINT&   aElement,
    ULONGLONG           nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Container with TcVnVector3_UINT elements
aElement	TcVnVector3_UINT [▸ 1590]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.26 GetAt (TcVnVector3_USINT)

Gets the TcVnVector3_USINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector3_USINT& aElement,
    ULONGLONG           nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector3_USINT elements
aElement	TcVnVector3_USINT [▶ 1590]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.27 GetAt (TcVnVector4_DINT)

Gets the TcVnVector4_DINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector4_DINT& aElement,
    ULONGLONG          nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector4_DINT elements
aElement	TcVnVector4_DINT [▶ 1590]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.28 GetAt (TcVnVector4_INT)

Gets the TcVnVector4_INT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector4_INT&  aElement,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector4_INT elements
aElement	TcVnVector4_INT [▶ 1590]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.29 GetAt (TcVnVector4_LREAL)

Gets the TcVnVector4_LREAL element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT              hrPrev,
    ITcVnContainer*      ipContainer,
    TcVnVector4_LREAL&   aElement,
    ULONGLONG            nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector4_LREAL elements
aElement	TcVnVector4_LREAL [▶ 1590]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.30 GetAt (TcVnVector4_SINT)

Gets the TcVnVector4_SINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector4_SINT&   aElement,
    ULONGLONG           nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector4_SINT elements
aElement	TcVnVector4_SINT [▶ 1590]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.31 GetAt (TcVnVector4_UINT)

Gets the TcVnVector4_UINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector4_UINT& aElement,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector4_UINT elements
aElement	TcVnVector4_UINT [▶ 1590]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.32 GetAt (TcVnVector4_USINT)

Gets the TcVnVector4_USINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector4_USINT&  aElement,
    ULONGLONG           nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector4_USINT elements
aElement	TcVnVector4_USINT [▶ 1590]&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.33 GetAt (ULONG)

Gets the UDINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    ULONG&          nElement,
    ULONGLONG       nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with UDINT elements
nElement	ULONG&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.34 GetAt (USHORT)

Gets the UINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    USHORT&         nElement,
    ULONGLONG       nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with UINT elements
nElement	USHORT&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.35 GetAt (ULONGLONG)

Gets the ULINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    ULONGLONG&       nElement,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with ULINT elements
nElement	ULONGLONG&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.4.36 GetAt (unsigned char)

Gets the USINT element at the specified index of the container.

Syntax

Definition:

```
HRESULT GetAt(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    unsigned char&   nElement,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with USINT elements
nElement	unsigned char&	Returns the element at the specified index
nIndex	ULONGLONG	Index of the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5 InsertIntoContainer

本章包含根据数据类型在特定的容器位置插入一个元素的功能。

6.2.3.4.5.1 InsertIntoContainer (LONG)

Insert an element of type DINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT          hrPrev,
    LONG             nElement,
    ITcVnContainer* ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	LONG	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.2 InsertIntoContainer (SHORT)

Insert an element of type INT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT          hrPrev,
    SHORT            nElement,
    ITcVnContainer* ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	SHORT	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.3 InsertIntoContainer (ITcVnContainer*)

Insert an element (or a range of elements) into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT          hrPrev,
    ITcVnContainer* ipElement,
    ITcVnContainer* ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipElement	ITcVnContainer* [▸ 1760]	Single element to insert into ipContainer or container with several elements
ipContainer	ITcVnContainer* [▸ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.4 InsertIntoContainer (ITcVnForwardIterator*)

Insert an element (represented by an iterator) into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT               hrPrev,
    ITcVnForwardIterator* ipElement,
    ITcVnContainer*       ipContainer,
    ULONGLONG             nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipElement	ITcVnForwardIterator* [▸ 1752]	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▸ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.5 InsertIntoContainer (ITcVnImage*)

Insert an element of type ITcVnImage into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT         hrPrev,
    ITcVnImage*     ipElement,
    ITcVnContainer* ipContainer,
    ULONGLONG       nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipElement	ITcVnImage* [▶ 1797]	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.6 InsertIntoContainer (double)

Insert an element of type LREAL into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT          hrPrev,
    double           fElement,
    ITcVnContainer* ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fElement	double	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.7 InsertIntoContainer (float)

Insert an element of type REAL into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT          hrPrev,
    float            fElement,
    ITcVnContainer* ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fElement	float	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.8 InsertIntoContainer (char)

Insert an element of type SINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT         hrPrev,
    char            nElement,
    ITcVnContainer* ipContainer,
    ULONGLONG       nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	char	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.9 InsertIntoContainer (TcVnDMatch)

Insert an element of type TcVnDMatch into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT         hrPrev,
    TcVnDMatch&     stElement,
    ITcVnContainer* ipContainer,
    ULONGLONG       nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stElement	TcVnDMatch [▶ 1643]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.10 InsertIntoContainer (TcVnKeyPoint)

Insert an element of type TcVnKeyPoint into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT          hrPrev,
    TcVnKeyPoint&    stElement,
    ITcVnContainer* ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stElement	TcVnKeyPoint [▶ 1643]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.11 InsertIntoContainer (TcVnPoint2_DINT)

Insert an element of type TcVnPoint2_DINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT          hrPrev,
    TcVnPoint2_DINT& aElement,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnPoint2_DINT [▶ 1588]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.12 InsertIntoContainer (TcVnPoint2_LREAL)

Insert an element of type TcVnPoint2_LREAL into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT            hrPrev,
    TcVnPoint2_LREAL&  aElement,
    ITcVnContainer*    ipContainer,
    ULONGLONG          nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnPoint2_LREAL [▶ 1588]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.13 InsertIntoContainer (TcVnPoint2_REAL)

Insert an element of type TcVnPoint2_REAL into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT           hrPrev,
    TcVnPoint2_REAL&  aElement,
    ITcVnContainer*   ipContainer,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnPoint2_REAL [▶ 1588]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.14 InsertIntoContainer (TcVnPoint3_LREAL)

Insert an element of type TcVnPoint3_LREAL into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT           hrPrev,
    TcVnPoint3_LREAL& aElement,
    ITcVnContainer*   ipContainer,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnPoint3_LREAL [▶ 1588]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.15 InsertIntoContainer (TcVnPoint3_REAL)

Insert an element of type TcVnPoint3_REAL into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT          hrPrev,
    TcVnPoint3_REAL& aElement,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnPoint3_REAL [▶ 1588]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.16 InsertIntoContainer (TcVnRectangle_DINT)

Insert an element of type TcVnRectangle_DINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT             hrPrev,
    TcVnRectangle_DINT& stElement,
    ITcVnContainer*     ipContainer,
    ULONGLONG           nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stElement	TcVnRectangle_DINT [▶ 1657]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.17 InsertIntoContainer (TcVnVector2_DINT)

Insert an element of type TcVnVector2_DINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT            hrPrev,
    TcVnVector2_DINT& aElement,
    ITcVnContainer*   ipContainer,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector2_DINT [▸ 1590]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▸ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.18 InsertIntoContainer (TcVnVector2_INT)

Insert an element of type TcVnVector2_INT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT          hrPrev,
    TcVnVector2_INT& aElement,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector2_INT [▸ 1590]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▸ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.19 InsertIntoContainer (TcVnVector2_REAL)

Insert an element of type TcVnVector2_REAL into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT             hrPrev,
    TcVnVector2_REAL&   aElement,
    ITcVnContainer*     ipContainer,
    ULONGLONG           nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector2_REAL [▶ 1590]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.20 InsertIntoContainer (TcVnVector2_SINT)

Insert an element of type TcVnVector2_SINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT           hrPrev,
    TcVnVector2_SINT& aElement,
    ITcVnContainer*   ipContainer,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector2_SINT [▶ 1590]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.21 InsertIntoContainer (TcVnVector2_UINT)

Insert an element of type TcVnVector2_UINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT           hrPrev,
    TcVnVector2_UINT& aElement,
    ITcVnContainer*   ipContainer,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector2_UINT [▶ 1590]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.22 InsertIntoContainer (TcVnVector2_USINT)

Insert an element of type TcVnVector2_USINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT             hrPrev,
    TcVnVector2_USINT&  aElement,
    ITcVnContainer*     ipContainer,
    ULONGLONG           nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector2_USINT [▶ 1590]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.23 InsertIntoContainer (TcVnVector3_INT)

Insert an element of type TcVnVector3_INT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT          hrPrev,
    TcVnVector3_INT& aElement,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector3_INT [▶ 1590]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.24 InsertIntoContainer (TcVnVector3_REAL)

Insert an element of type TcVnVector3_REAL into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT           hrPrev,
    TcVnVector3_REAL& aElement,
    ITcVnContainer*   ipContainer,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector3_REAL [▶ 1590]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.25 InsertIntoContainer (TcVnVector3_SINT)

Insert an element of type TcVnVector3_SINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT           hrPrev,
    TcVnVector3_SINT& aElement,
    ITcVnContainer*   ipContainer,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector3_SINT [▶ 1590]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.26 InsertIntoContainer (TcVnVector3_UINT)

Insert an element of type TcVnVector3_UINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT           hrPrev,
    TcVnVector3_UINT& aElement,
    ITcVnContainer*   ipContainer,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector3_UINT [▶ 1590]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.27 InsertIntoContainer (TcVnVector3_USINT)

Insert an element of type TcVnVector3_USINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT             hrPrev,
    TcVnVector3_USINT&  aElement,
    ITcVnContainer*     ipContainer,
    ULONGLONG           nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector3_USINT [▶ 1590]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.28 InsertIntoContainer (TcVnVector4_DINT)

Insert an element of type TcVnVector4_DINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT           hrPrev,
    TcVnVector4_DINT& aElement,
    ITcVnContainer*   ipContainer,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector4_DINT [▶ 1590]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.29 InsertIntoContainer (TcVnVector4_INT)

Insert an element of type TcVnVector4_INT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT          hrPrev,
    TcVnVector4_INT& aElement,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector4_INT [▶ 1590]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.30 InsertIntoContainer (TcVnVector4_LREAL)

Insert an element of type TcVnVector4_LREAL into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT              hrPrev,
    TcVnVector4_LREAL&  aElement,
    ITcVnContainer*      ipContainer,
    ULONGLONG            nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector4_LREAL [▶ 1590]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.31 InsertIntoContainer (TcVnVector4_SINT)

Insert an element of type TcVnVector4_SINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT           hrPrev,
    TcVnVector4_SINT& aElement,
    ITcVnContainer*   ipContainer,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector4_SINT [▶ 1590]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.32 InsertIntoContainer (TcVnVector4_UINT)

Insert an element of type TcVnVector4_UINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT           hrPrev,
    TcVnVector4_UINT& aElement,
    ITcVnContainer*   ipContainer,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector4_UINT [▶ 1590]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.33 InsertIntoContainer (TcVnVector4_USINT)

Insert an element of type TcVnVector4_USINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT            hrPrev,
    TcVnVector4_USINT& aElement,
    ITcVnContainer*    ipContainer,
    ULONGLONG          nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector4_USINT [▶ 1590]&	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.34 InsertIntoContainer (ULONG)

Insert an element of type UDINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT          hrPrev,
    ULONG            nElement,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	ULONG	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.35 InsertIntoContainer (USHORT)

Insert an element of type UINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT         hrPrev,
    USHORT          nElement,
    ITcVnContainer* ipContainer,
    ULONGLONG       nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	USHORT	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▸ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.36 InsertIntoContainer (ULONGLONG)

Insert an element of type ULINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT          hrPrev,
    ULONGLONG        nElement,
    ITcVnContainer* ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	ULONGLONG	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.5.37 InsertIntoContainer (unsigned char)

Insert an element of type USINT into a container before the specified position.

Syntax

Definition:

```
HRESULT InsertIntoContainer(
    HRESULT          hrPrev,
    unsigned char    nElement,
    ITcVnContainer* ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	unsigned char	Single element to insert into ipContainer
ipContainer	ITcVnContainer* [▶ 1760]	Container in which to insert the element
nIndex	ULONGLONG	Position, before which the element will be inserted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6 SetAt

如果需要访问一个容器中的单个元素，应该使用F_VN_SetAt函数。如果需要连续处理容器中的所有元素，使用迭代器 [▶ 2073]更好。

6.2.3.4.6.1 SetAt (LONG)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT          hrPrev,
    LONG             nElement,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	LONG	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with DINT elements, in which the element at position nIndex is replaced by nElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.2 SetAt (SHORT)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT          hrPrev,
    SHORT            nElement,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	SHORT	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with INT elements, in which the element at position nIndex is replaced by nElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.3 SetAt (ITcVnContainer*)

Sets the element at the specified index of the container. (Alternatively use interface method .SetAt.)

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT          hrPrev,
    ITcVnContainer*  ipElement,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipElement	ITcVnContainer* [▶ 1760]	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with ITcVnContainer elements, in which the element at position nIndex is replaced by ipElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.4 SetAt (ITcVnImage*)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT          hrPrev,
    ITcVnImage*      ipElement,
    ITcVnContainer* ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipElement	ITcVnImage* [▶ 1797]	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with ITcVnImage elements, in which the element at position nIndex is replaced by ipElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.5 SetAt (double)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT          hrPrev,
    double           fElement,
    ITcVnContainer* ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fElement	double	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with LREAL elements, in which the element at position nIndex is replaced by fElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.6 SetAt (float)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT          hrPrev,
    float            fElement,
    ITcVnContainer* ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fElement	float	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with REAL elements, in which the element at position nIndex is replaced by fElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.7 SetAt (char)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT          hrPrev,
    char             nElement,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	char	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with SINT elements, in which the element at position nIndex is replaced by nElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.8 SetAt (TcVnDMatch)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT          hrPrev,
    TcVnDMatch&      stElement,
    ITcVnContainer* ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stElement	TcVnDMatch [▶ 1643]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnDMatch elements, in which the element at position nIndex is replaced by stElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.9 SetAt (TcVnKeyPoint)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT          hrPrev,
    TcVnKeyPoint&    stElement,
    ITcVnContainer* ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stElement	TcVnKeyPoint [▶ 1643]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnKeyPoint elements, in which the element at position nIndex is replaced by stElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.10 SetAt (TcVnPoint2_DINT)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT          hrPrev,
    TcVnPoint2_DINT& aElement,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnPoint2_DINT [▶ 1588]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnPoint2_DINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.11 SetAt (TcVnPoint2_LREAL)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT             hrPrev,
    TcVnPoint2_LREAL&   aElement,
    ITcVnContainer*     ipContainer,
    ULONGLONG           nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnPoint2_LREAL [▶ 1588]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnPoint2_LREAL elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.12 SetAt (TcVnPoint2_REAL)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT          hrPrev,
    TcVnPoint2_REAL& aElement,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnPoint2_REAL [▶ 1588]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnPoint2_REAL elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.13 SetAt (TcVnPoint3_LREAL)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT           hrPrev,
    TcVnPoint3_LREAL& aElement,
    ITcVnContainer*   ipContainer,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnPoint3_LREAL [▶ 1588]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnPoint3_LREAL elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.14 SetAt (TcVnPoint3_REAL)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT          hrPrev,
    TcVnPoint3_REAL& aElement,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnPoint3_REAL [▶ 1588]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnPoint3_REAL elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.15 SetAt (TcVnRectangle_DINT)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT             hrPrev,
    TcVnRectangle_DINT& stElement,
    ITcVnContainer*     ipContainer,
    ULONGLONG           nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stElement	TcVnRectangle_DINT [▶ 1657]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnRectangle_DINT elements, in which the element at position nIndex is replaced by stElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.16 SetAt (TcVnVector2_DINT)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT           hrPrev,
    TcVnVector2_DINT& aElement,
    ITcVnContainer*   ipContainer,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector2_DINT [▶ 1590]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector2_DINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.17 SetAt (TcVnVector2_INT)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT          hrPrev,
    TcVnVector2_INT& aElement,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector2_INT [▶ 1590]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector2_INT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.18 SetAt (TcVnVector2_REAL)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT           hrPrev,
    TcVnVector2_REAL& aElement,
    ITcVnContainer*   ipContainer,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector2_REAL [▶ 1590]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector2_REAL elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.19 SetAt (TcVnVector2_SINT)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT            hrPrev,
    TcVnVector2_SINT& aElement,
    ITcVnContainer*    ipContainer,
    ULONGLONG          nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector2_SINT [▶ 1590]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector2_SINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.20 SetAt (TcVnVector2_UINT)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT            hrPrev,
    TcVnVector2_UINT&  aElement,
    ITcVnContainer*    ipContainer,
    ULONGLONG          nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector2_UINT [▶ 1590]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector2_UINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.21 SetAt (TcVnVector2_USINT)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT             hrPrev,
    TcVnVector2_USINT&  aElement,
    ITcVnContainer*     ipContainer,
    ULONGLONG           nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector2_USINT [▶ 1590]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector2_USINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.22 SetAt (TcVnVector3_INT)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT          hrPrev,
    TcVnVector3_INT& aElement,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector3_INT [▶ 1590]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector3_INT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.23 SetAt (TcVnVector3_REAL)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT            hrPrev,
    TcVnVector3_REAL&  aElement,
    ITcVnContainer*    ipContainer,
    ULONGLONG          nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector3_REAL [▶ 1590]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector3_REAL elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.24 SetAt (TcVnVector3_SINT)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT           hrPrev,
    TcVnVector3_SINT& aElement,
    ITcVnContainer*   ipContainer,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector3_SINT [▸ 1590]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▸ 1760]	Container with TcVnVector3_SINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.25 SetAt (TcVnVector3_UINT)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT           hrPrev,
    TcVnVector3_UINT& aElement,
    ITcVnContainer*   ipContainer,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector3_UINT [▶ 1590]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector3_UINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.26 SetAt (TcVnVector3_USINT)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT            hrPrev,
    TcVnVector3_USINT& aElement,
    ITcVnContainer*    ipContainer,
    ULONGLONG          nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector3_USINT [▶ 1590]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector3_USINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.27 SetAt (TcVnVector4_DINT)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT             hrPrev,
    TcVnVector4_DINT& aElement,
    ITcVnContainer*     ipContainer,
    ULONGLONG           nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector4_DINT [▶ 1590]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector4_DINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.28 SetAt (TcVnVector4_INT)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT          hrPrev,
    TcVnVector4_INT& aElement,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector4_INT [▶ 1590]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector4_INT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.29 SetAt (TcVnVector4_LREAL)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT            hrPrev,
    TcVnVector4_LREAL& aElement,
    ITcVnContainer*    ipContainer,
    ULONGLONG          nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector4_LREAL [▶ 1590]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector4_LREAL elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.30 SetAt (TcVnVector4_SINT)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT             hrPrev,
    TcVnVector4_SINT&   aElement,
    ITcVnContainer*     ipContainer,
    ULONGLONG           nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector4_SINT [▶ 1590]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector4_SINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.31 SetAt (TcVnVector4_UINT)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT           hrPrev,
    TcVnVector4_UINT& aElement,
    ITcVnContainer*   ipContainer,
    ULONGLONG         nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector4_UINT [▶ 1590]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector4_UINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.32 SetAt (TcVnVector4_USINT)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT            hrPrev,
    TcVnVector4_USINT& aElement,
    ITcVnContainer*    ipContainer,
    ULONGLONG          nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aElement	TcVnVector4_USINT [▶ 1590]&	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with TcVnVector4_USINT elements, in which the element at position nIndex is replaced by aElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.33 SetAt (ULONG)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT         hrPrev,
    ULONG           nElement,
    ITcVnContainer* ipContainer,
    ULONGLONG       nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	ULONG	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with UDINT elements, in which the element at position nIndex is replaced by nElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.34 SetAt (USHORT)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT          hrPrev,
    USHORT           nElement,
    ITcVnContainer* ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	USHORT	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with UINT elements, in which the element at position nIndex is replaced by nElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.35 SetAt (ULONGLONG)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT          hrPrev,
    ULONGLONG        nElement,
    ITcVnContainer* ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	ULONGLONG	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with ULINT elements, in which the element at position nIndex is replaced by nElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.6.36 SetAt (unsigned char)

Sets the element at the specified index of the container.

Syntax

Definition:

```
HRESULT SetAt(
    HRESULT          hrPrev,
    unsigned char    nElement,
    ITcVnContainer* ipContainer,
    ULONGLONG        nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nElement	unsigned char	Element to set at the specified container position
ipContainer	ITcVnContainer* [▶ 1760]	Container with USINT elements, in which the element at position nIndex is replaced by nElement
nIndex	ULONGLONG	Index

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.7 AdvanceIterator

Advance an iterator by the specified offset.

Syntax

Definition:

```
HRESULT AdvanceIterator(
    HRESULT                hrPrev,
    ITcVnForwardIterator* ipIterator,
    LONGLONG               nOffset
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipIterator	ITcVnForwardIterator* [▶ 1752]	Iterator to be advanced
nOffset	LONGLONG	Offset, by which ipIterator is advanced

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.8 CheckIfEmpty

Checks if the container is empty. (Alternatively use interface method .CheckIfEmpty.)

Syntax

Definition:

```
HRESULT CheckIfEmpty(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.9 CheckIfIteratorIsAtEnd

Checks if the iterator points to the past-the-end element. (Alternatively use interface method .CheckIfEnd.)

Syntax

Definition:

```
HRESULT CheckIfIteratorIsAtEnd(
    HRESULT                hrPrev,
    ITcVnForwardIterator* ipIterator
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipIterator	ITcVnForwardIterator* [▸ 1752]	Iterator

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.10 ConvertContainerType

Converts a container to another type (Struct element types are not supported).

Syntax

Definition:

```
HRESULT ConvertContainerType(
    HRESULT          hrPrev,
    ITcVnContainer*  ipSrcContainer,
    ITcVnContainer*& ipDestContainer,
    GUID             nDestTypeGuid
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▶ 1760]	Source container
ipDestContainer	ITcVnContainer* [▶ 1760]&	Returns the converted container
nDestTypeGuid	GUID	Specifies the destination container type for the conversion

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.11 CopyContainer

Copys a container.

Syntax

Definition:

```
HRESULT CopyContainer(
    HRESULT          hrPrev,
    ITcVnContainer*  ipSrcContainer,
    ITcVnContainer*& ipDestContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▶ 1760]	Source container
ipDestContainer	ITcVnContainer* [▶ 1760]&	Destination container (same type ID as ipSrcContainer; Non-zero interface pointers are reused.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.12 CopyContainerElementsConditional (ITcVnContainer*)

Copys container elements to a new container, depending on a custom condition.

Syntax

Definition:

```
HRESULT CopyContainerElementsConditional(
    HRESULT                                    hrPrev,
    ITcVnContainer*                            ipSrcContainer,
    ITcVnContainer*&                           ipDestContainer,
    ITcVnCustomElementCondition_ITcVnContainer* ipConditionFB
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▶ 1760]	Source container
ipDestContainer	ITcVnContainer* [▶ 1760]&	Returns a container with elements, that match the condition
ipConditionFB	ITcVnCustomElementCondition_ITcVnContainer* [▶ 1661]	Custom condition

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.13 CopyContainerElementsConditional (ITcVnForwardIterator*)

Copys container elements to a new container, depending on a custom condition.

Syntax

Definition:

```
HRESULT CopyContainerElementsConditional(
    HRESULT                                           hrPrev,
    ITcVnContainer*                                   ipSrcContainer,
    ITcVnContainer*&                                  ipDestContainer,
    ITcVnCustomElementCondition_ITcVnForwardIterator* ipConditionFB
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▶ 1760]	Source container
ipDestContainer	ITcVnContainer* [▶ 1760]&	Returns a container with elements, that match the condition
ipConditionFB	ITcVnCustomElementCondition_ITcVnForwardIterator* [▶ 1662]	Custom condition

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.14 CreateContainer

Creates a container with type GUID nTypeGuid and return its container interface.

Syntax

Definition:

```
HRESULT CreateContainer(
    HRESULT          hrPrev,
    ITcVnContainer*& ipContainer,
    GUID             nTypeGuid,
    ULONGLONG        nElementNum
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]&	Returns the created container
nTypeGuid	GUID	Type GUID of the container to be created
nElementNum	ULONGLONG	Number of elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.15 CreateContainerFromArray

Creates a container with type GUID nTypeGuid, initialize it with provided data and return its container interface. Only supported for basic container types, i.e. not for containers of containers.

Syntax

Definition:

```
HRESULT CreateContainerFromArray(
    HRESULT          hrPrev,
    PVOID            pData,
    ITcVnContainer*& ipContainer,
    GUID             nTypeGuid,
    ULONGLONG        nElementNum
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
pData	PVOID	Pointer to the data array. Make sure that the array contains at least nElementNum elements and that the array element type matches the container element type.
ipContainer	ITcVnContainer* [▶ 1760]&	Returns the created container
nTypeGuid	GUID	Type GUID of the container to be created
nElementNum	ULONGLONG	Number of elements to copy from pData

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.16 EraseFromContainer

Erase elements from a container.

Syntax

Definition:

```
HRESULT EraseFromContainer(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nStartIdx,
    ULONGLONG        nCount
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container from which to erase elements
nStartIdx	ULONGLONG	Index of the first element to erase
nCount	ULONGLONG	Number of elements to erase

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.17 ExportContainer

Export the container elements into a buffer (e.g. an array). Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT ExportContainer(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    PVOID            pBuffer,
    ULONGLONG        nBufferSize
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
pBuffer	PVOID	Buffer to store the container elements (Make sure to allocate enough memory! The required size in bytes can be determined using the function F_VN_ExportContainerSize)
nBufferSize	ULONGLONG	Size of the buffer memory in bytes

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.18 ExportContainer_String

Export the container elements into a string. Only possible for containers of type ContainerType_String_SINT.

Syntax

Definition:

```
HRESULT ExportContainer_String(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    PCHAR            sText,
    ULONGLONG        nMaxLength
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container of type ContainerType_String_SINT
sText	PCHAR	Make sure to choose a sufficient STRING size! The required size can be determined using the function VnExportContainerSize.
nMaxLength	ULONGLONG	Maximum string length to export (including 0 termination). If the container content is longer, the string is cut off at nMaxLength - 1 and 0 termination is appended. In this case, S_FALSE is returned.

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.19 ExportContainerSize

Determine the required buffer size in bytes to store all container elements (number_of_Elements * size_per_Element). Only possible for containers with basic elements.

Syntax

Definition:

```
HRESULT ExportContainerSize(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    ULONGLONG&      nBufferSize
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
nBufferSize	ULONGLONG&	Output parameter containing the required buffer size

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.20 ExportSubContainer

Export the container elements of a sub-container into a buffer (e.g. an array). Only possible for 2-dimensional containers with basic elements.

Syntax

Definition:

```
HRESULT ExportSubContainer(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nIndex,
    PVOID            pBuffer,
    ULONGLONG        nBufferSize
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
nIndex	ULONGLONG	Index of the requested element
pBuffer	PVOID	Buffer to store the container elements (Make sure to allocate enough memory! The required size in bytes can be determined using the function F_VN_ExportSubContainerSize)
nBufferSize	ULONGLONG	Size of the buffer memory in bytes

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.21 ExportSubContainer_String

Export the container elements of a sub-container into a string. Only possible for 2-dimensional containers of type ContainerType_Vector_String_SINT.

Syntax

Definition:

```
HRESULT ExportSubContainer_String(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    ULONGLONG       nIndex,
    PCHAR           sText,
    ULONGLONG       nMaxLength
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container of type ContainerType_Vector_String_SINT
nIndex	ULONGLONG	Index of the requested element
sText	PCHAR	Make sure to choose a sufficient STRING size! The required size can be determined using the function VnExportSubContainerSize.
nMaxLength	ULONGLONG	Maximum string length to export (including 0 termination). If the container content is longer, the string is cut off at nMaxLength - 1 and 0 termination is appended. In this case, S_FALSE is returned.

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.22 ExportSubContainerSize

Determine the required buffer size in bytes to store all container elements of a sub-container (number_of_Elements * size_per_Element). Only possible for 2-dimensional containers with basic elements.

Syntax

Definition:

```
HRESULT ExportSubContainerSize(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    ULONGLONG        nIndex,
    ULONGLONG&       nBufferSize
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container with basic elements
nIndex	ULONGLONG	Index of the requested element
nBufferSize	ULONGLONG&	Output parameter containing the required buffer size

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.23 ExtractContainerRange

Copy the specified range of the source container into the destination container. (If the destination container already contains elements, they will be removed.)

Syntax

Definition:

```
HRESULT ExtractContainerRange(
    HRESULT           hrPrev,
    ITcVnContainer*   ipSrcContainer,
    ITcVnContainer*&  ipDestContainer,
    ULONGLONG         nFirstIdx,
    ULONGLONG         nLastIdx
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▶ 1760]	Source container
ipDestContainer	ITcVnContainer* [▶ 1760]&	Destination container
nFirstIdx	ULONGLONG	Index of the first element to copy
nLastIdx	ULONGLONG	Index of the last element to copy

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.24 GetContainer

Gets a pointer to the current element converted into an ITcVnContainer interface and increment its reference counter (only possible for container types). (Alternatively use interface method .GetContainer.)

Syntax

Definition:

```
HRESULT GetContainer(
    HRESULT                hrPrev,
    ITcVnForwardIterator* ipIterator,
    ITcVnContainer*&       ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipIterator	ITcVnForwardIterator* [▶ 1752]	Iterator
ipContainer	ITcVnContainer* [▶ 1760]&	Returns the container interface

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.25 GetContainer (Exp)

Gets a pointer to the current element converted into an ITcVnContainer interface and increment its reference counter (only possible for container types). Additionally, an offset to increment or decrement the iterator afterwards can be provided.

Syntax

Definition:

```
HRESULT GetContainer(
    HRESULT                 hrPrev,
    ITcVnForwardIterator*  ipIterator,
    ITcVnContainer*&       ipContainer,
    LONG                    nOffset
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipIterator	ITcVnForwardIterator* [▸ 1752]	Iterator
ipContainer	ITcVnContainer* [▸ 1760]&	Returns the container interface
nOffset	LONG	Offset to increment (>0) or decrement (<0) the iterator afterwards

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.26 GetForwardIterator

Gets a forward iterator for the container. (Alternatively use interface method .GetForwardIterator.)

Syntax

Definition:

```
HRESULT GetForwardIterator(
    HRESULT                hrPrev,
    ITcVnContainer*        ipContainer,
    ITcVnForwardIterator*& ipIterator
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Container
ipIterator	ITcVnForwardIterator* [▸ 1752]&	Returns the iterator interface

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.27 GetNumberOfElements

Gets the number of elements in the container. (Alternatively use interface method .getElementNum.)

Syntax

Definition:

```
HRESULT GetNumberOfElements(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    ULONGLONG&      nNumberOfElements
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Container
nNumberOfElements	ULONGLONG&	Returns the number of elements in ipContainer

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.28 GetRandomAccessIterator

Gets a random access iterator for the container. (Alternatively use interface method .GetRandomAccessIterator.)

Syntax

Definition:

```
HRESULT GetRandomAccessIterator(
    HRESULT                    hrPrev,
    ITcVnContainer*            ipContainer,
    ITcVnRandomAccessIterator*& ipIterator
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container
ipIterator	ITcVnRandomAccessIterator* [▶ 1757]&	Returns the iterator interface

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.29 IncrementIterator

Increment the iterator. (Alternatively use interface method .Increment.)

Syntax

Definition:

```
HRESULT IncrementIterator(
    HRESULT                hrPrev,
    ITcVnForwardIterator*  ipIterator
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipIterator	ITcVnForwardIterator* [▸ 1752]	Iterator

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.30 IteratorDistance

Computes the distance between 2 iterators.

Syntax

Definition:

```
HRESULT IteratorDistance(
    HRESULT                hrPrev,
    ITcVnForwardIterator*  ipIteratorFirst,
    ITcVnForwardIterator*  ipIteratorLast,
    LONGLONG&              nDistance
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipIteratorFirst	ITcVnForwardIterator* [▸ 1752]	First iterator
ipIteratorLast	ITcVnForwardIterator* [▸ 1752]	Last iterator
nDistance	LONGLONG&	Returns the computed distance

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.31 ReserveContainerMemory

Reserve container memory (call with maximum required number of elements before manually appending elements for better performance)

Syntax

Definition:

```
HRESULT ReserveContainerMemory(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    ULONGLONG        nElements
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container for which to reserve the memory
nElements	ULONGLONG	Number of elements for which the container should reserve memory

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.32 ReverseContainer

Reverse the container elements.

Syntax

Definition:

```
HRESULT ReverseContainer(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Container to be reversed

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.33 SetContainer

Sets the current element using an ITcVnContainer interface (only possible for container types). (Alternatively use interface method .SetContainer.)

Syntax

Definition:

```
HRESULT SetContainer(
    HRESULT                hrPrev,
    ITcVnForwardIterator*  ipIterator,
    ITcVnContainer*        ipContainer
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipIterator	ITcVnForwardIterator* [▶ 1752]	Iterator
ipContainer	ITcVnContainer* [▶ 1760]	Container interface of which the content is to be assigned to the current element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.4.34 SetIteratorToBegin

Sets the iterator to the first element of the container. (Alternatively use interface method .SetToBegin.)

Syntax

Definition:

```
HRESULT SetIteratorToBegin(
    HRESULT               hrPrev,
    ITcVnForwardIterator* ipIterator
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipIterator	ITcVnForwardIterator* [▶ 1752]	Iterator

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5 Basic Image Operations

该组包含处理图像 [▶ 130]的函数。

函数

图像创建

- ConvertElementType [▶ 2081]
- CreateAssociatedImage [▶ 2085]
- CreateEmptyImage [▶ 2086]
- CreateImage [▶ 2087]
- CreateImageAndSetPixels [▶ 2087]
- CreateImageFromArray [▶ 2088]

图像信息

- GetImageHeight [▶ 2095]

复制图像

像素操作

感兴趣区域

将图像内容导出到内存区域

融合图像

处理图像通道

显示图像

6.2.3.5.1 CombineImageChannels

Combines several single-channel images into one multi-channel image.

Syntax

Definition:

```
HRESULT CombineImageChannels(
    HRESULT       hrPrev,
    PVOID         pSrcImages,
    ITcVnImage*&  ipDestImage,
    USHORT        nSrcArraySize
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
pSrcImages	PVOID	Pointer to an array of single-channel source images
ipDestImage	ITcVnImage* [▶ 1797]&	Multi-channel destination image (An appropriate destination image will be created if required.)
nSrcArraySize	USHORT	pSrcImages array size

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.2 ConvertElementType

Converts an image to another element type.

Syntax

Definition:

```
HRESULT ConvertElementType(
    HRESULT          hrPrev,
    ITcVnImage*      ipSrcImage,
    ITcVnImage*&     ipDestImage,
    ETcVnElementType eElementType,
    double           fScaleFactor = 1,
    double           fDelta = 0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (An appropriate destination image will be created if required.)
eElementType	ETcVnElementType [▶ 1615]		Desired element type of the destination image
fScaleFactor	double	1	Scale factor for the pixel values
fDelta	double	0	Value that is added to the scaled pixel values

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.3 CopyImage

Creates a deep copy of an image.

Syntax

Definition:

```
HRESULT CopyImage(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.4 CopyImageRegion

Deep copy the specified region of interest into a new image.

Syntax

Definition:

```
HRESULT CopyImageRegion(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage,
    ULONG         nX,
    ULONG         nY,
    ULONG         nWidth,
    ULONG         nHeight,
    ITcVnImage*&  ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
nX	ULONG	Left boundary (inclusive 0-based index)
nY	ULONG	Upper boundary (inclusive 0-based index)
nWidth	ULONG	ROI width
nHeight	ULONG	ROI height
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.5 CopyImageRegionToRegion

Copy an image region into another image region.

Syntax

Definition:

```
HRESULT CopyImageRegionToRegion(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage,
    ULONG        nXSrc,
    ULONG        nYSrc,
    ULONG        nWidth,
    ULONG        nHeight,
    ITcVnImage*  ipDestImage,
    ULONG        nXDest,
    ULONG        nYDest
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]	Source image
nXSrc	ULONG	Left boundary in ipSrcImage (inclusive 0-based index)
nYSrc	ULONG	Upper boundary in ipSrcImage (inclusive 0-based index)
nWidth	ULONG	ROI width
nHeight	ULONG	ROI height
ipDestImage	ITcVnImage* [▸ 1797]	Destination image (same type as ipSrcImage)
nXDest	ULONG	Left boundary in ipDestImage (inclusive 0-based index)
nYDest	ULONG	Upper boundary in ipDestImage (inclusive 0-based index)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.6 CopyIntoDisplayableImage

Copys an image into a displayable image. If you do not want to use ipSrcImage after this function call, you might want to use F_VN_TransformIntoDisplayableImage instead for better performance.

Syntax

Definition:

```
HRESULT CopyIntoDisplayableImage(
    HRESULT                 hrPrev,
    ITcVnImage*             ipSrcImage,
    ITcVnDisplayableImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnDisplayableImage* [▶ 1796]&	Returns the displayable image

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.7 CreateAssociatedImage

Creates a new image that shares its data with the source image. E.g. useful to work on different (disjoint) ROIs in parallel.

Syntax

Definition:

```
HRESULT CreateAssociatedImage(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Returns the created image (Non-zero interface pointers are reused.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.8 CreateEmptyImage

Creates an empty image without allocating any data buffer. The initial reference count is set to one if a new image interface is created and kept, otherwise.

Syntax

Definition:

```
HRESULT CreateEmptyImage(
    HRESULT      hrPrev,
    ITcVnImage*& ipImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▶ 1797]&	Returns the created image

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.9 CreateImage

Creates an image and allocate an appropriate data buffer. The initial reference count is set to one if a new image interface is created and kept, otherwise.

Syntax

Definition:

```
HRESULT CreateImage(
    HRESULT          hrPrev,
    ITcVnImage*&     ipImage,
    ULONG            nWidth,
    ULONG            nHeight,
    ETcVnElementType ePixelType,
    ULONG            nChannelNum
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▶ 1797]&	Returns the created image (Non-zero interface pointers are reused.)
nWidth	ULONG	Image width
nHeight	ULONG	Image height
ePixelType	ETcVnElementType [▶ 1615]	Pixel type
nChannelNum	ULONG	Number of channels (1 to 255)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.10 CreateImageAndSetPixels

Creates an image, allocates an appropriate data buffer and sets all pixels to the specified value. The initial reference count is set to one if a new image interface is created and kept, otherwise.

Syntax

Definition:

```
HRESULT CreateImageAndSetPixels(
    HRESULT            hrPrev,
    ITcVnImage*&       ipImage,
    ULONG              nWidth,
    ULONG              nHeight,
    ETcVnElementType   ePixelType,
    ULONG              nChannelNum,
    TcVnVector4_LREAL& aValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▶ 1797]&	Returns the created image (Non-zero interface pointers are reused.)
nWidth	ULONG	Image width
nHeight	ULONG	Image height
ePixelType	ETcVnElementType [▶ 1615]	Pixel type
nChannelNum	ULONG	Number of channels (1 to 4)
aValue	TcVnVector4_LREAL [▶ 1590]&	Pixel value

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.11 CreateImageFromArray

Creates an image and initialize it with the provided data. The initial reference count is set to one if a new image interface is created and kept, otherwise.

Syntax

Definition:

```
HRESULT CreateImageFromArray(
    HRESULT          hrPrev,
    PVOID            pData,
    ITcVnImage*&     ipImage,
    ULONG            nWidth,
    ULONG            nHeight,
    ETcVnElementType ePixelType,
    ULONG            nChannelNum
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
pData	PVOID	Pointer to the 1D data array. Make sure that the array contains at least nWidth * nHeight * nChannelNum elements and that the array element type matches ePixelType.
ipImage	ITcVnImage* [▶ 1797]&	Returns the created image (Non-zero interface pointers are reused.)
nWidth	ULONG	Image width
nHeight	ULONG	Image height
ePixelType	ETcVnElementType [▶ 1615]	Pixel type
nChannelNum	ULONG	Number of channels (1 to 255)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.12 ExportImage

Exports the image data into a given buffer (e.g. an array). F_VN_ExportImageSize should be called before to get the required buffer size.

Syntax

Definition:

```
HRESULT ExportImage(
    HRESULT      hrPrev,
    ITcVnImage*  ipImage,
    PVOID        pBuffer,
    ULONGLONG    nBufferSize
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▶ 1797]	Image
pBuffer	PVOID	Pointer to a buffer to store the image data (Make sure to allocate enough memory! The required size in bytes can be determined using the function F_VN_ExportImageSize)
nBufferSize	ULONGLONG	Size of the buffer memory in bytes

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.13 ExportImageAsBmp

Exports the image as bitmap into a given buffer (e.g. an array). F_VN_ExportImageAsBmpSizeExp should be called before to get the required buffer size.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT ExportImageAsBmp(
    HRESULT     hrPrev,
    ITcVnImage* ipImage,
    PVOID       pBuffer,
    ULONGLONG   nBufferSize,
    ULONG       nWidth = 0,
    ULONG       nHeight = 0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▶ 1797]		Image
pBuffer	PVOID		Pointer to a buffer to store the bitmap image (Make sure to allocate enough memory! The required size in bytes can be determined using the function F_VN_ExportImageAsBmpSizeExp)
nBufferSize	ULONGLONG		Size of the buffer memory in bytes
nWidth	ULONG	0	Desired width (or 0 to keep the original width)
nHeight	ULONG	0	Desired height (or 0 to keep the original height)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.14 ExportImageAsBmpSize

Get the required buffer size (in bytes) to store the image as bitmap.

Syntax

Definition:

```
HRESULT ExportImageAsBmpSize(
    HRESULT      hrPrev,
    ITcVnImage*  ipImage,
    ULONGLONG&   nBufferSize,
    ULONG        nWidth = 0,
    ULONG        nHeight = 0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▶ 1797]		Image
nBufferSize	ULONGLONG&		Returns the required buffer size in bytes
nWidth	ULONG	0	Desired width (or 0 to keep the original width)
nHeight	ULONG	0	Desired height (or 0 to keep the original height)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.15 ExportImageSize

Get the required buffer size (in bytes) to store the image data.

Syntax

Definition:

```
HRESULT ExportImageSize(
    HRESULT      hrPrev,
    ITcVnImage*  ipImage,
    ULONGLONG&   nBufferSize
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▶ 1797]	Image
nBufferSize	ULONGLONG&	Returns the required buffer size in bytes

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.16 FuseImages

Fuse 2 images vertically (intended for line scan cameras).

Syntax

Definition:

```
HRESULT FuseImages(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage1,
    ITcVnImage*  ipSrcImage2,
    ITcVnImage*& ipDestImage,
    ULONG        nFirstLine,
    ULONG        nNumLines
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage1	ITcVnImage* [▶ 1797]	Source image 1
ipSrcImage2	ITcVnImage* [▶ 1797]	Source image 2
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image
nFirstLine	ULONG	Line index in ipSrcImage1, which is the first line in ipDestImage
nNumLines	ULONG	The number of lines that should be copied to ipDestImage, starting with nFirstLine in ipSrcImage1. Once the last line of ipSrcImage1 was copied, the remaining lines are copied from ipSrcImage2, starting at line index 0.

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.17 FuseImagesArray

Fuse up to 10 images vertically (intended for line scan cameras).

Syntax

Definition:

```
HRESULT FuseImagesArray(
    HRESULT                   hrPrev,
    TcVnArray10_ITcVnImage&  aSrcImageArray,
    ULONG                     nValidImages,
    ITcVnImage*&              ipDestImage,
    ULONG                     nFirstLine,
    ULONG                     nNumLines
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aSrcImageArray	TcVnArray10_ITcVnImage [▸ 1590]&	Ten-element source image array (not all elements need to be filled, nValidImages specifies the actual amount of images)
nValidImages	ULONG	Number of valid images in aSrcImageArray
ipDestImage	ITcVnImage* [▸ 1797]&	Destination image
nFirstLine	ULONG	Line index in ipSrcImage1, which is the first line in ipDestImage
nNumLines	ULONG	The number of lines that should be copied to ipDestImage, starting with nFirstLine in aSrcImageArray[0]. Once the last line of aSrcImageArray[0] was copied, the remaining lines are copied from aSrcImageArray[1] (starting at line index 0) and so on.

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.18 GetImageChannel

Return the specified source image channel as a single-channel image.

Syntax

Definition:

```
HRESULT GetImageChannel(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage,
    USHORT       nChannelIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate image will be created if required.)
nChannelIndex	USHORT	Index of the requested source image channel

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.19 GetImageHeight

Return the height of an image. (Alternatively use interface method .GetHeight.)

Syntax

Definition:

```
HRESULT GetImageHeight(
    HRESULT     hrPrev,
    ITcVnImage* ipImage,
    ULONG&      nHeight
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▶ 1797]	Image
nHeight	ULONG&	Return the height of ipImage

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.20 GetImageInfo

Gets a struct containing all common meta infos of the image. (Alternatively use interface method .GetImageInfo.)

Syntax

Definition:

```
HRESULT GetImageInfo(
    HRESULT          hrPrev,
    ITcVnImage*      ipImage,
    TcVnImageInfo&   stImageInfo
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▶ 1797]	Image
stImageInfo	TcVnImageInfo [▶ 1643]&	Returns a struct describing the image

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.21 GetImageWidth

Return the width of an image. (Alternatively use interface method .GetWidth.)

Syntax

Definition:

```
HRESULT GetImageWidth(
    HRESULT      hrPrev,
    ITcVnImage* ipImage,
    ULONG&       nWidth
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▶ 1797]	Image
nWidth	ULONG&	Return the width of ipImage

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.22 GetPixel

Gets a specific pixel.

Syntax

Definition:

```
HRESULT GetPixel(
    HRESULT             hrPrev,
    ITcVnImage*         ipSrcImage,
    TcVnVector4_LREAL& aValue,
    ULONG               nX,
    ULONG               nY
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
aValue	TcVnVector4_LREAL [▶ 1590]&	Returns the pixel value (Unused channels are set to 0.)
nX	ULONG	x coordinate of the pixel
nY	ULONG	y coordinate of the pixel

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.23 GetPixelFormat

Gets the pixel format of an image. (Alternatively use interface method .GetPixelFormat.)

Syntax

Definition:

```
HRESULT GetPixelFormat(
    HRESULT           hrPrev,
    ITcVnImage*       ipImage,
    TcVnPixelFormat&  stPixelFormat
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▶ 1797]	Image
stPixelFormat	TcVnPixelFormat [▶ 1657]&	Returns a struct describing the pixel format

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.24 GetRoi

Gets the coordinates of the region of interest (ROI) within the image.

Syntax

Definition:

```
HRESULT GetRoi(
    HRESULT              hrPrev,
    ITcVnImage*          ipSrcImage,
    TcVnRectangle_UDINT& stRoi
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]	Source image
stRoi	TcVnRectangle_UDINT [▸ 1658]&	Returns the coordinates of the region of interest

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.25 MixImageChannels

Mix image channels by copying the specified channels of the source image into the specified channels of the destination image.

Syntax

Definition:

```
HRESULT MixImageChannels(
    HRESULT          hrPrev,
    ITcVnImage*      ipSrcImage,
    ITcVnImage*&     ipDestImage,
    ITcVnContainer* ipIndicesFromTo
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate image will be created if required.)
ipIndicesFromTo	ITcVnContainer* [▶ 1760]	Index pairs (ContainerType_Vector_TcVnVector2_DINT), specifying which source channel (TcVnVector2_DINT [0]) should be copied to which destination channel (TcVnVector2_DINT [1]).

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.26 ResetRoi

Reset the region of interest (ROI) of an image. (After this operation, the ROI is set to the entire image.)

Syntax

Definition:

```
HRESULT ResetRoi(
    HRESULT     hrPrev,
    ITcVnImage* ipImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▶ 1797]	Image

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.27 SetImageChannel

Sets the specified destination image channel to the values of the specified source image channel.

Syntax

Definition:

```
HRESULT SetImageChannel(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage,
    USHORT       nSrcChannelIndex,
    USHORT       nDestChannelIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]	Source image
ipDestImage	ITcVnImage* [▸ 1797]&	Destination image (An appropriate image will be created if required.)
nSrcChannelIndex	USHORT	Index of the source image channel
nDestChannelIndex	USHORT	Index of the destination image channel, which will be replaced by the specified source image channel

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.28 SetPixel

Sets a specific pixel.

Syntax

Definition:

```
HRESULT SetPixel(
    HRESULT             hrPrev,
    ITcVnImage*         ipSrcImage,
    TcVnVector4_LREAL&  aValue,
    ULONG               nX,
    ULONG               nY
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
aValue	TcVnVector4_LREAL [▶ 1590]&	The pixel value to set (additional channels are ignored.)
nX	ULONG	x coordinate of the pixel
nY	ULONG	y coordinate of the pixel

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.29 SetPixels

Sets all pixels of an image to a given value.

Syntax

Definition:

```
HRESULT SetPixels(
    HRESULT             hrPrev,
    ITcVnImage*         ipDestImage,
    TcVnVector4_LREAL&  aValue,
    ITcVnImage*         ipMask = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipDestImage	ITcVnImage* [▶ 1797]		Destination image
aValue	TcVnVector4_LREAL [▶ 1590]&		Pixel value
ipMask	ITcVnImage* [▶ 1797]	nullptr	Mask image

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.30 SetRoi

Sets a region of interest (ROI) within an image.

Syntax

Definition:

```
HRESULT SetRoi(
    HRESULT      hrPrev,
    ULONG        nX,
    ULONG        nY,
    ULONG        nWidth,
    ULONG        nHeight,
    ITcVnImage* ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nX	ULONG	Left boundary (inclusive 0-based index)
nY	ULONG	Upper boundary (inclusive 0-based index)
nWidth	ULONG	ROI width
nHeight	ULONG	ROI height
ipDestImage	ITcVnImage* [▶ 1797]	Destination image

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.31 SetRoi (TcVnRectangle_DINT)

Sets a region of interest (ROI) within an image.

Syntax

Definition:

```
HRESULT SetRoi(
    HRESULT             hrPrev,
    TcVnRectangle_DINT& stRoi,
    ITcVnImage*         ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stRoi	TcVnRectangle_DINT [▶ 1657]&	Region of interest
ipDestImage	ITcVnImage* [▶ 1797]	Destination image

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.32 SetRoi (TcVnRectangle_UDINT)

Sets a region of interest (ROI) within an image.

Syntax

Definition:

```
HRESULT SetRoi(
    HRESULT             hrPrev,
    TcVnRectangle_UDINT& stRoi,
    ITcVnImage*         ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stRoi	TcVnRectangle_UDINT [▶ 1658]&	Region of interest
ipDestImage	ITcVnImage* [▶ 1797]	Destination image

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.33 SplitImageChannels

Split a multi-channel image into multiple single channel images (1 for each source image channel).

Syntax

Definition:

```
HRESULT SplitImageChannels(
    HRESULT     hrPrev,
    ITcVnImage* ipSrcImage,
    PVOID       pDestImages,
    USHORT      nArraySize,
    USHORT&     nImageChannels
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image with multiple channels
pDestImages	PVOID	Pointer to an array of ITcVnImage (appropriate destination images will be created if required)
nArraySize	USHORT	Number of pDestImages array elements (array must be >= ipSourceImage channels)
nImageChannels	USHORT&	Actual source image channels

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.5.34 TransformIntoDisplayableImage

Transform an image into a displayable image. The source image will be released and zeroed while existing destination images will be released and overwritten with the function result. If bAllowDeepCopy equals false, the source image must not be used anywhere else and the transformation is always very fast. Otherwise, if bAllowDeepCopy equals true, a deep copy of the source image might be created if required, which will result in a longer execution time. If you want to use ipSrcImage after this function call, use F_VN_CopyIntoDisplayableImage instead.

Syntax

Definition:

```
HRESULT TransformIntoDisplayableImage(
    HRESULT                 hrPrev,
    ITcVnImage*&            ipSrcImage,
    ITcVnDisplayableImage*& ipDestImage,
    bool                    bAllowDeepCopy = false
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]&		Source image
ipDestImage	ITcVnDisplayableImage* [▶ 1796]&		Returns the displayable image
bAllowDeepCopy	bool	false	Allow deep image copies, if required

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.6 Code Quality

还请参阅有关此

- HRESULT [▶ 122]
- ADS 返回代码 [▶ 2753]
- GradeBarcode [▶ 2107]
- GradeDataMatrixCode [▶ 2110]
- GradeQRCode [▶ 2112]

6.2.3.6.1 GradeBarcode

Grades a 1D barcode according to ISO / IEC 15416:2016. Requires uniform illumination, good focus and the image sensor plane parallel to the code plane to achieve meaningful results.

Syntax

Definition:

```
HRESULT GradeBarcode(
    HRESULT                      hrPrev,
    ITcVnImage*                  ipSrcImage,
    ITcVnContainer*&             ipDecodedData,
    TcVnCodeGrades1D&            stCodeGrades,
    ETcVnBarcodeType             eBarcodeType,
    float&                       fAngleDeg,
    ETcVnBarcodeSearchDirection  eSearchDirection = BSD_ANY,
    ITcVnImage**                 pipResultVisualization = nullptr,
    ITcVnContainer**             pipScanLineInfo = nullptr,
    ITcVnContainer**             pipContour = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image, only containing the horizontally aligned barcode region (USINT, 1 channel, including quiet zones on left and right, excluding Text or empty space on more than 10% total height on top or bottom)
ipDecodedData	ITcVnContainer* [▶ 1760]&		Returns the decoded data (ContainerType_String_SINT)
stCodeGrades	TcVnCodeGrades1D [▶ 1641]&		Returns the code grades in the range from 0 (very bad) to 4 (very good)
eBarcodeType	ETcVnBarcodeType [▶ 1592]		Type of the barcode (supported: CODE39, CODE128, EAN8, EAN13, ITF, UPCA, UPCE)
fAngleDeg	float&		Returns the approximate clockwise rotation angle in degree
eSearchDirection	ETcVnBarcodeSearchDirection [▶ 1591]	BSD_ANY	Barcode search direction (BSD_ANY first tries horizontal, then vertical)
pipResultVisualization	ITcVnImage* [▶ 1797]*	nullptr	Returns a visualization of the scan lines (optional, set to 0 if not required). Grades are color coded: >= 3 as green, >= 2 as light blue, >= 1 as orange, < 1 as red
pipScanLineInfo	ITcVnContainer* [▶ 1760]*	nullptr	Returns additional scan line info for further evaluation, e.g. to use PlotIntensityProfile for visualization (optional, set to 0 if not required; ContainerType_Vector_TcVnVector3_REAL, containing [row index, overall grade, threshold])

Name	Type	Default	Description
pipContour	ITcVnContainer* [▶ 1760]*	nullptr	Returns the 4 approximate code corner points, including the quiet zone (optional, set to 0 if not required; ContainerType_Vector_TcVnPoint2_DINT)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Code Quality

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.6.2 GradeDataMatrixCode

Grades a Data Matrix code (ECC200) according to ISO / IEC 15415:2011. Requires uniform illumination, good focus and the image sensor plane parallel to the code plane to achieve meaningful results.

Syntax

Definition:

```
HRESULT GradeDataMatrixCode(
    HRESULT            hrPrev,
    ITcVnImage*        ipSrcImage,
    ITcVnContainer*&   ipDecodedData,
    TcVnCodeGradesDM&  stCodeGrades,
    float              fModuleWidth,
    float&             fAngleDeg,
    ITcVnImage**       pipResultVisualization = nullptr,
    ITcVnContainer**   pipMarginGrades = nullptr,
    ITcVnContainer**   pipContour = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image, only containing the Data Matrix code region (USINT, 1 channel, including quiet zone)
ipDecodedData	ITcVnContainer* [▶ 1760]&		Returns the decoded data (ContainerType_String_SINT)
stCodeGrades	TcVnCodeGradesDM [▶ 1641]&		Returns the code grades in the range from 0 (very bad) to 4 (very good)
fModuleWidth	float		Minimum module width of the code in the image (in pixels). Must be at least 3, preferably 5 - 8.
fAngleDeg	float&		Returns the clockwise rotation angle in degree
pipResultVisualization	ITcVnImage* [▶ 1797]*	nullptr	Returns a visualization of the scan grid (optional, set to 0 if not required). Module margin grades are color coded: >= 3 as green, >= 2 as light blue, >= 1 as orange, < 1 as red
pipMarginGrades	ITcVnContainer* [▶ 1760]*	nullptr	Returns the individual margin grade for each module, e.g. to create a custom visualization (optional, set to 0 if not required; ContainerType_Vector_TcVnVector3_REAL, containing [x-position, y-position, grade])
pipContour	ITcVnContainer* [▶ 1760]*	nullptr	Returns the 4 code corner points, excluding the quiet zone (optional, set to 0 if not required; ContainerType_Vector_TcVnPoint2_DINT)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Code Quality

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.6.3 GradeQRCode

Grades a QR code according to ISO / IEC 15415:2011. Requires uniform illumination, good focus and the image sensor plane parallel to the code plane to achieve meaningful results.

Syntax

Definition:

```
HRESULT GradeQRCode(
    HRESULT            hrPrev,
    ITcVnImage*        ipSrcImage,
    ITcVnContainer*&   ipDecodedData,
    TcVnCodeGradesQR&  stCodeGrades,
    float              fModuleWidth,
    float&             fAngleDeg,
    ITcVnImage**       pipResultVisualization = nullptr,
    ITcVnContainer**   pipMarginGrades = nullptr,
    ITcVnContainer**   pipContour = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]		Source image, only containing the QR code region (USINT, 1 channel, including quiet zone)
ipDecodedData	ITcVnContainer* [▸ 1760]&		Returns the decoded data (ContainerType_String_SINT)
stCodeGrades	TcVnCodeGradesQR [▸ 1642]&		Returns the code grades in the range from 0 (very bad) to 4 (very good)
fModuleWidth	float		Minimum module width of the code in the image (in pixels). Must be at least 3, preferably 5 - 8.
fAngleDeg	float&		Returns the clockwise rotation angle in degree
pipResultVisualization	ITcVnImage* [▸ 1797]*	nullptr	Returns a visualization of the scan grid (optional, set to 0 if not required). Module margin grades are color coded: >= 3 as green, >= 2 as light blue, >= 1 as orange, < 1 as red
pipMarginGrades	ITcVnContainer* [▸ 1760]*	nullptr	Returns the individual margin grade for each module, e.g. to create a custom visualization (optional, set to 0 if not required; ContainerType_Vector_TcVnVector3_REAL, containing [x-position, y-position, grade])
pipContour	ITcVnContainer* [▸ 1760]*	nullptr	Returns the 4 code corner points, excluding the quiet zone (optional, set to 0 if not required; ContainerType_Vector_TcVnPoint2_DINT)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Code Quality

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.7 Code Reading

该组包含检测和读取一维和二维码的函数。

函数

一维码

- ReadBarcode(Exp) [▸ 2116]
    - CodaBar
    - Code39
    - Code93
    - Code128
    - EAN8
    - EAN13
    - ITF
    - UPCA
    - UPCE
    - Code39Extended
- ReadPharmaCode [▸ 2119]

二维码

- ReadDataMatrixCode [▸ 2118]
- ReadQRCode [▸ 2120]

代码要求

一般来说，模块大小至少需要三个像素；建议超过三个像素，以获得更稳定的结果。图像应该清晰且具有很高的对比度。此外，建议代码方向为大约 0° 或 90°，以便使代码与图像轴同步。

HRESULT 的解释

对于读码函数，HRESULT [▸ 122] 的返回值如下使用：

代码	名称	描述
0x000	S_OK	函数已成功执行且所有预期代码已找到（对于标准函数为至少一个代码，而对于专家函数则为 nCodeNumber 个）。
0x001	S_FALSE	函数已成功执行且所有预期代码已找到（对于标准函数为至少一个代码，而对于专家函数则为 nCodeNumber 个）。
0x256	s_watchdogtimeout	函数被 Watchdog（看门狗）中止。一些读码函数可以返回部分结果（请参见单个函数描述）。
0x7xx	所有错误代码	函数未成功执行。请参见：ADS 返回代码 [▸ 2753]

因此，使用 SUCCEEDED() 进行常规检查不足以推断出找到的代码，也就无法推断出 ipDecodedData 中的现有数据。为此可使用以下查询：

```
if (hr == S_OK)
{
    // Export Code into String
    hr = ExportSubContainer_String(hr, ipDecodedCode, 0, sCodeAsString, 255);
    // Use sCodeAsString
}
else if (SUCCEEDED(hr))
{
    // Process partial results
}
else
{
    // Error handling
}
```

二维码搜索策略

对于二维码，会对输入图像的不同变体应用搜索算法，以便识别尽可能多的代码。图像反转和镜像都用于生成各种变体。决定测试哪个变体称为搜索策略。为了让您对函数运行时有更大的控制，可以对这种搜索策略进行配置。枚举 ETcVn2dCodeSearchStrategy [▶ 1590] 可用于此目的。借此，您可以为每个可能的图像变换（反转和镜像）指定是否应用，以及是否应首先检查原始图像或变换图像的编码。

函数参数 eSearchStrategy 精确定义相应读码专家函数的搜索策略。下表显示不同的组合：

表 3：反转

eSearchStrategy	原始图像	反转图像
CSS_ONLY_NOT_INVERTED	1.	–
CSS_FIRST_NOT_INVERTED	1.	2.
CSS_ONLY_INVERTED	–	1.
CSS_FIRST_INVERTED	2.	1.

表 4：镜像

eSearchStrategy	原始图像	镜像图像
CSS_ONLY_NOT_FLIPPED	1.	–
CSS_FIRST_NOT_FLIPPED	1.	2.
CSS_ONLY_FLIPPED	–	1.
CSS_FIRST_FLIPPED	2.	1.

可以为每种类型的变换定义搜索策略。不同搜索策略的搜索策略可如下链接：

```
eSearchStrategy = CSS_ONLY_FLIPPED | CSS_FIRST_INVERTED;
```

CSS_DEFAULT 选项可用于选择根据代码类型而变化的默认设置。

无法链接搜索策略

不允许为同一变换类型链接多个搜索策略。仅可以相互链接一个反转策略和一个镜像策略。

6.2.3.7.1 ReadBarcode

Detect and interpret a 1d barcode within the provided image. Can be canceled by Watchdog.

Syntax

Definition:

```
HRESULT ReadBarcode(
    HRESULT           hrPrev,
    ITcVnImage*       ipSrcImage,
    ITcVnContainer*&  ipDecodedData,
    ULONG             eBarcodeType
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (USINT elements, 1 channel or 3 channel (3 channel input is expected to be RGB and internally converted to Gray))
ipDecodedData	ITcVnContainer* [▶ 1760]&	Returns the decoded code (ContainerType_Vector_String_SINT)
eBarcodeType	ULONG	Types of barcode to search for (ETcVnBarcodeType)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Code Reading

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.7.2 ReadBarcode (Exp)

Detect and interpret a 1d barcode within the provided image. Can be canceled by Watchdog.

Syntax

Definition:

```
HRESULT ReadBarcode(
    HRESULT                      hrPrev,
    ITcVnImage*                  ipSrcImage,
    ITcVnContainer*&             ipDecodedData,
    ITcVnContainer*&             ipContours,
    ULONG                        eBarcodeType,
    LONG                         nCodeNumber,
    ETcVnBarcodeSearchDirection eSearchDirection,
    ITcVnContainer**             pipAngles = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (USINT elements, 1 channel or 3 channel (3 channel input is expected to be RGB and internally converted to Gray))
ipDecodedData	ITcVnContainer* [▶ 1760]&		Returns the decoded code (ContainerType_Vector_String_SINT)
ipContours	ITcVnContainer* [▶ 1760]&		Returns the code positions as contours (optional, set to 0 if not required; ContainerType_Vector_Vector_TcVnPoint2_DINT)
eBarcodeType	ULONG		Types of barcode to search for (ETcVnBarcodeType)
nCodeNumber	LONG		Number of 1d barcode that should be detected within the provided image (currently only 1 supported).
eSearchDirection	ETcVnBarcodeSearchDirection [▶ 1591]		Barcode search direction.
pipAngles	ITcVnContainer* [▶ 1760]*	nullptr	Returns the clockwise code rotation angles (optional, set to 0 if not required; ContainerType_Vector_REAL)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Code Reading

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.7.3 ReadDataMatrixCode

Detect and interpret a data matrix code (ECC200) within the provided image.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT ReadDataMatrixCode(
    HRESULT            hrPrev,
    ITcVnImage*        ipSrcImage,
    ITcVnContainer*&   ipDecodedData,
    ITcVnContainer**   pipContours = nullptr,
    LONG               nCodeNumber = 1,
    ULONG              eSearchStrategy = CSS_DEFAULT,
    ITcVnContainer**   pipAngles = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (USINT elements, 1 or 3 channels)
ipDecodedData	ITcVnContainer* [▶ 1760]&		Returns the decoded code (ContainerType_Vector_String_SINT)
pipContours	ITcVnContainer* [▶ 1760]*	nullptr	Returns the code positions as contours (optional, set to 0 if not required; ContainerType_Vector_Vector_TcVnPoint2_DINT)
nCodeNumber	LONG	1	Number of data matrix codes that should be detected within the provided image. (If set to -1, it tries to detect all data matrix codes.)
eSearchStrategy	ULONG	CSS_DEFAULT	Used search strategy (ETcVn2dCodeSearchStrategy)
pipAngles	ITcVnContainer* [▶ 1760]*	nullptr	Returns the clockwise code rotation angles (optional, set to 0 if not required; ContainerType_Vector_REAL)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Code Reading

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.7.4 ReadPharmaCode

Detect and interpret a pharma code within the provided image. Can be canceled by Watchdog. Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT ReadPharmaCode(
    HRESULT             hrPrev,
    ITcVnImage*         ipSrcImage,
    ITcVnContainer*&    ipDecodedData,
    ITcVnContainer**    pipContours = nullptr,
    LONG                nCodeNumber = 1,
    SHORT               nMinBarNumber = 4,
    ITcVnContainer**    pipAngles = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (1 channel (all element types) or 3 channel with elements of type TCVN_ET_USINT, TCVN_ET_UINT or TCVN_ET_REAL (3 channel input is expected to be RGB and internally converted to Gray))
ipDecodedData	ITcVnContainer* [▶ 1760]&		Returns the decoded code (ContainerType_Vector_String_SINT)
pipContours	ITcVnContainer* [▶ 1760]*	nullptr	Returns the code positions as contours (optional, set to 0 if not required; ContainerType_Vector_Vector_TcVnPoint2_DINT)
nCodeNumber	LONG	1	Number of pharma codes that should be detected within the provided image (currently only 1 supported).
nMinBarNumber	SHORT	4	Minimal number of (dark) bars that codes must have.
pipAngles	ITcVnContainer* [▶ 1760]*	nullptr	Returns the clockwise code rotation angles (optional, set to 0 if not required; ContainerType_Vector_REAL)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Code Reading

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.7.5 ReadQRCode

Detect and interpret a QR code within the provided image. Can be canceled by Watchdog.

Syntax

Definition:

```
HRESULT ReadQRCode(
    HRESULT           hrPrev,
    ITcVnImage*       ipSrcImage,
    ITcVnContainer*&  ipDecodedData,
    ITcVnContainer**  pipContours = nullptr,
    LONG              nCodeNumber = 1,
    ULONG             eSearchStrategy = CSS_DEFAULT,
    ITcVnContainer**  pipAngles = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (USINT elements, 1 channel or 3 channel (3 channel input is expected to be RGB and internally converted to Gray))
ipDecodedData	ITcVnContainer* [▶ 1760]&		Returns the decoded code (ContainerType_Vector_String_SINT)
pipContours	ITcVnContainer* [▶ 1760]*	nullptr	Returns the code positions as contours (optional, set to 0 if not required; ContainerType_Vector_Vector_TcVnPoint2_DINT)
nCodeNumber	LONG	1	Number of QR codes that should be detected within the provided image (currently only 1 supported).
eSearchStrategy	ULONG	CSS_DEFAULT	Used search strategy (ETcVn2dCodeSearchStrategy)
pipAngles	ITcVnContainer* [▶ 1760]*	nullptr	Returns the clockwise code rotation angles (optional, set to 0 if not required; ContainerType_Vector_REAL)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Code Reading

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8 Container Statistics

该组包含对容器 [▶ 132]进行统计分析的函数。

函数

使用欧几里得范数

- ContainerAverage [▶ 2184]
- ContainerAverageVariance [▶ 2187]
- MaxElement [▶ 2190]
- MedianElement [▶ 2190]
- MinElement [▶ 2191]

按元素

- ContainerAverageElementwise [▶ 2185]
- ContainerAverageVarianceElementwise [▶ 2187]
- MaxElementElementwise [▶ 2122]
- MedianElementElementwise [▶ 2143]
- MinElementElementwise [▶ 2163]

6.2.3.8.1 MaxElementElementwise

本章包含一些函数，通过这些函数可根据每个元素的数据类型来确定最大值。这个函数适用于包括基本元素的容器。

例如，对于 TcVnPoint2_DINT，最大 X 值和最大 Y 值相互独立确定，所以这些值不一定来自于同一个点。

6.2.3.8.1.1 MaxElementElementwise (LONG)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    LONG&            nMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMax	LONG&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.2 MaxElementElementwise (SHORT)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    SHORT&           nMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMax	SHORT&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.3 MaxElementElementwise (double)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    double&         fMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
fMax	double&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.4 MaxElementElementwise (float)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    float&          fMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
fMax	float&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.5 MaxElementElementwise (char)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    char&            nMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMax	char&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.6 MaxElementElementwise (TcVnPoint2_DINT)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnPoint2_DINT& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnPoint2_DINT [▶ 1588]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.7 MaxElementElementwise (TcVnPoint2_LREAL)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnPoint2_LREAL& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnPoint2_LREAL [▶ 1588]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.8 MaxElementElementwise (TcVnPoint2_REAL)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnPoint2_REAL& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Source container
aMax	TcVnPoint2_REAL [▸ 1588]&	Returns the requested element

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.9 MaxElementElementwise (TcVnPoint3_LREAL)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnPoint3_LREAL& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnPoint3_LREAL [▶ 1588]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.10 MaxElementElementwise (TcVnPoint3_REAL)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnPoint3_REAL& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnPoint3_REAL [▶ 1588]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.11 MaxElementElementwise (TcVnVector2_DINT)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector2_DINT& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnVector2_DINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.12 MaxElementElementwise (TcVnVector2_INT)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector2_INT& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnVector2_INT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.13 MaxElementElementwise (TcVnVector2_REAL)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector2_REAL& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnVector2_REAL [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.14 MaxElementElementwise (TcVnVector2_SINT)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector2_SINT&  aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnVector2_SINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.15 MaxElementElementwise (TcVnVector2_UINT)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector2_UINT&  aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnVector2_UINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.16 MaxElementElementwise (TcVnVector2_USINT)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector2_USINT& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnVector2_USINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.17 MaxElementElementwise (TcVnVector3_INT)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnVector3_INT& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnVector3_INT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.18 MaxElementElementwise (TcVnVector3_REAL)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector3_REAL& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnVector3_REAL [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.19 MaxElementElementwise (TcVnVector3_SINT)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector3_SINT&  aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnVector3_SINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.20 MaxElementElementwise (TcVnVector3_UINT)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector3_UINT& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnVector3_UINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.21 MaxElementElementwise (TcVnVector3_USINT)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector3_USINT& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnVector3_USINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.22 MaxElementElementwise (TcVnVector4_DINT)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector4_DINT& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnVector4_DINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.23 MaxElementElementwise (TcVnVector4_INT)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnVector4_INT& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnVector4_INT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.24 MaxElementElementwise (TcVnVector4_LREAL)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector4_LREAL& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnVector4_LREAL [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.25 MaxElementElementwise (TcVnVector4_SINT)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector4_SINT& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnVector4_SINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.26 MaxElementElementwise (TcVnVector4_UINT)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector4_UINT& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnVector4_UINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.27 MaxElementElementwise (TcVnVector4_USINT)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector4_USINT& aMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMax	TcVnVector4_USINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.28 MaxElementElementwise (ULONG)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    ULONG&           nMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMax	ULONG&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.29 MaxElementElementwise (USHORT)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    USHORT&          nMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMax	USHORT&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.30 MaxElementElementwise (ULONGLONG)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    ULONGLONG&       nMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMax	ULONGLONG&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.1.31 MaxElementElementwise (unsigned char)

Gets the element wise maximum container element.

Syntax

Definition:

```
HRESULT MaxElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    unsigned char&  nMax
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMax	unsigned char&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2 MedianElementElementwise

本章包含根据每个元素的数据类型确定中位数的函数。这个函数适用于包括基本元素的容器。

例如，对于 TcVnPoint2_DINT 来说，中位 X 值和中位 Y 值相互独立确定，所以这些值不一定来自于同一个点。

6.2.3.8.2.1 MedianElementElementwise (LONG)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    LONG&            nMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Source container
nMedian	LONG&	Returns the requested element

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.2 MedianElementElementwise (SHORT)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    SHORT&          nMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMedian	SHORT&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.3 MedianElementElementwise (double)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    double&         fMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
fMedian	double&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.4 MedianElementElementwise (float)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    float&          fMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
fMedian	float&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.5 MedianElementElementwise (char)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    char&           nMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMedian	char&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.6 MedianElementElementwise (TcVnPoint2_DINT)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnPoint2_DINT& aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMedian	TcVnPoint2_DINT [▶ 1588]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.7 MedianElementElementwise (TcVnPoint2_LREAL)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnPoint2_LREAL&  aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMedian	TcVnPoint2_LREAL [▶ 1588]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.8 MedianElementElementwise (TcVnPoint2_REAL)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnPoint2_REAL&  aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMedian	TcVnPoint2_REAL [▶ 1588]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.9 MedianElementElementwise (TcVnPoint3_LREAL)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnPoint3_LREAL& aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMedian	TcVnPoint3_LREAL [▶ 1588]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.10 MedianElementElementwise (TcVnPoint3_REAL)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnPoint3_REAL& aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Source container
aMedian	TcVnPoint3_REAL [▸ 1588]&	Returns the requested element

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.11 MedianElementElementwise (TcVnVector2_DINT)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector2_DINT& aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMedian	TcVnVector2_DINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.12 MedianElementElementwise (TcVnVector2_INT)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnVector2_INT& aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMedian	TcVnVector2_INT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.13 MedianElementElementwise (TcVnVector2_REAL)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector2_REAL& aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Source container
aMedian	TcVnVector2_REAL [▸ 1590]&	Returns the requested element

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.14 MedianElementElementwise (TcVnVector2_SINT)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector2_SINT& aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Source container
aMedian	TcVnVector2_SINT [▸ 1590]&	Returns the requested element

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.15 MedianElementElementwise (TcVnVector2_UINT)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector2_UINT& aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Source container
aMedian	TcVnVector2_UINT [▸ 1590]&	Returns the requested element

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.16 MedianElementElementwise (TcVnVector2_USINT)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector2_USINT& aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMedian	TcVnVector2_USINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.17 MedianElementElementwise (TcVnVector3_INT)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnVector3_INT& aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMedian	TcVnVector3_INT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.18 MedianElementElementwise (TcVnVector3_REAL)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector3_REAL& aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMedian	TcVnVector3_REAL [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.19 MedianElementElementwise (TcVnVector3_SINT)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector3_SINT& aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMedian	TcVnVector3_SINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.20 MedianElementElementwise (TcVnVector3_UINT)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector3_UINT& aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMedian	TcVnVector3_UINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.21 MedianElementElementwise (TcVnVector3_USINT)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector3_USINT& aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMedian	TcVnVector3_USINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.22 MedianElementElementwise (TcVnVector4_DINT)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector4_DINT&  aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMedian	TcVnVector4_DINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.23 MedianElementElementwise (TcVnVector4_INT)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector4_INT&  aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMedian	TcVnVector4_INT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.24 MedianElementElementwise (TcVnVector4_LREAL)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector4_LREAL& aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMedian	TcVnVector4_LREAL [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.25 MedianElementElementwise (TcVnVector4_SINT)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector4_SINT& aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Source container
aMedian	TcVnVector4_SINT [▸ 1590]&	Returns the requested element

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.26 MedianElementElementwise (TcVnVector4_UINT)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector4_UINT& aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMedian	TcVnVector4_UINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.27 MedianElementElementwise (TcVnVector4_USINT)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector4_USINT& aMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMedian	TcVnVector4_USINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.28 MedianElementElementwise (ULONG)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    ULONG&          nMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMedian	ULONG&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.29 MedianElementElementwise (USHORT)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    USHORT&         nMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMedian	USHORT&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.30 MedianElementElementwise (ULONGLONG)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    ULONGLONG&      nMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMedian	ULONGLONG&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.2.31 MedianElementElementwise (unsigned char)

Gets the element wise median container element.

Syntax

Definition:

```
HRESULT MedianElementElementwise(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    unsigned char&  nMedian
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMedian	unsigned char&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3 MinElementElementwise

本章包含根据每个元素的数据类型来确定最小值的函数。这个函数适用于包括基本元素的容器。

例如，对于 TcVnPoint2_DINT，最小 X 值和最小 Y 值相互独立确定，所以这些值不一定来自于同一个点。

6.2.3.8.3.1 MinElementElementwise (LONG)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    LONG&           nMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMin	LONG&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.2 MinElementElementwise (SHORT)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    SHORT&           nMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMin	SHORT&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.3 MinElementElementwise (double)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    double&         fMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
fMin	double&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.4 MinElementElementwise (float)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    float&          fMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
fMin	float&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.5 MinElementElementwise (char)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer* ipContainer,
    char&            nMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMin	char&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.6 MinElementElementwise (TcVnPoint2_DINT)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnPoint2_DINT& aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnPoint2_DINT [▶ 1588]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.7 MinElementElementwise (TcVnPoint2_LREAL)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnPoint2_LREAL& aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnPoint2_LREAL [▶ 1588]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.8 MinElementElementwise (TcVnPoint2_REAL)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnPoint2_REAL& aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnPoint2_REAL [▶ 1588]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.9 MinElementElementwise (TcVnPoint3_LREAL)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnPoint3_LREAL&  aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Source container
aMin	TcVnPoint3_LREAL [▸ 1588]&	Returns the requested element

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.10 MinElementElementwise (TcVnPoint3_REAL)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnPoint3_REAL&  aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnPoint3_REAL [▶ 1588]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.11 MinElementElementwise (TcVnVector2_DINT)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector2_DINT&  aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnVector2_DINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.12 MinElementElementwise (TcVnVector2_INT)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector2_INT& aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnVector2_INT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.13 MinElementElementwise (TcVnVector2_REAL)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector2_REAL& aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnVector2_REAL [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.14 MinElementElementwise (TcVnVector2_SINT)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector2_SINT& aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnVector2_SINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.15 MinElementElementwise (TcVnVector2_UINT)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector2_UINT&  aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnVector2_UINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.16 MinElementElementwise (TcVnVector2_USINT)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector2_USINT&  aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnVector2_USINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.17 MinElementElementwise (TcVnVector3_INT)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContainer,
    TcVnVector3_INT&  aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnVector3_INT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.18 MinElementElementwise (TcVnVector3_REAL)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector3_REAL&  aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnVector3_REAL [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.19 MinElementElementwise (TcVnVector3_SINT)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector3_SINT&  aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnVector3_SINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.20 MinElementElementwise (TcVnVector3_UINT)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector3_UINT&  aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnVector3_UINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.21 MinElementElementwise (TcVnVector3_USINT)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector3_USINT& aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnVector3_USINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.22 MinElementElementwise (TcVnVector4_DINT)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector4_DINT& aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Source container
aMin	TcVnVector4_DINT [▸ 1590]&	Returns the requested element

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.23 MinElementElementwise (TcVnVector4_INT)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnVector4_INT& aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▸ 1760]	Source container
aMin	TcVnVector4_INT [▸ 1590]&	Returns the requested element

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.24 MinElementElementwise (TcVnVector4_LREAL)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContainer,
    TcVnVector4_LREAL& aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnVector4_LREAL [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.25 MinElementElementwise (TcVnVector4_SINT)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector4_SINT& aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnVector4_SINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.26 MinElementElementwise (TcVnVector4_UINT)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    TcVnVector4_UINT& aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnVector4_UINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.27 MinElementElementwise (TcVnVector4_USINT)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContainer,
    TcVnVector4_USINT& aMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
aMin	TcVnVector4_USINT [▶ 1590]&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.28 MinElementElementwise (ULONG)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    ULONG&          nMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMin	ULONG&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.29 MinElementElementwise (USHORT)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContainer,
    USHORT&          nMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMin	USHORT&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.30 MinElementElementwise (ULONGLONG)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    ULONGLONG&      nMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMin	ULONGLONG&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.3.31 MinElementElementwise (unsigned char)

Gets the element wise minimum container element.

Syntax

Definition:

```
HRESULT MinElementElementwise(
    HRESULT         hrPrev,
    ITcVnContainer* ipContainer,
    unsigned char&  nMin
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer	ITcVnContainer* [▶ 1760]	Source container
nMin	unsigned char&	Returns the requested element

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.4 ContainerAverage

Calculate the average of elements in a container.

Syntax

Definition:

```
HRESULT ContainerAverage(
    HRESULT          hrPrev,
    ITcVnContainer*  ipSrcContainer,
    double&          fAverage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▶ 1760]	Source container (Arithmetic, point and vector element types are supported. For points and vectors, the euclidean norm is used.)
fAverage	double&	Returns the calculated average value

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.5 ContainerAverageElementwise (2)

Calculate the elementwise average of 2D elements in a container.

Syntax

Definition:

```
HRESULT ContainerAverageElementwise(
    HRESULT             hrPrev,
    ITcVnContainer*     ipSrcContainer,
    TcVnVector2_LREAL& aAverage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▸ 1760]	Source container with 2D points or vectors
aAverage	TcVnVector2_LREAL [▸ 1590]&	Returns the calculated average values

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.6 ContainerAverageElementwise (3)

Calculate the elementwise average of 3D elements in a container.

Syntax

Definition:

```
HRESULT ContainerAverageElementwise(
    HRESULT             hrPrev,
    ITcVnContainer*     ipSrcContainer,
    TcVnVector3_LREAL& aAverage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▶ 1760]	Source container with 3D points or vectors
aAverage	TcVnVector3_LREAL [▶ 1590]&	Returns the calculated average values

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.7 ContainerAverageElementwise (4)

Calculate the elementwise average of 4D elements in a container.

Syntax

Definition:

```
HRESULT ContainerAverageElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipSrcContainer,
    TcVnVector4_LREAL& aAverage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▶ 1760]	Source container with 4D vectors
aAverage	TcVnVector4_LREAL [▶ 1590]&	Returns the calculated average values

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.8 ContainerAverageVariance

Calculate the average and variance of elements in a container.

Syntax

Definition:

```
HRESULT ContainerAverageVariance(
    HRESULT         hrPrev,
    ITcVnContainer* ipSrcContainer,
    double&         fAverage,
    double&         fVariance
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▸ 1760]	Source container (Arithmetic, point and vector element types are supported. For points and vectors, the euclidean norm is used.)
fAverage	double&	Returns the calculated average value
fVariance	double&	Returns the calculated variance value

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.9 ContainerAverageVarianceElementwise (2)

Calculate the elementwise average and variance of 2D elements in a container.

Syntax

Definition:

```
HRESULT ContainerAverageVarianceElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipSrcContainer,
    TcVnVector2_LREAL& aAverage,
    TcVnVector2_LREAL& aVariance
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▶ 1760]	Source container with 2D points or vectors
aAverage	TcVnVector2_LREAL [▶ 1590]&	Returns the calculated average values
aVariance	TcVnVector2_LREAL [▶ 1590]&	Returns the calculated variance values

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.10 ContainerAverageVarianceElementwise (3)

Calculate the elementwise average and variance of 3D elements in a container.

Syntax

Definition:

```
HRESULT ContainerAverageVarianceElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipSrcContainer,
    TcVnVector3_LREAL& aAverage,
    TcVnVector3_LREAL& aVariance
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▶ 1760]	Source container with 3D points or vectors
aAverage	TcVnVector3_LREAL [▶ 1590]&	Returns the calculated average values
aVariance	TcVnVector3_LREAL [▶ 1590]&	Returns the calculated variance values

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.11 ContainerAverageVarianceElementwise (4)

Calculate the elementwise average and variance of 4D elements in a container.

Syntax

Definition:

```
HRESULT ContainerAverageVarianceElementwise(
    HRESULT            hrPrev,
    ITcVnContainer*    ipSrcContainer,
    TcVnVector4_LREAL& aAverage,
    TcVnVector4_LREAL& aVariance
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▶ 1760]	Source container with 4D vectors
aAverage	TcVnVector4_LREAL [▶ 1590]&	Returns the calculated average values
aVariance	TcVnVector4_LREAL [▶ 1590]&	Returns the calculated variance values

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.12 MaxElement

Gets the maximum element of a container.

Syntax

Definition:

```
HRESULT MaxElement(
    HRESULT                         hrPrev,
    ITcVnContainer*                 ipSrcContainer,
    ITcVnBidirectionalIterator*&    ipDestIterator,
    ULONGLONG&                      nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▸ 1760]	Source container (Arithmetic, point and vector element types are supported. For points and vectors, the euclidean norm is used.)
ipDestIterator	ITcVnBidirectionalIterator* [▸ 1750]&	Returns an iterator to the requested element
nIndex	ULONGLONG&	Returns the position of the requested element within the container

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.13 MedianElement

Gets the median element of a container.

Syntax

Definition:

```
HRESULT MedianElement(
    HRESULT                          hrPrev,
    ITcVnContainer*                  ipSrcContainer,
    ITcVnBidirectionalIterator*&     ipDestIterator,
    ULONGLONG&                       nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▶ 1760]	Source container (Arithmetic, point and vector element types are supported. For points and vectors, the euclidean norm is used.)
ipDestIterator	ITcVnBidirectionalIterator* [▶ 1750]&	Returns an iterator to the requested element
nIndex	ULONGLONG&	Returns the position of the requested element within the container

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.8.14 MinElement

Gets the minimum element of a container.

Syntax

Definition:

```
HRESULT MinElement(
    HRESULT                          hrPrev,
    ITcVnContainer*                  ipSrcContainer,
    ITcVnBidirectionalIterator*&     ipDestIterator,
    ULONGLONG&                       nIndex
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContainer	ITcVnContainer* [▶ 1760]	Source container (Arithmetic, point and vector element types are supported. For points and vectors, the euclidean norm is used.)
ipDestIterator	ITcVnBidirectionalIterator* [▶ 1750]&	Returns an iterator to the requested element
nIndex	ULONGLONG&	Returns the position of the requested element within the container

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9 Contour Analysis

该组包含用于分析轮廓的函数：

函数

形状近似

- ApproximatePolygon [▶ 2193]
- ConvexHullPoints [▶ 2206]
- EnclosingCircle [▶ 2210]
- EnclosingRectangle [▶ 2211]
- EnclosingTriangle [▶ 2211]
- FitEllipse [▶ 2212]
- FitLine [▶ 2213]
- UprightBoundingRectangle [▶ 2221]

点

- CheckIfPointIsInsideContour [▶ 2194]
- ContourExtremePoint [▶ 2200]
- ConvexityDefects [▶ 2207]

几何特征

- ContourArea [▶ 2195]
- ContourCenterOfMass [▶ 2196]

6.2.3.9.1 ApproximatePolygon

Approximate a contour to a simplified polygon (using the Douglas-Peucker algorithm).

Syntax

Definition:

```
HRESULT ApproximatePolygon(
    HRESULT          hrPrev,
    ITcVnContainer*  ipSrcContour,
    ITcVnContainer*& ipDestContour,
    double           fMaxDist,
    bool             bClosed
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcContour	ITcVnContainer* [▶ 1760]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
ipDestContour	ITcVnContainer* [▶ 1760]&	Returns a container which is filled with the approximated polygon points (same type ID as ipSrcContour; Non-zero interface pointers are reused.)
fMaxDist	double	Maximum distance between the original contour and its approximation
bClosed	bool	Specify, if the contour is closed (first and last points connected) or not

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.2 CheckIfPointIsInsideContour

Checks if a point is inside a contour (and optionally return the distance to it).

Syntax

Definition:

```
HRESULT CheckIfPointIsInsideContour(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContour,
    TcVnPoint2_REAL&  aPoint,
    double&           fDist,
    bool              bMeasureDistance
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContour	ITcVnContainer* [▶ 1760]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL)
aPoint	TcVnPoint2_REAL [▶ 1588]&	Point position to check
fDist	double&	Returns the result (> 0: the point is inside the contour; 0: the point is on the contour; < 0: the point is outside the contour)
bMeasureDistance	bool	If true, fDist returns the distance to the nearest contour edge. Otherwise, fDist only returns -1, 0 or +1 (recommended to set to false if the distance is not required, as this is faster).

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.3 ContourArea

Estimate the contour area using Green's formula.

Syntax

Definition:

```
HRESULT ContourArea(
    HRESULT          hrPrev,
    ITcVnContainer* ipContour,
    double&          fArea
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContour	ITcVnContainer* [▶ 1760]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
fArea	double&	Returns the estimated contour area (The actual contour area may differ depending on the shape of the contour.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.4 ContourCenterOfMass

Computes the center of mass of a contour.

Syntax

Definition:

```
HRESULT ContourCenterOfMass(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContour,
    TcVnPoint2_LREAL& aCenterOfMass
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContour	ITcVnContainer* [▶ 1760]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
aCenterOfMass	TcVnPoint2_LREAL [▶ 1588]&	Returns the center of mass of the contour

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.5 ContourCircularity

Computes the circularity of a contour (1.0: ideal circle, 0.0: straight line).

Syntax

Definition:

```
HRESULT ContourCircularity(
    HRESULT          hrPrev,
    ITcVnContainer* ipContour,
    double&          fCircularity
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContour	ITcVnContainer* [▶ 1760]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
fCircularity	double&	Returns the circularity of the contour [0..1]

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.6 ContourConvexity

Computes the convexity of a contour as the ratio between the contour's area and the area of the convex hull (1.0: convex shapes (i.e. rectangle, circle, ..., etc.), 0.0: zero area (i.e. a line or an empty contour)).

Syntax

Definition:

```
HRESULT ContourConvexity(
    HRESULT          hrPrev,
    ITcVnContainer* ipContour,
    double&          fConvexity
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContour	ITcVnContainer* [▶ 1760]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
fConvexity	double&	Returns the convexity of the contour [0..1]

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.7 ContourEccentricity

Computes the eccentricity of a contour (0.0: circular, 1.0: linear).

Syntax

Definition:

```
HRESULT ContourEccentricity(
    HRESULT         hrPrev,
    ITcVnContainer* ipContour,
    double&         fEccentricity
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContour	ITcVnContainer* [▶ 1760]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
fEccentricity	double&	Returns the eccentricity of the contour [0..1]

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.8 ContourElongation

Computes the elongation factor of a contour.

Syntax

Definition:

```
HRESULT ContourElongation(
    HRESULT         hrPrev,
    ITcVnContainer* ipContour,
    double&         fElongation
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContour	ITcVnContainer* [▶ 1760]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
fElongation	double&	Returns the elongation of the contour

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.9 ContourExtremePoint

Finds the contour extreme point.

Syntax

Definition:

```
HRESULT ContourExtremePoint(
    HRESULT                     hrPrev,
    ITcVnContainer*             ipContour,
    ETcVnExtremePointDirection eDirection,
    TcVnPoint2_LREAL&           aExtremePoint
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContour	ITcVnContainer* [▶ 1760]	Source container with 2D points (TcVnPoint2_DINT or TcVnPoint2_REAL or TcVnPoint2_LREAL)
eDirection	ETcVnExtremePointDirection [▶ 1616]	Selects the search direction for the extreme point
aExtremePoint	TcVnPoint2_LREAL [▶ 1588]&	Returns the extreme point of the contour, according to eDirection

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.10 ContourInertiaRatio

Computes the inertia ratio of a contour which reflects how a contour/shape is elongated (i.e. circle: 1.0, line: 0.0).

Syntax

Definition:

```
HRESULT ContourInertiaRatio(
    HRESULT         hrPrev,
    ITcVnContainer* ipContour,
    double&         fInertiaRatio
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContour	ITcVnContainer* [▶ 1760]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
fInertiaRatio	double&	Returns the inertia raio of the contour [0..1]

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.11 ContourMoments

Computes the spatial moments, the central moments, and the central normalized moments of a contour up to the third order.

Syntax

Definition:

```
HRESULT ContourMoments(
    HRESULT          hrPrev,
    ITcVnContainer* ipContour,
    TcVnMoments&     stMoments
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContour	ITcVnContainer* [▶ 1760]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
stMoments	TcVnMoments [▶ 1644]&	Returns a struct containing the moments

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.12 ContourOrientation

Calculate the orientation of a contour or a set of points.

Syntax

Definition:

```
HRESULT ContourOrientation(
    HRESULT                 hrPrev,
    ITcVnContainer*         ipContour,
    TcVnRotatedRectangle&   stOrientation,
    ETcVnOrientationMethod eMethod = OM_PCA
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContour	ITcVnContainer* [▶ 1760]		Source point set (ContainerType_Vector_TcVnPoint2_DINT, ContainerType_Vector_TcVnPoint2_REAL, or ContainerType_Vector_TcVnPoint2_LREAL. The elements of this container are the individual points.)
stOrientation	TcVnRotatedRectangle [▶ 1658]&		Resulting rotated rectangle, containig center of mass, lenghts of axes, and rotation angle of the contour in clockwise direction.
eMethod	ETcVnOrientationMethod [▶ 1626]	OM_PCA	Method for calculating the orientation.

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.13 ContourPerimeter

Computes the perimeter of a contour (curve length if the contour is not closed).

Syntax

Definition:

```
HRESULT ContourPerimeter(
    HRESULT         hrPrev,
    ITcVnContainer* ipContour,
    double&         fPerimeter,
    bool            bClosed
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContour	ITcVnContainer* [▶ 1760]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
fPerimeter	double&	Returns the perimeter of the contour
bClosed	bool	Specifys, if the contour is closed (first and last points connected) or not

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.14 ContourRoundness

Computes the roundness of a contour (perimeter^2 / area).

Syntax

Definition:

```
HRESULT ContourRoundness(
    HRESULT         hrPrev,
    ITcVnContainer* ipContour,
    double&         fRoundness
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContour	ITcVnContainer* [▶ 1760]	Single contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour.)
fRoundness	double&	Returns the roundness of the contour

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.15 ConvexHullPoints

Determines the convex hull of a point set.

Syntax

Definition:

```
HRESULT ConvexHullPoints(
    HRESULT          hrPrev,
    ITcVnContainer*  ipPointSet,
    ITcVnContainer*& ipConvexHull,
    bool             bClockwise = false
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipPointSet	ITcVnContainer* [▶ 1760]		Source point set (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the individual points.)
ipConvexHull	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the sorted convex hull points (same type ID as ipPointSet; Non-zero interface pointers are reused.)
bClockwise	bool	false	Selects the sorting direction of the hull points, assuming cartesian coordinates.

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.16 ConvexityDefects

Finds the convexity defects of a point set.

Syntax

Definition:

```
HRESULT ConvexityDefects(
    HRESULT          hrPrev,
    ITcVnContainer*  ipContour,
    ITcVnContainer*& ipConvexityDefects
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContour	ITcVnContainer* [▶ 1760]	Source contour (ContainerType_Vector_TcVnPoint2_DINT; The elements of this container are the individual points.)
ipConvexityDefects	ITcVnContainer* [▶ 1760]&	Returns a container which is filled with the convexity defects (ContainerType_Vector_TcVnVector4_DINT with [startIndex, endIndex, furthestPointIndex, fixedPointDistance (8 fraction bits, i.e. divide by 256 to get the distance in pixels)]; Non-zero interface pointers are reused.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.17 CustomElementWiseContainerOperation (ITcVnContainer*)

Performs a custom, element wise operation on a set of up to 3 containers.

Syntax

Definition:

```
HRESULT CustomElementWiseContainerOperation(
    HRESULT                                       hrPrev,
    ITcVnContainer*                               ipContainer1,
    ITcVnContainer*                               ipContainer2,
    ITcVnContainer*                               ipContainer3,
    ITcVnCustomContainerOperation_ITcVnContainer* ipOperationFB
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer1	ITcVnContainer* [▸ 1760]	Container 1
ipContainer2	ITcVnContainer* [▸ 1760]	Container 2
ipContainer3	ITcVnContainer* [▸ 1760]	Container 3
ipOperationFB	ITcVnCustomContainerOperation_ITcVnContainer* [▸ 1659]	Custom operation on the container elements

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.18 CustomElementWiseContainerOperation (ITcVnForwardIterator*)

Performs a custom, element wise operation on a set of up to 3 containers.

Syntax

Definition:

```
HRESULT CustomElementWiseContainerOperation(
    HRESULT                                              hrPrev,
    ITcVnContainer*                                      ipContainer1,
    ITcVnContainer*                                      ipContainer2,
    ITcVnContainer*                                      ipContainer3,
    ITcVnCustomContainerOperation_ITcVnForwardIterator* ipOperationFB
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContainer1	ITcVnContainer* [▸ 1760]	Container 1
ipContainer2	ITcVnContainer* [▸ 1760]	Container 2
ipContainer3	ITcVnContainer* [▸ 1760]	Container 3
ipOperationFB	ITcVnCustomContainerOperation_ITcVnForwardIterator* [▸ 1660]	Custom operation on the elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.19 EnclosingCircle

Searchs for a minimum area circle enclosing a set of points.

Syntax

Definition:

```
HRESULT EnclosingCircle(
    HRESULT          hrPrev,
    ITcVnContainer*  ipPointSet,
    TcVnPoint2_REAL& aCenter,
    float&           fRadius
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipPointSet	ITcVnContainer* [▶ 1760]	Source point set (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the individual points.)
aCenter	TcVnPoint2_REAL [▶ 1588]&	Returns the center of the circle
fRadius	float&	Returns the radius of the circle

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.20 EnclosingRectangle

Searches for a minimum area rectangle enclosing a set of points.

Syntax

Definition:

```
HRESULT EnclosingRectangle(
    HRESULT               hrPrev,
    ITcVnContainer*       ipPointSet,
    TcVnRotatedRectangle& stRectangle
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipPointSet	ITcVnContainer* [▶ 1760]	Source point set (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the individual points.)
stRectangle	TcVnRotatedRectangle [▶ 1658]&	Returns the determined rectangle

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.21 EnclosingTriangle

Searches for a minimum area triangle enclosing a set of points.

Syntax

Definition:

```
HRESULT EnclosingTriangle(
    HRESULT                 hrPrev,
    ITcVnContainer*         ipPointSet,
    TcVnArray3_Point2_REAL& aTriangleVertices,
    double&                 fArea
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipPointSet	ITcVnContainer* [▶ 1760]	Source point set (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the individual points.)
aTriangleVertices	TcVnArray3_Point2_REAL [▶ 1590]&	Returns the 3 triangle vertices
fArea	double&	Returns the triangle area

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.22 FitEllipse

Fit an ellipse in a point set.

Syntax

Definition:

```
HRESULT FitEllipse(
    HRESULT               hrPrev,
    ITcVnContainer*       ipPointSet,
    TcVnRotatedRectangle& stEllipse
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipPointSet	ITcVnContainer* [▶ 1760]	Source point set (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; At least 5 reasonable points are required! The elements of this container are the individual points.)
stEllipse	TcVnRotatedRectangle [▶ 1658]&	Resulting ellipse, described by a rotated rectangle.

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.23 FitLine

Fit a line into a point set.

Syntax

Definition:

```
HRESULT FitLine(
    HRESULT             hrPrev,
    ITcVnContainer*     ipPointSet,
    TcVnVector4_LREAL&  aFitLine,
    ETcVnDistanceType   eDistanceType = DT_L2,
    double              fParam = 0,
    double              fEpsRadius = 0.001,
    double              fEpsAngle = 0.001
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipPointSet	ITcVnContainer* [▶ 1760]		Source point set (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the individual points.)
aFitLine	TcVnVector4_LREAL [▶ 1590]&		Resulting line. The first and second element describe the x and y component of a vector collinear to the line. The third and fourth element describe the x and y component of a point on the line.
eDistanceType	ETcVnDistanceType [▶ 1613]	DT_L2	Distance computation method (supported: L1, L2, L12, FAIR, WELSCH, HUBER)
fParam	double	0	Numerical parameter (c) for some eDistanceType (should be >= 0). If 0, an optimal value is chosen internally.
fEpsRadius	double	0.001	Accuracy of the radius (distance of the line from the coordinate origin, should be > 0). A smaller value means higher accuracy.
fEpsAngle	double	0.001	Accuracy of the angle (should be > 0). A smaller value means higher accuracy.

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.24 FourierDescriptors

Computes the fourier descriptors for a closed contour.

Syntax

Definition:

```
HRESULT FourierDescriptors(
    HRESULT           hrPrev,
    ITcVnContainer*   ipContour,
    ITcVnContainer*&  ipDescriptors,
    LONG              nDescriptors
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContour	ITcVnContainer* [▶ 1760]	Closed contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_LREAL. Providing the full contour is strongly recommended, i.e. use TCVN_CAM_NONE for contour detection algorithms.)
ipDescriptors	ITcVnContainer* [▶ 1760]&	Returns the fourier descriptors for ipContour (Re0, Im0, Re1, Im1, ...; ContainerType_Vector_LREAL if ipContour is of type ContainerType_Vector_TcVnPoint2_LREAL, else ContainerType_Vector_REAL.)
nDescriptors	LONG	Specifies how many descriptors should be returned (set to -1 to return all computed descriptors, i.e. at least as many as the number of contour points)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.25 MatchContours

Compare contours using the Hu moment invariants (and optionally further aspects). In case of multiple contours in each container, the best matches are found and the average dissimilarity over all matched contours is returned.

Syntax

Definition:

```
HRESULT MatchContours(
    HRESULT                            hrPrev,
    ITcVnContainer*                    ipContour1,
    ITcVnContainer*                    ipContour2,
    ETcVnContoursMatchComparisonMethod eComparisonMethod,
    double&                            fDissimilarity,
    double                             fAreaFactor = 0,
    double                             fAbsPositionFactor = 0,
    double                             fRelPositionFactor = 0,
    double                             fNumDiffFactor = 0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContour1	ITcVnContainer* [▶ 1760]		First contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour) or collection of multiple contours (ContainerType_Vector_Vector_TcVnPoint2_DINT)
ipContour2	ITcVnContainer* [▶ 1760]		Second contour (same type as ipContour1)
eComparisonMethod	ETcVnContoursMatchComparisonMethod [▶ 1611]		Method used for comparing the Hu moment invariants of the contours
fDissimilarity	double&		Returns the dissimilarity of the contours depending on the chosen comparison method
fAreaFactor	double	0	If > 0, the relative area difference between contours is scaled by this factor and added to the computed dissimilarity
fAbsPositionFactor	double	0	If > 0, the absolute position difference between contours (i.e. the coordinates of the geometric contour centers) is scaled by this factor and added to the computed dissimilarity
fRelPositionFactor	double	0	If > 0, the relative position difference between contours is scaled by this factor and added to the computed dissimilarity (only used if matching multiple contours)

Name	Type	Default	Description
fNumDiffFactor	double	0	If > 0, the difference between the number of contours in both containers is scaled by this factor and added to the computed dissimilarity (only used if matching multiple contours)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.26 MatchContours1vsN

Compare a reference contour with multiple other contours using the Hu moment invariants (and optionally further aspects). Returns a sorted list of best matches.

Syntax

Definition:

```
HRESULT MatchContours1vsN(
    HRESULT                            hrPrev,
    ITcVnContainer*                    ipRefContour,
    ITcVnContainer*                    ipContours,
    ITcVnContainer*&                   ipMatchIndexes,
    ITcVnContainer*&                   ipDissimilarities,
    double                             fDissimilarityThreshold,
    ETcVnContoursMatchComparisonMethod eComparisonMethod,
    double                             fAreaFactor = 0,
    double                             fAbsPositionFactor = 0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipRefContour	ITcVnContainer* [▶ 1760]		Reference contour (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the points of the contour)
ipContours	ITcVnContainer* [▶ 1760]		Collection of multiple contours (CTcVnContainer_Vector_Vector_TcVnPoint2_DINT)
ipMatchIndexes	ITcVnContainer* [▶ 1760]&		Returns the indexes of the best matches (CTcVnContainer_Vector_ULINT; sorted, first element is best match, i.e. has lowest dissimilarity)
ipDissimilarities	ITcVnContainer* [▶ 1760]&		Returns the computed dissimilarities of the best matches (CTcVnContainer_Vector_LREAL; Optional, set to 0 if not required; sorted corresponding to ipMatchIndexes)
fDissimilarityThreshold	double		Neglect irrelevant matches, i.e. dissimilarity > fDissimilarityThreshold
eComparisonMethod	ETcVnContoursMatchComparisonMethod [▶ 1611]		Method used for comparing the Hu moment invariants of the contours
fAreaFactor	double	0	If > 0, the relative area difference between contours is scaled by this factor and added to the computed dissimilarity
fAbsPositionFactor	double	0	If > 0, the absolute position difference between contours (i.e. the coordinates of the geometric contour centers) is scaled by this factor and added to the computed dissimilarity

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.9.27 UprightBoundingRectangle

Determines the upright bounding rectangle of a set of points.

Syntax

Definition:

```
HRESULT UprightBoundingRectangle(
    HRESULT               hrPrev,
    ITcVnContainer*       ipPointSet,
    TcVnRectangle_UDINT& stRectangle
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipPointSet	ITcVnContainer* [▶ 1760]	Source point set (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL; The elements of this container are the individual points.)
stRectangle	TcVnRectangle_UDINT [▶ 1658]&	Returns the determined rectangle

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.10 Control Functions

该组包含用于顺序控制的函数，例如用于处理 Watchdog（看门狗） [▶ 127]的函数。

函数

Watchdog（看门狗）

- StartAbsWatchdog [▶ 2222]
- StartRelWatchdog [▶ 2222]
- StopWatchdog [▶ 2223]

6.2.3.10.1 StartAbsWatchdog

Start a cooperative watchdog given an absolute stop time.

Syntax

Definition:

```
HRESULT StartAbsWatchdog(
    HRESULT                     hrPrev,
    LONG                        tStop,
    ETcWatchdogAccumulationType eWatchdogAccType = WD_ACC_TYPE_PRODUCT
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
tStop	LONG		Stop time in us
eWatchdogAccType	ETcWatchdogAccumulationType [▶ 1636]	WD_ACC_TYPE_PRODUCT	Accumalation method used for combining the results of multiple functions enclosed by this watchdog

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.10.2 StartRelWatchdog

Start a cooperative watchdog given a stop time relative to the current time.

Syntax

Definition:

```
HRESULT StartRelWatchdog(
    HRESULT                     hrPrev,
    LONG                        tStop,
    ETcWatchdogAccumulationType eWatchdogAccType = WD_ACC_TYPE_PRODUCT
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
tStop	LONG		Stop time in us
eWatchdogAccType	ETcWatchdogAccumulationType [▶ 1636]	WD_ACC_TYPE_PRODUCT	Accumalation method used for combining the results of multiple functions enclosed by this watchdog

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.10.3 StopWatchdog

Stops a watchdog and provide runtime information.

Syntax

Definition:

```
HRESULT StopWatchdog(
    HRESULT    hrStartWatchdog,
    ULONGLONG& nFunctionsMonitored,
    ULONG&     nFractionProcessed,
    LONG&      tRest
)
```

Parameters

Name	Type	Description
hrStartWatchdog	HRESULT [▶ 122]	HRESULT indicating the result of the function used to start the watchdog (If SUCCEEDED(hrStartWatchdog) equals false, no operation is executed.)
nFunctionsMonitored	ULONGLONG&	Returns the number of functions monitored
nFractionProcessed	ULONG&	Returns the fraction processed accumulated over the monitored functions in percent
tRest	LONG&	Returns the remaining computation time in us (may be negative)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11 Drawing

该组包含绘制和描述图像的函数。

处理 4 通道颜色

4 通道颜色直接接受到结果图像中。不会进行阿尔法混合或类似操作。

函数

箭头

- DrawArrow [▶ 2225]
- DrawArrow_TcVnVector4_DINT [▶ 2227]

圆圈和弧形

- DrawCircle [▶ 2227]
- DrawCircles [▶ 2228]
- DrawCircularArc [▶ 2229]
- FillCircle [▶ 2249]

组件

- DrawComponents [▶ 2230]

轮廓

- DrawContours [▶ 2231]

椭圆

关键点和匹配

线

定向

点

多边形

矩形

强度分布

文本

绘图和填充

对于许多形状，可以选择同时绘制形状轮廓和填充形状。为此提供了两个函数组（绘制……和填充……）。此外，将轮廓厚度 nThickness 设置为 -1 会填充形状。

6.2.3.11.1 DrawArrow

Draws an arrowed line.

Syntax

Definition:

```
HRESULT DrawArrow(
    HRESULT             hrPrev,
    ULONG               nX1,
    ULONG               nY1,
    ULONG               nX2,
    ULONG               nY2,
    ITcVnImage*         ipDestImage,
    TcVnVector4_LREAL&  aColor,
    LONG                nThickness,
    bool                bDoubleHead,
    ETcVnLineType       eLineType = LT_8_CONNECTED,
    ULONG               nShift = 0,
    double              fTipLenght = 0.1
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nX1	ULONG		x coordinate of the start point
nY1	ULONG		y coordinate of the start point
nX2	ULONG		x coordinate of the end point
nY2	ULONG		y coordinate of the end point
ipDestImage	ITcVnImage* [▶ 1797]		Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&		Color
nThickness	LONG		Line thickness
bDoubleHead	bool		To draw an arrow on each side of the line.
eLineType	ETcVnLineType [▶ 1622]	LT_8_CONNECTED	Line type
nShift	ULONG	0	Fractional bits of the coordinates (bit shift)
fTipLenght	double	0.1	Lenght of the arrow's tip as a fraction of the line's lenght

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.2 DrawArrow (TcVnVector4_DINT)

Draws an arrowed line.

Syntax

Definition:

```
HRESULT DrawArrow(
    HRESULT            hrPrev,
    TcVnVector4_DINT&  aLine,
    ITcVnImage*        ipDestImage,
    TcVnVector4_LREAL& aColor,
    LONG               nThickness,
    bool               bDoubleHead,
    ETcVnLineType      eLineType = LT_8_CONNECTED,
    ULONG              nShift = 0,
    double             fTipLenght = 0.1
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aLine	TcVnVector4_DINT [▶ 1590]&		The start and end point of a line segment [x1, y1, x2, y2]
ipDestImage	ITcVnImage* [▶ 1797]		Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&		Color
nThickness	LONG		Line thickness
bDoubleHead	bool		To draw an arrow on each side of the line.
eLineType	ETcVnLineType [▶ 1622]	LT_8_CONNECTED	Line type
nShift	ULONG	0	Fractional bits of the coordinates (bit shift)
fTipLenght	double	0.1	Lenght of the arrow's tip as a fraction of the line's lenght

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.3 DrawCircle

Draws a circle.

Syntax

Definition:

```
HRESULT DrawCircle(
    HRESULT              hrPrev,
    ULONG                nCenterX,
    ULONG                nCenterY,
    ULONG                nRadius,
    ITcVnImage*          ipDestImage,
    TcVnVector4_LREAL&   aColor,
    LONG                 nThickness,
    ETcVnLineType        eLineType = LT_8_CONNECTED
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nCenterX	ULONG		x coordinate of the center
nCenterY	ULONG		y coordinate of the center
nRadius	ULONG		Radius
ipDestImage	ITcVnImage* [▶ 1797]		Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&		Color
nThickness	LONG		Line thickness (if negative, the circle is filled)
eLineType	ETcVnLineType [▶ 1622]	LT_8_CONNECTED	Line type

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.4 DrawCircles

Draws circles.

Syntax

Definition:

```
HRESULT DrawCircles(
    HRESULT              hrPrev,
    ITcVnContainer*      ipCircles,
    LONG                 nCircleIndex,
    ITcVnImage*          ipDestImage,
```

```
    TcVnVector4_LREAL& aColor,
    LONG               nThickness,
    ETcVnLineType      eLineType = LT_8_CONNECTED
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipCircles	ITcVnContainer* [▶ 1760]		Container with circles (ContainerType_Vector_TcVnVector3_REAL; Each container element contains the x coordinate of the circle center [0], the y coordinate of the circle center [1], and the radius [2].)
nCircleIndex	LONG		Index of a specific circle to be drawn (if negative, all circles within the container are drawn)
ipDestImage	ITcVnImage* [▶ 1797]		Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&		Color
nThickness	LONG		Line thickness (if negative, the circle is filled)
eLineType	ETcVnLineType [▶ 1622]	LT_8_CONNECTED	Line type

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.5 DrawCircularArc

Draws a circular arc.

Syntax

Definition:

```
HRESULT DrawCircularArc(
    HRESULT             hrPrev,
    TcVnCircularArc&    stCircularArc,
    ITcVnImage*         ipDestImage,
    TcVnVector4_LREAL&  aColor,
    LONG                nThickness,
    ETcVnLineType       eLineType = LT_8_CONNECTED
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stCircularArc	TcVnCircularArc [▶ 1640]&		Circular arc definition
ipDestImage	ITcVnImage* [▶ 1797]		Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&		Color
nThickness	LONG		Line thickness (if negative, the arc is filled)
eLineType	ETcVnLineType [▶ 1622]	LT_8_CONNECTED	Line type

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.6 DrawComponents

Draw the results of connected component function.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT DrawComponents(
    HRESULT             hrPrev,
    ITcVnImage*         ipLabelImage,
    LONG                nLabelID,
    ITcVnImage*&        ipDestImage,
    bool                bShowLabelID,
    TcVnVector4_LREAL&  aColor = {-1, -1, -1, -1},
    LONG                nThickness = 1,
    double              fFontScale = 1
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipLabelImage	ITcVnImage* [▶ 1797]		Image contains labels for each pixel (1 channel DINT or UINT).
nLabelID	LONG		Value of a specific label to be drawn (if negative, all components within the image are drawn)
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (An appropriate image will be created if required. The output image is 1 or 3 channels USINT).
bShowLabelID	bool		Show lable ID in the center of the component on the destination image
aColor	TcVnVector4_LREAL [▶ 1590]&	{-1, -1, -1, -1}	Color to draw the components (for {-1, -1, -1, -1}, a random color is chosen for each component)
nThickness	LONG	1	Text line thickness
fFontScale	double	1	Scaling factor for the text

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.7 DrawContours

Draws a single point set or multiple point sets that are interpreted as contours.

Syntax

Definition:

```
HRESULT DrawContours(
    HRESULT            hrPrev,
    ITcVnContainer*    ipContours,
    LONG               nContourIndex,
```

```
    ITcVnImage*         ipDestImage,
    TcVnVector4_LREAL&  aColor,
    LONG                nThickness,
    ETcVnLineType       eLineType = LT_8_CONNECTED,
    ITcVnContainer*     ipHierarchy = nullptr,
    LONG                nMaxLevel = 2147483647,
    TcVnPoint&          aOffset = {0, 0}
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContours	ITcVnContainer* [▶ 1760]		Single contour (ContainerType_Vector_TcVnPoint2_DINT) or multiple contours (ContainerType_Vector_Vector_TcVnPoint2_DINT)
nContourIndex	LONG		Index of a specific contour to be drawn (if negative, all contours within the container are drawn)
ipDestImage	ITcVnImage* [▶ 1797]		Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&		Color
nThickness	LONG		Line thickness (if negative, the contours are filled)
eLineType	ETcVnLineType [▶ 1622]	LT_8_CONNECTED	Line type
ipHierarchy	ITcVnContainer* [▶ 1760]	nullptr	Contour hierarchy (ContainerType_Vector_TcVnVector4_DINT)
nMaxLevel	LONG	2147483647	Maximum level of contours to be drawn
aOffset	TcVnPoint [▶ 1588]&	{0, 0}	Offset by which every contour point is shifted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.8 DrawEllipse

Draws an ellipse.

Syntax

Definition:

```
HRESULT DrawEllipse(
    HRESULT                 hrPrev,
    TcVnRotatedRectangle&   stEllipse,
    ITcVnImage*             ipDestImage,
    TcVnVector4_LREAL&      aColor,
    LONG                    nThickness,
    ETcVnLineType           eLineType = LT_8_CONNECTED
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stEllipse	TcVnRotatedRectangle [▸ 1658]&		Ellipse to be drawn (rotation angle in degrees)
ipDestImage	ITcVnImage* [▸ 1797]		Destination image
aColor	TcVnVector4_LREAL [▸ 1590]&		Color
nThickness	LONG		Line thickness (if negative, the ellipse is filled)
eLineType	ETcVnLineType [▸ 1622]	LT_8_CONNECTED	Line type

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.9 DrawKeypoints

Draws the keypoints for visualization purpose.

Syntax

Definition:

```
HRESULT DrawKeypoints(
    HRESULT                 hrPrev,
    ITcVnImage*             ipSrcImage,
    ITcVnContainer*         ipKeyPoints,
    ITcVnImage*&            ipDestImage,
```

```
    TcVnVector4_LREAL&    aColor = {-1, -1, -1, -1},
    ETcVnDrawMatchesFlags eFlags = DMF_DEFAULT
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]		Source image (elements of type USINT)
ipKeyPoints	ITcVnContainer* [▸ 1760]		Container with the keypoints (ContainerType_Vector_TcVnKeyPoint)
ipDestImage	ITcVnImage* [▸ 1797]&		Destination image (An appropriate color image will be created if required.)
aColor	TcVnVector4_LREAL [▸ 1590]&	{-1, -1, -1, -1}	Color to draw the keypoints (for {-1, -1, -1, -1}, a random color is chosen for each point)
eFlags	ETcVnDrawMatchesFlags [▸ 1613]	DMF_DEFAULT	A combination of flags to support overdrawing an existing destination image and/or drawing additional (Rich-)Keypoint information (size and orientation)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.10 DrawLine

Draws a line.

Syntax

Definition:

```
HRESULT DrawLine(
    HRESULT             hrPrev,
    ULONG               nX1,
```

```
    ULONG              nY1,
    ULONG              nX2,
    ULONG              nY2,
    ITcVnImage*        ipDestImage,
    TcVnVector4_LREAL& aColor,
    LONG               nThickness,
    ETcVnLineType      eLineType = LT_8_CONNECTED,
    ULONG              nShift = 0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nX1	ULONG		x coordinate of the start point
nY1	ULONG		y coordinate of the start point
nX2	ULONG		x coordinate of the end point
nY2	ULONG		y coordinate of the end point
ipDestImage	ITcVnImage* [▶ 1797]		Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&		Color
nThickness	LONG		Line thickness
eLineType	ETcVnLineType [▶ 1622]	LT_8_CONNECTED	Line type
nShift	ULONG	0	Fractional bits of the coordinates (bit shift)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.11 DrawLine (TcVnVector4_DINT)

Draws a line.

Syntax

Definition:

```
HRESULT DrawLine(
    HRESULT            hrPrev,
    TcVnVector4_DINT&  aLine,
    ITcVnImage*        ipDestImage,
    TcVnVector4_LREAL& aColor,
```

```
    LONG                nThickness,
    ETcVnLineType       eLineType = LT_8_CONNECTED,
    ULONG               nShift = 0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aLine	TcVnVector4_DINT [▶ 1590]&		The start and end point of a line segment [x1, y1, x2, y2]
ipDestImage	ITcVnImage* [▶ 1797]		Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&		Color
nThickness	LONG		Line thickness
eLineType	ETcVnLineType [▶ 1622]	LT_8_CONNECTED	Line type
nShift	ULONG	0	Fractional bits of the coordinates (bit shift)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.12 DrawLine (TcVnVector4_LREAL)

Draws a line.

Syntax

Definition:

```
HRESULT DrawLine(
    HRESULT             hrPrev,
    TcVnVector4_LREAL&  aLine,
    ITcVnImage*         ipDestImage,
    TcVnVector4_LREAL&  aColor,
    LONG                nThickness,
    ETcVnLineType       eLineType = LT_8_CONNECTED,
    ULONG               nShift = 0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aLine	TcVnVector4_LREAL [▸ 1590]&		The first and second element describe the x and y component of the direction vector. The third and fourth element describe the x and y component of the position vector.
ipDestImage	ITcVnImage* [▸ 1797]		Destination image
aColor	TcVnVector4_LREAL [▸ 1590]&		Color
nThickness	LONG		Line thickness
eLineType	ETcVnLineType [▸ 1622]	LT_8_CONNECTED	Line type
nShift	ULONG	0	Fractional bits of the coordinates (bit shift)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.13 DrawLines

Draws lines.

Syntax

Definition:

```
HRESULT DrawLines(
    HRESULT             hrPrev,
    ITcVnContainer*     ipLines,
    LONG                nLineIndex,
    ITcVnImage*         ipDestImage,
    TcVnVector4_LREAL&  aColor,
    LONG                nThickness,
    ETcVnLineType       eLineType = LT_8_CONNECTED,
    ULONG               nShift = 0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipLines	ITcVnContainer* [▶ 1760]		Container with line descriptions (ContainerType_Vector_TcVnVector2_REAL: the distance from the origin [0] in pixels and the rotation angle [1] in radians. ContainerType_Vector_TcVnVector4_LREAL: the first and second element describe the x and y component of a vector collinear to the line. The third and fourth element describe the x and y component of a point on the line. ContainerType_Vector_TcVnVector4_DINT: start and end point [x1, y1, x2, y2])
nLineIndex	LONG		Index of a specific line to be drawn (if negative, all lines within the container are drawn)
ipDestImage	ITcVnImage* [▶ 1797]		Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&		Color
nThickness	LONG		Line thickness
eLineType	ETcVnLineType [▶ 1622]	LT_8_CONNECTED	Line type
nShift	ULONG	0	Fractional bits of the coordinates (bit shift)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.14 DrawMatches

Draws the keypoints and matches for visualization purpose.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT DrawMatches(
    HRESULT               hrPrev,
    ITcVnImage*           ipSrcImage1,
    ITcVnContainer*       ipKeyPoints1,
    ITcVnImage*           ipSrcImage2,
    ITcVnContainer*       ipKeyPoints2,
    ITcVnContainer*       ipMatches1To2,
    ITcVnImage*&          ipDestImage,
    TcVnVector4_LREAL&    aMatchColor = {-1, -1, -1, -1},
    TcVnVector4_LREAL&    aSingleColor = {-1, -1, -1, -1},
    ITcVnContainer*       ipMatchesMask = nullptr,
    ETcVnDrawMatchesFlags eFlags = DMF_DEFAULT
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage1	ITcVnImage* [▶ 1797]		Source image 1 (elements of type USINT)
ipKeyPoints1	ITcVnContainer* [▶ 1760]		Container with the keypoints 1 (ContainerType_Vector_TcVnKeyPoint)
ipSrcImage2	ITcVnImage* [▶ 1797]		Source image 2 (elements of type USINT, must have the same number of channels as ipSrcImage1)
ipKeyPoints2	ITcVnContainer* [▶ 1760]		Container with the keypoints 2 (ContainerType_Vector_TcVnKeyPoint)
ipMatches1To2	ITcVnContainer* [▶ 1760]		Container with the descriptor matches (ContainerType_Vector_TcVnDMatch)
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (An appropriate color image will be created if required.)
aMatchColor	TcVnVector4_LREAL [▶ 1590]&	{-1, -1, -1, -1}	Color to draw the matches (for {-1, -1, -1, -1}, a random color is chosen for each point and line)
aSingleColor	TcVnVector4_LREAL [▶ 1590]&	{-1, -1, -1, -1}	Color to draw the single keypoints (for {-1, -1, -1, -1}, a random color is chosen for each point)
ipMatchesMask	ITcVnContainer* [▶ 1760]	nullptr	Mask to select the matches to be drawn (ContainerType_Vector_SINT; Set to 0 if all matches should be drawn.)
eFlags	ETcVnDrawMatchesFlags [▶ 1613]	DMF_DEFAULT	A combination of flags to support overdrawing an existing destination image and/or drawing additional (Rich-)Keypoint information (size and orientation) and/or skipping single keypoints

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.15 DrawOrientation

Draw the main axis of a set of points/contour based on a rotated rectangle.

Syntax

Definition:

```
HRESULT DrawOrientation(
    HRESULT                 hrPrev,
    TcVnRotatedRectangle&  stOrientation,
    ITcVnImage*             ipDestImage,
    TcVnVector4_LREAL&      aColorMainAxis,
    LONG                    nThickness
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stOrientation	TcVnRotatedRectangle [▶ 1658]&	Rotated rectangle, containig center of mass, lenghts of axes, and rotation angle in clockwise direction
ipDestImage	ITcVnImage* [▶ 1797]	Destination image
aColorMainAxis	TcVnVector4_LREAL [▶ 1590]&	Color of the major axis
nThickness	LONG	Line thickness

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.16 DrawOrientation (Exp)

Draw the main axis, the horizontal axis and the angle of a set of points/contour based on a rotated rectangle.

Syntax

Definition:

```
HRESULT DrawOrientation(
    HRESULT                  hrPrev,
    TcVnRotatedRectangle&  stOrientation,
    ITcVnImage*              ipDestImage,
    TcVnVector4_LREAL&       aColorMainAxis,
    LONG                     nThickness,
    TcVnVector4_LREAL&       aColorHorizontalAxis,
    double                   fFontScale
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stOrientation	TcVnRotatedRectangle [▶ 1658]&	Rotated rectangle, containig center of mass, lenghts of axes, and rotation angle in clockwise direction
ipDestImage	ITcVnImage* [▶ 1797]	Destination image
aColorMainAxis	TcVnVector4_LREAL [▶ 1590]&	Color of the major axis
nThickness	LONG	Line thickness
aColorHorizontalAxis	TcVnVector4_LREAL [▶ 1590]&	Color of the horizontal axis
fFontScale	double	Scaling factor for the text

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.17 DrawPoint

Draw a point.

Syntax

Definition:

```
HRESULT DrawPoint(
    HRESULT              hrPrev,
    ULONG                nX,
    ULONG                nY,
    ITcVnImage*          ipDestImage,
    ETcVnDrawShape       eShape,
```

```
    TcVnVector4_LREAL& aColor,
    ULONG              nSize = 3,
    LONG               nThickness = 1,
    ETcVnLineType      eLineType = LT_8_CONNECTED
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nX	ULONG		x coordinate of the point
nY	ULONG		y coordinate of the point
ipDestImage	ITcVnImage* [▸ 1797]		Destination image
eShape	ETcVnDrawShape [▸ 1614]		Point shape
aColor	TcVnVector4_LREAL [▸ 1590]&		Color (for {-1, -1, -1, -1}, a random color is chosen)
nSize	ULONG	3	Size of the shape (half width)
nThickness	LONG	1	Line thickness (if negative, the shape is filled if it is closed)
eLineType	ETcVnLineType [▸ 1622]	LT_8_CONNECTED	Line type

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.18 DrawPoints

Draw a collection of points.

Syntax

Definition:

```
HRESULT DrawPoints(
    HRESULT            hrPrev,
    ITcVnContainer*    ipPoints,
    ITcVnImage*        ipDestImage,
    ETcVnDrawShape     eShape,
    TcVnVector4_LREAL& aColor,
    ULONG              nSize = 3,
```

```
    LONG               nThickness = 1,
    ETcVnLineType      eLineType = LT_8_CONNECTED
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipPoints	ITcVnContainer* [▶ 1760]		Container with TcVnPoint2_DINT or TcVnPoint2_REAL elements
ipDestImage	ITcVnImage* [▶ 1797]		Destination image
eShape	ETcVnDrawShape [▶ 1614]		Point shape
aColor	TcVnVector4_LREAL [▶ 1590]&		Color (for {-1, -1, -1, -1}, a random color is chosen for each point)
nSize	ULONG	3	Size of the shape (half width)
nThickness	LONG	1	Line thickness (if negative, the shape is filled if it is closed)
eLineType	ETcVnLineType [▶ 1622]	LT_8_CONNECTED	Line type

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.19 DrawPolygon

Draw a pologon using collection of points.

Syntax

Definition:

```
HRESULT DrawPolygon(
    HRESULT            hrPrev,
    ITcVnContainer*    ipPoints,
    ITcVnImage*        ipDestImage,
    TcVnVector4_LREAL& aColor,
    LONG               nThickness,
    bool               bIsClosed = true,
    ETcVnLineType      eLineType = LT_8_CONNECTED,
    ULONG              nShift = 0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipPoints	ITcVnContainer* [▶ 1760]		Container with TcVnPoint2_DINT, TcVnPoint2_REAL, or TcVnPoint2_LREAL elements
ipDestImage	ITcVnImage* [▶ 1797]		Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&		Color
nThickness	LONG		Line thickness (if negative, the shape is filled)
bIsClosed	bool	true	If it is true, the function draws a line from the last point to the first point to close the polygon.
eLineType	ETcVnLineType [▶ 1622]	LT_8_CONNECTED	Line type
nShift	ULONG	0	Fractional bits of the coordinates (bit shift)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.20 DrawRectangle

Draw a rectangle.

Syntax

Definition:

```
HRESULT DrawRectangle(
    HRESULT            hrPrev,
    ULONG              nTopLeftX,
    ULONG              nTopLeftY,
    ULONG              nBottomRightX,
    ULONG              nBottomRightY,
    ITcVnImage*        ipDestImage,
    TcVnVector4_LREAL& aColor,
    LONG               nThickness
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nTopLeftX	ULONG	x coordinate of the top left corner
nTopLeftY	ULONG	y coordinate of the top left corner
nBottomRightX	ULONG	x coordinate of the bottom right corner
nBottomRightY	ULONG	y coordinate of the bottom right corner
ipDestImage	ITcVnImage* [▶ 1797]	Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&	Color
nThickness	LONG	Line thickness (if negative, the rectangle is filled)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.21 DrawRectangle (TcVnRectangle_DINT)

Draw a rectangle.

Syntax

Definition:

```
HRESULT DrawRectangle(
    HRESULT             hrPrev,
    TcVnRectangle_DINT& stRectangle,
    ITcVnImage*         ipDestImage,
    TcVnVector4_LREAL&  aColor,
    LONG                nThickness
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stRectangle	TcVnRectangle_DINT [▶ 1657]&	Rectangle to be drawn
ipDestImage	ITcVnImage* [▶ 1797]	Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&	Color
nThickness	LONG	Line thickness (if negative, the rectangle is filled)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.22 DrawRectangle (TcVnRectangle_UDINT)

Draw a rectangle.

Syntax

Definition:

```
HRESULT DrawRectangle(
    HRESULT              hrPrev,
    TcVnRectangle_UDINT& stRectangle,
    ITcVnImage*          ipDestImage,
    TcVnVector4_LREAL&   aColor,
    LONG                 nThickness
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stRectangle	TcVnRectangle_UDINT [▶ 1658]&	Rectangle to be drawn
ipDestImage	ITcVnImage* [▶ 1797]	Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&	Color
nThickness	LONG	Line thickness (if negative, the rectangle is filled)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.23 DrawRotatedRectangle

Draw a rectangle.

Syntax

Definition:

```
HRESULT DrawRotatedRectangle(
    HRESULT                hrPrev,
    TcVnRotatedRectangle&  stRectangle,
    ITcVnImage*            ipDestImage,
    TcVnVector4_LREAL&     aColor,
    LONG                   nThickness,
    ETcVnLineType          eLineType = LT_8_CONNECTED
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stRectangle	TcVnRotatedRectangle [▶ 1658]&		Rectangle to be drawn (rotation angle in degrees)
ipDestImage	ITcVnImage* [▶ 1797]		Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&		Color
nThickness	LONG		Line thickness (if negative, the rectangle is filled)
eLineType	ETcVnLineType [▶ 1622]	LT_8_CONNECTED	Line type

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.24 FillCircle

Paint a filled circle.

Syntax

Definition:

```
HRESULT FillCircle(
    HRESULT             hrPrev,
    ULONG               nCenterX,
    ULONG               nCenterY,
    ULONG               nRadius,
    ITcVnImage*         ipDestImage,
    TcVnVector4_LREAL&  aColor
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nCenterX	ULONG	x coordinate of the center
nCenterY	ULONG	y coordinate of the center
nRadius	ULONG	Radius
ipDestImage	ITcVnImage* [▶ 1797]	Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&	Color

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.25 FillContours

Paint a single point set or multiple point sets that are interpreted as contours.

Syntax

Definition:

```
HRESULT FillContours(
    HRESULT             hrPrev,
    ITcVnContainer*     ipContours,
    LONG                nContourIndex,
    ITcVnImage*         ipDestImage,
    TcVnVector4_LREAL&  aColor
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipContours	ITcVnContainer* [▶ 1760]	Single contour (ContainerType_Vector_TcVnPoint2_DINT) or multiple contours (ContainerType_Vector_Vector_TcVnPoint2_DINT)
nContourIndex	LONG	Index of a specific contour to be drawn (if negative, all contours within the container are drawn)
ipDestImage	ITcVnImage* [▶ 1797]	Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&	Color

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.26 FillEllipse

Paint a filled ellipse.

Syntax

Definition:

```
HRESULT FillEllipse(
    HRESULT                 hrPrev,
    TcVnRotatedRectangle& stEllipse,
    ITcVnImage*             ipDestImage,
    TcVnVector4_LREAL&      aColor
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stEllipse	TcVnRotatedRectangle [▶ 1658]&	Ellipse to be painted (rotation angle in degrees)
ipDestImage	ITcVnImage* [▶ 1797]	Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&	Color

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.27 FillPolygon

Paint a filled pologon using a set of points.

Syntax

Definition:

```
HRESULT FillPolygon(
    HRESULT            hrPrev,
    ITcVnContainer*    ipPoints,
    ITcVnImage*        ipDestImage,
    TcVnVector4_LREAL& aColor
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipPoints	ITcVnContainer* [▶ 1760]	Container with TcVnPoint2_DINT, TcVnPoint2_REAL, or TcVnPoint2_LREAL elements
ipDestImage	ITcVnImage* [▶ 1797]	Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&	Color

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.28 FillRectangle

Paint a filled rectangle.

Syntax

Definition:

```
HRESULT FillRectangle(
    HRESULT             hrPrev,
    ULONG               nTopLeftX,
    ULONG               nTopLeftY,
    ULONG               nBottomRightX,
    ULONG               nBottomRightY,
    ITcVnImage*         ipDestImage,
    TcVnVector4_LREAL&  aColor
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nTopLeftX	ULONG	x coordinate of the top left corner
nTopLeftY	ULONG	y coordinate of the top left corner
nBottomRightX	ULONG	x coordinate of the bottom right corner
nBottomRightY	ULONG	y coordinate of the bottom right corner
ipDestImage	ITcVnImage* [▸ 1797]	Destination image
aColor	TcVnVector4_LREAL [▸ 1590]&	Color

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.29 FillRotatedRectangle

Paint a filled rectangle.

Syntax

Definition:

```
HRESULT FillRotatedRectangle(
    HRESULT                 hrPrev,
    TcVnRotatedRectangle&   stRectangle,
    ITcVnImage*             ipDestImage,
    TcVnVector4_LREAL&      aColor
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stRectangle	TcVnRotatedRectangle [▶ 1658]&	Rectangle to be painted (rotation angle in degrees)
ipDestImage	ITcVnImage* [▶ 1797]	Destination image
aColor	TcVnVector4_LREAL [▶ 1590]&	Color

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.30 PlotIntensityProfile

Plots the pixel intensity profile along a line segment in an image.

Syntax

Definition:

```
HRESULT PlotIntensityProfile(
    HRESULT             hrPrev,
    ITcVnImage*         ipSrcImage,
    ITcVnImage*&        ipDestImage,
    TcVnPoint2_REAL&    aStartPoint,
    TcVnPoint2_REAL&    aEndPoint,
    TcVnVector4_LREAL&  aBackgroundColor = {0, 0, 0, 0},
    TcVnVector4_LREAL&  aLineColor = {255, 255, 255, 255},
    TcVnVector4_LREAL&  aGridColor = {64, 64, 64, 64},
    LONG                nScaleX = 2,
    LONG                nScaleY = 2,
    LONG                nThickness = 1,
    LONG                nDestChannels = 1
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (USINT, 1 channel)
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (USINT, 1 channel. x: position, y: intensity, origin: bottom left. An appropriate destination image will be created if required.)
aStartPoint	TcVnPoint2_REAL [▶ 1588]&		Start point of the line segment
aEndPoint	TcVnPoint2_REAL [▶ 1588]&		End point of the line segment
aBackgroundColor	TcVnVector4_LREAL [▶ 1590]&	{0, 0, 0, 0}	Background color
aLineColor	TcVnVector4_LREAL [▶ 1590]&	{255, 255, 255, 255}	Line color
aGridColor	TcVnVector4_LREAL [▶ 1590]&	{64, 64, 64, 64}	Grid color
nScaleX	LONG	2	Scale in x direction (ipDestImage will have a width of nScaleX * (ipSrcImage width - 1) + 1)
nScaleY	LONG	2	Scale in y direction (ipDestImage will have a height of nScaleY * 255 + 1)
nThickness	LONG	1	Line thickness
nDestChannels	LONG	1	ipDestImage channels (1 or 3)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.31 PutLabel

Write a label (text on unified background) into an image.

Syntax

Definition:

```
HRESULT PutLabel(
    HRESULT             hrPrev,
    PCCH                sText,
    ITcVnImage*         ipDestImage,
    ULONG               nX,
    ULONG               nY,
    double              fFontScale,
    ULONG               nThickness = 1,
    ETcVnFontType       eFontType = FT_HERSHEY_SIMPLEX,
    TcVnVector4_LREAL&  aFontColor = {0, 0, 0, 0},
    TcVnVector4_LREAL&  aBackgroundColor = {255, 255, 255, 255},
    ETcVnLineType       eLineType = LT_4_CONNECTED
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
sText	PCCH		Text
ipDestImage	ITcVnImage* [▸ 1797]		Destination image
nX	ULONG		x coordinate (bottom left)
nY	ULONG		y coordinate (bottom left)
fFontScale	double		Scaling factor
nThickness	ULONG	1	Line thickness
eFontType	ETcVnFontType [▸ 1619]	FT_HERSHEY_SIMPLEX	Font type
aFontColor	TcVnVector4_LREAL [▸ 1590]&	{0, 0, 0, 0}	Font color
aBackgroundColor	TcVnVector4_LREAL [▸ 1590]&	{255, 255, 255, 255}	Background color
eLineType	ETcVnLineType [▸ 1622]	LT_4_CONNECTED	Line type

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.11.32 PutText

Write text into an image.

Syntax

Definition:

```
HRESULT PutText(
    HRESULT              hrPrev,
    PCCH                 sText,
    ITcVnImage*          ipDestImage,
    ULONG                nX,
    ULONG                nY,
    ETcVnFontType        eFontType,
    double               fFontScale,
    TcVnVector4_LREAL&   aColor,
    LONG                 nThickness = 1,
    ETcVnLineType        eLineType = LT_8_CONNECTED,
    bool                 bBottomLeftOrigin = false
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
sText	PCCH		Text
ipDestImage	ITcVnImage* [▶ 1797]		Destination image
nX	ULONG		x coordinate (bottom left)
nY	ULONG		y coordinate (bottom left)
eFontType	ETcVnFontType [▶ 1619]		Font type
fFontScale	double		Scaling factor
aColor	TcVnVector4_LREAL [▶ 1590]&		Text color
nThickness	LONG	1	Line thickness
eLineType	ETcVnLineType [▶ 1622]	LT_8_CONNECTED	Line type
bBottomLeftOrigin	bool	false	Sets the image origin to the bottom left corner, if true

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.12 Fourier Analysis

该组包含用于图像频率分析和频率滤波的函数。

函数

巴特沃斯滤波器

- CreateBandpassButterworthFilter [▶ 2257]
- CreateBandrejectButterworthFilter [▶ 2259]
- CreateHighpassButterworthFilter [▶ 2262]
- CreateLowpassButterworthFilter [▶ 2264]

高斯滤波器

- CreateBandpassGaussianFilter [▶ 2258]
- CreateBandrejectGaussianFilter [▶ 2261]
- CreateHighpassGaussianFilter [▶ 2263]
- CreateLowpassGaussianFilter [▶ 2265]

离散傅立叶变换

- Dft [▶ 2266]
- InverseDft [▶ 2267]
- OptimalDftSize [▶ 2268]

其它

- PadImageBorder [▶ 2268]

6.2.3.12.1 CreateBandpassButterworthFilter

Creates a bandpass Butterworth filter, which can be applied to a frequency domain image by element-wise multiplication.

Syntax

Definition:

```
HRESULT CreateBandpassButterworthFilter(
    HRESULT       hrPrev,
    ITcVnImage*&  ipFilter,
    ULONG         nWidth,
    ULONG         nHeight,
    bool          bDoublePrecision,
    bool          bOriginAtCenter,
    double        fCutoffDistance,
    double        fBandWidth,
    ULONG         nOrder,
    double        fScale
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipFilter	ITcVnImage* [▶ 1797]&	Returns the created filter (1 channel of type REAL or LREAL, depending on bDoublePrecision. Non-zero interface pointers are reused.).
nWidth	ULONG	Filter width (even, >= 2, must match the width of the image it is applied to)
nHeight	ULONG	Filter height (even, >= 2, must match the height of the image it is applied to)
bDoublePrecision	bool	If true, the filter is generated with double precision (LREAL) instead of single precision (REAL).
bOriginAtCenter	bool	If true, the filter origin is shifted to the image center.
fCutoffDistance	double	The cutoff distance of the Butterworth filter (> 0).
fBandWidth	double	The band width of the Butterworth filter (> 0).
nOrder	ULONG	The order of the Butterworth filter (> 0).
fScale	double	The scale factor of the Butterworth filter denominator term (> 0, e.g. 0.414 or 1.0).

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.12.2 CreateBandpassGaussianFilter

Creates a bandpass Gaussian filter, which can be applied to a frequency domain image by element-wise multiplication.

Syntax

Definition:

```
HRESULT CreateBandpassGaussianFilter(
    HRESULT      hrPrev,
    ITcVnImage*& ipFilter,
```

```
    ULONG         nWidth,
    ULONG         nHeight,
    bool          bDoublePrecision,
    bool          bOriginAtCenter,
    double        fCutoffDistance,
    double        fBandWidth
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipFilter	ITcVnImage* [▶ 1797]&	Returns the created filter (1 channel of type REAL or LREAL, depending on bDoublePrecision. Non-zero interface pointers are reused.).
nWidth	ULONG	Filter width (even, >= 2, must match the width of the image it is applied to)
nHeight	ULONG	Filter height (even, >= 2, must match the height of the image it is applied to)
bDoublePrecision	bool	If true, the filter is generated with double precision (LREAL) instead of single precision (REAL).
bOriginAtCenter	bool	If true, the filter origin is shifted to the image center.
fCutoffDistance	double	The cutoff distance of the Gaussian filter (> 0).
fBandWidth	double	The band width of the Gaussian filter (> 0).

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.12.3 CreateBandrejectButterworthFilter

Creates a bandreject Butterworth filter, which can be applied to a frequency domain image by element-wise multiplication.

Syntax

Definition:

```
HRESULT CreateBandrejectButterworthFilter(
    HRESULT      hrPrev,
    ITcVnImage*& ipFilter,
    ULONG        nWidth,
    ULONG        nHeight,
    bool         bDoublePrecision,
    bool         bOriginAtCenter,
    double       fCutoffDistance,
    double       fBandWidth,
    ULONG        nOrder,
    double       fScale
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipFilter	ITcVnImage* [▶ 1797]&	Returns the created filter (1 channel of type REAL or LREAL, depending on bDoublePrecision. Non-zero interface pointers are reused.).
nWidth	ULONG	Filter width (even, >= 2, must match the width of the image it is applied to)
nHeight	ULONG	Filter height (even, >= 2, must match the height of the image it is applied to)
bDoublePrecision	bool	If true, the filter is generated with double precision (LREAL) instead of single precision (REAL).
bOriginAtCenter	bool	If true, the filter origin is shifted to the image center.
fCutoffDistance	double	The cutoff distance of the Butterworth filter (> 0).
fBandWidth	double	The band width of the Butterworth filter (> 0).
nOrder	ULONG	The order of the Butterworth filter (> 0).
fScale	double	The scale factor of the Butterworth filter denominator term (> 0, e.g. 0.414 or 1.0).

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.12.4 CreateBandrejectGaussianFilter

Creates a bandreject Gaussian filter, which can be applied to a frequency domain image by element-wise multiplication.

Syntax

Definition:

```
HRESULT CreateBandrejectGaussianFilter(
    HRESULT       hrPrev,
    ITcVnImage*&  ipFilter,
    ULONG         nWidth,
    ULONG         nHeight,
    bool          bDoublePrecision,
    bool          bOriginAtCenter,
    double        fCutoffDistance,
    double        fBandWidth
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipFilter	ITcVnImage* [▶ 1797]&	Returns the created filter (1 channel of type REAL or LREAL, depending on bDoublePrecision. Non-zero interface pointers are reused.).
nWidth	ULONG	Filter width (even, >= 2, must match the width of the image it is applied to)
nHeight	ULONG	Filter height (even, >= 2, must match the height of the image it is applied to)
bDoublePrecision	bool	If true, the filter is generated with double precision (LREAL) instead of single precision (REAL).
bOriginAtCenter	bool	If true, the filter origin is shifted to the image center.
fCutoffDistance	double	The cutoff distance of the Gaussian filter (> 0).
fBandWidth	double	The band width of the Gaussian filter (> 0).

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.12.5 CreateHighpassButterworthFilter

Creates a highpass Butterworth filter, which can be applied to a frequency domain image by element-wise multiplication.

Syntax

Definition:

```
HRESULT CreateHighpassButterworthFilter(
    HRESULT       hrPrev,
    ITcVnImage*&  ipFilter,
    ULONG         nWidth,
    ULONG         nHeight,
    bool          bDoublePrecision,
    bool          bOriginAtCenter,
    double        fCutoffDistance,
    ULONG         nOrder,
    double        fScale
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipFilter	ITcVnImage* [▶ 1797]&	Returns the created filter (1 channel of type REAL or LREAL, depending on bDoublePrecision. Non-zero interface pointers are reused.).
nWidth	ULONG	Filter width (even, >= 2, must match the width of the image it is applied to)
nHeight	ULONG	Filter height (even, >= 2, must match the height of the image it is applied to)
bDoublePrecision	bool	If true, the filter is generated with double precision (LREAL) instead of single precision (REAL).
bOriginAtCenter	bool	If true, the filter origin is shifted to the image center.
fCutoffDistance	double	The cutoff distance of the Butterworth filter (> 0).
nOrder	ULONG	The order of the Butterworth filter (> 0).
fScale	double	The scale factor of the Butterworth filter denominator term (> 0, e.g. 0.414 or 1.0).

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.12.6 CreateHighpassGaussianFilter

Creates a highpass Gaussian filter, which can be applied to a frequency domain image by element-wise multiplication.

Syntax

Definition:

```
HRESULT CreateHighpassGaussianFilter(
    HRESULT      hrPrev,
    ITcVnImage*& ipFilter,
    ULONG        nWidth,
    ULONG        nHeight,
    bool         bDoublePrecision,
    bool         bOriginAtCenter,
    double       fCutoffDistance
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipFilter	ITcVnImage* [▶ 1797]&	Returns the created filter (1 channel of type REAL or LREAL, depending on bDoublePrecision. Non-zero interface pointers are reused.).
nWidth	ULONG	Filter width (even, >= 2, must match the width of the image it is applied to)
nHeight	ULONG	Filter height (even, >= 2, must match the height of the image it is applied to)
bDoublePrecision	bool	If true, the filter is generated with double precision (LREAL) instead of single precision (REAL).
bOriginAtCenter	bool	If true, the filter origin is shifted to the image center.
fCutoffDistance	double	The cutoff distance of the Gaussian filter (> 0).

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.12.7 CreateLowpassButterworthFilter

Creates a lowpass Butterworth filter, which can be applied to a frequency domain image by element-wise multiplication.

Syntax

Definition:

```
HRESULT CreateLowpassButterworthFilter(
    HRESULT      hrPrev,
    ITcVnImage*& ipFilter,
    ULONG        nWidth,
    ULONG        nHeight,
    bool         bDoublePrecision,
    bool         bOriginAtCenter,
    double       fCutoffDistance,
    ULONG        nOrder,
    double       fScale
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipFilter	ITcVnImage* [▶ 1797]&	Returns the created filter (1 channel of type REAL or LREAL, depending on bDoublePrecision. Non-zero interface pointers are reused.).
nWidth	ULONG	Filter width (even, >= 2, must match the width of the image it is applied to)
nHeight	ULONG	Filter height (even, >= 2, must match the height of the image it is applied to)
bDoublePrecision	bool	If true, the filter is generated with double precision (LREAL) instead of single precision (REAL).
bOriginAtCenter	bool	If true, the filter origin is shifted to the image center.
fCutoffDistance	double	The cutoff distance of the Butterworth filter (> 0).
nOrder	ULONG	The order of the Butterworth filter (> 0).
fScale	double	The scale factor of the Butterworth filter denominator term (> 0, e.g. 0.414 or 1.0).

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.12.8 CreateLowpassGaussianFilter

Creates a lowpass Gaussian filter, which can be applied to a frequency domain image by element-wise multiplication.

Syntax

Definition:

```
HRESULT CreateLowpassGaussianFilter(
    HRESULT       hrPrev,
    ITcVnImage*&  ipFilter,
    ULONG         nWidth,
    ULONG         nHeight,
    bool          bDoublePrecision,
    bool          bOriginAtCenter,
    double        fCutoffDistance
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipFilter	ITcVnImage* [▸ 1797]&	Returns the created filter (1 channel of type REAL or LREAL, depending on bDoublePrecision. Non-zero interface pointers are reused.).
nWidth	ULONG	Filter width (even, >= 2, must match the width of the image it is applied to)
nHeight	ULONG	Filter height (even, >= 2, must match the height of the image it is applied to)
bDoublePrecision	bool	If true, the filter is generated with double precision (LREAL) instead of single precision (REAL).
bOriginAtCenter	bool	If true, the filter origin is shifted to the image center.
fCutoffDistance	double	The cutoff distance of the Gaussian filter (> 0).

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.12.9 Dft

Computes the DFT for a given image.

Syntax

Definition:

```
HRESULT Dft(
    HRESULT       hrPrev,
    ITcVnImage*   ipSpatialImage,
    ITcVnImage*&  ipFrequencyImage,
    bool          bPackedCCS,
    bool          bAutoPadding
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSpatialImage	ITcVnImage* [▸ 1797]	Source image (elements of type REAL or LREAL, 1 (Re) or 2 (Re + Im) channels)
ipFrequencyImage	ITcVnImage* [▸ 1797]&	Destination image (same type as ipSpatialImage, but number of channels and size can vary depending on bPackedCCS and bAutoPadding.
bPackedCCS	bool	If true and ipSpatialImage has only 1 channel, the result image will have 1 channel with packed complex-conjugate-symmetrical format results. Otherwise, the result image will have 2 separate channels (Re + Im), containing the full spectrum.
bAutoPadding	bool	If true, the input image is automatically padded (with 0s) to optimal size if required, to speed up dft (Creates a temporary copy so that ipSpatialImage content stays unchanged, which also requires some additional computation power. Therefore, it is recommended to compare execution times with and without padding.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.12.10 InverseDft

Compute the inverse DFT for a given frequency image.

Syntax

Definition:

```
HRESULT InverseDft(
    HRESULT       hrPrev,
    ITcVnImage*   ipFrequencyImage,
    ITcVnImage*&  ipSpatialImage,
    bool          bRealOutput
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipFrequencyImage	ITcVnImage* [▶ 1797]	Source frequency image (elements of type REAL or LREAL, 1 (packed CCS) or 2 (Re + Im) channels)
ipSpatialImage	ITcVnImage* [▶ 1797]&	Destination image (Same type as ipFrequencyImage, but number of channels can vary depending on bRealOutput. An appropriate image will be created if required.)
bRealOutput	bool	Only relevant if ipFrequencyImage has 2 channels. If true, the result image will have only 1 channel. Otherwise, the result image will have 2 separate channels (Re + Im). If ipFrequencyImage has only 1 channel, packed CCS format is assumed and the result image will always have only 1 channel.

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.12.11 OptimalDftSize

Computes the optimal image size for applying a dft (can lead to better performance).

Syntax

Definition:

```
HRESULT OptimalDftSize(
    HRESULT      hrPrev,
    ITcVnImage*  ipImage,
    ULONG&       nOptWidth,
    ULONG&       nOptHeight
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▶ 1797]	Source image for which to compute the optimal width and height
nOptWidth	ULONG&	Returns the optimal width of the image
nOptHeight	ULONG&	Returns the optimal height of the image

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.12.12 PadImageBorder

Add padding around the original image borders.

Syntax

Definition:

```
HRESULT PadImageBorder(
    HRESULT                        hrPrev,
    ITcVnImage*                    ipSrcImage,
    ITcVnImage*&                   ipPaddedImage,
    ULONG                          nTopBorder,
    ULONG                          nBottomBorder,
    ULONG                          nLeftBorder,
    ULONG                          nRightBorder,
    ETcVnBorderInterpolationMethod ePaddingType = BIM_CONSTANT,
    TcVnVector4_LREAL&             aPaddingValue = {0, 0, 0, 0}
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipPaddedImage	ITcVnImage* [▶ 1797]&		Padded destination image (Same type as ipSrcImage, an appropriate destination image will be created if required.)
nTopBorder	ULONG		Padding height in pixels above top border
nBottomBorder	ULONG		Padding height in pixels below bottom border
nLeftBorder	ULONG		Padding width in pixels before left border
nRightBorder	ULONG		Padding width in pixels after right border
ePaddingType	ETcVnBorderInterpolationMethod [▶ 1593]	BIM_CONSTANT	Specifies how the pixel values of the padding area are determined
aPaddingValue	TcVnVector4_LREAL [▶ 1590]&	{0, 0, 0, 0}	Specifies the padding value if CONSTANT is used

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13 Geometric and Coordinate Transformations

该组包含用于图像和像素转换的函数。

函数

标定 [▶ 2271]

- 内部和外部标定
- 光学几何畸变补偿
- 图像坐标与世界坐标之间的转换

仿射转换

笛卡尔坐标和极坐标之间的转换

简单图像运算

图像对齐和旋转

透视转换

6.2.3.13.1 Calibration

该子组包含用于像素与现实世界像素之间相关性的函数。它们通常与测量函数 [▶ 2496]一起使用。

函数

标定

镜头几何畸变补偿

图像坐标与世界坐标之间的转换

输入图像要求

输入图像不应使用压缩或滤波格式，如Bayer格式。就标定函数而言，这样做的后果是干扰模板识别，从而导致标定结果较差或根本没有。由于在补偿函数过程中进行了插值，改变了像素的原始排列，因此之后无法再正确转换图像。因此，建议在使用该子组中的函数之前一定要先进行转换。

相机矩阵要求

$$CameraMatrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

相机矩阵由焦距 fx 和 fy 以及主点（cx,cy）组成，以像素单位表示。主点是光轴与图像的交点，通常位于图像中心附近。必须满足以下条件：

- 0 < fx 和 fy
- cx >= 0 和 <= 图像宽度 -1
- cy >= 0 和 <= 图像高度
- 如图所示，矩阵中的其他位置必须填 0 或 1。不允许其他值。

6.2.3.13.1.1 CalibrateCamera

Compute the camera parameters (intrinsic + extrinsic) by evaluating an image containing a calibration pattern (circles).
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT CalibrateCamera(
    HRESULT                         hrPrev,
    ITcVnImage*                     ipSrcImage,
    ITcVnContainer*                 ipReferencePoints,
    TcVnMatrix3x3_LREAL&            aCameraMatrix,
    TcVnArray8_LREAL&               aDistortionCoefficients,
    TcVnMatrix3x3_LREAL&            aRotationMatrix,
    TcVnVector3_LREAL&              aTranslationVector,
    double&                         fReprojError,
    TcVnParamsBlobDetection&        stBlobDetectionParams,
    TcVnCameraCalibrationOptions&  stCalibrationOptions
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]	Source image containing a calibration pattern (1 channel, USINT elements)
ipReferencePoints	ITcVnContainer* [▸ 1760]	Reference calibration pattern point positions (ContainerType_Vector_TcVnPoint3_REAL)
aCameraMatrix	TcVnMatrix3x3_LREAL [▸ 1590]&	Returns the camera matrix
aDistortionCoefficients	TcVnArray8_LREAL [▸ 1590]&	Returns the lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▸ 1590]&	Returns the rotation matrix
aTranslationVector	TcVnVector3_LREAL [▸ 1590]&	Returns the translation vector
fReprojError	double&	Returns the reprojection error
stBlobDetectionParams	TcVnParamsBlobDetection [▸ 1646]&	Parameters for the internally used F_VN_DetectBlobs function
stCalibrationOptions	TcVnCameraCalibrationOptions [▸ 1639]&	Calibration options

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.2 CalibrateCamera (Exp)

Compute the camera parameters (intrinsic + extrinsic) by evaluating images containing a calibration pattern (circles). The extrinsic parameters are computed for the first image in the array.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT CalibrateCamera(
    HRESULT                        hrPrev,
    PVOID                          pSrcImages,
    USHORT                         nSrcArraySize,
    ITcVnContainer*                ipReferencePoints,
    TcVnMatrix3x3_LREAL&           aCameraMatrix,
    TcVnArray8_LREAL&              aDistortionCoefficients,
    TcVnMatrix3x3_LREAL&           aRotationMatrix,
    TcVnVector3_LREAL&             aTranslationVector,
    double&                        fReprojError,
    TcVnParamsBlobDetection&       stBlobDetectionParams,
    TcVnCameraCalibrationOptions&  stCalibrationOptions,
    bool                           bSubpixelAccuracy = false,
    ETcVnEdgeDirection             eEdgeDirection = ED_DARK_TO_LIGHT,
    float                          fMinStrength = 50,
    ULONG                          nMaxThickness = 10,
    ULONG                          nSubpixelsIterations = 10,
    ULONG                          nSearchLines = 92,
    float                          fApproxPrecision = 0.001f,
    ETcVnEdgeDetectionAlgorithm    eAlgorithm = EDA_INTERPOLATION,
    ETcVnWorldCoordinateSystem     eWorldSystem = WCS_X_RIGHT_Y_DOWN
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
pSrcImages	PVOID		Pointer to an array of source images, each containing the same calibration pattern (1 channel, USINT elements)
nSrcArraySize	USHORT		pSrcImages array size
ipReferencePoints	ITcVnContainer* [▸ 1760]		Reference calibration pattern point positions (ContainerType_Vector_TcVnPoint3_REAL)
aCameraMatrix	TcVnMatrix3x3_LREAL [▸ 1590]&		Returns the camera matrix
aDistortionCoefficients	TcVnArray8_LREAL [▸ 1590]&		Returns the lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▸ 1590]&		Returns the rotation matrix
aTranslationVector	TcVnVector3_LREAL [▸ 1590]&		Returns the translation vector
fReprojError	double&		Returns the reprojection error
stBlobDetectionParams	TcVnParamsBlobDetection [▸ 1646]&		Parameters for the internally used F_VN_DetectBlobs function
stCalibrationOptions	TcVnCameraCalibrationOptions [▸ 1639]&		Calibration options
bSubpixelAccuracy	bool	false	If true, the pattern points are detected with subpixel accuracy
eEdgeDirection	ETcVnEdgeDirection [▸ 1615]	ED_DARK_TO_LIGHT	Specification of the edge direction to search for (from center to outside ellipse)
fMinStrength	float	50	Specification of the minimum strength (intensity difference) of the edge to search for
nMaxThickness	ULONG	10	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels

Name	Type	Default	Description
nSubpixelsIterations	ULONG	10	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)
nSearchLines	ULONG	92	Specifies the amount of search lines, which are equally distributed in all directions (must be >= 8 and a multiple of 4)
fApproxPrecision	float	0.001f	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▸ 1614]	EDA_INTERPOLATION	Selection of the edge detection algorithm
eWorldSystem	ETcVnWorldCoordinateSystem [▸ 1636]	WCS_X_RIGHT_Y_DOWN	World coordinate system orientation

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.3 CalibrateCameraManually

Compute the camera parameters (intrinsic + extrinsic). The extrinsic parameters are computed for the first inner container.

Syntax

Definition:

```
HRESULT CalibrateCameraManually(
    HRESULT                    hrPrev,
    ITcVnContainer*            ipImagePoints,
    ULONG                      nImageWidth,
    ULONG                      nImageHeight,
    ITcVnContainer*            ipReferencePoints,
```

```
    TcVnMatrix3x3_LREAL&           aCameraMatrix,
    TcVnArray8_LREAL&              aDistortionCoefficients,
    TcVnMatrix3x3_LREAL&           aRotationMatrix,
    TcVnVector3_LREAL&             aTranslationVector,
    double&                        fReprojError,
    TcVnCameraCalibrationOptions&  stCalibrationOptions,
    ETcVnWorldCoordinateSystem     eWorldSystem = WCS_X_RIGHT_Y_DOWN
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImagePoints	ITcVnContainer* [▸ 1760]		Provided image points of multiple images (ContainerType_Vector_Vector_TcVnPoint2_REAL, each inner container with at least 6 points). The container and point order must match ipReferencePoints.
nImageWidth	ULONG		Image width
nImageHeight	ULONG		Image height
ipReferencePoints	ITcVnContainer* [▸ 1760]		Reference world points (ContainerType_Vector_Vector_TcVnPoint3_REAL). The number of inner containers and their amount of points must match ipImagePoints.
aCameraMatrix	TcVnMatrix3x3_LREAL [▸ 1590]&		Returns the camera matrix
aDistortionCoefficients	TcVnArray8_LREAL [▸ 1590]&		Returns the lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▸ 1590]&		Returns the rotation matrix (for the first inner container)
aTranslationVector	TcVnVector3_LREAL [▸ 1590]&		Returns the translation vector (for the first inner container)
fReprojError	double&		Returns the reprojection error
stCalibrationOptions	TcVnCameraCalibrationOptions [▸ 1639]&		Calibration options
eWorldSystem	ETcVnWorldCoordinateSystem [▸ 1636]	WCS_X_RIGHT_Y_DOWN	World coordinate system orientation

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.4 CalibrateCameraPlanar

Calibrate camera using a planar calibration pattern comprised of four circles marking the corners of a rectangle.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT CalibrateCameraPlanar(
    HRESULT                 hrPrev,
    ITcVnImage*             ipSrcImage,
    TcVnArray4_Point2_REAL& aMarkerPositions,
    TcVnMatrix3x3_LREAL&    aTransformationMatrix,
    TcVnArray4_Point2_REAL& aMarkerImagePositions,
    ULONG                   nMinMarkerArea = 100,
    ULONG                   nMaxMarkerArea = 10000,
    double                  fMaxMarkerEccentricity = 0.05
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Binary source image (background zero, markers non-zero)
aMarkerPositions	TcVnArray4_Point2_REAL [▶ 1590]&		Marker positions within their plane using world units (right-handed coordinate system)
aTransformationMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&		Returns a 3-by-3 matrix for the transformation from image coordinates into the reference coordinate system
aMarkerImagePositions	TcVnArray4_Point2_REAL [▶ 1590]&		Returns the marker positions in image coordinates (left-handed coordinate system)
nMinMarkerArea	ULONG	100	Minimum area for the detection of markers (Smaller markers are ignored.)
nMaxMarkerArea	ULONG	10000	Maximum area for the detection of markers (Larger markers are ignored.)
fMaxMarkerEccentricity	double	0.05	Maximum eccenticity for the detection of markers (The eccentricity measures deviations from a round shape (eccentricity = 0: round object; eccentricity = 1.0: linear object).)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.5 CalibrateLinescanCamera

Compute the camera parameters (intrinsic + extrinsic) by evaluating images containing a line calibration pattern. The pattern must consist of alternating vertical and diagonal lines, with each diagonal having the same constant slope: |\|\|...\|. The results are only valid for the x-direction of a line scan image. To get accurate x-values for a transformation into world coordinates, the y-coordinate must be set to half the sensor width and so equal to the cy-value of the camera matrix. For other y-values, the results are approximated and can be inaccurate. The extrinsic parameters are computed for the first image in the array.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT CalibrateLinescanCamera(
    HRESULT                         hrPrev,
    PVOID                           pSrcImages,
    USHORT                          nSrcArraySize,
    ITcVnContainer*                 ipPatternLineOrigins,
    double                          fSlopeRad,
    TcVnMatrix3x3_LREAL&            aCameraMatrix,
    TcVnArray8_LREAL&               aDistortionCoefficients,
    TcVnMatrix3x3_LREAL&            aRotationMatrix,
    TcVnVector3_LREAL&              aTranslationVector,
    double&                         fReprojError,
    TcVnCameraCalibrationOptions&   stCalibrationOptions,
    ETcVnEdgeDirection              eEdgeDirection,
    float                           fMinStrength = 100,
    ULONG                           nMaxSearchLines = 11,
    ULONG                           nMaxThickness = 5,
    ULONG                           nSubpixelsIterations = 50,
    float                           fApproxPrecision = 0.001f,
    ETcVnEdgeDetectionAlgorithm     eAlgorithm = EDA_APPROX_ERF
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
pSrcImages	PVOID		Pointer to an array of 1 or more source images (1 channel). If the images have more than 1 row, each row must contain an image of the same pattern position.
nSrcArraySize	USHORT		pSrcImages array size
ipPatternLineOrigins	ITcVnContainer* [▶ 1760]		X position of the pattern line origins (ContainerType_Vector_REAL, usually same origin for vertical and following diagonal line)
fSlopeRad	double		Slope of the diagonal line in rad (> 0, < PI/2), relative to the vertical line (i.e., 0 would be a vertical line, PI/2 a horizontal line)
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&		Returns the camera matrix
aDistortionCoefficients	TcVnArray8_LREAL [▶ 1590]&		Returns the lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&		Returns the rotation matrix
aTranslationVector	TcVnVector3_LREAL [▶ 1590]&		Returns the translation vector
fReprojError	double&		Returns the reprojection error
stCalibrationOptions	TcVnCameraCalibrationOptions [▶ 1639]&		Calibration options (recommended to set bFixAspectRatio, bFixPrincipalPoint, bZeroTangentDist)
eEdgeDirection	ETcVnEdgeDirection [▶ 1615]		Specification of the edge direction to search for
fMinStrength	float	100	Specification of the minimum strength (intensity difference) of the edge to search for

Name	Type	Default	Description
nMaxSearchLines	ULONG	11	Maximum number of search lines (equally distributed over the image height, at most 1 searchline per image row)
nMaxThickness	ULONG	5	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels
nSubpixelsIterations	ULONG	50	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)
fApproxPrecision	float	0.001f	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▶ 1614]	EDA_APPROX_ERF	Selection of the edge detection algorithm

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.6 CompensateLensDistortion

Transforms an image to compensate the lens distortion.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT CompensateLensDistortion(
    HRESULT                hrPrev,
    ITcVnImage*            ipSrcImage,
    ITcVnImage*&           ipDestImage,
    TcVnMatrix3x3_LREAL&   aCameraMatrix,
    TcVnArray4_LREAL&      aDistortionCoefficients
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]	Source image
ipDestImage	ITcVnImage* [▸ 1797]&	Destination image (An appropriate destination image will be created if required.)
aCameraMatrix	TcVnMatrix3x3_LREAL [▸ 1590]&	Camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray4_LREAL [▸ 1590]&	Lens distortion coefficients [k1, k2, p1, p2]

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.7 CompensateLensDistortion (Exp1)

Transforms an image to compensate the lens distortion.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT CompensateLensDistortion(
    HRESULT                hrPrev,
    ITcVnImage*            ipSrcImage,
    ITcVnImage*&           ipDestImage,
    TcVnMatrix3x3_LREAL&   aCameraMatrix,
    TcVnArray8_LREAL&      aDistortionCoefficients
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	Camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray8_LREAL [▶ 1590]&	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.8 CompensateLensDistortion (Exp2)

Transforms an image to compensate the lens distortion.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT CompensateLensDistortion(
    HRESULT               hrPrev,
    ITcVnImage*           ipSrcImage,
    ITcVnImage*&          ipDestImage,
    TcVnMatrix3x3_LREAL&  aCameraMatrix,
    TcVnArray8_LREAL&     aDistortionCoefficients,
    TcVnMatrix3x3_LREAL&  aNewCameraMatrix
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	Camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray8_LREAL [▶ 1590]&	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aNewCameraMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	Allows additional scaling and shifting of the result image

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.9 CompensateLensDistortionForPoints

Transforms point coordinates to compensate the lens distortion.

Syntax

Definition:

```
HRESULT CompensateLensDistortionForPoints(
    HRESULT                hrPrev,
    ITcVnContainer*        ipSrcPoints,
    ITcVnContainer*&       ipDestPoints,
    TcVnMatrix3x3_LREAL& aCameraMatrix,
    TcVnArray4_LREAL&    aDistortionCoefficients
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcPoints	ITcVnContainer* [▶ 1760]	Container with source point coordinates (ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_DINT)
ipDestPoints	ITcVnContainer* [▶ 1760]&	Returns a container with the transformed point coordinates (ContainerType_Vector_TcVnPoint2_REAL)
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	Camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray4_LREAL [▶ 1590]&	Lens distortion coefficients [k1, k2, p1, p2]

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.10 CompensateLensDistortionForPoints (Exp1)

Transforms point coordinates to compensate the lens distortion.

Syntax

Definition:

```
HRESULT CompensateLensDistortionForPoints(
    HRESULT              hrPrev,
    ITcVnContainer*      ipSrcPoints,
    ITcVnContainer*&     ipDestPoints,
    TcVnMatrix3x3_LREAL& aCameraMatrix,
    TcVnArray8_LREAL&    aDistortionCoefficients
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcPoints	ITcVnContainer* [▶ 1760]	Container with source point coordinates (ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_DINT)
ipDestPoints	ITcVnContainer* [▶ 1760]&	Returns a container with the transformed point coordinates (ContainerType_Vector_TcVnPoint2_REAL)
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	Camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray8_LREAL [▶ 1590]&	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.11 CompensateLensDistortionForPoints (Exp2)

Transforms point coordinates to compensate the lens distortion.

Syntax

Definition:

```
HRESULT CompensateLensDistortionForPoints(
    HRESULT              hrPrev,
    ITcVnContainer*      ipSrcPoints,
    ITcVnContainer*&     ipDestPoints,
    TcVnMatrix3x3_LREAL& aCameraMatrix,
    TcVnArray8_LREAL&    aDistortionCoefficients,
    TcVnMatrix3x3_LREAL& aNewCameraMatrix
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcPoints	ITcVnContainer* [▶ 1760]	Container with source point coordinates (ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_DINT)
ipDestPoints	ITcVnContainer* [▶ 1760]&	Returns a container with the transformed point coordinates (ContainerType_Vector_TcVnPoint2_REAL)
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	Camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray8_LREAL [▶ 1590]&	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aNewCameraMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	Allows additional shifting of the result point coordinates

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.12 DetectPatternPoints

Detects calibration pattern points (circles) within the provided image.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT DetectPatternPoints(
    HRESULT                      hrPrev,
    ITcVnImage*                  ipSrcImage,
    ITcVnContainer*&             ipPatternPoints,
    TcVnParamsBlobDetection&     stBlobDetectionParams,
    ULONG                        nNumberOfPoints,
    float                        fSizeDiffThresFactor = 0.3f,
    bool                         bSubpixelAccuracy = false,
    ETcVnEdgeDirection           eEdgeDirection = ED_DARK_TO_LIGHT,
    float                        fMinStrength = 50,
    ULONG                        nMaxThickness = 10,
    ULONG                        nSubpixelsIterations = 10,
    ULONG                        nSearchLines = 92,
    float                        fApproxPrecision = 0.001f,
    ETcVnEdgeDetectionAlgorithm eAlgorithm = EDA_INTERPOLATION
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (USINT, 1 channel)
ipPatternPoints	ITcVnContainer* [▶ 1760]&		Returns the pattern point positions (ContainerType_Vector_TcVnPoint2_REAL; Non-zero interface pointers are reused.)
stBlobDetectionParams	TcVnParamsBlobDetection [▶ 1646]&		Parameters to detect and filter contours.
nNumberOfPoints	ULONG		Expected number of pattern points
fSizeDiffThresFactor	float	0.3f	Threshold for filtering contours by average size
bSubpixelAccuracy	bool	false	If true, the pattern points are detected with subpixel accuracy
eEdgeDirection	ETcVnEdgeDirection [▶ 1615]	ED_DARK_TO_LIGHT	Specification of the edge direction to search for (from center to outside circle)
fMinStrength	float	50	Specification of the minimum strength (intensity difference) of the edge to search for
nMaxThickness	ULONG	10	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels
nSubpixelsIterations	ULONG	10	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)
nSearchLines	ULONG	92	Specifies the amount of search lines, which are equally distributed in all directions (must be >= 8 and a multiple of 4)

Name	Type	Default	Description
fApproxPrecision	float	0.001f	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▶ 1614]	EDA_INTERPOLATION	Selection of the edge detection algorithm

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.13 GenerateCalibrationPatternReferencePoints

Generate reference points for common calibration patterns.

Syntax

Definition:

```
HRESULT GenerateCalibrationPatternReferencePoints(
    HRESULT                        hrPrev,
    ITcVnContainer*&               ipReferencePoints,
    ETcVnCalibrationPattern        ePattern,
    ETcVnCalibrationPatternOrigin  eOrigin,
    ULONG                          nWidth,
    ULONG                          nHeight,
    float                          fDistX,
    float                          fDistY
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipReferencePoints	ITcVnContainer* [▶ 1760]&	Returns the generated reference points (ContainerType_Vector_TcVnPoint3_REAL)
ePattern	ETcVnCalibrationPattern [▶ 1594]	Calibration pattern type
eOrigin	ETcVnCalibrationPatternOrigin [▶ 1595]	Calibration pattern origin
nWidth	ULONG	Pattern width, i.e. number of pattern points in x direction
nHeight	ULONG	Pattern height, i.e. number of pattern points in y direction
fDistX	float	Distance between pattern points in x direction
fDistY	float	Distance between pattern points in y direction

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.14 ImagePointsWorldDistance

Computes the distance in the world coordinate system between two points (in the same world coordinate z layer) given in image coordinates.

Syntax

Definition:

```
HRESULT ImagePointsWorldDistance(
    HRESULT               hrPrev,
    TcVnPoint2_REAL&      aImagePoint1,
    TcVnPoint2_REAL&      aImagePoint2,
    double&               fWorldDistance,
    TcVnMatrix3x3_LREAL& aCameraMatrix,
    TcVnArray8_LREAL&     aDistortionCoefficients,
    TcVnMatrix3x3_LREAL& aRotationMatrix,
    TcVnVector3_LREAL&    aTranslationVector,
    double                fZ
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aImagePoint1	TcVnPoint2_REAL [▶ 1588]&	First point in the image coordinate system
aImagePoint2	TcVnPoint2_REAL [▶ 1588]&	Second point in the image coordinate system
fWorldDistance	double&	Returns the distance between aImagePoint1 and aImagePoint2 in the world coordinate system
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	3x3 camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray8_LREAL [▶ 1590]&	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	3x3 rotation matrix
aTranslationVector	TcVnVector3_LREAL [▶ 1590]&	Translation vector
fZ	double	z coordinate (world coordinate system) of the given points (0 would be at (toplevel) calibration pattern)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.15 SolvePnP

Computes the rotation matrix and translation vector from 3D-2D point correspondences.

Syntax

Definition:

```
HRESULT SolvePnP(
    HRESULT                hrPrev,
    ITcVnContainer*        ipImagePoints,
    ITcVnContainer*        ipReferencePoints,
    TcVnMatrix3x3_LREAL& aCameraMatrix,
    TcVnArray8_LREAL&      aDistortionCoefficients,
    TcVnMatrix3x3_LREAL& aRotationMatrix,
    TcVnVector3_LREAL&     aTranslationVector,
    double&                fReprojError,
    ETcVnSolvePnPMethod    eMethod = SPM_ITERATIVE,
    bool                   bUseExtrinsicGuess = false
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImagePoints	ITcVnContainer* [▶ 1760]		Image points (ContainerType_Vector_TcVnPoint2_REAL, requires at least 4 points in the same order as in ipReferencePoints)
ipReferencePoints	ITcVnContainer* [▶ 1760]		Reference world points (ContainerType_Vector_TcVnPoint3_REAL, same number of points as in ipImagePoints)
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&		Camera matrix (e.g. computed with one of the CalibrateCamera functions)
aDistortionCoefficients	TcVnArray8_LREAL [▶ 1590]&		Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6] (e.g. computed with one of the CalibrateCamera functions)
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&		Returns the rotation matrix
aTranslationVector	TcVnVector3_LREAL [▶ 1590]&		Returns the translation vector
fReprojError	double&		Returns the reprojection error
eMethod	ETcVnSolvePnPMethod [▶ 1630]	SPM_ITERATIVE	Solve PnP method
bUseExtrinsicGuess	bool	false	If true, the provided rotation matrix and translation vector are used as an initial guess for further optimization (only supported for the SPM_ITERATIVE method).

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.16 SortDetectedPatternPoints

Sort the detected pattern points according to the relative positions of the calibration pattern points.

Syntax

Definition:

```
HRESULT SortDetectedPatternPoints(
    HRESULT         hrPrev,
    ITcVnContainer* ipImagePoints,
    ITcVnContainer* ipPatternPoints
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImagePoints	ITcVnContainer* [▶ 1760]	Detected pattern point positions in the image coordinate system (will be sorted by this function, ContainerType_Vector_TcVnPoint2_REAL)
ipPatternPoints	ITcVnContainer* [▶ 1760]	The pattern points in the calibration pattern coordinate system, used as a reference to sort ipImagePoints (ContainerType_Vector_TcVnPoint3_REAL)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.17 TransformCoordinatesImageToWorld (Container)

Transform 2D image point coordinates to 3D world coordinates (using intrinsic + extrinsic calibration results).

Syntax

Definition:

```
HRESULT TransformCoordinatesImageToWorld(
    HRESULT                hrPrev,
    ITcVnContainer*        ipSrcPoints2D,
    ITcVnContainer*&       ipDestPoints3D,
    TcVnMatrix3x3_LREAL& aCameraMatrix,
    TcVnArray8_LREAL&    aDistortionCoefficients,
    TcVnMatrix3x3_LREAL& aRotationMatrix,
    TcVnVector3_LREAL&   aTranslationVector,
    double               fZ
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcPoints2D	ITcVnContainer* [▸ 1760]	Container with 2D source points (TcVnPoint2_DINT or TcVnPoint2_REAL or TcVnPoint2_LREAL)
ipDestPoints3D	ITcVnContainer* [▸ 1760]&	Returns the transformed 3D points (TcVnPoint3_LREAL)
aCameraMatrix	TcVnMatrix3x3_LREAL [▸ 1590]&	3x3 camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray8_LREAL [▸ 1590]&	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▸ 1590]&	3x3 rotation matrix
aTranslationVector	TcVnVector3_LREAL [▸ 1590]&	Translation vector
fZ	double	z coordinate (world coordinate system) of the transformed points (0 would be at (toplevel) calibration pattern)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.18 TransformCoordinatesImageToWorld (Point)

Transform 2D image point coordinate to 3D world coordinate (using intrinsic + extrinsic calibration results).

Syntax

Definition:

```
HRESULT TransformCoordinatesImageToWorld(
    HRESULT                hrPrev,
    TcVnPoint2_LREAL&      aSrcPoint,
    TcVnPoint3_LREAL&      aDestPoint,
    TcVnMatrix3x3_LREAL& aCameraMatrix,
    TcVnArray8_LREAL&      aDistortionCoefficients,
    TcVnMatrix3x3_LREAL& aRotationMatrix,
    TcVnVector3_LREAL&     aTranslationVector,
    double                 fZ
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aSrcPoint	TcVnPoint2_LREAL [▶ 1588]&	Point in the image coordinate system
aDestPoint	TcVnPoint3_LREAL [▶ 1588]&	Returns the point in the world coordinate system
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	3x3 camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray8_LREAL [▶ 1590]&	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	3x3 rotation matrix
aTranslationVector	TcVnVector3_LREAL [▶ 1590]&	Translation vector
fZ	double	z coordinate (world coordinate system) of the transformed points (0 would be at (toplevel) calibration pattern)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.19 TransformCoordinatesPlanar (Container)

Compute real-world coordinates for given image points or vice versa.

Syntax

Definition:

```
HRESULT TransformCoordinatesPlanar(
    HRESULT                hrPrev,
    ITcVnContainer*        ipSrcPoints,
    ITcVnContainer*&       ipDestPoints,
    TcVnMatrix3x3_LREAL& aTransformationMatrix
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcPoints	ITcVnContainer* [▶ 1760]	Container with 2D source points (TcVnPoint2_DINT or TcVnPoint2_REAL or TcVnPoint2_LREAL)
ipDestPoints	ITcVnContainer* [▶ 1760]&	Returns the transformed 2D points (TcVnPoint2_LREAL)
aTransformationMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	Transformation matrix obtained with VnCalibrateCameraPlanar() or its inverse matrix obtained with VnInvertMatrix3x3()

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.20 TransformCoordinatesPlanar (Point)

Compute real-world coordinates for a given image point or vice versa.

Syntax

Definition:

```
HRESULT TransformCoordinatesPlanar(
    HRESULT              hrPrev,
    TcVnPoint2_LREAL&    aSrcPoint,
    TcVnPoint2_LREAL&    aDestPoint,
    TcVnMatrix3x3_LREAL& aTransformationMatrix
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aSrcPoint	TcVnPoint2_LREAL [▶ 1588]&	Point in the source coordinate system
aDestPoint	TcVnPoint2_LREAL [▶ 1588]&	Returns point in the destination coordinate system
aTransformationMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	Transformation matrix obtained with VnCalibrateCameraPlanar() or its inverse matrix obtained with VnInvertMatrix3x3()

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.21 TransformCoordinatesWorldToImage (Container)

Transform 3D world coordinates to 2D image coordinates (using intrinsic + extrinsic calibration results).

Syntax

Definition:

```
HRESULT TransformCoordinatesWorldToImage(
    HRESULT              hrPrev,
    ITcVnContainer*      ipSrcPoints3D,
    ITcVnContainer*&     ipDestPoints2D,
    TcVnMatrix3x3_LREAL& aCameraMatrix,
    TcVnArray8_LREAL&    aDistortionCoefficients,
    TcVnMatrix3x3_LREAL& aRotationMatrix,
    TcVnVector3_LREAL&   aTranslationVector,
    bool                 bInteger
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcPoints3D	ITcVnContainer* [▶ 1760]	Container with 3D source points (TcVnPoint3_LREAL)
ipDestPoints2D	ITcVnContainer* [▶ 1760]&	Returns the transformed 2D points (TcVnPoint2_REAL or TcVnPoint2_DINT, depending on bInteger)
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	3x3 camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray8_LREAL [▶ 1590]&	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	3x3 rotation matrix
aTranslationVector	TcVnVector3_LREAL [▶ 1590]&	Translation vector
bInteger	bool	Selects whether the function returns a container of TcVnPoint2_DINT or TcVnPoint2_REAL

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.1.22 TransformCoordinatesWorldToImage (Point)

Transform 3D world coordinates to 2D image coordinates (using intrinsic + extrinsic calibration results).

Syntax

Definition:

```
HRESULT TransformCoordinatesWorldToImage(
    HRESULT             hrPrev,
    TcVnPoint3_LREAL&   aSrcPoint,
    TcVnPoint2_REAL&    aDestPoint,
    TcVnMatrix3x3_LREAL& aCameraMatrix,
    TcVnArray8_LREAL&   aDistortionCoefficients,
    TcVnMatrix3x3_LREAL& aRotationMatrix,
    TcVnVector3_LREAL&  aTranslationVector
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aSrcPoint	TcVnPoint3_LREAL [▶ 1588]&	3D source point (world coordinate system)
aDestPoint	TcVnPoint2_REAL [▶ 1588]&	Returns the transformed 2D point (image coordinate system)
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	3x3 camera matrix containing the intrinsic parameters
aDistortionCoefficients	TcVnArray8_LREAL [▶ 1590]&	Lens distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	3x3 rotation matrix
aTranslationVector	TcVnVector3_LREAL [▶ 1590]&	Translation vector

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.2 AlignRotatedImageRegion

Copies the provided image region and aligns it to the image axes.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT AlignRotatedImageRegion(
    HRESULT                        hrPrev,
    ITcVnImage*                    ipSrcImage,
    ITcVnImage*&                   ipDestImage,
    TcVnRotatedRectangle&          stRegion,
    ETcVnInterpolationType         eInterpolationType = IT_BILINEAR,
    ETcVnBorderInterpolationMethod eBorderInterpolation = BIM_CONSTANT,
    TcVnVector4_LREAL&             aBorderValue = {0, 0, 0, 0}
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (An appropriate image will be created if required.)
stRegion	TcVnRotatedRectangle [▶ 1658]&		Image region to be aligned
eInterpolationType	ETcVnInterpolationType [▶ 1621]	IT_BILINEAR	Interpolation method
eBorderInterpolation	ETcVnBorderInterpolationMethod [▶ 1593]	BIM_CONSTANT	Border interpolation method (ISOLATED not supported)
aBorderValue	TcVnVector4_LREAL [▶ 1590]&	{0, 0, 0, 0}	Border value, if CONSTANT is used

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.3 ApplyRotationToAffineTransformation

Apply a rotation to an existing 2D affine transformation matrix.

Syntax

Definition:

```
HRESULT ApplyRotationToAffineTransformation(
    HRESULT              hrPrev,
    TcVnMatrix2x3_LREAL& aTransformationMatrix,
    double               fAngle,
    bool                 bUsePreMultiplication = true
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 1590]&		Affine transformation matrix to which the rotation is applied
fAngle	double		Angle in radians (positive means counter-clockwise)
bUsePreMultiplication	bool	true	Select if pre- or post-multiplication should be used

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.4 ApplyScalingToAffineTransformation

Apply a scaling to an existing 2D affine transformation matrix.

Syntax

Definition:

```
HRESULT ApplyScalingToAffineTransformation(
    HRESULT              hrPrev,
    TcVnMatrix2x3_LREAL& aTransformationMatrix,
    double               fScaleX,
    double               fScaleY,
    bool                 bUsePreMultiplication = true
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 1590]&		Affine transformation matrix to which the scaling is applied
fScaleX	double		Scaling factor in x direction
fScaleY	double		Scaling factor in y direction
bUsePreMultiplication	bool	true	Select if pre- or post-multiplication should be used

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.5 ApplyTranslationToAffineTransformation

Apply a translation to an existing 2D affine transformation matrix.

Syntax

Definition:

```
HRESULT ApplyTranslationToAffineTransformation(
    HRESULT              hrPrev,
    TcVnMatrix2x3_LREAL& aTransformationMatrix,
    double               fDeltaX,
    double               fDeltaY,
    bool                 bUsePreMultiplication = true
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 1590]&		Affine transformation matrix to which the translation is applied
fDeltaX	double		Translation in x direction
fDeltaY	double		Translation in y direction
bUsePreMultiplication	bool	true	Select if pre- or post-multiplication should be used

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.6 ApplyYAxisInversionToAffineTransformation

Apply an inversion of the y-axis direction to an existing 2D affine transformation matrix.

Syntax

Definition:

```
HRESULT ApplyYAxisInversionToAffineTransformation(
    HRESULT              hrPrev,
    TcVnMatrix2x3_LREAL& aTransformationMatrix,
    bool                 bUsePreMultiplication = true
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 1590]&		Affine transformation matrix to which the inversion of the y-axis direction is applied
bUsePreMultiplication	bool	true	Select if pre- or post-multiplication should be used

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.7 ConvertCartesianToPolarAngleImage

Converts cartesian coordinates (x, y) to polar angle.

Syntax

Definition:

```
HRESULT ConvertCartesianToPolarAngleImage(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImageX,
    ITcVnImage*   ipSrcImageY,
    ITcVnImage*&  ipDestImageAng,
    bool          bAngleInDegrees = false
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImageX	ITcVnImage* [▶ 1797]		Source image containing the x values (elements of type REAL or LREAL)
ipSrcImageY	ITcVnImage* [▶ 1797]		Source image containing the y values (Same element type as ipSrcImageX)
ipDestImageAng	ITcVnImage* [▶ 1797]&		Destination image containing the angles (Same element type as source images. An appropriate destination image will be created if required.)
bAngleInDegrees	bool	false	Specifys, if the angles should be in degrees or radians

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.8 ConvertCartesianToPolarAngles

Converts cartesian coordinates (x, y) to polar angle.

Syntax

Definition:

```
HRESULT ConvertCartesianToPolarAngles(
    HRESULT            hrPrev,
    ITcVnContainer*    ipSrcPointsCart,
    ITcVnContainer*&   ipDestAngles,
    bool               bAngleInDegrees = false
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcPointsCart	ITcVnContainer* [▶ 1760]		Container with cartesian points (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_LREAL)
ipDestAngles	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the angles (ContainerType_Vector_REAL, if source points are of type TcVnPoint2_DINT or TcVnPoint2_REAL; else ContainerType_Vector_LREAL. Non-zero interface pointers are reused.)
bAngleInDegrees	bool	false	Specify, if the angles should be in degrees or radians

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.9 ConvertCartesianToPolarImages

Converts cartesian coordinates (x, y) to polar coordinates (magnitude, angle).

Syntax

Definition:

```
HRESULT ConvertCartesianToPolarImages(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImageX,
    ITcVnImage*  ipSrcImageY,
    ITcVnImage*& ipDestImageMag,
    ITcVnImage*& ipDestImageAng,
    bool         bAngleInDegrees = false
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImageX	ITcVnImage* [▶ 1797]		Source image containing the x values (elements of type REAL or LREAL)
ipSrcImageY	ITcVnImage* [▶ 1797]		Source image containing the y values (Same element type as ipSrcImageX)
ipDestImageMag	ITcVnImage* [▶ 1797]&		Destination image containing the magnitudes (Same element type as source images. An appropriate destination image will be created if required.)
ipDestImageAng	ITcVnImage* [▶ 1797]&		Destination image containing the angles (Same element type as source images. An appropriate destination image will be created if required.)
bAngleInDegrees	bool	false	Specifys, if the angles should be in degrees or radians

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.10 ConvertCartesianToPolarMagnitudeImage

Converts cartesian coordinates (x, y) to polar magnitude.

Syntax

Definition:

```
HRESULT ConvertCartesianToPolarMagnitudeImage(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImageX,
```

```
    ITcVnImage*  ipSrcImageY,
    ITcVnImage*& ipDestImageMag
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImageX	ITcVnImage* [▶ 1797]	Source image containing the x values (elements of type REAL or LREAL)
ipSrcImageY	ITcVnImage* [▶ 1797]	Source image containing the y values (Same element type as ipSrcImageX)
ipDestImageMag	ITcVnImage* [▶ 1797]&	Destination image containing the magnitudes (Same element type as source images. An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.11 ConvertCartesianToPolarMagnitudes

Converts cartesian coordinates (x, y) to polar magnitude.

Syntax

Definition:

```
HRESULT ConvertCartesianToPolarMagnitudes(
    HRESULT          hrPrev,
    ITcVnContainer*  ipSrcPointsCart,
    ITcVnContainer*& ipDestMags
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcPointsCart	ITcVnContainer* [▶ 1760]	Container with cartesian points (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_LREAL)
ipDestMags	ITcVnContainer* [▶ 1760]&	Returns a container which is filled with the magnitudes (ContainerType_Vector_REAL, if source points are of type TcVnPoint2_DINT or TcVnPoint2_REAL; else ContainerType_Vector_LREAL. Non-zero interface pointers are reused.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.12 ConvertCartesianToPolarPoints

Converts cartesian coordinates (x, y) to polar coordinates (magnitude, angle).

Syntax

Definition:

```
HRESULT ConvertCartesianToPolarPoints(
    HRESULT           hrPrev,
    ITcVnContainer*   ipSrcPointsCart,
    ITcVnContainer*&  ipDestPointsPolar,
    bool              bAngleInDegrees = false
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcPointsCart	ITcVnContainer* [▶ 1760]		Container with cartesian points (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_LREAL)
ipDestPointsPolar	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the converted points in polar coordinates (ContainerType_Vector_TcVnPoint2_REAL, if source points are of type TcVnPoint2_DINT or TcVnPoint2_REAL; else ContainerType_Vector_TcVnPoint2_LREAL. Non-zero interface pointers are reused.)
bAngleInDegrees	bool	false	Specifys, if the angles should be in degrees or radians

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.13 ConvertPolarToCartesianImages

Converts polar coordinates (magnitude, angle) to cartesian coordinates (x, y).

Syntax

Definition:

```
HRESULT ConvertPolarToCartesianImages(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImageMag,
    ITcVnImage*  ipSrcImageAng,
    ITcVnImage*& ipDestImageX,
```

```
    ITcVnImage*&  ipDestImageY,
    bool          bAngleInDegrees = false
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImageMag	ITcVnImage* [▶ 1797]		Source image containing the magnitudes (elements of type REAL or LREAL)
ipSrcImageAng	ITcVnImage* [▶ 1797]		Source image containing the angles (Same element type as ipSrcImageMag)
ipDestImageX	ITcVnImage* [▶ 1797]&		Destination image containing the x values (Same element type as source images. An appropriate destination image will be created if required.)
ipDestImageY	ITcVnImage* [▶ 1797]&		Destination image containing the y values (Same element type as source images. An appropriate destination image will be created if required.)
bAngleInDegrees	bool	false	Specifys, if the angles are in degrees or radians

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.14 ConvertPolarToCartesianPoints

Converts polar coordinates (magnitude, angle) to cartesian coordinates (x, y).

Syntax

Definition:

```
HRESULT ConvertPolarToCartesianPoints(
    HRESULT           hrPrev,
    ITcVnContainer*   ipSrcPointsPolar,
    ITcVnContainer*&  ipDestPointsCart,
    bool              bAngleInDegrees = false
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcPointsPolar	ITcVnContainer* [▶ 1760]		Container with polar coordinates (ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_LREAL)
ipDestPointsCart	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the converted points in cartesian coordinates (same type ID as ipSrcPointsPolar. Non-zero interface pointers are reused.)
bAngleInDegrees	bool	false	Specifys, if the angles are in degrees or radians

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.15 DecomposeAffineTransformation

Decompose an affine transformation matrix and return the rotation angle (clockwise) in radian, translation vector (pixels), scale, and skew vectors (angle in radian). The decomposition method assumes the following order of transformation: Translation, Rotation, Scale, and Skew.

Syntax

Definition:

```
HRESULT DecomposeAffineTransformation(
    HRESULT              hrPrev,
    TcVnMatrix2x3_LREAL& aAffineTransform,
    double&              fRotationAngle,
    TcVnVector2_LREAL&   aTranslationVector = {0, 0},
```

```
    TcVnVector2_LREAL&    aScaleVector = {0, 0},
    TcVnVector2_LREAL&    aSkewVector = {0, 0}
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aAffineTransform	TcVnMatrix2x3_LREAL [▶ 1590]&		Affine transformation matrix
fRotationAngle	double&		Rotation angle (radian) in the clockwise direction. The rotation center is the origin point (0,0)
aTranslationVector	TcVnVector2_LREAL [▶ 1590]&	{0, 0}	2D translation vector. The values are in pixels
aScaleVector	TcVnVector2_LREAL [▶ 1590]&	{0, 0}	2D scale vector
aSkewVector	TcVnVector2_LREAL [▶ 1590]&	{0, 0}	2D skew vector. The values are the skew angles in x and y directions in radian

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.16 DecomposeHomography

Decompose a homography matrix and verify four solutions of rotation, translation, and plane normal. It uses visible reference points being in front of the camera to confirm which solution(s) of the four solutions (maximum two solutions) are consistent with all reference points.

Syntax

Definition:

```
HRESULT DecomposeHomography(
    HRESULT               hrPrev,
    TcVnMatrix3x3_LREAL& aHomography,
    TcVnMatrix3x3_LREAL& aCameraMatrix,
    TcVnMatrix3x3_LREAL& aRotationMatrixA,
    TcVnVector3_LREAL&    aTranslationVectorA,
    TcVnVector3_LREAL&    aNormVectorA,
    TcVnMatrix3x3_LREAL& aRotationMatrixB,
```

```
    TcVnVector3_LREAL&    aTranslationVectorB,
    TcVnVector3_LREAL&    aNormVectorB,
    ULONG&                nSolutions,
    ITcVnContainer*       ipPointsSrc = nullptr,
    ITcVnContainer*       ipPointsDes = nullptr,
    ITcVnContainer*       ipInlierMask = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aHomography	TcVnMatrix3x3_LREAL [▶ 1590]&		Homography matrix (a perspective transformation between two planes)
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&		Camera matrix
aRotationMatrixA	TcVnMatrix3x3_LREAL [▶ 1590]&		Rotation matrix of the first solution (Ra)
aTranslationVectorA	TcVnVector3_LREAL [▶ 1590]&		Translation vector of the first solution (Ta)
aNormVectorA	TcVnVector3_LREAL [▶ 1590]&		Norm vector of the first solution (Na)
aRotationMatrixB	TcVnMatrix3x3_LREAL [▶ 1590]&		Rotation matrix of the second solution (Rb)
aTranslationVectorB	TcVnVector3_LREAL [▶ 1590]&		Translation vector of the second solution (Tb)
aNormVectorB	TcVnVector3_LREAL [▶ 1590]&		Norm vector of the second solution (Nb)
nSolutions	ULONG&		Return the number of true potential solutions [0,1, or 2]. 0: no true potential solutions for the given corresponding points are confirmed. 1: the solution (Ra, Ta, Na) is confirmed. 2: both solution (Ra, Ta, Na) and (Rb, Tb, Nb) are confirmed. In all cases the function returns two solutions (Ra, Ta, Nb) and (Rb, Tb, Nb). The third and the fourth solutions can be calculated as follows: (Ra, -Ta, -Na) and (Rb, -Tb, -Nb).
ipPointsSrc	ITcVnContainer* [▶ 1760]	nullptr	Container with source points (ContainerType_Vector_TcVnPoint2_REAL)
ipPointsDes	ITcVnContainer* [▶ 1760]	nullptr	Container with destination points (same number as ipPoints1, ContainerType_Vector_TcVnPoint2_REAL)

Name	Type	Default	Description
ipInlierMask	ITcVnContainer* [▶ 1760]	nullptr	A mask marking the inliers (optional, set to 0 if not available; ContainerType_Vector_SINT or ContainerType_Vector_USINT)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.17 FlipImage

Flip an image (mirror and shift back to prior coordinates).

Syntax

Definition:

```
HRESULT FlipImage(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage,
    ITcVnImage*&  ipDestImage,
    ETcVnFlipAxis eFlipAxis
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
eFlipAxis	ETcVnFlipAxis [▶ 1619]	Selects the axis around which to flip the image

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.18 GenerateAffineTransformationUnitMatrix2D

Generate an affine transformation 2D unit matrix.

Syntax

Definition:

```
HRESULT GenerateAffineTransformationUnitMatrix2D(
    HRESULT              hrPrev,
    TcVnMatrix2x3_LREAL& aTransformationMatrix
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 1590]&	Resulting affine transformation unit matrix

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.19 GetAffineTransformation

Calculate the affine transformation between three corresponding point pairs. The points mark the corners of the corresponding triangles.

Syntax

Definition:

```
HRESULT GetAffineTransformation(
    HRESULT                 hrPrev,
    TcVnArray3_Point2_REAL& aSrcPoints,
    TcVnArray3_Point2_REAL& aDestPoints,
    TcVnMatrix2x3_LREAL&    aTransformationMatrix
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aSrcPoints	TcVnArray3_Point2_REAL [▶ 1590]&	Source points
aDestPoints	TcVnArray3_Point2_REAL [▶ 1590]&	Destination points
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 1590]&	Resulting transformation matrix

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.20 GetAffineTransformation2D

Estimate the affine transformation between two planar point sets.

Syntax

Definition:

```
HRESULT GetAffineTransformation2D(
    HRESULT                  hrPrev,
    ITcVnContainer*          ipSrcPoints,
    ITcVnContainer*          ipDestPoints,
    TcVnMatrix2x3_LREAL&     aAffineTransform,
    ETcVnEstimationAlgorithm eAlgorithm = EA_RANSAC,
    double                   fReprojThreshold = 3,
    ITcVnContainer**         pipInlierMask = nullptr,
    ULONG                    nMaxIterations = 2000,
    double                   fConfidence = 0.995,
    ULONG                    nRefineIters = 10
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcPoints	ITcVnContainer* [▶ 1760]		Container with at least 3 source points (ContainerType_Vector_TcVnPoint2_REAL)
ipDestPoints	ITcVnContainer* [▶ 1760]		Container with destination points (same number as ipSrcPoints, ContainerType_Vector_TcVnPoint2_REAL)
aAffineTransform	TcVnMatrix2x3_LREAL [▶ 1590]&		Returns the affine transformation matrix, which transforms the source points to the destination points
eAlgorithm	ETcVnEstimationAlgorithm [▶ 1616]	EA_RANSAC	Estimation algorithm (only RANSAC and LMEDS are supported)
fReprojThreshold	double	3	Maximum allowed reprojection error to treat a point pair as an inlier
pipInlierMask	ITcVnContainer* [▶ 1760]*	nullptr	Returns a mask marking the inliers (optional, set to 0 if not required; ContainerType_Vector_USINT)
nMaxIterations	ULONG	2000	Maximum number of iterations
fConfidence	double	0.995	Confidence (0..1)
nRefineIters	ULONG	10	Maximum number of iterations of Levenberg-Marquardt algorithm to refine further the affine transform (using only inliers). Set to 0 to disable refining.

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.21 GetPerspectiveTransformation

Calculate the perspective transformation between four corresponding point pairs. The points mark the corners of the corresponding rectangles.

Syntax

Definition:

```
HRESULT GetPerspectiveTransformation(
    HRESULT                hrPrev,
    TcVnArray4_Point2_REAL& aSrcPoints,
    TcVnArray4_Point2_REAL& aDestPoints,
    TcVnMatrix3x3_LREAL&    aTransformationMatrix
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aSrcPoints	TcVnArray4_Point2_REAL [▶ 1590]&	Source points
aDestPoints	TcVnArray4_Point2_REAL [▶ 1590]&	Destination points
aTransformationMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	Resulting transformation matrix

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.22 Homography

Find the homography (perspective transformation) between two planar point sets.

Syntax

Definition:

```
HRESULT Homography(
    HRESULT                  hrPrev,
    ITcVnContainer*          ipSrcPoints,
    ITcVnContainer*          ipDestPoints,
    TcVnMatrix3x3_LREAL&     aPerspectiveTransform,
    ETcVnEstimationAlgorithm eAlgorithm = EA_DEFAULT,
```

```
    double                  fReprojThreshold = 3,
    ITcVnContainer**        pipInlierMask = nullptr,
    ULONG                   nMaxIterations = 2000,
    double                  fConfidence = 0.995
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcPoints	ITcVnContainer* [▶ 1760]		Container with at least 4 source points (ContainerType_Vector_TcVnPoint2_REAL)
ipDestPoints	ITcVnContainer* [▶ 1760]		Container with destination points (same number as ipSrcPoints, ContainerType_Vector_TcVnPoint2_REAL)
aPerspectiveTransform	TcVnMatrix3x3_LREAL [▶ 1590]&		Returns the perspective transformation matrix, which transforms the source points to the destination points
eAlgorithm	ETcVnEstimationAlgorithm [▶ 1616]	EA_DEFAULT	Estimation algorithm
fReprojThreshold	double	3	Maximum allowed reprojection error to treat a point pair as an inlier (only for RANSAC, RHO)
pipInlierMask	ITcVnContainer* [▶ 1760]*	nullptr	Returns a mask marking the inliers (optional, set to 0 if not required; ContainerType_Vector_SINT; only for RANSAC, LMEDS)
nMaxIterations	ULONG	2000	Maximum number of RANSAC iterations
fConfidence	double	0.995	Confidence (0..1)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.23 InvertAffineTransformation

Invert a 2D affine transformation matrix. The inverted transformation compensates the affine transformation on an image if it is used in the F_VN_WarpAffine function.

Syntax

Definition:

```
HRESULT InvertAffineTransformation(
    HRESULT               hrPrev,
    TcVnMatrix2x3_LREAL& aTransformationMatrix,
    TcVnMatrix2x3_LREAL& aInvertedTransformationMatrix
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 1590]&	Source affine transformation matrix
aInvertedTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 1590]&	Resulting inverted affine transformation matrix

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.24 PerspectiveTransformation

Apply a perspective transformation (rotation + translation, e.g. from extrinsic calibration) to 3D point coordinates.

Syntax

Definition:

```
HRESULT PerspectiveTransformation(
    HRESULT               hrPrev,
    ITcVnContainer*       ipSrcPoints,
    ITcVnContainer*&      ipDestPoints,
    TcVnMatrix3x3_LREAL& aRotationMatrix,
    TcVnVector3_LREAL&   aTranslationVector
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcPoints	ITcVnContainer* [▶ 1760]	Container with 3D source points (TcVnPoint3_REAL or TcVnPoint3_LREAL)
ipDestPoints	ITcVnContainer* [▶ 1760]&	Returns the transformed points (same type as ipSrcPoints)
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	3x3 rotation matrix
aTranslationVector	TcVnVector3_LREAL [▶ 1590]&	Translation vector

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.25 PyramidDown

Downsamples an image to half width and height.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT PyramidDown(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.26 PyramidUp

Upsamples an image to double width and height.

Syntax

Definition:

```
HRESULT PyramidUp(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.27 RemapImageToLogPolarSpace

Remap an image to log-polar space.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT RemapImageToLogPolarSpace(
    HRESULT                 hrPrev,
    ITcVnImage*             ipSrcImage,
    ITcVnImage*&            ipDestImage,
```

```
    TcVnPoint2_REAL&         aCenter,
    double                   fScale = 0,
    ETcVnInterpolationType eInterpolationType = IT_BILINEAR,
    LONG                     nDestWidth = -1,
    LONG                     nDestHeight = -1
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]		Source image
ipDestImage	ITcVnImage* [▸ 1797]&		Destination image (Must not be the same as ipSrcImage! An appropriate destination image will be created if required.)
aCenter	TcVnPoint2_REAL [▸ 1588]&		Center point for the transformation
fScale	double	0	Magnitude scale parameter, i.e. image width / ln(radius) (set to 0 for auto select)
eInterpolationType	ETcVnInterpolationType [▸ 1621]	IT_BILINEAR	Interpolation type
nDestWidth	LONG	-1	Destination image width (set to -1 for source image width, 0 to auto scale to transformed radius, > 0 for a user defined width)
nDestHeight	LONG	-1	destination image height (set to -1 for source image height, 0 to auto scale to transformed radius * PI, > 0 for a user defined height)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.28 RemapImageToPolarSpace

Remap an image to polar space.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT RemapImageToPolarSpace(
    HRESULT                 hrPrev,
    ITcVnImage*             ipSrcImage,
    ITcVnImage*&            ipDestImage,
    TcVnPoint2_REAL&        aCenter,
    double                  fMaxRadius = 0,
    ETcVnInterpolationType eInterpolationType = IT_BILINEAR,
    LONG                    nDestWidth = -1,
    LONG                    nDestHeight = -1
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]		Source image
ipDestImage	ITcVnImage* [▸ 1797]&		Destination image (Must not be the same as ipSrcImage! An appropriate destination image will be created if required.)
aCenter	TcVnPoint2_REAL [▸ 1588]&		Center point for the transformation
fMaxRadius	double	0	Maximum radius for the transformation (set to 0 for auto select)
eInterpolationType	ETcVnInterpolationType [▸ 1621]	IT_BILINEAR	Interpolation type
nDestWidth	LONG	-1	Destination image width (set to -1 for source image width, 0 to auto scale to transformed radius, > 0 for a user defined width)
nDestHeight	LONG	-1	destination image height (set to -1 for source image height, 0 to auto scale to transformed radius * PI, > 0 for a user defined height)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.29 ResizeImage

Resize an image using a specific interpolation type while maintaining its aspect ratio.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT ResizeImage(
    HRESULT                 hrPrev,
    ITcVnImage*             ipSrcImage,
    ITcVnImage*&            ipDestImage,
    ULONG                   nWidth,
    ULONG                   nHeight,
    ETcVnInterpolationType  eInterpolationType,
    ETcVnPaddingMode        ePaddingMode = PM_NONE,
    TcVnVector4_LREAL&      aBorderValue = {0, 0, 0, 0}
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (An appropriate image will be created if required.)
nWidth	ULONG		New width
nHeight	ULONG		New height
eInterpolationType	ETcVnInterpolationType [▶ 1621]		Interpolation type
ePaddingMode	ETcVnPaddingMode [▶ 1627]	PM_NONE	Image padding mode
aBorderValue	TcVnVector4_LREAL [▶ 1590]&	{0, 0, 0, 0}	Border value, if TCVN_PM_LETTERBOX is used

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.30 RotateImage

Rotate an image by 90, 180, or 270 degrees in clockwise direction.

Syntax

Definition:

```
HRESULT RotateImage(
    HRESULT             hrPrev,
    ITcVnImage*         ipSrcImage,
    ITcVnImage*&        ipDestImage,
    ETcVnRotationAngle eAngle
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]	Source image
ipDestImage	ITcVnImage* [▸ 1797]&	Destination image (An appropriate image will be created if required.)
eAngle	ETcVnRotationAngle [▸ 1629]	Rotation angle

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.31 RotateImage (Exp)

Rotate an image by an individual rotation angle in degrees.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT RotateImage(
    HRESULT                         hrPrev,
    ITcVnImage*                     ipSrcImage,
    ITcVnImage*&                    ipDestImage,
    double                          fAngleDeg,
    bool                            bAdjustDestSize,
```

```
    ETcVnInterpolationType          eInterpolationType = IT_BILINEAR,
    ETcVnBorderInterpolationMethod eBorderInterpolation = BIM_CONSTANT,
    TcVnVector4_LREAL&              aBorderValue = {0, 0, 0, 0}
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (An appropriate image will be created if required.)
fAngleDeg	double		Rotation angle in degree (positive: counter-clockwise, negative: clockwise)
bAdjustDestSize	bool		If true, ipDestImage size is adjusted so that the whole rotated ipSrcImage is contained. Otherwise, ipDestImage size is set to ipSrcImage size.
eInterpolationType	ETcVnInterpolationType [▶ 1621]	IT_BILINEAR	Interpolation method
eBorderInterpolation	ETcVnBorderInterpolationMethod [▶ 1593]	BIM_CONSTANT	Border interpolation method (ISOLATED not supported)
aBorderValue	TcVnVector4_LREAL [▶ 1590]&	{0, 0, 0, 0}	Border value, if CONSTANT is used

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.32 WarpAffine

Apply an affine transformation to an image.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT WarpAffine(
    HRESULT             hrPrev,
    ITcVnImage*         ipSrcImage,
    ITcVnImage*&        ipDestImage,
    TcVnMatrix2x3_LREAL& aTransformationMatrix
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]	Source image
ipDestImage	ITcVnImage* [▸ 1797]&	Destination image (An appropriate image will be created if required.)
aTransformationMatrix	TcVnMatrix2x3_LREAL [▸ 1590]&	Affine transformation matrix

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.33 WarpAffine (Container)

Apply an affine transformation to a container of 2D points.

Syntax

Definition:

```
HRESULT WarpAffine(
    HRESULT             hrPrev,
    ITcVnContainer*     ipSrcPoints,
    ITcVnContainer*&    ipDestPoints,
    TcVnMatrix2x3_LREAL& aTransformationMatrix
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcPoints	ITcVnContainer* [▶ 1760]	Source points (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_LREAL)
ipDestPoints	ITcVnContainer* [▶ 1760]&	Returns the transformed points (same type ID as ipSrcPoints; Non-zero interface pointers are reused.)
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 1590]&	Affine transformation matrix

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.34 WarpAffine (Point)

Apply an affine transformation to a 2D point.

Syntax

Definition:

```
HRESULT WarpAffine(
    HRESULT             hrPrev,
    TcVnPoint2_LREAL&    aSrcPoint,
    TcVnPoint2_LREAL&    aDestPoint,
    TcVnMatrix2x3_LREAL& aTransformationMatrix
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aSrcPoint	TcVnPoint2_LREAL [▶ 1588]&	Source point
aDestPoint	TcVnPoint2_LREAL [▶ 1588]&	Destination point
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 1590]&	Affine transformation matrix

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.35 WarpAffine (Rectangle)

Use an affine transform to warp a rectangle.

Syntax

Definition:

```
HRESULT WarpAffine(
    HRESULT             hrPrev,
    ULONG               nTopLeftX,
    ULONG               nTopLeftY,
    ULONG               nBottomRightX,
    ULONG               nBottomRightY,
    ITcVnContainer*&    ipDestPoints,
    TcVnMatrix2x3_LREAL& aTransformationMatrix
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nTopLeftX	ULONG	x coordinate of the top left corner
nTopLeftY	ULONG	y coordinate of the top left corner
nBottomRightX	ULONG	x coordinate of the bottom right corner
nBottomRightY	ULONG	y coordinate of the bottom right corner
ipDestPoints	ITcVnContainer* [▶ 1760]&	Returns the 4 transformed points (ContainerType_Vector_TcVnPoint2_REAL)
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 1590]&	Affine transformation matrix

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.36 WarpAffine (Exp)

Apply an affine transformation to an image.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT WarpAffine(
    HRESULT                        hrPrev,
    ITcVnImage*                    ipSrcImage,
    ITcVnImage*&                   ipDestImage,
    TcVnMatrix2x3_LREAL&           aTransformationMatrix,
    ULONG                          nDestWidth,
    ULONG                          nDestHeight,
    ETcVnInterpolationType         eInterpolationType,
    bool                           bWarpInverse,
    ETcVnBorderInterpolationMethod eBorderInterpolation,
    double                         fBorderValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate image will be created if required.)
aTransformationMatrix	TcVnMatrix2x3_LREAL [▶ 1590]&	Affine transformation matrix
nDestWidth	ULONG	Width of the destination image
nDestHeight	ULONG	Height of the destination image
eInterpolationType	ETcVnInterpolationType [▶ 1621]	Interpolation method
bWarpInverse	bool	Handle aTransformationMatrix as the inverse transformation
eBorderInterpolation	ETcVnBorderInterpolationMethod [▶ 1593]	Border interpolation method (ISOLATED not supported)
fBorderValue	double	Border value, if CONSTANT is used

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.37 WarpPerspective

Apply a perspective transformation to an image.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT WarpPerspective(
    HRESULT               hrPrev,
    ITcVnImage*           ipSrcImage,
    ITcVnImage*&          ipDestImage,
    TcVnMatrix3x3_LREAL&  aTransformationMatrix
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]	Source image
ipDestImage	ITcVnImage* [▸ 1797]&	Destination image (An appropriate image will be created if required.)
aTransformationMatrix	TcVnMatrix3x3_LREAL [▸ 1590]&	Perspective transformation matrix

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.38 WarpPerspective (Container)

Apply a perspective transformation to a container of 2D points.

Syntax

Definition:

```
HRESULT WarpPerspective(
    HRESULT               hrPrev,
    ITcVnContainer*       ipSrcPoints,
```

```
    ITcVnContainer*&     ipDestPoints,
    TcVnMatrix3x3_LREAL& aTransformationMatrix
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcPoints	ITcVnContainer* [▶ 1760]	Source points (ContainerType_Vector_TcVnPoint2_DINT or ContainerType_Vector_TcVnPoint2_REAL or ContainerType_Vector_TcVnPoint2_LREAL)
ipDestPoints	ITcVnContainer* [▶ 1760]&	Returns the transformed points (same type ID as ipSrcPoints; Non-zero interface pointers are reused.)
aTransformationMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	Perspective transformation matrix

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.39 WarpPerspective (Point)

Apply a perspective transformation to a 2D point.

Syntax

Definition:

```
HRESULT WarpPerspective(
    HRESULT              hrPrev,
    TcVnPoint2_LREAL&    aSrcPoint,
    TcVnPoint2_LREAL&    aDestPoint,
    TcVnMatrix3x3_LREAL& aTransformationMatrix
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aSrcPoint	TcVnPoint2_LREAL [▶ 1588]&	Source point
aDestPoint	TcVnPoint2_LREAL [▶ 1588]&	Destination point
aTransformationMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	Perspective transformation matrix

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.40 WarpPerspective (Rectangle)

Use a perspective transform to warp a rectangle.

Syntax

Definition:

```
HRESULT WarpPerspective(
    HRESULT              hrPrev,
    ULONG                nTopLeftX,
    ULONG                nTopLeftY,
    ULONG                nBottomRightX,
    ULONG                nBottomRightY,
    ITcVnContainer*&     ipDestPoints,
    TcVnMatrix3x3_LREAL& aTransformationMatrix
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nTopLeftX	ULONG	x coordinate of the top left corner
nTopLeftY	ULONG	y coordinate of the top left corner
nBottomRightX	ULONG	x coordinate of the bottom right corner
nBottomRightY	ULONG	y coordinate of the bottom right corner
ipDestPoints	ITcVnContainer* [▸ 1760]&	Returns the 4 transformed points (ContainerType_Vector_TcVnPoint2_REAL)
aTransformationMatrix	TcVnMatrix3x3_LREAL [▸ 1590]&	Perspective transformation matrix

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.13.41 WarpPerspective (Exp)

Apply a perspective transformation to an image.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT WarpPerspective(
    HRESULT                         hrPrev,
    ITcVnImage*                     ipSrcImage,
    ITcVnImage*&                    ipDestImage,
    TcVnMatrix3x3_LREAL&            aTransformationMatrix,
    ULONG                           nDestWidth,
    ULONG                           nDestHeight,
    ETcVnInterpolationType          eInterpolationType,
    bool                            bWarpInverse,
    ETcVnBorderInterpolationMethod eBorderInterpolation,
    double                          fBorderValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate image will be created if required.)
aTransformationMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	Perspective transformation matrix
nDestWidth	ULONG	Width of the destination image
nDestHeight	ULONG	Height of the destination image
eInterpolationType	ETcVnInterpolationType [▶ 1621]	Interpolation method
bWarpInverse	bool	Handle aTransformationMatrix as the inverse transformation
eBorderInterpolation	ETcVnBorderInterpolationMethod [▶ 1593]	Border interpolation method (ISOLATED not supported)
fBorderValue	double	Border value, if CONSTANT is used

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14 Image Analysis

该组包含从图像中提取信息的函数。

函数

对象检测 [▶ 2343]

- 轮廓搜索
- 通过 Hough 变换查找对象
- 通过图像对比进行匹配

统计图像特征

- ImageAverage [▶ 2364]
- ImageAverageStdDev [▶ 2365]
- ImageMedian [▶ 2367]
- MaxPixelValue [▶ 2369]
- MinPixelValue [▶ 2370]

边缘检测

查找连续图像区域

区域对齐

其它

6.2.3.14.1 Object Detection

该子组包含对象检测函数。

函数

对象检测

匹配

其它

6.2.3.14.1.1 AdjustActiveContour

Adjust active contour (snake) in order to minimize its cummulative (internal and external) energy.

Syntax

Definition:

```
HRESULT AdjustActiveContour(
    HRESULT         hrPrev,
    ITcVnImage*     ipImage,
    ITcVnContainer* ipActiveContour,
    float           fAlpha,
    float           fBeta,
    float           fGamma,
    ULONG           nWindowWidth,
```

```
    ULONG           nWindowHeight,
    ULONG           nMaxIterations,
    bool            bUseGradient
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▸ 1797]	Source image (1 channel, USINT elements)
ipActiveContour	ITcVnContainer* [▸ 1760]	Initial contour, which will be adjusted
fAlpha	float	Continuity energy coefficient
fBeta	float	Curvature energy coefficient
fGamma	float	Image energy coefficient
nWindowWidth	ULONG	Window width (3, 5, 7, ...)
nWindowHeight	ULONG	Window height (3, 5, 7, ...)
nMaxIterations	ULONG	Maximum iterations
bUseGradient	bool	If true, the gradient magnitude is used as image energy (otherwise: pixel intensity)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.1.2 DetectBlobs

Detects blob-contours. Applies a threshold, a contour search and offers several options for filtering the found contours. Provides easy setup for multiple thresholds and combination of results.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT DetectBlobs(
    HRESULT                  hrPrev,
    ITcVnImage*              ipSrcImage,
    ITcVnContainer*&         ipBlobContours,
    TcVnParamsBlobDetection& stParams,
    TcVnPoint&               aOffset = {0, 0}
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (USINT or REAL, 1 channel or 3 channel (3 channel input is expected to be RGB and internally converted to Gray))
ipBlobContours	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the found contours (ContainerType_Vector_Vector_TcVnPoint2_DINT; The elements of this container are single contours. Non-zero interface pointers are reused.)
stParams	TcVnParamsBlobDetection [▶ 1646]&		Parameters to filter the detected contours.
aOffset	TcVnPoint [▶ 1588]&	{0, 0}	Offset by which every contour point is shifted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.1.3 FindContourHierarchy (Exp)

Search for object contours in a binary image and determine their hierarchical relationship.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT FindContourHierarchy(
    HRESULT                         hrPrev,
    ITcVnImage*                     ipSrcImage,
    ITcVnContainer*&                ipContours,
    ITcVnContainer*&                ipHierarchy,
    ETcVnContourRetrievalMode       eRetrievalMode,
```

```
    ETcVnContourApproximationMethod eApproximationMethod,
    TcVnPoint&                      aOffset
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (1 channel, binary)
ipContours	ITcVnContainer* [▶ 1760]&	Returns a container which is filled with the found contours (ContainerType_Vector_Vector_TcVnPoint2_DINT; The elements of this container are single contours. Non-zero interface pointers are reused.)
ipHierarchy	ITcVnContainer* [▶ 1760]&	Returns a container which is filled with information on the found contours' hierarchy (ContainerType_Vector_TcVnVector4_DINT; The elements of this container are four-dimensional vectors containing the 0-based indices of the next [0] and the previous contour [1] at the same level, the first child [2], and the parent [3]. Non-zero interface pointers are reused.)
eRetrievalMode	ETcVnContourRetrievalMode [▶ 1611]	Specifies which contours are retrieved and how their relationship is encoded
eApproximationMethod	ETcVnContourApproximationMethod [▶ 1610]	Contour encoding
aOffset	TcVnPoint [▶ 1588]&	Offset by which every contour point is shifted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.1.4 FindContours

Search for object contours in a binary image.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT FindContours(
    HRESULT                             hrPrev,
    ITcVnImage*                         ipSrcImage,
    ITcVnContainer*&                    ipContours,
    ETcVnContourRetrievalMode           eRetrievalMode = CRM_EXTERNAL,
    ETcVnContourApproximationMethod eApproximationMethod = CAM_SIMPLE,
    TcVnPoint&                          aOffset = {0, 0}
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (1 channel, binary)
ipContours	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the found contours (ContainerType_Vector_Vector_TcVnPoint2_DINT; The elements of this container are single contours. Non-zero interface pointers are reused.)
eRetrievalMode	ETcVnContourRetrievalMode [▶ 1611]	CRM_EXTERNAL	Specifies which contours are retrieved and how their relationship is encoded
eApproximationMethod	ETcVnContourApproximationMethod [▶ 1610]	CAM_SIMPLE	Contour encoding
aOffset	TcVnPoint [▶ 1588]&	{0, 0}	Offset by which every contour point is shifted

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.1.5 HoughCircles

Search for circles using the Hough transform.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT HoughCircles(
    HRESULT           hrPrev,
    ITcVnImage*       ipSrcImage,
    ITcVnContainer*&  ipCircles,
    double            fInvAccuRatio,
    double            fMinDist,
    ETcVnHoughMethod  eHoughMethod = HM_GRADIENT,
    double            fParam1 = 100,
    double            fParam2 = 100,
    ULONG             nMinRadius = 0,
    ULONG             nMaxRadius = 0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (USINT, 1 channel, gray-level)
ipCircles	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the found circles (ContainerType_Vector_TcVnVector3_REAL; Each container element contains the x coordinate of the circle center [0], the y coordinate of the circle center [1], and the radius [2], respectively. Non-zero interface pointers are reused.)
fInvAccuRatio	double		Inverted ratio of the accumulator size in relation to the source image's size (must be > 0. A value of 2 means that the size is halved in both directions.)
fMinDist	double		Smallest allowed distance of two circles (must be > 0)
eHoughMethod	ETcVnHoughMethod [▶ 1620]	HM_GRADIENT	Hough method to use (GRADIENT or GRADIENT_ALT)
fParam1	double	100	First method specific parameter (GRADIENT, GRADIENT_ALT: upper threshold for canny edge detection, which must be > 0)
fParam2	double	100	Second method specific parameter (GRADIENT: accumulator threshold for detecting circle centers, which must be > 0. GRADIENT_ALT: required circle perfectness, which must be > 0 and < 1, 1 would be a perfect circle)
nMinRadius	ULONG	0	Minimum circle radius allowed
nMaxRadius	ULONG	0	Maximum circle radius allowed (if 0, the value is internally set to max(image rows, image columns))

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.1.6 HoughLines

Search for lines using the standard Hough transform.

Syntax

Definition:

```
HRESULT HoughLines(
    HRESULT           hrPrev,
    ITcVnImage*       ipSrcImage,
    ITcVnContainer*&  ipLines,
    double            fDistRes,
    double            fAngleRes,
    ULONG             nAccuThreshold,
    double            fDistResDiv = 0,
    double            fAngleResDiv = 0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]		Source image (USINT, 1 channel, binary, may be modified by the function)
ipLines	ITcVnContainer* [▸ 1760]&		Returns a container which is filled with the found lines (ContainerType_Vector_TcVnVector2_REAL; Each container element contains the distance from the origin [0] in pixels and the rotation angle [1] in radians, respectively. Non-zero interface pointers are reused.)
fDistRes	double		Distance resolution of the accumulator (in pixels, must be > 0)
fAngleRes	double		Angle resolution of the accumulator (in radians, must be > 0)
nAccuThreshold	ULONG		Accumulator threshold
fDistResDiv	double	0	Divisor of the distance resolution for the multi-scale Hough transform (>= 0, must be > 0 if fAngleResDiv is > 0)
fAngleResDiv	double	0	Divisor of the angle resolution for the multi-scale Hough transform (>= 0, must be > 0 if fDistResDiv is > 0)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.1.7 HoughLinesP

Search for line segments using the probabilistic Hough transform.

Syntax

Definition:

```
HRESULT HoughLinesP(
    HRESULT          hrPrev,
    ITcVnImage*      ipSrcImage,
    ITcVnContainer*& ipLines,
    double           fDistRes,
    double           fAngleRes,
    ULONG            nAccuThreshold,
    double           fMinLineLength = 0,
    double           fMaxLineGap = 0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (USINT, 1 channel, binary, may be modified by the function)
ipLines	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the found line segments (ContainerType_Vector_TcVnVector4_DINT; Each container element contains the two ending points. Non-zero interface pointers are reused.)
fDistRes	double		Distance resolution of the accumulator (in pixels, must be > 0)
fAngleRes	double		Angle resolution of the accumulator (in radians, must be > 0)
nAccuThreshold	ULONG		Accumulator threshold
fMinLineLength	double	0	Minimum line length to search for
fMaxLineGap	double	0	Maximum gap between points on the same line

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.1.8 MatchImageHuMoments

Compare two images using the Hu moment invariants.

Syntax

Definition:

```
HRESULT MatchImageHuMoments(
    HRESULT                            hrPrev,
    ITcVnImage*                        ipImage1,
    ITcVnImage*                        ipImage2,
    ETcVnContoursMatchComparisonMethod eComparisonMethod,
    double&                            fDissimilarity
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage1	ITcVnImage* [▶ 1797]	First image (1 channel)
ipImage2	ITcVnImage* [▶ 1797]	Second image (1 channel)
eComparisonMethod	ETcVnContoursMatchComparisonMethod [▶ 1611]	Method used for comparing the Hu moment invariants of the images
fDissimilarity	double&	Returns the dissimilarity of the image Hu moment invariants depending on the chosen comparison method

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.1.9 MatchTemplate

Match a template image with every location in the source image and save the comparison results. Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT MatchTemplate(
    HRESULT                    hrPrev,
    ITcVnImage*                ipSrcImage,
    ITcVnImage*                ipTemplateImage,
    ITcVnImage*&               ipResultImage,
    ETcVnTemplateMatchMethod eMatchMethod = TMM_CCORR_NORMED,
    ITcVnImage*                ipTemplateMask = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (USINT or REAL, 1 or 3 channels)
ipTemplateImage	ITcVnImage* [▶ 1797]		Template image (same type as ipSrcImage, smaller width and height)
ipResultImage	ITcVnImage* [▶ 1797]&		Returns the result image (REAL, 1 channel, dimensions: (ipSrcImage.width - ipTemplateImage.width + 1) x (ipSrcImage.height - ipTemplateImage.height + 1). The best match is the global minimum (SQDIFF(_NORMED)) or maximum (CCORR(_NORMED), CCOEFF(_NORMED)). The position in ipResultImage is the top-left corner of ipTemplateImage position in ipSrcImage)
eMatchMethod	ETcVnTemplateMatchMethod [▶ 1633]	TMM_CCORR_NORMED	Specifies the template match method
ipTemplateMask	ITcVnImage* [▶ 1797]	nullptr	Optional mask for ipTemplateImage (same type and size as ipTemplateImage)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.1.10 MatchTemplateAndEvaluate

Match a template image with every location in the source image and evaluate the comparison results. Returns a sorted list of possible matches (best match first).
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT MatchTemplateAndEvaluate(
    HRESULT                  hrPrev,
    ITcVnImage*              ipSrcImage,
    ITcVnImage*              ipTemplateImage,
    ITcVnContainer*&         ipMatches,
    float                    fMatchThreshold,
    ETcVnTemplateMatchMethod eMatchMethod = TMM_CCORR_NORMED,
    ITcVnImage*              ipTemplateMask = nullptr,
    float                    fScaleFactor = 1,
    ETcVnInterpolationType   eInterpolationType = IT_BILINEAR,
    ITcVnContainer**         pipMatchValues = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (USINT or REAL, 1 or 3 channels)
ipTemplateImage	ITcVnImage* [▶ 1797]		Template image (same type as ipSrcImage, smaller width and height)
ipMatches	ITcVnContainer* [▶ 1760]&		Returns the matching positions (ContainerType_Vector_TcVnPoint2_DINT, where each element represents the top-left corner of ipTemplateImage) in ipSrcImage, sorted by relevance (best match first)
fMatchThreshold	float		Threshold to separate relevant from irrelevant matches (0..1 for NORMED methods, otherwise dependent on template size and content. To find a suitable value, you could evaluate some sample result images of F_VN_MatchTemplateExp.)
eMatchMethod	ETcVnTemplateMatchMethod [▶ 1633]	TMM_CCORR_NORMED	Specifies the template match method
ipTemplateMask	ITcVnImage* [▶ 1797]	nullptr	Optional mask for ipTemplateImage (same type and size as ipTemplateImage)
fScaleFactor	float	1	Factor (0..1] to reduce source and template image width and height for better performance (but less accuracy!)
eInterpolationType	ETcVnInterpolationType [▶ 1621]	IT_BILINEAR	Image resize interpolation type (only used if fScaleFactor != 1, TCVN_IT_BILINEAR recommended for most cases)
pipMatchValues	ITcVnContainer* [▶ 1760]*	nullptr	Optionally returns the matching values (ContainerType_Vector_REAL, same size and sort order as ipMatches. Set to 0 if not required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.2 CannyEdgeDetection

Find edges using the Canny edge detection algorithm.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT CannyEdgeDetection(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage,
    double       fThresholdLow,
    double       fThresholdHigh,
    ULONG        nApertureSize = 3,
    bool         bL2Gradient = false
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (elements of type USINT)
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (elements of type USINT. An appropriate destination image will be created if required.)
fThresholdLow	double		Low threshold
fThresholdHigh	double		High threshold
nApertureSize	ULONG	3	Aperture size for the Sobel operator (3, 5, 7)
bL2Gradient	bool	false	If true, the more accurate (and slower) L2 norm is used instead of the L1 norm

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.3 ConnectedComponents

Computes the connected components of an image.

Syntax

Definition:

```
HRESULT ConnectedComponents(
    HRESULT                              hrPrev,
    ITcVnImage*                          ipSrcImage,
    ITcVnImage*&                         ipLabelImage,
    LONG&                                nLabels,
    ETcVnPixelConnectivity               eConnectivity = PC_8,
    ETcVnElementType                     eLabelType = TCVN_ET_DINT,
    ETcVnConnectedComponentsAlgorithm eAlgorithm = CCA_GRANA
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (USINT, 1 channel, binary)
ipLabelImage	ITcVnImage* [▶ 1797]&		Returns the labels for each source image pixel (1 channel, type depends on eLabelType. An appropriate destination image will be created if required.)
nLabels	LONG&		Returns the number of labels
eConnectivity	ETcVnPixelConnectivity [▶ 1627]	PC_8	Selects if 4- or 8-way pixel connectivity should be used
eLabelType	ETcVnElementType [▶ 1615]	TCVN_ET_DINT	Selects the type of ipLabelImage (only UINT or DINT supported)
eAlgorithm	ETcVnConnectedComponentsAlgorithm [▶ 1609]	CCA_GRANA	Selects the applied algorithm

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.4 ConnectedComponentsWithStats

Computes the connected components and corresponding statistics of an image.

Syntax

Definition:

```
HRESULT ConnectedComponentsWithStats(
    HRESULT                               hrPrev,
    ITcVnImage*                           ipSrcImage,
    ITcVnImage*&                          ipLabelImage,
    ITcVnContainer*&                      ipBoundingBoxes,
    ITcVnContainer*&                      ipNumPixels,
    ITcVnContainer*&                      ipCentroids,
    LONG&                                 nLabels,
    ETcVnPixelConnectivity                eConnectivity = PC_8,
    ETcVnElementType                      eLabelType = TCVN_ET_DINT,
    ETcVnConnectedComponentsAlgorithm eAlgorithm = CCA_GRANA
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]		Source image (USINT, 1 channel, binary)
ipLabelImage	ITcVnImage* [▸ 1797]&		Returns the labels for each source image pixel (1 channel, DINT. An appropriate destination image will be created if required.)
ipBoundingBoxes	ITcVnContainer* [▸ 1760]&		Returns the bounding boxes for each labeled region. (ContainerType_Vector_TcVnRectangle_DINT; Non-zero interface pointers are reused.)
ipNumPixels	ITcVnContainer* [▸ 1760]&		Returns the number of pixels for each labeled region. (ContainerType_Vector_DINT; Non-zero interface pointers are reused.)
ipCentroids	ITcVnContainer* [▸ 1760]&		Returns the centroids for each labeled region. (ContainerType_Vector_TcVnPoint2_LREAL; Non-zero interface pointers are reused.)
nLabels	LONG&		Returns the number of labels
eConnectivity	ETcVnPixelConnectivity [▸ 1627]	PC_8	Selects if 4- or 8-way pixel connectivity should be used
eLabelType	ETcVnElementType [▸ 1615]	TCVN_ET_DINT	Selects the type of ipLabelImage (only UINT or DINT supported)
eAlgorithm	ETcVnConnectedComponentsAlgorithm [▸ 1609]	CCA_GRANA	Selects the applied algorithm

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.5 CountNonZeroPixels

Counts the non-zero pixels in a single-channel image (e.g. useful to analyze threshold results).

Syntax

Definition:

```
HRESULT CountNonZeroPixels(
    HRESULT     hrPrev,
    ITcVnImage* ipSrcImage,
    ULONGLONG&  nNonZero
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (1 channel)
nNonZero	ULONGLONG&	Returns the number of non-zero pixels in ipSrcImage

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.6 DistanceTransformation

Calculates the distance transformation, which is the distance to the closest zero pixel in a binary image. Additionally, an image with component labels is created (discrete Voronoi diagram).

Syntax

Definition:

```
HRESULT DistanceTransformation(
    HRESULT                          hrPrev,
    ITcVnImage*                      ipSrcImage,
    ITcVnImage*&                     ipDestImage,
    ETcVnDistanceType                eDistanceType,
    ETcVnDistanceTransformationMask  eMaskSize,
    ITcVnImage**                     pipDestLabels = nullptr,
    ETcVnDistanceTransformationLabel eLabelType = DTL_CCOMP
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]		Source image (USINT, 1 channel)
ipDestImage	ITcVnImage* [▸ 1797]&		Destination image (REAL, 1 channel. An appropriate destination image will be created if required.)
eDistanceType	ETcVnDistanceType [▸ 1613]		Distance computation method (supported: L1, L2, C)
eMaskSize	ETcVnDistanceTransformationMask [▸ 1612]		Size of the distance transformation mask (PRECISE not supported for label computation)
pipDestLabels	ITcVnImage* [▸ 1797]*	nullptr	Returns the component labels (Discrete Voronoi diagram; DINT, 1 channel. An appropriate image will be created if required.)
eLabelType	ETcVnDistanceTransformationLabel [▸ 1612]	DTL_CCOMP	Type of the labels

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.7 GetConnectedComponent

Get the mask image (and optionally the outer contour) of a connected component from a label image based on the label ID.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT GetConnectedComponent(
    HRESULT         hrPrev,
    ITcVnImage*     ipLabelImage,
```

```
    LONG             nLabelID,
    ITcVnImage*&     ipMaskImage,
    ITcVnContainer** pipContourPoints = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipLabelImage	ITcVnImage* [▶ 1797]		Image containing labels for each pixel (1 channel DINT or UINT).
nLabelID	LONG		Value of the requested label
ipMaskImage	ITcVnImage* [▶ 1797]&		Returns a mask image marking the pixels that belong to the component (1 channel USINT. Mask is optional, set to 0 if not required.)
pipContourPoints	ITcVnContainer* [▶ 1760]*	nullptr	Returns the outer contour of the component(ContainerType_Vector_TcVnPoint2_DINT. Contour is optional, set to 0 if not required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.8 ImageAverage

Computes the (channel-wise) average pixel value of an image.

Syntax

Definition:

```
HRESULT ImageAverage(
    HRESULT            hrPrev,
    ITcVnImage*        ipSrcImage,
    TcVnVector4_LREAL& aAverage,
    ITcVnImage*        ipMask = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (1 - 4 channels)
aAverage	TcVnVector4_LREAL [▶ 1590]&		Returns the (channel-wise) average pixel value of ipSrcImage
ipMask	ITcVnImage* [▶ 1797]	nullptr	Mask of type USINT (1 channel)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.9 ImageAverageStdDev

Computes the (channel-wise) average pixel value and the corresponding standard deviation of an image.

Syntax

Definition:

```
HRESULT ImageAverageStdDev(
    HRESULT            hrPrev,
    ITcVnImage*        ipSrcImage,
    TcVnVector4_LREAL& aAverage,
    TcVnVector4_LREAL& aStdDev,
    ITcVnImage*        ipMask = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (1 - 4 channels)
aAverage	TcVnVector4_LREAL [▶ 1590]&		Returns the (channel-wise) average pixel value of ipSrcImage
aStdDev	TcVnVector4_LREAL [▶ 1590]&		Returns the (channel-wise) pixel value standard deviation of ipSrcImage
ipMask	ITcVnImage* [▶ 1797]	nullptr	Mask of type USINT (1 channel)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.10 ImageCenterOfMass

Computes the center of mass of an image.

Syntax

Definition:

```
HRESULT ImageCenterOfMass(
    HRESULT           hrPrev,
    ITcVnImage*       ipSrcImage,
    TcVnPoint2_LREAL& aCenterOfMass,
    ITcVnImage*       ipMask = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (1 channel)
aCenterOfMass	TcVnPoint2_LREAL [▶ 1588]&		Returns the center of mass of the image
ipMask	ITcVnImage* [▶ 1797]	nullptr	Optional mask (1 channel of type USINT, same width and height as ipSrcImage)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.11 ImageMedian

Computes the (approximated, channel-wise) median pixel value of an image.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT ImageMedian(
    HRESULT             hrPrev,
    ITcVnImage*         ipSrcImage,
    TcVnVector4_LREAL&  aMedian,
    ITcVnImage*         ipMask = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (1 - 4 channels)
aMedian	TcVnVector4_LREAL [▶ 1590]&		Returns the (approximated, channel-wise) median pixel value of ipSrcImage
ipMask	ITcVnImage* [▶ 1797]	nullptr	Mask of type USINT (1 channel)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.12 ImageMoments

Computes the spatial moments, the central moments, and the central normalized moments of an image up to the third order.

Syntax

Definition:

```
HRESULT ImageMoments(
    HRESULT      hrPrev,
    ITcVnImage*  ipImage,
    TcVnMoments& stMoments,
    bool         bBinaryImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▶ 1797]	Source image (1 channel, elements of type USINT or REAL)
stMoments	TcVnMoments [▶ 1644]&	Returns a struct containing the moments
bBinaryImage	bool	If true, all non-zero pixels are treated as ones

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.13 MaxPixelValue

Finds the maximum pixel value in an image (1 - 4 channels supported).

Syntax

Definition:

```
HRESULT MaxPixelValue(
    HRESULT                   hrPrev,
    ITcVnImage*               ipImage,
    TcVnVector4_LREAL&        aMaxValue,
    TcVnPoint2_DINT&          aPosition,
    ITcVnImage*               ipMask = nullptr,
    ETcVnVectorCompareMethod eVectorCompareMethod = VCM_EUCLIDEAN
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▸ 1797]		Source image
aMaxValue	TcVnVector4_LREAL [▸ 1590]&		Returns the maximum pixel value
aPosition	TcVnPoint2_DINT [▸ 1588]&		Returns the first found position of aMaxValue (not supported for multi-channel images with ELEMENTWISE)
ipMask	ITcVnImage* [▸ 1797]	nullptr	Optional mask to specify which pixel positions are considered (USINT, set parameter to 0 if not required)
eVectorCompareMethod	ETcVnVectorCompareMethod [▸ 1635]	VCM_EUCLIDEAN	Select a vector compare method for multi-channel images

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.14 MinPixelValue

Finds the minimum pixel value in an image (1 - 4 channels supported).

Syntax

Definition:

```
HRESULT MinPixelValue(
    HRESULT                    hrPrev,
    ITcVnImage*                ipImage,
    TcVnVector4_LREAL&         aMinValue,
    TcVnPoint2_DINT&           aPosition,
    ITcVnImage*                ipMask = nullptr,
    ETcVnVectorCompareMethod eVectorCompareMethod = VCM_EUCLIDEAN
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▶ 1797]		Source image
aMinValue	TcVnVector4_LREAL [▶ 1590]&		Returns the minimum pixel value
aPosition	TcVnPoint2_DINT [▶ 1588]&		Returns the first found position of aMinValue (not supported for multi-channel images with ELEMENTWISE)
ipMask	ITcVnImage* [▶ 1797]	nullptr	Optional mask to specify which pixel positions are considered (USINT, set parameter to 0 if not required)
eVectorCompareMethod	ETcVnVectorCompareMethod [▶ 1635]	VCM_EUCLIDEAN	Select a vector compare method for multi-channel images

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.14.15 RegionOrientation

Calculate the orientation of a region based on a binary image that contains a non zero value for every pixel in that region.

Syntax

Definition:

```
HRESULT RegionOrientation(
    HRESULT                  hrPrev,
    ITcVnImage*              ipSrcImage,
    TcVnRotatedRectangle&    stOrientation,
    ETcVnOrientationMethod eMethod = OM_PCA
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (1 channel, binary)
stOrientation	TcVnRotatedRectangle [▶ 1658]&		Resulting rotated rectangle, containig center of mass, lenghts of axes, and rotation angle of the region in clockwise direction.
eMethod	ETcVnOrientationMethod [▶ 1626]	OM_PCA	Method for calculating the orientation.

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.15 Image Color and Contrast Processing

该组包含色彩和对比处理函数。

函数

图像的频率分布、强度和对比调整

- Clahe [▸ 2374]
- Histogram [▸ 2378]
- HistogramEqualization [▸ 2379]
- NormalizeImage [▸ 2381]
- NormalizeImageForDisplay [▸ 2382]

查找表和色图

- ApplyColorMap [▸ 2372]
- ApplyLut [▸ 2373]
- GenerateColorMap [▸ 2376]
- GenerateCustomColorMap [▸ 2377]

颜色匹配

- ReferenceColorSimilarity [▸ 2382]
- TrainImageColor [▸ 2386]
- TrainImageColor_ITcVnMlModel [▸ 2389]

其它

- ConvertColorSpace [▸ 2375]
- InvertImageColor [▸ 2380]

还请参阅有关此

WhiteBalance [▸ 2392]

6.2.3.15.1 ApplyColorMap

Apply a color map to a gray-level image.

Syntax

Definition:

```
HRESULT ApplyColorMap(
    HRESULT         hrPrev,
    ITcVnImage*     ipSrcImage,
    ITcVnImage*&    ipDestImage,
    ITcVnContainer* ipColorMap
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (TCVN_ET_USINT or TCVN_ET_UINT, 1 channel)
ipDestImage	ITcVnImage* [▶ 1797]&	Returns the colored image (TCVN_ET_USINT, 3 channel. An appropriate image will be created if required.)
ipColorMap	ITcVnContainer* [▶ 1760]	Color map to be applied to ipSrcImage (ContainerType_Vector_TcVnVector3_REAL with 256 or 65536 elements, dependent on ipSrcImage type). Can be either custom or created with F_VN_GenerateColorMap.

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.15.2 ApplyLut

Apply a lookup table to an image to manipulate its colors.

Syntax

Definition:

```
HRESULT ApplyLut(
    HRESULT         hrPrev,
    ITcVnImage*     ipSrcImage,
    ITcVnImage*&    ipDestImage,
    ITcVnContainer* ipLut
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (1-4 channels of 8 or 16 bit types SINT, USINT, INT, UINT)
ipDestImage	ITcVnImage* [▶ 1797]&	Returns the result image (same size, type and channels as ipSrcImage. An appropriate image will be created if required.)
ipLut	ITcVnContainer* [▶ 1760]	Lookup table with elements matching ipSrcImage type and either 256 (8 bit images) or 65536 (16 bit images) elements. E.g. ContainerType_Vector_USINT for 1-4 channel USINT image (in this case, the same lookup values are used for each channel) or ContainerType_Vector_TcVnVector3_USINT for a 3 channel USINT image (in this case, each channel uses an individual lookup value).

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.15.3 Clahe

Apply Constrast Limited Adaptive Histogram Equalization. This applies local histogram equalization to individual tiles.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT Clahe(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage,
    ITcVnImage*&  ipDestImage,
    double        fClipLimit = 40,
    ULONG         nTilesX = 8,
    ULONG         nTilesY = 8
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (TCVN_ET_USINT or TCVN_ET_UINT, 1 channel)
ipDestImage	ITcVnImage* [▶ 1797]&		Returns the resulting image (An appropriate image will be created if required.)
fClipLimit	double	40	Threshold for contrast limit (set to <= 0 to disable clipping)
nTilesX	ULONG	8	Number of tiles in x direction
nTilesY	ULONG	8	Number of tiles in y direction

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.15.4 ConvertColorSpace

Convert image from one color space to another.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT ConvertColorSpace(
    HRESULT                  hrPrev,
    ITcVnImage*              ipSrcImage,
    ITcVnImage*&             ipDestImage,
    ETcVnColorSpaceTransform eTransform
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
eTransform	ETcVnColorSpaceTransform [▶ 1598]	Transforms to be applied

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.15.5 GenerateColorMap

Generates a pre-defined color map.

Syntax

Definition:

```
HRESULT GenerateColorMap(
    HRESULT           hrPrev,
    ITcVnContainer*&  ipColorMap,
    ETcVnColorMap     eColorMap,
    ETcVnColorMapSize eColorMapSize
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipColorMap	ITcVnContainer* [▶ 1760]&	Returns the color map (ContainerType_Vector_TcVnVector3_REAL with 256 or 65536 elements)
eColorMap	ETcVnColorMap [▶ 1596]	Selects a color map (similar to GNU Octave/MATLAB types)
eColorMapSize	ETcVnColorMapSize [▶ 1597]	Defines how many elements the generated color map should have

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.15.6 GenerateCustomColorMap

Generate a custom color map by interpolating between user defined colors (equally distributed, linear for each channel).

Syntax

Definition:

```
HRESULT GenerateCustomColorMap(
    HRESULT           hrPrev,
    ITcVnContainer*&  ipColorMap,
    ITcVnContainer*   ipInitialColors,
    ETcVnColorMapSize eColorMapSize
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipColorMap	ITcVnContainer* [▶ 1760]&	Returns the color map (ContainerType_Vector_TcVnVector3_REAL with 256 or 65536 elements)
ipInitialColors	ITcVnContainer* [▶ 1760]	User defined colors (ContainerType_Vector_TcVnVector3_REAL, at least 2 elements)
eColorMapSize	ETcVnColorMapSize [▶ 1597]	Defines how many elements the generated color map should have

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.15.7 Histogram

Calculate the (multi-channel) histogram of an image.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT Histogram(
    HRESULT          hrPrev,
    ITcVnImage*      ipSrcImage,
    ITcVnContainer*& ipDestHistogram,
    ULONG&           nBins = 0,
    double&          fLowerBound = 0,
    double&          fUpperBound = 0,
    ITcVnImage*      ipMask = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]		Source image
ipDestHistogram	ITcVnContainer* [▸ 1760]&		Returns a container with a multi-channel histogram, where every channel is represented as a vector of UDINT (ContainerType_Vector_Vector_UDINT. Non-zero interface pointers are reused.)
nBins	ULONG&	0	Desired number of bins or 0 to keep the default for the corresponding image format (in) and default number of bins (out)
fLowerBound	double&	0	Lower (inclusive) boundary of the 0-th histogram bin (in), or receive the default if fLowerBound AND fUpperBound are set to 0 (out)
fUpperBound	double&	0	Upper (exclusive) boundary of the last histogram bin nBins-1 (in), or receive the default if fLowerBound AND fUpperBound are set to 0 (out)
ipMask	ITcVnImage* [▸ 1797]	nullptr	Source image mask (TCVN_ET_USINT, 1 channel. Mask is optional, set to 0 if not required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.15.8 HistogramEqualization

Equalize the histogram of a grayscale or color image, which normalizes the brightness and improves the contrast. The channel index that should be equalized has to be specified (-1 expects a RGB image, converts it to YCbCr, equalizes the Y channel and converts the image back to RGB).
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT HistogramEqualization(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage,
    ITcVnImage*&  ipDestImage,
    SHORT         nChannelIdx = -1
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (USINT)
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (An appropriate destination image will be created if required.)
nChannelIdx	SHORT	-1	Index of the image channel that should be equalized (-1 expects a RGB image, converts it to YCbCr, equalizes the Y channel and converts the image back to RGB)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.15.9 InvertImageColor

Invert the color of an image. For signed integer or negative floating point values, only the signs will be switched. If the image contains only positive floating point values, each pixel value is subtracted from the maximum available pixel value (or 1.0, whatever is higher).

Syntax

Definition:

```
HRESULT InvertImageColor(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage,
    double       fMaxValue = -1
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]		Source image
ipDestImage	ITcVnImage* [▸ 1797]&		Destination image (An appropriate image will be created if required.)
fMaxValue	double	-1	Maximum pixel value (e.g. if a 16 bit image contains 12 bit values). -1 means the default values are used.

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.15.10 NormalizeImage

Normalize an image regarding its value range (e.g. stretch pixel values [50..150] to full range [0..255]) or scale the values regarding a specific normalization (e.g. L2-norm).

Syntax

Definition:

```
HRESULT NormalizeImage(
    HRESULT                 hrPrev,
    ITcVnImage*             ipSrcImage,
    ITcVnImage*&            ipDestImage,
    double                  fAlpha = 0,
    double                  fBeta = 255,
    ETcVnNormalizationType  eNormType = NT_MINMAX,
    ETcVnElementType        eDestType = TCVN_ET_SAME_AS_SOURCE,
    ITcVnImage*             ipMask = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (An appropriate destination image will be created if required.)
fAlpha	double	0	Lower range boundary (in case of normalizing the value range) or value to normalize to (||ipDestImage|| = fAlpha)
fBeta	double	255	Upper range boundary (in case of normalizing the value range)
eNormType	ETcVnNormalizationType [▶ 1624]	NT_MINMAX	Normalization type (only INF, L1, L2 or MINMAX)
eDestType	ETcVnElementType [▶ 1615]	TCVN_ET_SAME_AS_SOURCE	Destination image depth (usually SAME_AS_SOURCE)
ipMask	ITcVnImage* [▶ 1797]	nullptr	Mask to restrict the normalization to specific pixel positions (set 0 to normalize the whole image)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.15.11 NormalizeImageForDisplay

Normalize an image for display, i.e. scale it to the full value range of the underlying data type (-1 to 1 for floating point).
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT NormalizeImageForDisplay(
    HRESULT                   hrPrev,
    ITcVnImage*               ipSrcImage,
    ITcVnImage*&              ipDestImage,
    ETcVnSignedNormalization eSignedNormalization
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]	Source image
ipDestImage	ITcVnImage* [▸ 1797]&	Destination image (An appropriate destination image will be created if required.)
eSignedNormalization	ETcVnSignedNormalization [▸ 1630]	Option for normalizing signed values

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.15.12 ReferenceColorSimilarity (ITcVnColorModel*)

Computes the similarity to a reference color model for each pixel in the source image.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT ReferenceColorSimilarity(
    HRESULT          hrPrev,
    ITcVnImage*      ipSrcImage,
    ITcVnImage*&     ipDestImage,
    ITcVnColorModel* ipColorModel,
    float            fVariance = 0.15f,
    float            fLuminanceWeight = 0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]		RGB source image (USINT, 3 channels)
ipDestImage	ITcVnImage* [▸ 1797]&		Returns the similarity to ipColorModel for each pixel in ipSrcImage (USINT, 1 channel. An appropriate destination image will be created if required.)
ipColorModel	ITcVnColorModel* [▸ 1802]		Color model
fVariance	float	0.15f	Allowed color variance (0.1 - 0.3 might be a good start to try)
fLuminanceWeight	float	0	Weight the impact of the luminance ([0..1], e.g. set to 0 to be more resistant to unequal illumination, but might be required to differentiate between some colors. Ignored if ipColorModel is of type RGB.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.15.13 ReferenceColorSimilarity (ITcVnMlModel*)

Computes the similarity to a reference color model for each pixel in the source image.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT ReferenceColorSimilarity(
    HRESULT        hrPrev,
    ITcVnImage*    ipSrcImage,
    ITcVnImage*&   ipDestImage,
    ITcVnMlModel*  ipColorModel,
    float          fVariance = 0.15f,
    float          fLuminanceWeight = 0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]		RGB source image (USINT, 3 channels)
ipDestImage	ITcVnImage* [▸ 1797]&		Returns the similarity to ipColorModel for each pixel in ipSrcImage (USINT, 1 channel. An appropriate destination image will be created if required.)
ipColorModel	ITcVnMlModel* [▸ 1804]		Color model
fVariance	float	0.15f	Allowed color variance (0.1 - 0.3 might be a good start to try)
fLuminanceWeight	float	0	Weight the impact of the luminance ([0..1], e.g. set to 0 to be more resistant to unequal illumination, but might be required to differentiate between some colors. Ignored if ipColorModel is of type RGB.)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.15.14 ReferenceColorSimilarity (TcVnVector3_LREAL)

Computes the similarity to a reference color for each pixel in the source image.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT ReferenceColorSimilarity(
    HRESULT            hrPrev,
    ITcVnImage*        ipSrcImage,
    ITcVnImage*&       ipDestImage,
    TcVnVector3_LREAL& aRefColor,
    float              fVariance = 0.15f,
    float              fLuminanceWeight = 0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		RGB source image (USINT, 3 channels)
ipDestImage	ITcVnImage* [▶ 1797]&		Returns the similarity to aRefColor for each pixel in ipSrcImage (USINT, 1 channel. An appropriate destination image will be created if required.)
aRefColor	TcVnVector3_LREAL [▶ 1590]&		Reference color (RGB, [0..255])
fVariance	float	0.15f	Allowed color variance (0.1 - 0.3 might be a good start to try)
fLuminanceWeight	float	0	Weight the impact of the luminance ([0..1], e.g. set to 0 to be more resistant to unequal illumination, but might be required to differentiate between some colors)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.15.15 TrainImageColor

Create a new color model, describing the image color.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT TrainImageColor(
    HRESULT                   hrPrev,
    ITcVnImage*               ipSrcImage,
    ITcVnColorModel*&         ipColorModel,
    ULONG                     nDifferentColors,
    ETcVnColorTrainingMethod eMethod = CTM_LAB,
    ITcVnImage*               ipMask = nullptr,
    ULONG                     nSkipPixels = 0,
    ETcVnClusteringAlgorithm eClusteringAlgorithm = CA_KMEANSPP,
    LONG                      nMaxClusterRadius = 50,
    bool                      bSingleSplitSteps = true
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (3 channels (RGB) of type USINT)
ipColorModel	ITcVnColorModel* [▶ 1802]&		Returns the color model
nDifferentColors	ULONG		Maximum number of different colors to distinguish (if LBG is used as a clustering algorithm, the result might have less different colors, depending on nMaxClusterRadius)
eMethod	ETcVnColorTrainingMethod [▶ 1609]	CTM_LAB	Color training method
ipMask	ITcVnImage* [▶ 1797]	nullptr	Optional image mask (1 channel of type USINT, set to 0 if not required)
nSkipPixels	ULONG	0	Number of pixels to skip between each evaluated color sample (to achieve a better performance). 0 takes every pixel into account and tends to be more accurate.
eClusteringAlgorithm	ETcVnClusteringAlgorithm [▶ 1596]	CA_KMEANSPP	Clustering algorithm
nMaxClusterRadius	LONG	50	Only used for the LBG clustering algorithm. Maximum allowed radius (> 0) of a single cluster, i.e. clusters with a higher radius will be split into smaller ones, until a global number of nDifferentColors is reached.

Name	Type	Default	Description
bSingleSplitSteps	bool	true	Only used for the LBG clustering algorithm. If true, the global optimization is always run after a single cluster has been split. If false, several clusters are split within the same step before applying the global optimization. Applying the global optimization less often is faster, but can lead to less optimal results, especially having 2 nearby clusters that could be represented by 1)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.15.16 TrainImageColor (ITcVnMlModel*)

Create a new color model, describing the image color.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT TrainImageColor(
    HRESULT                    hrPrev,
    ITcVnImage*                ipSrcImage,
    ITcVnMlModel*&             ipColorModel,
    ULONG                      nDifferentColors,
    ETcVnColorTrainingMethod eMethod = CTM_LAB,
    ITcVnImage*                ipMask = nullptr,
    ULONG                      nSkipPixels = 0,
    ETcVnClusteringAlgorithm eClusteringAlgorithm = CA_KMEANSPP,
    LONG                       nMaxClusterRadius = 50,
    bool                       bSingleSplitSteps = true
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]		Source image (3 channels (RGB) of type USINT)
ipColorModel	ITcVnMlModel* [▸ 1804]&		Returns the color model
nDifferentColors	ULONG		Maximum number of different colors to distinguish (if LBG is used as a clustering algorithm, the result might have less different colors, depending on nMaxClusterRadius)
eMethod	ETcVnColorTrainingMethod [▸ 1609]	CTM_LAB	Color training method
ipMask	ITcVnImage* [▸ 1797]	nullptr	Optional image mask (1 channel of type USINT, set to 0 if not required)
nSkipPixels	ULONG	0	Number of pixels to skip between each evaluated color sample (to achieve a better performance). 0 takes every pixel into account and tends to be more accurate.
eClusteringAlgorithm	ETcVnClusteringAlgorithm [▸ 1596]	CA_KMEANSPP	Clustering algorithm
nMaxClusterRadius	LONG	50	Only used for the LBG clustering algorithm. Maximum allowed radius (> 0) of a single cluster, i.e. clusters with a higher radius will be split into smaller ones, until a global number of nDifferentColors is reached.

Name	Type	Default	Description
bSingleSplitSteps	bool	true	Only used for the LBG clustering algorithm. If true, the global optimization is always run after a single cluster has been split. If false, several clusters are split within the same step before applying the global optimization. Applying the global optimization less often is faster, but can lead to less optimal results, especially having 2 nearby clusters that could be represented by 1)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.15.17 WhiteBalance

Calculates the white balance ratios for a RGB image. Uses the green channel as a reference and computes the relative ratios for the red and blue channels. Requires a non-reflecting, not overexposed white colored object (e.g. a sheet of paper) in at least 100 pixels of the image to provide reasonable results. The resulting ratios can then be set on the camera (multiply with existing ratios), or applied in the PLC via MultiplyImageWithVector.

Syntax

Definition:

```
HRESULT WhiteBalance(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage,
    ITcVnImage** ppDestImage,
    float&       fRatioRed,
    float&       fRatioBlue,
    ULONG        nMaxValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (3 channels (RGB), USINT or UINT)
ppDestImage	ITcVnImage* [▶ 1797]*	Destination image with white balance applied (optional, set to 0 if not required. An appropriate image will be created if required.)
fRatioRed	float&	Returns the ratio for the red channel
fRatioBlue	float&	Returns the ratio for the blue channel
nMaxValue	ULONG	Specify the maximum pixel value that would be achieved by overexposure (usually 255 for 8 bit or 4095 for 12 bit data)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16 Image Filtering

该组包含用于滤波图像的函数。

函数

模糊化

- BilateralFilter [▶ 2394]
- BoxFilter [▶ 2395]
- GaussianFilter [▶ 2401]
- MedianFilter [▶ 2404]

边缘检测

- LaplacianFilter [▶ 2402]
- ScharrFilter [▶ 2408]
- SobelFilter [▶ 2410]

用户定义滤波

- CustomFilter [▶ 2398]
- SeparableCustomFilter [▶ 2409]

填充区域

形态学算子

局部极值

局部方差

6.2.3.16.1 BilateralFilter

Apply a Bilateral filter to smooth the image but preserve edges.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT BilateralFilter(
    HRESULT                         hrPrev,
    ITcVnImage*                     ipSrcImage,
    ITcVnImage*&                    ipDestImage,
    LONG                            nDiameter,
    double                          fSigmaColor,
    double                          fSigmaSpace,
    ETcVnBorderInterpolationMethod eBorderType = BIM_DEFAULT
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (Must be different from ipSrcImage! An appropriate destination image will be created if required.)
nDiameter	LONG		Diameter of the pixel neighborhood used for filtering (e.g. 5, 7, 9). If <= 0, it is automatically chosen dependend on fSigmaSpace.
fSigmaColor	double		Sigma used for color space filtering (> 0). A larger value means that farther colors are mixed together.
fSigmaSpace	double		Sigma used for coordinate space filtering (> 0). A larger value means that farther pixels can influence each other.
eBorderType	ETcVnBorderInterpolationMethod [▶ 1593]	BIM_DEFAULT	Image border handling

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16.2 BoxFilter

Apply a box filter to an image.

Syntax

Definition:

```
HRESULT BoxFilter(
    HRESULT                           hrPrev,
    ITcVnImage*                       ipSrcImage,
    ITcVnImage*&                      ipDestImage,
    ULONG                             nFilterWidth,
    ULONG                             nFilterHeight,
    ETcVnElementType                  eDestDepth = TCVN_ET_SAME_AS_SOURCE,
    TcVnPoint&                        aAnchor = {-1, -1},
    bool                              bNormalize = true,
    ETcVnBorderInterpolationMethod eBorderType = BIM_DEFAULT
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (An appropriate destination image will be created if required.)
nFilterWidth	ULONG		Filter width in pixels
nFilterHeight	ULONG		Filter height in pixels
eDestDepth	ETcVnElementType [▶ 1615]	TCVN_ET_SAME_AS_SOURCE	Destination image depth
aAnchor	TcVnPoint [▶ 1588]&	{-1, -1}	Anchor point of the kernel ([-1, -1] for center)
bNormalize	bool	true	If true, the kernel is normalized by nFilterWidth * nFilterHeight
eBorderType	ETcVnBorderInterpolationMethod [▶ 1593]	BIM_DEFAULT	Image border handling

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16.3 BrightBorderObjects

Find bright objects connected to the image border within a gray-scale single-channel image.

Syntax

Definition:

```
HRESULT BrightBorderObjects(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (USINT, UINT, INT, REAL, or LREAL, 1 channel)
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16.4 CreateStructuringElement

Creates a structuring element for the usage with morphological operators and allocate an appropriate data buffer. The initial reference count is set to one if a new image interface is created and kept, otherwise.

Syntax

Definition:

```
HRESULT CreateStructuringElement(
    HRESULT                        hrPrev,
    ITcVnImage*&                   ipStructuringElement,
    ETcVnStructuringElementShape   eShape,
    ULONG                          nWidth,
    ULONG                          nHeight
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipStructuringElement	ITcVnImage* [▶ 1797]&	Returns the created structuring element (Non-zero interface pointers are reused.)
eShape	ETcVnStructuringElementShape [▶ 1631]	Shape of the structuring element (rectangle, cross, or ellipse)
nWidth	ULONG	Width
nHeight	ULONG	Height

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16.5 CustomFilter

Apply a custom filter to the image.

Syntax

Definition:

```
HRESULT CustomFilter(
    HRESULT                         hrPrev,
    ITcVnImage*                     ipSrcImage,
    ITcVnImage*&                    ipDestImage,
    ETcVnElementType                eDestDepth,
    TcVnMatrix&                     stKernel,
    TcVnPoint&                      aAnchor = {-1, -1},
    double                          fDelta = 0,
    ETcVnBorderInterpolationMethod eBorderType = BIM_DEFAULT
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (An appropriate destination image will be created if required.)
eDestDepth	ETcVnElementType [▶ 1615]		Destination image depth
stKernel	TcVnMatrix [▶ 1644]&		Custom filter kernel with values of type REAL or LREAL
aAnchor	TcVnPoint [▶ 1588]&	{-1, -1}	Anchor point of the kernel
fDelta	double	0	Value added to each pixel after filtering
eBorderType	ETcVnBorderInterpolationMethod [▶ 1593]	BIM_DEFAULT	Image border handling

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16.6 DarkBorderObjects

Find dark objects connected to the image border within a gray-scale single-channel image. (equivalent to filling holes)

Syntax

Definition:

```
HRESULT DarkBorderObjects(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage,
    ITcVnImage*&  ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (USINT, UINT, INT, REAL, or LREAL, 1 channel)
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16.7 FillHoles

Fill holes within a gray-scale single-channel image. (equivalent to finding dark border objects)

Syntax

Definition:

```
HRESULT FillHoles(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage,
    ITcVnImage*&  ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (USINT, UINT, INT, REAL, or LREAL, 1 channel)
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16.8 GaussianFilter

Apply a Gaussian filter to smooth the image.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT GaussianFilter(
    HRESULT                         hrPrev,
    ITcVnImage*                     ipSrcImage,
    ITcVnImage*&                    ipDestImage,
    ULONG                           nFilterWidth,
    ULONG                           nFilterHeight,
    double                          fSigmaX = 0,
    double                          fSigmaY = 0,
    ETcVnBorderInterpolationMethod eBorderType = BIM_DEFAULT
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (An appropriate destination image will be created if required.)
nFilterWidth	ULONG		Filter width in pixels (1, 3, 5, 7, ...)
nFilterHeight	ULONG		Filter height in pixels (1, 3, 5, 7, ...)
fSigmaX	double	0	Gaussian kernel standard deviation in X direction (>= 0, automatically chosen if 0)
fSigmaY	double	0	Gaussian kernel standard deviation in Y direction (>= 0, automatically chosen if 0)
eBorderType	ETcVnBorderInterpolationMethod [▶ 1593]	BIM_DEFAULT	Image border handling

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16.9 LaplacianFilter

Apply a Laplacian filter to an image.

Syntax

Definition:

```
HRESULT LaplacianFilter(
    HRESULT                        hrPrev,
    ITcVnImage*                    ipSrcImage,
    ITcVnImage*&                   ipDestImage,
    ETcVnElementType               eDestDepth,
    ULONG                          nKernelSize = 1,
    double                         fScale = 1,
    double                         fDelta = 0,
    ETcVnBorderInterpolationMethod eBorderType = BIM_DEFAULT
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (An appropriate destination image will be created if required.)
eDestDepth	ETcVnElementType [▶ 1615]		Destination image depth
nKernelSize	ULONG	1	Aperture size used to compute the second-derivative filters (1, 3, 5, ..., 31)
fScale	double	1	Scale factor for the computed derivative values
fDelta	double	0	Delta value that is added to the results before storing them
eBorderType	ETcVnBorderInterpolationMethod [▶ 1593]	BIM_DEFAULT	Image border handling

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16.10 LocalMaxima

Find local maxima in a gray-scale single-channel image. The found maxima are marked by a value of 1 in the destination image.

Syntax

Definition:

```
HRESULT LocalMaxima(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage,
    ITcVnImage*&  ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (USINT or UINT, 1 channel)
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (Must be different from ipSrcImage! An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16.11 LocalMinima

Find local minima in a gray-scale single-channel image. The found minima are marked by a value of 1 in the destination image.

Syntax

Definition:

```
HRESULT LocalMinima(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (USINT or UINT, 1 channel)
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (Must be different from ipSrcImage! An appropriate destination image will be created if required.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16.12 MedianFilter

Apply a Median filter to an image.

Syntax

Definition:

```
HRESULT MedianFilter(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage,
    ULONG        nFilterSize
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (for nFilterSize 3 or 5: USINT, UINT, REAL. For bigger filters, only USINT is supported.
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
nFilterSize	ULONG	Size (width and height) of the filter (3, 5, 7, ...)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16.13 MorphologicalOperator

Apply a morphological operator.

Syntax

Definition:

```
HRESULT MorphologicalOperator(
    HRESULT                    hrPrev,
    ITcVnImage*                ipSrcImage,
    ITcVnImage*&               ipDestImage,
    ETcVnMorphologicalOperator eOperator,
    ITcVnImage*                ipStructuringElement
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (only 1 channel supported for reconstruction operators)
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
eOperator	ETcVnMorphologicalOperator [▶ 1623]	Operator type
ipStructuringElement	ITcVnImage* [▶ 1797]	Structuring element to be applied (Typically created via F_VN_CreateStructuringElement.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16.14 RemoveLocalMaxima

Remove local maxima up to a given height from a gray-scale single-channel image.

Syntax

Definition:

```
HRESULT RemoveLocalMaxima(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage,
    double       fHeight
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (USINT, UINT, INT, REAL, or LREAL, 1 channel)
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
fHeight	double	Maximum height of the maxima to be removed (must be greater than 0)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16.15 RemoveLocalMinima

Remove local minima up to a given height from a gray-scale single-channel image.

Syntax

Definition:

```
HRESULT RemoveLocalMinima(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage,
    ITcVnImage*&  ipDestImage,
    double        fHeight
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (USINT, UINT, INT, REAL, or LREAL, 1 channel)
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
fHeight	double	Maximum height of the minima to be removed (must be greater than 0)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16.16 ScharrFilter

Calculates the first order derivative in x or y direction using a Scharr filter.

Syntax

Definition:

```
HRESULT ScharrFilter(
    HRESULT                        hrPrev,
    ITcVnImage*                    ipSrcImage,
    ITcVnImage*&                   ipDestImage,
    ETcVnElementType               eDestDepth,
    ETcVnFilterDirection           eFilterDirection,
    double                         fScale = 1,
    double                         fDelta = 0,
    ETcVnBorderInterpolationMethod eBorderType = BIM_DEFAULT
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (An appropriate destination image will be created if required.)
eDestDepth	ETcVnElementType [▶ 1615]		Destination image depth
eFilterDirection	ETcVnFilterDirection [▶ 1618]		Filter direction
fScale	double	1	Scale factor for the computed derivative values
fDelta	double	0	Delta value that is added to the results prior to storing them in dest
eBorderType	ETcVnBorderInterpolationMethod [▶ 1593]	BIM_DEFAULT	Image border handling

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16.17 SeparableCustomFilter

Apply a separable custom filter to the image.

Syntax

Definition:

```
HRESULT SeparableCustomFilter(
    HRESULT                       hrPrev,
    ITcVnImage*                   ipSrcImage,
    ITcVnImage*&                  ipDestImage,
    ETcVnElementType              eDestDepth,
    TcVnMatrix&                   stKernelX,
    TcVnMatrix&                   stKernelY,
    TcVnPoint&                    aAnchor = {-1, -1},
    double                        fDelta = 0,
    ETcVnBorderInterpolationMethod eBorderType = BIM_DEFAULT
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (An appropriate destination image will be created if required.)
eDestDepth	ETcVnElementType [▶ 1615]		Destination image depth
stKernelX	TcVnMatrix [▶ 1644]&		1D custom row-filter kernel with values of type REAL or LREAL
stKernelY	TcVnMatrix [▶ 1644]&		1D custom column-filter kernel with values of type REAL or LREAL
aAnchor	TcVnPoint [▶ 1588]&	{-1, -1}	Anchor point of the kernel
fDelta	double	0	Value added to each pixel after filtering
eBorderType	ETcVnBorderInterpolationMethod [▶ 1593]	BIM_DEFAULT	Image border handling

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16.18 SobelFilter

Calculates the first, second, third, or mixed image derivatives using an extended Sobel filter.

Syntax

Definition:

```
HRESULT SobelFilter(
    HRESULT                     hrPrev,
    ITcVnImage*                 ipSrcImage,
    ITcVnImage*&                ipDestImage,
    ETcVnElementType            eDestDepth,
    ULONG                       nXOrder,
    ULONG                       nYOrder,
    ULONG                       nKernelSize = 3,
    double                      fScale = 1,
```

```
    double                          fDelta = 0,
    ETcVnBorderInterpolationMethod eBorderType = BIM_DEFAULT
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (An appropriate destination image will be created if required.)
eDestDepth	ETcVnElementType [▶ 1615]		Destination image depth
nXOrder	ULONG		Order of the x-derivative (must be < nKernelSize)
nYOrder	ULONG		Order of the y-derivative (must be < nKernelSize)
nKernelSize	ULONG	3	Size of the extended Sobel kernel (3, 5, 7, ..., 31)
fScale	double	1	Scale factor for the computed derivative values
fDelta	double	0	Delta value that is added to the results prior to storing them in dest
eBorderType	ETcVnBorderInterpolationMethod [▶ 1593]	BIM_DEFAULT	Image border handling

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.16.19 VarianceFilter

Calculates the local variance of an image.

Syntax

Definition:

```
HRESULT VarianceFilter(
    HRESULT                         hrPrev,
    ITcVnImage*                     ipSrcImage,
    ITcVnImage*&                    ipDestImage,
    ULONG                           nKernelSize,
    double                          fScale = 1,
    ETcVnBorderInterpolationMethod eBorderType = BIM_DEFAULT
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image. Only TCVN_ET_USINT Type is supported.
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image of type TCVN_ET_UINT (An appropriate destination image will be created if required.)
nKernelSize	ULONG		Size of the kernel
fScale	double	1	Scale factor for the computed variance values
eBorderType	ETcVnBorderInterpolationMethod [▶ 1593]	BIM_DEFAULT	Image border handling

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.17 Image Segmentation

该组包含用于图像分割的函数。

函数

Segmentation

- AdaptiveThreshold [▶ 2413]
- CheckColorRange [▶ 2415]

- Threshold [▶ 2416]
- WatershedSegmentationExp [▶ 2417]

还请参阅有关此

DoubleThreshold [▶ 2415]

6.2.3.17.1 AdaptiveThreshold

Apply an adaptive threshold to a gray level image.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT AdaptiveThreshold(
    HRESULT                       hrPrev,
    ITcVnImage*                   ipSrcImage,
    ITcVnImage*&                  ipDestImage,
    double                        fMaxValue,
    ETcVnAdaptiveThresholdMethod eAdaptiveMethod = ATM_MEAN,
    ETcVnThresholdType            eThresholdType = TT_BINARY,
    ULONG                         nBlockSize = 3,
    double                        fConstant = 5
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (1 channel, elements of type USINT)
ipDestImage	ITcVnImage* [▶ 1797]&		Destination image (An appropriate destination image will be created if required.)
fMaxValue	double		Value assigned to pixels for which the threshold condition is true
eAdaptiveMethod	ETcVnAdaptiveThresholdMethod [▶ 1591]	ATM_MEAN	Adaptive threshold method to be applied (MEAN: threshold value will be calculated as a mean of the nBlockSize x nBlockSize neighborhood of (x,y) minus fConstant. GAUSSIAN: threshold value is a weighted sum (cross-correlation with a Gaussian window) of the nBlockSize x nBlockSize neighborhood of (x,y) minus fConstant.
eThresholdType	ETcVnThresholdType [▶ 1634]	TT_BINARY	Threshold type to be applied (only BINARY and BINARY_INV are supported)
nBlockSize	ULONG	3	Size of the pixel neighborhood to calculate the local threshold (3, 5, 7, ...)
fConstant	double	5	Constant that is subtracted from the weighted mean of the pixel neighborhood, which leads to the local threshold

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.17.2 CheckColorRange

Check if the pixel values of an image lie in a given range. The destination image has the same size as the source image but only one channel with a pixel element size of 8 bit. Its elements are set to 255 if the corresponding pixels of the source image are in the checked range and set to 0 otherwise.

Syntax

Definition:

```
HRESULT CheckColorRange(
    HRESULT             hrPrev,
    ITcVnImage*         ipSrcImage,
    ITcVnImage*&        ipDestImage,
    TcVnVector4_LREAL&  aLowerBounds,
    TcVnVector4_LREAL&  aUpperBounds
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
aLowerBounds	TcVnVector4_LREAL [▶ 1590]&	Channel-wise lower bounds (Unused channels are ignored.)
aUpperBounds	TcVnVector4_LREAL [▶ 1590]&	Channel-wise upper bounds (Unused channels are ignored.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.17.3 DoubleThreshold

Apply a double threshold (also called hysteresis threshold) using morphological reconstruction. Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT DoubleThreshold(
    HRESULT      hrPrev,
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipDestImage,
    double       fLowThreshold,
    double       fHighThreshold,
    double       fMaxValue
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (USINT, UINT, INT, REAL, or LREAL, 1 channel)
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
fLowThreshold	double	Low threshold determinining the shape of found regions
fHighThreshold	double	High threshold selecting found regions (Each region must at least contain a single pixel if the high threshold is applied.)
fMaxValue	double	Maximum pixel value

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.17.4 Threshold

Apply a fixed threshold or a dynamic threshold according to Otsu.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT Threshold(
    HRESULT            hrPrev,
    ITcVnImage*        ipSrcImage,
    ITcVnImage*&       ipDestImage,
    double             fThreshold,
```

```
    double             fMaxValue,
    ETcVnThresholdType eThresholdType
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (For Otsu and Triangle threshold types, only 1 channel of TCVN_ET_USINT is supported)
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (An appropriate destination image will be created if required.)
fThreshold	double	Fixed threshold (unused if dynamic thresholding is selected)
fMaxValue	double	Maximum pixel value
eThresholdType	ETcVnThresholdType [▶ 1634]	Threshold type to be applied

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.17.5 WatershedSegmentation (Exp)

Apply a marker-based watershed segmentation.

Syntax

Definition:

```
HRESULT WatershedSegmentation(
    HRESULT       hrPrev,
    ITcVnImage*   ipSrcImage,
    ITcVnImage*&  ipDestImage,
    ITcVnImage*   ipMarkers
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (USINT, 3 channels)
ipDestImage	ITcVnImage* [▶ 1797]&	Destination image (DINT, 1 channel. An appropriate image will be created if required.)
ipMarkers	ITcVnImage* [▶ 1797]	Marker image (DINT, 1 channel)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18 Keypoint Features

该组包含用于查找和匹配图像中明显特征的函数。

注意
计算时间长 许多查找关键点特征的函数都需要很长的计算时间。因此，请注意设定循环时间，且必要时使用Watchdog（看门狗） [▶ 127]，以保证实时行为。

函数

关键点

- KeyPointsAGAST [▶ 2436]
- KeyPointsFAST [▶ 2441]
- KeyPointsGFTT [▶ 2442]
- KeyPointsMSER [▶ 2443]
- KeyPointsSB [▶ 2444]

关键点和描述子

- KeyPointsAndDescriptorsAKAZE [▶ 2437]
- KeyPointsAndDescriptorsBRISK [▶ 2438]
- KeyPointsAndDescriptorsaKAZE [▶ 2439]
- KeyPointsAndDescriptorsORB [▶ 2440]

描述子匹配

- FilterGoodMatches [▶ 2419]
- GetMatchCoordinates [▶ 2435]

- MatchDescriptorsBF [▶ 2445]
- MatchDescriptorsFlannLsh [▶ 2447]
- MatchDescriptorsKnnBF [▶ 2448]
- MatchDescriptorsKnnFlannLsh [▶ 2450]

关键点引用匹配

- FindReferenceKeyPointsInImage [▶ 2420]
- FindReferenceKeyPointsInImageAKAZE [▶ 2423]
- FindReferenceKeyPointsInImageBRISK [▶ 2427]
- FindReferenceKeyPointsInImageORB [▶ 2431]

其它

- RegionsMSER [▶ 2451]

6.2.3.18.1 FilterGoodMatches

Filter the descriptor matches and return only good ones.

Syntax

Definition:

```
HRESULT FilterGoodMatches(
    HRESULT           hrPrev,
    ITcVnContainer*   ipMatches,
    ITcVnContainer*&  ipGoodMatches,
    float             fMaxDist,
    float             fMaxKnnRatio
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipMatches	ITcVnContainer* [▶ 1760]	Container with descriptor matches (ContainerType_Vector_TcVnDMatch or ContainerType_Vector_Vector_TcVnDMatch)
ipGoodMatches	ITcVnContainer* [▶ 1760]&	Returns a container with good matches (ContainerType_Vector_TcVnDMatch)
fMaxDist	float	Maximum allowed descriptor distance (-1 disables this filter criterion)
fMaxKnnRatio	float	Maximum allowed distance ratio between first and second best match ([0..1], used for knn match results only, -1 disables this filter criterion)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.2 FindReferenceKeyPointsInImage

Searches a reference image, represented by its keypoints and corresponding descriptors, in a given source image.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT FindReferenceKeyPointsInImage(
    HRESULT                   hrPrev,
    ITcVnContainer*           ipSrcKeyPoints,
    ITcVnImage*               ipSrcDescriptors,
    ITcVnContainer*           ipRefKeyPoints,
    ITcVnImage*               ipRefDescriptors,
    ULONG                     nRefImageWidth,
    ULONG                     nRefImageHeight,
    ITcVnContainer*&          ipEdgePoints,
    float                     fMaxDist,
    float                     fMaxKnnRatio,
    ETcVnNormalizationType    eNormType,
    TcVnMatrix3x3_LREAL&      aPerspectiveTransform,
    ULONG&                    nNumberOfGoodMatches,
    ULONG&                    nNumberOfInliers,
    ETcVnEstimationAlgorithm eAlgorithm = EA_RANSAC,
    double                    fReprojThreshold = 3,
    ULONG                     nMaxIterations = 2000,
    double                    fConfidence = 0.995,
    ITcVnContainer**          pipGoodMatches = nullptr,
    ITcVnContainer**          pipInlierMask = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcKeyPoints	ITcVnContainer* [▸ 1760]		KeyPoints of the source image, e.g. (ContainerType_Vector_TcVnKeyPoint)
ipSrcDescriptors	ITcVnImage* [▸ 1797]		Descriptors of the source image KeyPoints
ipRefKeyPoints	ITcVnContainer* [▸ 1760]		KeyPoints of the reference image (ContainerType_Vector_TcVnKeyPoint)
ipRefDescriptors	ITcVnImage* [▸ 1797]		Descriptors of the reference image KeyPoints
nRefImageWidth	ULONG		Width of the reference image in pixels
nRefImageHeight	ULONG		Height of the reference image in pixels
ipEdgePoints	ITcVnContainer* [▸ 1760]&		Returns a container with the 4 edge points of the reference image transformed into the coordinates of the source image (ContainerType_Vector_TcVnPoint2_REAL)
fMaxDist	float		Maximum allowed descriptor distance (-1 disables this filter criterion)
fMaxKnnRatio	float		Maximum allowed distance ratio between first and second best match ([0..1], -1 disables this filter criterion)
eNormType	ETcVnNormalizationType [▸ 1624]		Normalization type used for descriptor matching (HAMMING recommended for AKAZE, ORB, and BRISK. HAMMING2 is recommended for ORB if the ORB nBriefPoints parameter is 3 or 4.)
aPerspectiveTransform	TcVnMatrix3x3_LREAL [▸ 1590]&		Returns the perspective transformation matrix, which transforms the reference points to the source points
nNumberOfGoodMatches	ULONG&		Return the number of good matches
nNumberOfInliers	ULONG&		Return the number of inlier matches

Name	Type	Default	Description
eAlgorithm	ETcVnEstimationAlgorithm [▶ 1616]	EA_RANSAC	Estimation algorithm used for computing the perspective transformation between the two point sets
fReprojThreshold	double	3	Maximum allowed reprojection error to treat a point pair as an inlier (only used if eAlgorithm is RANSAC, RHO)
nMaxIterations	ULONG	2000	Maximum number of RANSAC iterations
fConfidence	double	0.995	Confidence (0..1)
pipGoodMatches	ITcVnContainer* [▶ 1760]*	nullptr	Returns a container with good matches (optional, set to 0 if not required; ContainerType_Vector_TcVnDMatch)
pipInlierMask	ITcVnContainer* [▶ 1760]*	nullptr	Returns a mask marking the inliers (optional, set to 0 if not required; ContainerType_Vector_SINT; only for RANSAC, LMEDS)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.3 FindReferenceKeyPointsInImageAKAZE

Searches a reference image, represented by its AKAZE keypoints and corresponding descriptors, in a given source image.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT FindReferenceKeyPointsInImageAKAZE(
    HRESULT                 hrPrev,
    ITcVnImage*             ipSrcImage,
    ITcVnContainer*         ipRefKeyPoints,
    ITcVnImage*             ipRefDescriptors,
    ULONG                   nRefImageWidth,
    ULONG                   nRefImageHeight,
    ITcVnContainer*&        ipEdgePoints,
    float                   fMaxDist,
```

```
    float                       fMaxKnnRatio,
    TcVnMatrix3x3_LREAL&        aPerspectiveTransform,
    ULONG&                      nNumberOfGoodMatches,
    ULONG&                      nNumberOfInliers,
    ETcVnNormalizationType      eNormType = NT_HAMMING,
    ITcVnImage*                 ipSrcImageMask = nullptr,
    TcVnParamsAKAZE&            stAkazeParams = {FDT_AKAZE_MLDB, 0, 3, 0.001f, 2, 1, DT1_KAZE_PM_G2},
    ETcVnEstimationAlgorithm eAlgorithm = EA_RANSAC,
    double                      fReprojThreshold = 3,
    ULONG                       nMaxIterations = 2000,
    double                      fConfidence = 0.995
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipRefKeyPoints	ITcVnContainer* [▶ 1760]		KeyPoints of the reference image, e.g. computed with F_VN_KeyPointsAndDescriptorsAKAZE (ContainerType_Vector_TcVnKeyPoint)
ipRefDescriptors	ITcVnImage* [▶ 1797]		Descriptors of the reference image KeyPoints, e.g. computed with F_VN_KeyPointsAndDescriptorsAKAZE
nRefImageWidth	ULONG		Width of the reference image in pixels
nRefImageHeight	ULONG		Height of the reference image in pixels
ipEdgePoints	ITcVnContainer* [▶ 1760]&		Returns a container with the 4 edge points of the reference image transformed into ipSrcImage coordinates (ContainerType_Vector_TcVnPoint2_REAL)
fMaxDist	float		Maximum allowed descriptor distance (-1 disables this filter criterion)
fMaxKnnRatio	float		Maximum allowed distance ratio between first and second best match ([0..1], -1 disables this filter criterion)
aPerspectiveTransform	TcVnMatrix3x3_LREAL [▶ 1590]&		Returns the perspective transformation matrix, which transforms the reference points to the source points
nNumberOfGoodMatches	ULONG&		Return the number of good matches
nNumberOfInliers	ULONG&		Return the number of inlier matches
eNormType	ETcVnNormalizationType [▶ 1624]	NT_HAMMING	Normalization type used for descriptor matching
ipSrcImageMask	ITcVnImage* [▶ 1797]	nullptr	Mask to specify, where to look for keypoints in ipSrcImage (set to 0 if no mask required)

Name	Type	Default	Description
stAkazeParams	TcVnParamsAKAZE [▶ 1646]&	{FDT_AKAZE_MLDB, 0, 3, 0.001f, 2, 1, DT1_KAZE_PM_G2}	Parameters to configure the keypoint and descriptor computation for ipSrcImage (resulting descriptors must be compatible to ipRefDescriptors!)
eAlgorithm	ETcVnEstimationAlgorithm [▶ 1616]	EA_RANSAC	Estimation algorithm used for computing the perspective transformation between the two point sets
fReprojThreshold	double	3	Maximum allowed reprojection error to treat a point pair as an inlier (only used if eAlgorithm is RANSAC, RHO)
nMaxIterations	ULONG	2000	Maximum number of RANSAC iterations
fConfidence	double	0.995	Confidence (0..1)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.4 FindReferenceKeyPointsInImageBRISK

Searches a reference image, represented by its BRISK keypoints and corresponding descriptors, in a given source image.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT FindReferenceKeyPointsInImageBRISK(
    HRESULT                  hrPrev,
    ITcVnImage*              ipSrcImage,
    ITcVnContainer*          ipRefKeyPoints,
    ITcVnImage*              ipRefDescriptors,
    ULONG                    nRefImageWidth,
    ULONG                    nRefImageHeight,
    ITcVnContainer*&         ipEdgePoints,
    float                    fMaxDist,
    float                    fMaxKnnRatio,
    TcVnMatrix3x3_LREAL&     aPerspectiveTransform,
    ULONG&                   nNumberOfGoodMatches,
    ULONG&                   nNumberOfInliers,
    ETcVnNormalizationType   eNormType = NT_HAMMING,
    ITcVnImage*              ipSrcImageMask = nullptr,
    TcVnParamsBRISK&         stBriskParams = {30, 3, 1},
    ETcVnEstimationAlgorithm eAlgorithm = EA_RANSAC,
```

```
    double                    fReprojThreshold = 3,
    ULONG                     nMaxIterations = 2000,
    double                    fConfidence = 0.995
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipRefKeyPoints	ITcVnContainer* [▶ 1760]		KeyPoints of the reference image, e.g. computed with F_VN_KeyPointsAndDescriptorsBRISK (ContainerType_Vector_TcVnKeyPoint)
ipRefDescriptors	ITcVnImage* [▶ 1797]		Descriptors of the reference image KeyPoints, e.g. computed with F_VN_KeyPointsAndDescriptorsBRISK
nRefImageWidth	ULONG		Width of the reference image in pixels
nRefImageHeight	ULONG		Height of the reference image in pixels
ipEdgePoints	ITcVnContainer* [▶ 1760]&		Returns a container with the 4 edge points of the reference image transformed into ipSrcImage coordinates (ContainerType_Vector_TcVnPoint2_REAL)
fMaxDist	float		Maximum allowed descriptor distance (-1 disables this filter criterion)
fMaxKnnRatio	float		Maximum allowed distance ratio between first and second best match ([0..1], -1 disables this filter criterion)
aPerspectiveTransform	TcVnMatrix3x3_LREAL [▶ 1590]&		Returns the perspective transformation matrix, which transforms the reference points to the source points
nNumberOfGoodMatches	ULONG&		Return the number of good matches
nNumberOfInliers	ULONG&		Return the number of inlier matches
eNormType	ETcVnNormalizationType [▶ 1624]	NT_HAMMING	Normalization type used for descriptor matching
ipSrcImageMask	ITcVnImage* [▶ 1797]	nullptr	Mask to specify, where to look for keypoints in ipSrcImage (set to 0 if no mask required)

Name	Type	Default	Description
stBriskParams	TcVnParamsBRISK [▶ 1650]&	{30, 3, 1}	Parameters to configure the keypoint and descriptor computation for ipSrcImage (resulting descriptors must be compatible to ipRefDescriptors!)
eAlgorithm	ETcVnEstimationAlgorithm [▶ 1616]	EA_RANSAC	Estimation algorithm used for computing the perspective transformation between the two point sets
fReprojThreshold	double	3	Maximum allowed reprojection error to treat a point pair as an inlier (only used if eAlgorithm is RANSAC, RHO)
nMaxIterations	ULONG	2000	Maximum number of RANSAC iterations
fConfidence	double	0.995	Confidence (0..1)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.5 FindReferenceKeyPointsInImageORB

Searches a reference image, represented by its ORB keypoints and corresponding descriptors, in a given source image.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT FindReferenceKeyPointsInImageORB(
    HRESULT                  hrPrev,
    ITcVnImage*              ipSrcImage,
    ITcVnContainer*          ipRefKeyPoints,
    ITcVnImage*              ipRefDescriptors,
    ULONG                    nRefImageWidth,
    ULONG                    nRefImageHeight,
    ITcVnContainer*&         ipEdgePoints,
    float                    fMaxDist,
    float                    fMaxKnnRatio,
    TcVnMatrix3x3_LREAL&     aPerspectiveTransform,
    ULONG&                   nNumberOfGoodMatches,
    ULONG&                   nNumberOfInliers,
    ETcVnNormalizationType   eNormType = NT_HAMMING,
    ITcVnImage*              ipSrcImageMask = nullptr,
    TcVnParamsORB&           stOrbParams = {500, 1.2f, 8, 31, 0, 2, FST_ORB_HARRIS, 31, 20},
    ETcVnEstimationAlgorithm eAlgorithm = EA_RANSAC,
```

```
    double                    fReprojThreshold = 3,
    ULONG                     nMaxIterations = 2000,
    double                    fConfidence = 0.995
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▸ 1797]		Source image
ipRefKeyPoints	ITcVnContainer* [▸ 1760]		KeyPoints of the reference image, e.g. computed with F_VN_KeyPointsAndDescriptorsORB (ContainerType_Vector_TcVnKeyPoint)
ipRefDescriptors	ITcVnImage* [▸ 1797]		Descriptors of the reference image KeyPoints, e.g. computed with F_VN_KeyPointsAndDescriptorsORB
nRefImageWidth	ULONG		Width of the reference image in pixels
nRefImageHeight	ULONG		Height of the reference image in pixels
ipEdgePoints	ITcVnContainer* [▸ 1760]&		Returns a container with the 4 edge points of the reference image transformed into ipSrcImage coordinates (ContainerType_Vector_TcVnPoint2_REAL)
fMaxDist	float		Maximum allowed descriptor distance (-1 disables this filter criterion)
fMaxKnnRatio	float		Maximum allowed distance ratio between first and second best match ([0..1], -1 disables this filter criterion)
aPerspectiveTransform	TcVnMatrix3x3_LREAL [▸ 1590]&		Returns the perspective transformation matrix, which transforms the reference points to the source points
nNumberOfGoodMatches	ULONG&		Return the number of good matches
nNumberOfInliers	ULONG&		Return the number of inlier matches
eNormType	ETcVnNormalizationType [▸ 1624]	NT_HAMMING	Normalization type used for descriptor matching (HAMMING recommended for ORB, HAMMING2 if the ORB nBriefPoints parameter is 3 or 4)

Name	Type	Default	Description
ipSrcImageMask	ITcVnImage* [▶ 1797]	nullptr	Mask to specify, where to look for keypoints in ipSrcImage (set to 0 if no mask required)
stOrbParams	TcVnParamsORB [▶ 1653]&	{500, 1.2f, 8, 31, 0, 2, FST_ORB_HARRIS, 31, 20}	Parameters to configure the keypoint and descriptor computation for ipSrcImage (resulting descriptors must be compatible to ipRefDescriptors!)
eAlgorithm	ETcVnEstimationAlgorithm [▶ 1616]	EA_RANSAC	Estimation algorithm used for computing the perspective transformation between the two point sets
fReprojThreshold	double	3	Maximum allowed reprojection error to treat a point pair as an inlier (only used if eAlgorithm is RANSAC, RHO)
nMaxIterations	ULONG	2000	Maximum number of RANSAC iterations
fConfidence	double	0.995	Confidence (0..1)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.6 GetMatchCoordinates

Return the coordinates of keypoints that match each other.

Syntax

Definition:

```
HRESULT GetMatchCoordinates(
    HRESULT          hrPrev,
    ITcVnContainer*  ipQueryKeyPoints,
    ITcVnContainer*  ipTrainKeyPoints,
    ITcVnContainer*  ipMatches,
    ITcVnContainer*& ipQueryCoordinates,
    ITcVnContainer*& ipTrainCoordinates
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipQueryKeyPoints	ITcVnContainer* [▶ 1760]	Container with query Keypoints (i.e. keypoints of a reference/template image) of type CTcVnContainer_Vector_TcVnKeyPoint
ipTrainKeyPoints	ITcVnContainer* [▶ 1760]	Container with training Keypoints (i.e. keypoints of a source/input image) of type CTcVnContainer_Vector_TcVnKeyPoint
ipMatches	ITcVnContainer* [▶ 1760]	Container with matches between query and train Keypoints (CTcVnContainer_Vector_TcVnDMatch).
ipQueryCoordinates	ITcVnContainer* [▶ 1760]&	Returns a container with the coordinates of keypoints from the query Keypoints that exist in ipMatches (CTcVnContainer_Vector_TcVnPoint2_REAL)
ipTrainCoordinates	ITcVnContainer* [▶ 1760]&	Returns a container with the coordinates of keypoints from the training Keypoints that exist in ipMatches (CTcVnContainer_Vector_TcVnPoint2_REAL)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.7 KeyPointsAGAST

Detects keypoints using the AGAST method.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT KeyPointsAGAST(
    HRESULT          hrPrev,
    ITcVnImage*      ipSrcImage,
    ITcVnContainer*& ipKeyPoints,
```

```
    ITcVnImage*       ipMask = nullptr,
    TcVnParamsAGAST& stParams = {10, true, KDT_AGAST_9_16}
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipKeyPoints	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipMask	ITcVnImage* [▶ 1797]	nullptr	Mask to specify, where to look for keypoints (set to 0 if no mask required)
stParams	TcVnParamsAGAST [▶ 1645]&	{10, true, KDT_AGAST_9_16}	Additional expert parameters

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.8 KeyPointsAndDescriptorsAKAZE

Detects keypoints and compute descriptors using the AKAZE method.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT KeyPointsAndDescriptorsAKAZE(
    HRESULT           hrPrev,
    ITcVnImage*       ipSrcImage,
    ITcVnContainer*& ipKeyPoints,
    ITcVnImage*&      ipDescriptors,
    ITcVnImage*       ipMask = nullptr,
    TcVnParamsAKAZE& stParams = {FDT_AKAZE_MLDB, 0, 3, 0.001f, 2, 1, DT1_KAZE_PM_G2}
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipKeyPoints	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipDescriptors	ITcVnImage* [▶ 1797]&		Descriptor image (set to 0 if not required; 1 descriptor in each row; An appropriate image will be created if required.)
ipMask	ITcVnImage* [▶ 1797]	nullptr	Mask to specify, where to look for keypoints (set to 0 if no mask required)
stParams	TcVnParamsAKAZE [▶ 1646]&	{FDT_AKAZE_MLDB, 0, 3, 0.001f, 2, 1, DT1_KAZE_PM_G2}	Additional expert parameters

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.9 KeyPointsAndDescriptorsBRISK

Detects keypoints and compute descriptors using the BRISK method.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT KeyPointsAndDescriptorsBRISK(
    HRESULT          hrPrev,
    ITcVnImage*      ipSrcImage,
    ITcVnContainer*& ipKeyPoints,
    ITcVnImage*&     ipDescriptors,
    ITcVnImage*      ipMask = nullptr,
    TcVnParamsBRISK& stParams = {30, 3, 1}
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipKeyPoints	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipDescriptors	ITcVnImage* [▶ 1797]&		Descriptor image (set to 0 if not required; 1 descriptor in each row; An appropriate image will be created if required.)
ipMask	ITcVnImage* [▶ 1797]	nullptr	Mask to specify, where to look for keypoints (set to 0 if no mask required)
stParams	TcVnParamsBRISK [▶ 1650]&	{30, 3, 1}	Additional expert parameters

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.10 KeyPointsAndDescriptorsKAZE

Detects keypoints and compute descriptors using the KAZE method.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT KeyPointsAndDescriptorsKAZE(
    HRESULT          hrPrev,
    ITcVnImage*      ipSrcImage,
    ITcVnContainer*& ipKeyPoints,
    ITcVnImage*&     ipDescriptors,
    ITcVnImage*      ipMask = nullptr,
    TcVnParamsKAZE&  stParams = {false, false, 0.001f, 4, 2, DT1_KAZE_PM_G2}
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipKeyPoints	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipDescriptors	ITcVnImage* [▶ 1797]&		Descriptor image (set to 0 if not required; 1 descriptor in each row; An appropriate image will be created if required.)
ipMask	ITcVnImage* [▶ 1797]	nullptr	Mask to specify, where to look for keypoints (set to 0 if no mask required)
stParams	TcVnParamsKAZE [▶ 1652]&	{false, false, 0.001f, 4, 2, DT1_KAZE_PM_G2}	Additional expert parameters

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.11 KeyPointsAndDescriptorsORB

Detect keypoints and compute descriptors using the ORB method.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT KeyPointsAndDescriptorsORB(
    HRESULT           hrPrev,
    ITcVnImage*       ipSrcImage,
    ITcVnContainer*&  ipKeyPoints,
    ITcVnImage*&      ipDescriptors,
    ITcVnImage*       ipMask = nullptr,
    TcVnParamsORB&    stParams = {500, 1.2f, 8, 31, 0, 2, FST_ORB_HARRIS, 31, 20}
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipKeyPoints	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipDescriptors	ITcVnImage* [▶ 1797]&		Descriptor image (set to 0 if not required; 1 descriptor in each row; An appropriate image will be created if required.)
ipMask	ITcVnImage* [▶ 1797]	nullptr	Mask to specify, where to look for keypoints (set to 0 if no mask required)
stParams	TcVnParamsORB [▶ 1653]&	{500, 1.2f, 8, 31, 0, 2, FST_ORB_HARRIS, 31, 20}	Additional expert parameters

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.12 KeyPointsFAST

Detects keypoints using the FAST method.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT KeyPointsFAST(
    HRESULT           hrPrev,
    ITcVnImage*       ipSrcImage,
    ITcVnContainer*&  ipKeyPoints,
    ITcVnImage*       ipMask = nullptr,
    TcVnParamsFAST&   stParams = {10, true, KDT_FAST_9_16}
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipKeyPoints	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipMask	ITcVnImage* [▶ 1797]	nullptr	Mask to specify, where to look for keypoints (set to 0 if no mask required)
stParams	TcVnParamsFAST [▶ 1651]&	{10, true, KDT_FAST_9_16}	Additional expert parameters

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.13 KeyPointsGFTT

Detects keypoints using the GFTT method, which detects strong corners.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT KeyPointsGFTT(
    HRESULT          hrPrev,
    ITcVnImage*      ipSrcImage,
    ITcVnContainer*& ipKeyPoints,
    ITcVnImage*      ipMask = nullptr,
    TcVnParamsGFTT&  stParams = {1000, 0.01, 1, 3, false, 0.04}
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipKeyPoints	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipMask	ITcVnImage* [▶ 1797]	nullptr	Mask to specify, where to look for keypoints (set to 0 if no mask required)
stParams	TcVnParamsGFTT [▶ 1651]&	{1000, 0.01, 1, 3, false, 0.04}	Additional expert parameters

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.14 KeyPointsMSER

Detects keypoints using the MSER method.

Syntax

Definition:

```
HRESULT KeyPointsMSER(
    HRESULT           hrPrev,
    ITcVnImage*       ipSrcImage,
    ITcVnContainer*&  ipKeyPoints,
    ITcVnImage*       ipMask = nullptr,
    TcVnParamsMSER&   stParams = {5, 60, 14400, 0.25, 0.2, 200, 1.01, 0.003, 5}
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipKeyPoints	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipMask	ITcVnImage* [▶ 1797]	nullptr	Mask to specify, where to look for keypoints (set to 0 if no mask required)
stParams	TcVnParamsMSER [▶ 1653]&	{5, 60, 14400, 0.25, 0.2, 200, 1.01, 0.003, 5}	Additional expert parameters

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.15 KeyPointsSB

Detects keypoints using a Simple Blob method - several iterations apply different thresholds to source image - connected components (blobs) are detected - the center and radius of the blobs are returned as keypoints
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT KeyPointsSB(
    HRESULT          hrPrev,
    ITcVnImage*      ipSrcImage,
    ITcVnContainer*& ipKeyPoints,
    ITcVnImage*      ipMask = nullptr,
    TcVnParamsSB&    stParams = {true, false, false, false, false, 25, 15000, 0, 1, 255, 0, 1, 0, 1,
5, 2, 30, 225, 10}
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipKeyPoints	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the keypoints (ContainerType_Vector_TcVnKeyPoint; Non-zero interface pointers are reused.)
ipMask	ITcVnImage* [▶ 1797]	nullptr	Mask to specify, where to look for keypoints (optional, set to 0 if not required)
stParams	TcVnParamsSB [▶ 1654]&	{true, false, false, false, false, 25, 15000, 0, 1, 255, 0, 1, 0, 1, 5, 2, 30, 225, 10}	Several parameters to filter the detected blobs

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.16 MatchDescriptorsBF

Match descriptors using a brute force approach.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT MatchDescriptorsBF(
    HRESULT                 hrPrev,
    ITcVnImage*             ipQueryDescriptors,
    ITcVnImage*             ipTrainDescriptors,
    ITcVnContainer*&        ipMatches,
    ITcVnImage*             ipMask = nullptr,
    ETcVnNormalizationType eNormType = NT_L2,
    bool                    bCrossCheck = false
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipQueryDescriptors	ITcVnImage* [▸ 1797]		Query descriptors (i.e. descriptors of a reference/template image)
ipTrainDescriptors	ITcVnImage* [▸ 1797]		Training descriptors (i.e. descriptors of a source/input image)
ipMatches	ITcVnContainer* [▸ 1760]&		Returns a container which is filled with the descriptor matches (ContainerType_Vector_TcVnDMatch; Non-zero interface pointers are reused.)
ipMask	ITcVnImage* [▸ 1797]	nullptr	Mask to specify permissable matches, i.e. query[i] can be matched with train[j] only if mask[i][j] != 0 (set to 0 if no mask required)
eNormType	ETcVnNormalizationType [▸ 1624]	NT_L2	Normalization type (only L1, L2, L2SQR, HAMMING, HAMMING2 supported). HAMMING2 should be used for ORB descriptors if the ORB nBriefPoints parameter is 3 or 4)
bCrossCheck	bool	false	If true, only consistent matches are returned, i.e. query->train and train->query detect the same match

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.17 MatchDescriptorsFlannLsh

Match descriptors using a FLANN based approach with LSH index.

Syntax

Definition:

```
HRESULT MatchDescriptorsFlannLsh(
    HRESULT          hrPrev,
    ITcVnImage*      ipQueryDescriptors,
    ITcVnImage*      ipTrainDescriptors,
    ITcVnContainer*& ipMatches,
    ULONG            nTableNumber = 12,
    ULONG            nKeySize = 20,
    ULONG            nMultiProbeLevel = 2,
    ULONG            nChecks = 32
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipQueryDescriptors	ITcVnImage* [▶ 1797]		Query descriptors (i.e. descriptors of a reference/template image)
ipTrainDescriptors	ITcVnImage* [▶ 1797]		Training descriptors (i.e. descriptors of a source/input image)
ipMatches	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the descriptor matches (ContainerType_Vector_TcVnDMatch; Non-zero interface pointers are reused.)
nTableNumber	ULONG	12	Number of tables
nKeySize	ULONG	20	Key size
nMultiProbeLevel	ULONG	2	Multi-probe level
nChecks	ULONG	32	Maximum number of visited leafs when searching for neighbors

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.18 MatchDescriptorsKnnBF

Match descriptors (k nearest neighbors) using a brute force approach.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT MatchDescriptorsKnnBF(
    HRESULT                 hrPrev,
    ITcVnImage*             ipQueryDescriptors,
    ITcVnImage*             ipTrainDescriptors,
    ITcVnContainer*&        ipMatches,
    ULONG                   nK,
    ITcVnImage*             ipMask = nullptr,
    bool                    bCompactResult = false,
    ETcVnNormalizationType eNormType = NT_L2,
    bool                    bCrossCheck = false
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipQueryDescriptors	ITcVnImage* [▶ 1797]		Query descriptors (i.e. descriptors of a reference/template image)
ipTrainDescriptors	ITcVnImage* [▶ 1797]		Training descriptors (i.e. descriptors of a source/input image)
ipMatches	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the descriptor matches (ContainerType_Vector_Vector_TcVnDMatch; Non-zero interface pointers are reused.)
nK	ULONG		Number of required best train matches for each query descriptor (i.e. nk := 2)
ipMask	ITcVnImage* [▶ 1797]	nullptr	Mask to specify permissable matches, i.e. query[i] can be matched with train[j] only if mask[i][j] != 0 (set to 0 if no mask required)
bCompactResult	bool	false	If true, matches vector does not contain entries for fully masked-out query descriptors
eNormType	ETcVnNormalizationType [▶ 1624]	NT_L2	Normalization type (only L1, L2, L2SQR, HAMMING, HAMMING2 supported). HAMMING2 should be used for ORB descriptors if the ORB nBriefPoints parameter is 3 or 4)
bCrossCheck	bool	false	If true, only consistent matches are returned, i.e. query->train and train->query detect the same match (only used if nK = 1)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.19 MatchDescriptorsKnnFlannLsh

Match descriptors (k nearest neighbors) using a FLANN based approach with LSH index.

Syntax

Definition:

```
HRESULT MatchDescriptorsKnnFlannLsh(
    HRESULT          hrPrev,
    ITcVnImage*      ipQueryDescriptors,
    ITcVnImage*      ipTrainDescriptors,
    ITcVnContainer*& ipMatches,
    ULONG            nK,
    ULONG            nTableNumber = 12,
    ULONG            nKeySize = 20,
    ULONG            nMultiProbeLevel = 2,
    ULONG            nChecks = 32
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipQueryDescriptors	ITcVnImage* [▶ 1797]		Query descriptors (i.e. descriptors of a reference/template image)
ipTrainDescriptors	ITcVnImage* [▶ 1797]		Training descriptors (i.e. descriptors of a source/input image)
ipMatches	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the descriptor matches (ContainerType_Vector_Vector_TcVnDMatch; Non-zero interface pointers are reused.)
nK	ULONG		Number of required best train matches for each query descriptor (i.e. nk := 2)
nTableNumber	ULONG	12	Number of tables
nKeySize	ULONG	20	Key size
nMultiProbeLevel	ULONG	2	Multi-probe level
nChecks	ULONG	32	Maximum number of visited leafs when searching for neighbors

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.18.20 RegionsMSER

Detects regions using the MSER method.

Syntax

Definition:

```
HRESULT RegionsMSER(
    HRESULT           hrPrev,
    ITcVnImage*       ipSrcImage,
    ITcVnContainer*&  ipRegions,
    ITcVnContainer*&  ipBoundingBoxes,
    TcVnParamsMSER&   stParams = {5, 60, 14400, 0.25, 0.2, 200, 1.01, 0.003, 5}
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image
ipRegions	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the region points (ContainerType_Vector_Vector_TcVnPoint2_DINT; Non-zero interface pointers are reused.)
ipBoundingBoxes	ITcVnContainer* [▶ 1760]&		Returns a container which is filled with the region bounding boxes (ContainerType_Vector_TcVnRectangle_DINT; Non-zero interface pointers are reused.)
stParams	TcVnParamsMSER [▶ 1653]&	{5, 60, 14400, 0.25, 0.2, 200, 1.01, 0.003, 5}	Additional expert parameters

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19 Machine Learning

该组包含用于创建、培训和应用机器学习模型的函数，以及用于特征预处理的函数。关于机器学习 [▶ 125]的可能应用的简要概述和进一步说明可在软件概念 [▶ 119]下找到。

模型创建函数：

- CreateKmppModel [▶ 2454]
- CreateKnnModel [▶ 2456]
- CreateLbgModel [▶ 2456]
- CreateLdaTransform [▶ 2459]
- CreateLdaTransformViaComponentNum [▶ 2460]
- CreateNbcModel [▶ 2461]
- CreatePcaTransform [▶ 2461]
- CreatePcaTransformViaComponentNum [▶ 2462]
- CreatePcaTransformViaVariance [▶ 2463]
- CreateRTreesModel [▶ 2464]
- CreateStaModel [▶ 2466]
- CreateSvmModel [▶ 2469]
- CreateSvmSgdClassifier [▶ 2473]

模型培训函数：

- TrainBatch [▶ 2490]
- TrainBatchClusters [▶ 2492]
- TrainSample [▶ 2492]
- TrainSampleClass [▶ 2493]
- TrainSampleCluster [▶ 2494]
- TrainSampleScalar [▶ 2495]
- TrainSampleVector [▶ 2495]

用于获取聚类模型上信息的函数：

- GetClusterCenter [▶ 2478]
- GetClusterNum [▶ 2479]

将模型应用于样本的函数：

- GetSampleCluster [▶ 2481]
- GetSampleNovelty [▶ 2482]
- PredictSampleClass [▶ 2488]
- PredictSampleScalar [▶ 2488]
- PredictSampleVector [▶ 2489]

将模型应用于批处理的函数：

- GetBatchClusters [▶ 2477]

- GetBatchNovelties [▶ 2478]
- PredictBatch [▶ 2486]

将模型应用于特征转换/降维的函数：

- FeatureTransform [▶ 2476]
- InverseFeatureTransform [▶ 2485]

特征标准化函数：

- GetFeatureScales [▶ 2480]
- FeatureScaling [▶ 2475]
- InverseFeatureScaling [▶ 2483]
- InverseFeatureScaling_REAL [▶ 2484]

还请参阅有关此

CreateBoostClassifier [▶ 2453]

6.2.3.19.1 CreateBoostClassifier

Create a Boost classifier. The initial reference count is set to one if a new model is created and kept, otherwise. The Boost classifier is only applicable to binary classification problems. It learns to distinguish between samples labelled with two user-defined class labels by incrementally adding weak classifiers to improve the classification results. Models of this type neither support on-line training (sample by sample) nor retraining.

Syntax

Definition:

```
HRESULT CreateBoostClassifier(
    HRESULT                  hrPrev,
    ITcVnMlModel*&           ipMlModel,
    ETcVnBoostClassifierType eType = BCT_REAL,
    ULONG                    nMaxDepth = 1,
    ULONG                    nMinSamples = 10,
    ULONG                    nWeakClassifiers = 100,
    double                   fWeightTrimRate = 0.95,
    ITcVnContainer*          ipClassPriors = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipMlModel	ITcVnMlModel* [▶ 1804]&		Returns the created model (Non-zero interface pointers are reused.)
eType	ETcVnBoostClassifierType [▶ 1593]	BCT_REAL	Learning algorithm type (default: TCVN_BCT_REAL)
nMaxDepth	ULONG	1	Maximum tree depth (default: 1)
nMinSamples	ULONG	10	Minimum number of samples within a node required for splitting (default: 10)
nWeakClassifiers	ULONG	100	Number of weak classifiers (default: 100)
fWeightTrimRate	double	0.95	Weight threshold used during training (off: 0; default: 0.95).
ipClassPriors	ITcVnContainer* [▶ 1760]	nullptr	Class priors (ContainerType_Vector_REAL or ContainerType_Vector_LREAL; only for classifiers; optional, set to 0 if not required or not allowed; default: 0)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.2 CreateKmppModel

Create a k-means++ model of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type neither support on-line training (sample by sample) nor retraining.

Syntax

Definition:

```
HRESULT CreateKmppModel(
    HRESULT                    hrPrev,
    ITcVnMlModel*&             ipMlModel,
    ETcVnPrototypeClusterer eKmppType,
    ULONG                      nK,
    bool                       bDoublePrecision = false,
    ULONG                      nMaxIterations = 0,
    double                     fEpsilon = 0.0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipMlModel	ITcVnMlModel* [▸ 1804]&		Returns the created model (Non-zero interface pointers are reused.)
eKmppType	ETcVnPrototypeClusterer [▸ 1629]		k-means++ model type
nK	ULONG		Parameter k (number of clusters)
bDoublePrecision	bool	false	If true, the model is generated with double precision (LREAL) instead of single precision (REAL). (default: FALSE)
nMaxIterations	ULONG	0	Maximum number of iterations (triggers the usage of the default value of 10 if it equals 0)
fEpsilon	double	0.0	Maximum allowed difference of the error between two successive iterations (triggers the usage of the default value of 0.001 if it equals 0)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.3 CreateKnnModel

Create a k-nearest neighbors model of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type support on-line training (sample by sample) and retraining. Predictions can only be scalar.

Syntax

Definition:

```
HRESULT CreateKnnModel(
    HRESULT         hrPrev,
    ITcVnMlModel*&  ipMlModel,
    ETcVnKnn        eKnnType,
    ULONG           nK
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipMlModel	ITcVnMlModel* [▶ 1804]&	Returns the created model (Non-zero interface pointers are reused.)
eKnnType	ETcVnKnn [▶ 1622]	k-nearest neighbors model type
nK	ULONG	Parameter k used for prediction (number of considered neighbors)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.4 CreateLbgModel

Create a LBG model of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type neither support on-line training (sample by sample) nor retraining.

Syntax

Definition:

```
HRESULT CreateLbgModel(
    HRESULT                  hrPrev,
    ITcVnMlModel*&           ipMlModel,
    ETcVnPrototypeClusterer eLbgType,
    ULONG                    nMaxClusters,
    double                   fMaxClusterRadius,
    bool                     bSingleSplitSteps,
    bool                     bDoublePrecision = false,
```

```
    ULONG                   nMaxIterations = 0,
    double                  fEpsilon = 0.0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipMlModel	ITcVnMlModel* [▶ 1804]&		Returns the created model (Non-zero interface pointers are reused.)
eLbgType	ETcVnPrototypeClusterer [▶ 1629]		LBG model type
nMaxClusters	ULONG		Maximum number of clusters
fMaxClusterRadius	double		Maximum allowed radius (L2 norm) of a single cluster, i.e. clusters with a higher radius will be split into smaller ones, until a global number of nMaxClusters is reached.
bSingleSplitSteps	bool		If true, the global optimization is always run after a single cluster has been split. If false, several clusters are split within the same step before applying the global optimization. Applying the global optimization less often is faster, but can lead to less optimal results, especially having 2 nearby clusters that could be represented by 1.
bDoublePrecision	bool	false	If true, the model is generated with double precision (LREAL) instead of single precision (REAL). (default: FALSE)
nMaxIterations	ULONG	0	Maximum number of iterations (triggers the usage of the default value of 10 if it equals 0)
fEpsilon	double	0.0	Maximum allowed difference of the error between two successive iterations (triggers the usage of the default value of 0.001 if it equals 0)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.5 CreateLdaTransform

Create an LDA-based feature transform from the provided data. The number of samples must be >= the number of features and the number of classes must be >= 2. The initial reference count is set to one if a new model is created and kept, otherwise.

Syntax

Definition:

```
HRESULT CreateLdaTransform(
    HRESULT          hrPrev,
    ITcVnMlModel*&   ipMlModel,
    ITcUnknown*      ipSamples,
    ITcVnContainer*  ipClasses
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipMlModel	ITcVnMlModel* [▶ 1804]&	Returns the created feature transform (Non-zero interface pointers are reused.)
ipSamples	ITcUnknown* [▶ 1810]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
ipClasses	ITcVnContainer* [▶ 1760]	Class labels corresponding to the input samples (ContainerType_Vector_DINT)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.6 CreateLdaTransformViaComponentNum

Create an LDA-based feature transform from the provided data where the number of components to be used is predefined. The number of samples must be >= the number of features and the number of classes must be >= 2. The initial reference count is set to one if a new model is created and kept, otherwise.

Syntax

Definition:

```
HRESULT CreateLdaTransformViaComponentNum(
    HRESULT          hrPrev,
    ITcVnMlModel*&   ipMlModel,
    ITcUnknown*      ipSamples,
    ITcVnContainer*  ipClasses,
    ULONG            nComponentNum
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipMlModel	ITcVnMlModel* [▶ 1804]&	Returns the created feature transform (Non-zero interface pointers are reused.)
ipSamples	ITcUnknown* [▶ 1810]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
ipClasses	ITcVnContainer* [▶ 1760]	Class labels corresponding to the input samples (ContainerType_Vector_DINT)
nComponentNum	ULONG	Number of components (nComponentNum must be smaller than the number of classes. A value of 0 results in the automatic computation of the maximum number of components.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.7 CreateNbcModel

Create a normal Bayes classifier of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. In order to train normal Bayes classifiers, a sufficiently high number of samples is required for each class. It depends on the number of features and the distribution of the data. Hence, it needs to be tested for each application. Models of this type do not support on-line training (sample by sample). For the retraining of such classifier models, the set of presented classes must be identical to the previous learning steps. Otherwise, an exception is raised.

Syntax

Definition:

```
HRESULT CreateNbcModel(
    HRESULT        hrPrev,
    ITcVnMlModel*& ipMlModel,
    ETcVnNbc       eNbcType
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipMlModel	ITcVnMlModel* [▶ 1804]&	Returns the created model (Non-zero interface pointers are reused.)
eNbcType	ETcVnNbc [▶ 1624]	Normal Bayes classifier type

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.8 CreatePcaTransform

Create a PCA-based feature transform from the provided data. The maximum number of principal components that can be computed equals the minimum of the number of samples and the number of features. The initial reference count is set to one if a new model is created and kept, otherwise.

Syntax

Definition:

```
HRESULT CreatePcaTransform(
    HRESULT         hrPrev,
    ITcVnMlModel*&  ipMlModel,
    ITcUnknown*     ipSamples
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipMlModel	ITcVnMlModel* [▶ 1804]&	Returns the created feature transform (Non-zero interface pointers are reused.)
ipSamples	ITcUnknown* [▶ 1810]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.9 CreatePcaTransformViaComponentNum

Create a PCA-based feature transform from the provided data where the number of prinicpal components to be used is predefined. The maximum number of principal components that can be computed equals the minimum of the number of samples and the number of features. The initial reference count is set to one if a new model is created and kept, otherwise.

Syntax

Definition:

```
HRESULT CreatePcaTransformViaComponentNum(
    HRESULT         hrPrev,
    ITcVnMlModel*&  ipMlModel,
    ITcUnknown*     ipSamples,
    ULONG           nComponentNum
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipMlModel	ITcVnMlModel* [▶ 1804]&	Returns the created feature transform (Non-zero interface pointers are reused.)
ipSamples	ITcUnknown* [▶ 1810]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
nComponentNum	ULONG	Number of principal components (A value of 0 results in the automatic computation of the maximum number of components.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.10 CreatePcaTransformViaVariance

Create a PCA-based feature transform from the provided data based on a given fraction of variance to be retained. The maximum number of principal components that can be computed equals the minimum of the number of samples and the number of features. The initial reference count is set to one if a new model is created and kept, otherwise.

Syntax

Definition:

```
HRESULT CreatePcaTransformViaVariance(
    HRESULT         hrPrev,
    ITcVnMlModel*&  ipMlModel,
    ITcUnknown*     ipSamples,
    double          fRetainedVariance
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipMlModel	ITcVnMlModel* [▶ 1804]&	Returns the created feature transform (Non-zero interface pointers are reused.)
ipSamples	ITcUnknown* [▶ 1810]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
fRetainedVariance	double	Fraction of variance that is to be retained by the PCA (A value of 1.0 signifies 100%.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.11 CreateRTreesModel

Create a random trees model of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type neither support on-line training (sample by sample) nor retraining. Predictions can only be scalar.

Syntax

Definition:

```
HRESULT CreateRTreesModel(
    HRESULT          hrPrev,
    ITcVnMlModel*&   ipMlModel,
    ETcVnRTrees      eRTreesType,
    ULONG            nMaxDepth = 5,
    ULONG            nMinSamples = 10,
    ULONG            nActiveVariables = 0,
    ULONG            nMaxIterations = 0,
    double           fEpsilon = 0.0,
    float            fRegressionAccuracy = 0,
    ITcVnContainer* ipClassPriors = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipMlModel	ITcVnMlModel* [▶ 1804]&		Returns the created model (Non-zero interface pointers are reused.)
eRTreesType	ETcVnRTrees [▶ 1630]		Random trees model type
nMaxDepth	ULONG	5	Maximum tree depth (default: 5)
nMinSamples	ULONG	10	Minimum number of samples within a node required for splitting (default: 10)
nActiveVariables	ULONG	0	Number of variables considered for splitting (0 means sqrt(total number of variables); default: 0)
nMaxIterations	ULONG	0	Maximum number of iterations (disabled if it equals 0 and fEpsilon is different from 0.0; triggers the usage of the default value of 50 if nMaxIterations and fEpsilon equal 0)
fEpsilon	double	0.0	Maximum allowed difference of the error between two successive iterations (disabled if it equals 0.0 and nMaxIterations is different from 0; triggers the usage of the default value of 0.1 if nMaxIterations and fEpsilon equal 0)
fRegressionAccuracy	float	0	Termination criterion for regressors (only for regressors; set to default if not allowed; default: 0.0)
ipClassPriors	ITcVnContainer* [▶ 1760]	nullptr	Class priors (ContainerType_Vector_REAL or ContainerType_Vector_LREAL; only for classifiers; optional, set to 0 if not required or not allowed; default: 0)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.12 CreateStaModel

Create a Simplified TopoART neural network of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type support on-line training (sample by sample), retraining, as well as scalar and vectorial predictions. It requires all input except class labels (i.e., samples and training outputs/predictions) to lie in the interval [0.0, 1.0]. The predictions of regressors need to be rescaled from the interval [0.0, 1.0] to their respective value range before usage. Depending on the parameter settings and the number of available training samples, repeated training with the same data may improve the results. Like other neural networks based on the Adaptive Resonance Theory (ART), Simplified TopoART neural networks are not prone to catastrophic inference and patricularly well-suited to incremental learning tasks. (additional expert function providing parameters for fine-tuning and noise reduction)

Syntax

Definition:

```
HRESULT CreateStaModel(
    HRESULT        hrPrev,
    ITcVnMlModel*& ipMlModel,
    ETcVnSta       eStaType,
    double         fRho,
    ULONG          nNu = 3,
    bool           bDoublePrecision = false,
    double         fBetaSbm = 0.0,
    ULONG          nPhi = 1,
    ULONG          nTau = 100
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipMlModel	ITcVnMlModel* [▶ 1804]&		Returns the created model (Non-zero interface pointers are reused.)
eStaType	ETcVnSta [▶ 1631]		Simpified TopoART model type
fRho	double		Vigilance parameter (controls the number of neurons that are inserted and the maximum size of the formed categories; valid range: [0.0, 1.0]; suggested range: [0.8, 1.0))
nNu	ULONG	3	Number of neurons used for classification and prediction (must be larger than or equal to 1; suggested range: [1, 10); default: 3)
bDoublePrecision	bool	false	If true, the model is generated with double precision (LREAL) instead of single precision (REAL). (default: FALSE)
fBetaSbm	double	0.0	Learning rate of the second best-matching neuron (learning the second best-matching neuron keeps related categories closer together and might improve the results but may require a higher number of neurons and training runs; a value of 0.0 disables learning of the second best-matching neuron; valid range: [0.0, 1.0]; suggested range: [0.0, 0.5]); default: 0.0

Name	Type	Default	Description
nPhi	ULONG	1	Number of samples a neuron must have learnt to become permanent (required for noise reduction; works in conjunction with nTau; must be larger than or equal to 1; higher values intensify noise reduction; a value of 1 disables removal of neuron candidates; default: 1)
nTau	ULONG	100	Number of learning steps after which node removal is performed (required for noise reduction; works in conjunction with nPhi; must be large enough to allow neuron candidates representing non-noise samples to become the best-matching neuron at least nPhi times before node removal; default: 100)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.13 CreateSvmModel

Create an SVM model of the specified type. The initial reference count is set to one if a new model is created and kept, otherwise. Models of this type neither support on-line training (sample by sample) nor retraining. Predictions can only be scalar. (additional expert function for C support vector classifiers)

Syntax

Definition:

```
HRESULT CreateSvmModel(
    HRESULT             hrPrev,
    ITcVnMlModel*&      ipMlModel,
    ETcVnSvm            eSvmType,
    double              fC,
    double              fNu,
    double              fP,
    ETcVnSvmKernelType eKernelType,
    double              fGamma,
```

```
    double              fCoef0,
    double              fDegree,
    ULONG               nMaxIterations = 0,
    double              fEpsilon = 0.0,
    ITcVnContainer*     ipClassWeights = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipMlModel	ITcVnMlModel* [▶ 1804]&		Returns the created model (Non-zero interface pointers are reused.)
eSvmType	ETcVnSvm [▶ 1632]		SVM model type
fC	double		Parameter C (required for TCVN_SVM_C_CLASSIFIER, TCVN_SVM_EPS_REGRESSOR, and TCVN_SVM_NU_REGRESSOR; ignored otherwise)
fNu	double		Parameter nu (required for TCVN_SVM_NU_CLASSIFIER, TCVN_SVM_NOVELTY_DETECTOR, and TCVN_SVM_NU_REGRESSOR; ignored otherwise)
fP	double		Parameter p (required for TCVN_SVM_EPS_REGRESSOR; ignored otherwise)
eKernelType	ETcVnSvmKernelType [▶ 1632]		Kernel type
fGamma	double		Parameter gamma (used by polynomial, RBF, sigmoid, and chi-squared kernels; ignored otherwise)
fCoef0	double		Parameter coef0 (used by polynomial and sigmoid kernels; ignored otherwise)
fDegree	double		Degree (used by polynomial kernels; ignored otherwise)
nMaxIterations	ULONG	0	Maximum number of iterations (disabled if it equals 0 and fEpsilon is different from 0.0; triggers the usage of the default value of 100000 if nMaxIterations and fEpsilon equal 0)

Name	Type	Default	Description
fEpsilon	double	0.0	Maximum allowed difference of the error between two successive iterations (disabled if it equals 0.0 and nMaxIterations is different from 0; triggers the usage of the default value of 0.00001 if nMaxIterations and fEpsilon equal 0)
ipClassWeights	ITcVnContainer* [▶ 1760]	nullptr	Class weights (ContainerType_Vector_REAL or ContainerType_Vector_LREAL; only valid if eSvmType equals TCVN_SVM_C_CLASSIFIER; optional, set to 0 if not required or not allowed; default: 0)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.14 CreateSvmSgdClassifier

Create a linear SVM classifier using stochastic gradient descent for training. The initial reference count is set to one if a new model is created and kept, otherwise. This SVM classifier is only applicable to binary classification problems. It learns a separating hyperplane between a class with label -1 and a class with label 1. These class labels are predefined. For training, any positive class labels are mapped to 1 and any negative class labels are mapped to -1. Models of this type neither support on-line training (sample by sample) nor retraining.

Syntax

Definition:

```
HRESULT CreateSvmSgdClassifier(
    HRESULT                         hrPrev,
    ITcVnMlModel*&                  ipMlModel,
    ETcVnSvmSgdClassifierType       eType = SSCT_ASGD,
    ETcVnSvmSgdClassifierMarginType eMarginType = SSCMT_SOFT_MARGIN,
    float                           fMarginRegularization = 0.00001f,
    float                           fInitialStepSize = 0.05f,
    float                           fStepDecreasingPower = 0.75f,
    ULONG                           nMaxIterations = 0,
    double                          fEpsilon = 0.0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipMlModel	ITcVnMlModel* [▶ 1804]&		Returns the created model (Non-zero interface pointers are reused.)
eType	ETcVnSvmSgdClassifierType [▶ 1633]	SSCT_ASGD	Learning algorithm type (default: TCVN_SSCT_ASGD)
eMarginType	ETcVnSvmSgdClassifierMarginType [▶ 1633]	SSCMT_SOFT_MARGIN	Margin type (default: TCVN_SSCMT_SOFT_MARGIN)
fMarginRegularization	float	0.00001f	Margin regularization parameter (default: 0.00001)
fInitialStepSize	float	0.05f	Initial step size (default: 0.05)
fStepDecreasingPower	float	0.75f	Power parameter (default: 0.75)
nMaxIterations	ULONG	0	Maximum number of iterations (disabled if it equals 0 and fEpsilon is different from 0.0; triggers the usage of the default value of 100000 if nMaxIterations and fEpsilon equal 0)
fEpsilon	double	0.0	Maximum allowed difference of the error between two successive iterations (disabled if it equals 0.0 and nMaxIterations is different from 0; triggers the usage of the default value of 0.00001 if nMaxIterations and fEpsilon equal 0)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.15 FeatureScaling

Apply a feature scaling to one or more sample(s) and get output range between [fA, fB] in the case of TCVN_FST1_MINMAX.

Syntax

Definition:

```
HRESULT FeatureScaling(
    HRESULT           hrPrev,
    ITcVnContainer*   ipSamples,
    ITcVnContainer*   ipScales,
    ITcVnContainer*&  ipScaledSamples,
    double            fA = 0.0,
    double            fB = 1.0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSamples	ITcVnContainer* [▶ 1760]		Container holding one or more input sample(s) (ContainerType_Vector_REAL, ContainerType_Vector_LREAL, ContainerType_Vector_Vector_REAL, or ContainerType_Vector_Vector_LREAL)
ipScales	ITcVnContainer* [▶ 1760]		Container with the scaling parameters of each feature (type ContainerType_Vector_REAL or ContainerType_Vector_LREAL).
ipScaledSamples	ITcVnContainer* [▶ 1760]&		Returns the scaled sample(s) using the same type like ipSamples. If the same container ipSamples is uesd, the source data will be replaced.
fA	double	0.0	represents the lower bound of the range for TCVN_FST1_MINMAX
fB	double	1.0	represents the upper bound of the range for TCVN_FST1_MINMAX

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.16 FeatureTransform

Apply a feature transform to one or more sample(s).

Syntax

Definition:

```
HRESULT FeatureTransform(
    HRESULT        hrPrev,
    ITcVnMlModel* ipFeatureTransform,
    ITcUnknown*    ipSamples,
    ITcUnknown**   pipTransformed
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipFeatureTransform	ITcVnMlModel* [▶ 1804]	Feature transform instance to be used
ipSamples	ITcUnknown* [▶ 1810]	Container holding one or more of input sample(s) (ContainerType_Vector_REAL, ContainerType_Vector_LREAL, ContainerType_Vector_Vector_REAL, or ContainerType_Vector_Vector_LREAL)
pipTransformed	ITcUnknown* [▶ 1810]*	Returns the transformed sample(s) using the same type like ipSamples

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.17 GetBatchClusters

Get the IDs of the best-matching clusters of a batch of samples.

Syntax

Definition:

```
HRESULT GetBatchClusters(
    HRESULT          hrPrev,
    ITcVnMlModel*    ipClusterer,
    ITcUnknown*      ipSamples,
    ITcVnContainer*& ipClusters,
    ITcVnContainer** pipNovelties = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipClusterer	ITcVnMlModel* [▶ 1804]		Clusterer to be used
ipSamples	ITcUnknown* [▶ 1810]		Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
ipClusters	ITcVnContainer* [▶ 1760]&		Returns the cluster IDs (ContainerType_Vector_DINT)
pipNovelties	ITcVnContainer* [▶ 1760]*	nullptr	Returns the degree of novelty (0.0 if a sample is completely known; > 0.0 otherwise) of each sample (ContainerType_Vector_REAL; optional, set to 0 if not required)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.18 GetBatchNovelties

Get the degree of novelty of multiple samples.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT GetBatchNovelties(
    HRESULT          hrPrev,
    ITcVnMlModel*    ipNoveltyDetector,
    ITcUnknown*      ipSamples,
    ITcVnContainer*& ipNovelties
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipNoveltyDetector	ITcVnMlModel* [▶ 1804]	Novelty detector to be used
ipSamples	ITcUnknown* [▶ 1810]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
ipNovelties	ITcVnContainer* [▶ 1760]&	Returns the degree of novelty (0.0 if a sample is completely known; > 0.0 otherwise) of each sample (ContainerType_Vector_REAL)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.19 GetClusterCenter

Get the center of a cluster.

Syntax

Definition:

```
HRESULT GetClusterCenter(
    HRESULT           hrPrev,
    ITcVnMlModel*     ipClusterer,
    LONG              nCluster,
    ITcVnContainer*&  ipCenter
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipClusterer	ITcVnMlModel* [▶ 1804]	Clusterer to be used
nCluster	LONG	Cluster ID of the cluster the center of which is requested
ipCenter	ITcVnContainer* [▶ 1760]&	Returns the cluster center (ContainerType_Vector_REAL or ContainerType_Vector_LREAL depending on the floating point type internally used by the clusterer)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.20 GetClusterNum

Get the number of clusters used by this clusterer.

Syntax

Definition:

```
HRESULT GetClusterNum(
    HRESULT       hrPrev,
    ITcVnMlModel* ipClusterer,
    ULONGLONG&    nClusterNum
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipClusterer	ITcVnMlModel* [▶ 1804]	Clusterer to be used
nClusterNum	ULONGLONG&	Returns the number of clusters

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.21 GetFeatureScales

Calculate the scaling parameters for each feature in the input samples based on the scaling type.

Syntax

Definition:

```
HRESULT GetFeatureScales(
    HRESULT                  hrPrev,
    ITcVnContainer*          ipSamples,
    ITcVnContainer*&         ipScales,
    ETcVnFeatureScalingType eFeatureScalingType
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSamples	ITcVnContainer* [▶ 1760]	Container holding input sample(s) (ContainerType_Vector_REAL, ContainerType_Vector_LREAL, ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
ipScales	ITcVnContainer* [▶ 1760]&	Returns a container with the scaling parameters for each feature (Vector_REAL or Vector_LREAL depending on the type of ipSamples).
eFeatureScalingType	ETcVnFeatureScalingType [▶ 1617]	Feature scaling type

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.22 GetSampleCluster

Get the ID of the best-matching cluster of a single sample.

Syntax

Definition:

```
HRESULT GetSampleCluster(
    HRESULT       hrPrev,
    ITcVnMlModel* ipClusterer,
    ITcUnknown*   ipSample,
    LONG&         nCluster,
    float*        pNovelty = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipClusterer	ITcVnMlModel* [▶ 1804]		Clusterer to be used
ipSample	ITcUnknown* [▶ 1810]		Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
nCluster	LONG&		Returns the ID of the cluster the sample has been assigned to
pNovelty	float*	nullptr	Returns the degree of novelty (0.0 if a sample is completely known; > 0.0 otherwise) of the presented sample (optional, set to 0 if not required)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.23 GetSampleNovelty

Get the degree of novelty of a single sample.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT GetSampleNovelty(
    HRESULT         hrPrev,
    ITcVnMlModel*  ipNoveltyDetector,
    ITcUnknown*    ipSample,
    float&         fNovelty
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipNoveltyDetector	ITcVnMlModel* [▶ 1804]	Novelty detector to be used
ipSample	ITcUnknown* [▶ 1810]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
fNovelty	float&	Returns the degree of novelty (0.0 if a sample is completely known; > 0.0 otherwise) of the presented sample

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.24 InverseFeatureScaling

Apply the inverse of feature scaling to one or more sample(s). [fA, fB] should be used if the scaling has been performed using TCVN_FST1_MINMAX with a predetermined output range..

Syntax

Definition:

```
HRESULT InverseFeatureScaling(
    HRESULT          hrPrev,
    ITcVnContainer*  ipSamples,
    ITcVnContainer*  ipScales,
    ITcVnContainer*& ipInverseSamples,
    double           fA = 0.0,
    double           fB = 1.0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSamples	ITcVnContainer* [▶ 1760]		Container holding one or more input sample(s) (ContainerType_Vector_REAL, ContainerType_Vector_LREAL, ContainerType_Vector_Vector_REAL, or ContainerType_Vector_Vector_LREAL)
ipScales	ITcVnContainer* [▶ 1760]		Container with the scaling parameters of each feature (type ContainerType_Vector_REAL or ContainerType_Vector_LREAL).
ipInverseSamples	ITcVnContainer* [▶ 1760]&		Returns the inversed sample(s) using the scaling type and parameters of ipScales. It has the same type of ipSamples. If the same container ipSamples is used, the source data will be replaced.
fA	double	0.0	represents the lower bound of the range for TCVN_FST1_MINMAX
fB	double	1.0	represents the upper bound of the range for TCVN_FST1_MINMAX

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.25 InverseFeatureScaling (float)

Apply the inverse of feature scaling to a single value. [fA, fB] should be used if the scaling has been performed using TCVN_FST1_MINMAX with a predetermined output range.

Syntax

Definition:

```
HRESULT InverseFeatureScaling(
    HRESULT          hrPrev,
    float            fSample,
    ITcVnContainer* ipScales,
    float&           fInverseSample,
    double           fA = 0.0,
    double           fB = 1.0
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
fSample	float		Sample value
ipScales	ITcVnContainer* [▶ 1760]		Container with the scaling parameters of each feature (type ContainerType_Vector_REAL or ContainerType_Vector_LREAL).
fInverseSample	float&		Returns the inversed sample.
fA	double	0.0	represents the lower bound of the range for TCVN_FST1_MINMAX
fB	double	1.0	represents the upper bound of the range for TCVN_FST1_MINMAX

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.26 InverseFeatureTransform

Apply an inverse feature transform to one or more sample(s).

Syntax

Definition:

```
HRESULT InverseFeatureTransform(
    HRESULT       hrPrev,
    ITcVnMlModel* ipFeatureTransform,
    ITcUnknown*   ipSamples,
    ITcUnknown**  pipTransformed
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipFeatureTransform	ITcVnMlModel* [▶ 1804]	Feature transform instance to be used
ipSamples	ITcUnknown* [▶ 1810]	Container holding one or more of input sample(s) (ContainerType_Vector_REAL, ContainerType_Vector_LREAL, ContainerType_Vector_Vector_REAL, or ContainerType_Vector_Vector_LREAL)
pipTransformed	ITcUnknown* [▶ 1810]*	Returns the transformed sample(s) using the same type like ipSamples

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.27 PredictBatch

Compute predictions for a batch of samples.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT PredictBatch(
    HRESULT           hrPrev,
    ITcVnMlModel*     ipMlModel,
    ITcUnknown*       ipSamples,
    ITcVnContainer*& ipPredictions,
    ITcVnContainer** pipNovelties = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipMlModel	ITcVnMlModel* [▶ 1804]		Classifier or regressor to be used
ipSamples	ITcUnknown* [▶ 1810]		Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
ipPredictions	ITcVnContainer* [▶ 1760]&		Returns the predicted outputs (depending on ipSamples; class labels (for classification, ContainerType_Vector_DINT) or real-valued predictions (for regression with scalar output, ContainerType_Vector_REAL or ContainerType_Vector_LREAL; for regression with vectorial output, ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL))
pipNovelties	ITcVnContainer* [▶ 1760]*	nullptr	Returns the degree of novelty (0.0 if a sample is completely known; > 0.0 otherwise) of each sample (ContainerType_Vector_REAL; optional, set to 0 if not required)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.28 PredictSampleClass

Classify a single sample.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT PredictSampleClass(
    HRESULT       hrPrev,
    ITcVnMlModel* ipClassifier,
    ITcUnknown*   ipSample,
    LONG&         nClass,
    float*        pNovelty = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipClassifier	ITcVnMlModel* [▶ 1804]		Classifier to be used
ipSample	ITcUnknown* [▶ 1810]		Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
nClass	LONG&		Returns the classification result
pNovelty	float*	nullptr	Returns the degree of novelty (0.0 if a sample is completely known; > 0.0 otherwise) of the presented sample (optional, set to 0 if not required)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.29 PredictSampleScalar

Compute a scalar prediction for a single sample.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT PredictSampleScalar(
    HRESULT       hrPrev,
    ITcVnMlModel* ipRegressor,
    ITcUnknown*   ipSample,
    float&        fPrediction,
    float*        pNovelty = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipRegressor	ITcVnMlModel* [▶ 1804]		Regressor to be used
ipSample	ITcUnknown* [▶ 1810]		Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
fPrediction	float&		Returns the predicted output
pNovelty	float*	nullptr	Returns the degree of novelty (0.0 if a sample is completely known; > 0.0 otherwise) of the presented sample (optional, set to 0 if not required)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.30 PredictSampleVector

Compute a vectorial prediction for a single sample.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT PredictSampleVector(
    HRESULT           hrPrev,
    ITcVnMlModel*     ipRegressor,
    ITcUnknown*       ipSample,
    ITcVnContainer*&  ipPrediction,
    float*            pNovelty = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▸ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipRegressor	ITcVnMlModel* [▸ 1804]		Regressor to be used
ipSample	ITcUnknown* [▸ 1810]		Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
ipPrediction	ITcVnContainer* [▸ 1760]&		Returns the predicted output (ContainerType_Vector_REAL or ContainerType_Vector_LREAL, depending on ipSample)
pNovelty	float*	nullptr	Returns the degree of novelty (0.0 if a sample is completely known; > 0.0 otherwise) of the presented sample (optional, set to 0 if not required)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.31 TrainBatch

Train a classifier, a clusterer, a novelty detector, or a regressor with a batch of samples. Online trainable models are trained once with each sample. Depending on the application and the number of available training samples, repeated training of such models with the same data may improve the results.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT TrainBatch(
    HRESULT          hrPrev,
    ITcVnMlModel*    ipMlModel,
    ITcUnknown*      ipSamples,
    ITcVnContainer* ipOutputs
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipMlModel	ITcVnMlModel* [▶ 1804]	Classifier, clusterer, novelty detector, or regressor to be used
ipSamples	ITcUnknown* [▶ 1810]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
ipOutputs	ITcVnContainer* [▶ 1760]	Class labels (for classification, ContainerType_Vector_DINT) or target outputs (for regression with scalar output, ContainerType_Vector_REAL or ContainerType_Vector_LREAL; for regression with vectorial output, ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL) to be learnt (must be 0 for clusterers and for novelty detectors)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.32 TrainBatchClusters

Train a clusterer with a batch of samples and return the IDs of the clusters the samples have been assigned to, if requested. On-line trainable clusterers are trained once with each sample. Depending on the application and the number of available training samples, repeated training of such models with the same data may improve the results.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT TrainBatchClusters(
    HRESULT           hrPrev,
    ITcVnMlModel*     ipClusterer,
    ITcUnknown*       ipSamples,
    ITcVnContainer*& ipClusters
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipClusterer	ITcVnMlModel* [▶ 1804]	Clusterer to be used
ipSamples	ITcUnknown* [▶ 1810]	Container holding a batch of input samples (ContainerType_Vector_Vector_REAL or ContainerType_Vector_Vector_LREAL)
ipClusters	ITcVnContainer* [▶ 1760]&	Returns the IDs of the clusters the samples have been assigned to (ContainerType_Vector_DINT; optional, set to 0 if not required)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.33 TrainSample

Train a clusterer or a novelty detector with a single sample.

Syntax

Definition:

```
HRESULT TrainSample(
    HRESULT       hrPrev,
    ITcVnMlModel* ipMlModel,
    ITcUnknown*   ipSample
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipMlModel	ITcVnMlModel* [▶ 1804]	Clusterer or novelty detector to be used
ipSample	ITcUnknown* [▶ 1810]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.34 TrainSampleClass

Train a classifier with a single sample.

Syntax

Definition:

```
HRESULT TrainSampleClass(
    HRESULT       hrPrev,
    ITcVnMlModel* ipClassifier,
    ITcUnknown*   ipSample,
    LONG          nClass
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipClassifier	ITcVnMlModel* [▶ 1804]	Classifier to be used
ipSample	ITcUnknown* [▶ 1810]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
nClass	LONG	Class label to be learnt

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.35 TrainSampleCluster

Train a clusterer with a single sample and return the ID of the cluster the sample has been assigned to, if requested. The clusterer must be online trainable.

Syntax

Definition:

```
HRESULT TrainSampleCluster(
    HRESULT       hrPrev,
    ITcVnMlModel* ipClusterer,
    ITcUnknown*   ipSample,
    LONG*         pCluster
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipClusterer	ITcVnMlModel* [▶ 1804]	Clusterer to be used
ipSample	ITcUnknown* [▶ 1810]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
pCluster	LONG*	Returns the ID of the cluster the sample has been assigned to (If the pointer is 0, no cluster ID is requested.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.36 TrainSampleScalar

Train a regressor with a single sample and scalar output.

Syntax

Definition:

```
HRESULT TrainSampleScalar(
    HRESULT        hrPrev,
    ITcVnMlModel*  ipRegressor,
    ITcUnknown*    ipSample,
    float          fOutput
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipRegressor	ITcVnMlModel* [▶ 1804]	Regressor to be used
ipSample	ITcUnknown* [▶ 1810]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
fOutput	float	Scalar output to be learnt

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.19.37 TrainSampleVector

Train a regressor with a single sample and vectorial output.

Syntax

Definition:

```
HRESULT TrainSampleVector(
    HRESULT          hrPrev,
    ITcVnMlModel*    ipRegressor,
    ITcUnknown*      ipSample,
    ITcVnContainer* ipOutput
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipRegressor	ITcVnMlModel* [▶ 1804]	Regressor to be used
ipSample	ITcUnknown* [▶ 1810]	Container holding a single input sample (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)
ipOutput	ITcVnContainer* [▶ 1760]	Vectorial output to be learnt (ContainerType_Vector_REAL or ContainerType_Vector_LREAL)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.20 Measurement

该组包含用于测量图像中对象的函数。

函数

边缘定位

- LocateCircularArc [▶ 2502]，用于定位比例圆弧
- LocateEdge [▶ 2505]，用于定位线性边缘
- LocateEdges [▶ 2509]，用于定位几个线性边缘
- LocateEllipse [▶ 2512]，用于定位完整椭圆

几何特征

- ClosestPointsBF [▶ 2501]，用于计算两个点云中最近的点
- MeasureAngleBetweenEdges [▶ 2515]，用于测量两条边之间的角度
- MeasureEdgeDistance [▶ 2520]，用于测量两条边之间的平均距离
- MeasureMinEdgeDistance [▶ 2524]，用于测量两条边之间的最小距离

其它

- AdjustSearchWindowOrientationToLinearEdge [▶ 2500]，用于将搜索栏与边缘对齐

边缘定位参数

图像对象的边界往往不完全位于两个像素之间的边界上。相反，它们与像素网格独立运行。因此，有必要使用亚像素精度进行边缘定位。这不仅考虑两个像素之间的转换，还根据对象边界的整体强度梯度来确定像素内的边缘位置。

注意
测量前不要处理图像
在调用测量函数之前，不要处理图像。处理过程可能会导致信息丢失，并相应降低准确性。

所有测量函数都使用边缘定位这一基本功能。这些函数的区别主要在于执行边缘定位的几何排列（例如：LocateEllipse [▸ 2512]，用于圆形物体）和从中计算出的属性（例如：MeasureAngleBetweenEdges [▸ 2515]，用于角度测量）。

必须为每个使用边缘定位的函数定义某些参数。其中包括：

- 搜索区域的定义：在哪寻找边缘？
- 边缘对比的定义：什么会被检测为边缘？
- 测量算法的定义：如何确定边缘的准确位置。

这些参数如下所述。对于专家参数，还指定了标准值。

搜索区域

参数	描述
搜索线的位置	搜索线的位置根据待定位特征定义。 线性边缘的搜索线位置定义为带有起点 aStartPoint 和终点 aEndPoint 的单线。 椭圆的搜索线位置由中心点 aCenter 和半径 fSearchRadius 定义。 角搜索线的位置由一个内点 aInnerPoint 和两个外点 aOuterPoint1 和 aOuterPoint2 定义。 搜索线的位置应始终设置为搜索线切割待正交定位的边缘。矩形搜索区域可通过 AdjustSearchWindowOrientationToLinearEdge （[▸ 2500] 函数调整为与边缘正交。
搜索线数 nSearchLines	这指定要用于边缘定位的搜索线数量。最简单的边缘定位只有一条搜索线，必须在这条搜索线上找到边缘。根据需要确定的几何特征，在局部分布多条搜索线，以便对多个边缘转换进行定位。 在以下情况下可能有必要这样做： • 不仅需要位置，还需要边缘渐变或旋转。 • 几何特征（如椭圆）需要多条搜索线。 • 需要一个平均值。 在定位线性边缘时，搜索线对称排列在中心搜索线的两侧，中心搜索线包含在 nSearchLines 中。因此，总是需要奇数条搜索线。
搜索线之间的距离 fSearchlineDist （专家） 默认：1	搜索线距离以像素为单位定义相邻搜索线之间的距离（大于 0，默认为 1）。这样就可以灵活平衡精度和计算时间。
反转搜索方向 bInvertSearchDirection （专家） 默认：false	定位算法始终从搜索线的一端到另一端搜索边缘，例如，从 aStartPoint 到 aEndPoint 进行搜索。如果由于图像属性的原因需要反转搜索方向（即从搜索线的另一端开始），该参数会有所帮助。

边缘对比

参数	描述
边缘方向 eEdgeDirection	边缘方向 ETcVnEdgeDirection 决定应找到从亮到暗还是从暗到亮的边缘过渡。为此，有以下值可供选择： • ED_DARK_TO_LIGHT • ED_LIGHT_TO_DARK
最低强度 fMinStrength	最低强度定义边缘想要被找到所需的最小内部强度差。 这样就可以控制边缘为被找到所必须要具备的差异程度。
最大厚度 nMaxThickness （专家） 默认：10	最大厚度指定 fMinStrength 内部必须达到的像素数。 这样就可以控制是只寻找锋利边缘，还是也寻找非常模糊的边缘。

算法

参数	描述
算法 eAlgorithm *（专家）* 默认：EDA_INTERPOLATION	该参数 ETcVnEdgeDetectionAlgorithm 定义所使用的定位算法。提供以下选项： • EDA_INTERPOLATION • EDA_APPROX_ERF • EDA_APPROX_GAUSSIAN EDA_APPROX_ERF 和 EDA_APPROX_GAUSSIAN 比标准算法 EDA_INTERPOLATION 慢，但可以实现更高的精度。不过，这样的前提条件是边缘的强度梯度也尽可能符合指定模型。误差函数 erf 或高斯函数被用作模型。 如果从亮到暗的过渡跨越多个像素，并且在亮区或暗区两侧分别保持近似恒定，则 erf 函数就非常适合边缘。  高斯函数非常适用于线，即暗-亮-暗的过渡，反之亦然。不过，亮度高原可能只有几个像素宽，否则应使用 erf 函数。 
子像素/迭代 nSubpixelsIterations *（专家）* 默认：10	该参数的含义取决于所使用的定位算法。 如为 EDA_INTERPOLATION： 所考虑的子像素数定义搜索线评估底层像素强度所使用的细粒度。例如，值 10 指定在 1 像素的长度上取 10 个测量值。 建议：5 — 10 如为 EDA_APPROX_ERF 和EDA_APPROX_GAUSSIAN： 迭代数是一个终止标准，定义针对像素强度对边缘模型进行近似所使用的最大迭代数。更多的迭代会增加运行时；如果迭代次过多，则应使用插值法。 建议：50 — 100
近似精度 fApproxPrecision *（专家）* 默认：0.001	近似精度是边缘定位的另一个终止标准。它只有在定位算法为 EDA_APPROX_ERF 或EDA_APPROX_GAUSSIAN 的情况下才能被观察到。 如果达到 nSubPixelIterations 的迭代数，或者连续两次迭代的像素相对偏差小于 fApproximationPrecision，边缘或线的近似就会中止。 建议：0.01 — 0.0001

HRESULT 的解释

HRESULT [▸ 122] 返回值对测量函数的意义如下：

代码	名称	描述
0x000	S_OK	函数成功执行，并在所有搜索线上定位了一条边。
0x001	S_FALSE	函数成功执行，但没有在每条搜索线上定位到边。
0x256	s_watchdogtimeout	函数被 Watchdog（看门狗）中止。在这种情况下，所有测量函数都可以返回部分结果。
0x70C	NOTFOUND	函数成功执行，但在任何搜索行上都未找到边。假设只有在确保图像包含具有参数设置边缘属性的对象时才会调用测量函数。如果没有找到任何搜索线，则表明参数设置不正确或函数调用不合适。因此，在这种情况下会输出错误代码。
0x7xx	除 NOTFOUND 外的所有错误代码	函数未成功执行。请参见：ADS 返回代码 [▶ 2753]

例如，可以使用下方查询来处理返回值：

```
if (SUCCEEDED(hr))
{
    // Process results
    if (hr != S_OK)
    {
        // Emit warning that not all search lines were found
    }
}
else if (hr == ADS_E_NOTFOUND)
{
    // Emit warning for wrong parametrization or missing piece
}
else
{
    // Error handling
}
```

缩短执行时间

固定搜索窗口的执行时间根据组件在图像中的位置而略有不同。如果找到的边离起点较近，执行时间就会比离起点较远的边短。

两种近似算法会出现较大的波动，因为这些算法需要不同的迭代，以便利用周围的像素强度近似模型参数。

一般来说，如果搜索窗口的搜索线尽可能正交地击中边缘，则需要更少的迭代。如果图像中物体的位置和方向波动很小或没有波动，可以相应地调整搜索窗口以实现更短的执行时间。此外，可以通过减少最大迭代次数来减少所需的最大时间，但这可能会导致结果不太准确。

此外，可以使用外部看门狗限制时间。如有必要，看门狗会终止函数的执行并返回现有的部分结果。

6.2.3.20.1 AdjustSearchWindowOrientationToLinearEdge

Adjust the search window to be orthogonal to a linear edge. Might be helpful before calling edge localization and distance measurement functions that rely on the search window.
Can use available TwinCAT Job Tasks for executing parallel code regions.

Syntax

Definition:

```
HRESULT AdjustSearchWindowOrientationToLinearEdge(
    HRESULT            hrPrev,
    ITcVnImage*        ipSrcImage,
    TcVnPoint2_REAL&   aStartPoint,
    TcVnPoint2_REAL&   aEndPoint,
    ETcVnEdgeDirection eEdgeDirection,
    float              fMinStrength,
    ULONG              nMaxThickness,
    ULONG              nSearchLines
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]	Source image
aStartPoint	TcVnPoint2_REAL [▶ 1588]&	Position from which to start the search process (in the direction of aEndPoint), which is not changed
aEndPoint	TcVnPoint2_REAL [▶ 1588]&	Position where the search process ends, which is adjusted by this function
eEdgeDirection	ETcVnEdgeDirection [▶ 1615]	Specification of the edge direction to search for
fMinStrength	float	Specification of the minimum strength (intensity difference) of the edge to search for
nMaxThickness	ULONG	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels
nSearchLines	ULONG	Width of the search window, i.e. the number of search lines (3, 5, 7, ...), centered around the line specified by aStartPoint and aEndPoint

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.20.2 ClosestPointsBF

Find the closest distance between two 2d point sets of the same type, using a brute force approach.

Syntax

Definition:

```
HRESULT ClosestPointsBF(
    HRESULT             hrPrev,
    ITcVnContainer*     ipPointSet1,
    ITcVnContainer*     ipPointSet2,
    double&             fMinDist,
    TcVnPoint2_LREAL& aPoint1,
    TcVnPoint2_LREAL& aPoint2
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipPointSet1	ITcVnContainer* [▶ 1760]	Container with 1st set of 2d points
ipPointSet2	ITcVnContainer* [▶ 1760]	Container with 2nd set of 2d points
fMinDist	double&	Returns the minimum distance between 2 points of the different point sets
aPoint1	TcVnPoint2_LREAL [▶ 1588]&	Returns the point out of ipPointSet1, for which fMinDist is achieved
aPoint2	TcVnPoint2_LREAL [▶ 1588]&	Returns the point out of ipPointSet2, for which fMinDist is achieved

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.20.3 LocateCircularArc

Locate a circular arc.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT LocateCircularArc(
    HRESULT                         hrPrev,
    ITcVnImage*                     ipSrcImage,
    TcVnCircularArc&                stCircularArc,
    TcVnPoint2_REAL&                aCenter,
    float                           fSearchRadius,
    double                          fArcDirectionRad,
    ETcVnEdgeDirection              eEdgeDirection,
    float                           fMinStrength,
    double                          fAngleStepRad = 0.1,
    ULONG                           nMaxThickness = 10,
    bool                            bInvertSearchDirection = false,
    ULONG                           nSubpixelsIterations = 10,
    float                           fApproxPrecision = 0.001f,
    ETcVnEdgeDetectionAlgorithm     eAlgorithm = EDA_INTERPOLATION,
    ITcVnContainer**                pipContourPoints = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (1 channel)
stCircularArc	TcVnCircularArc [▶ 1640]&		Returns the circular arc parameters
aCenter	TcVnPoint2_REAL [▶ 1588]&		Input estimated circle center (only used as a starting point to search for the circle contour, not used for the circle center estimation)
fSearchRadius	float		Input search radius (starting from aCenter, should be greater than the actual circle radius but aCenter + fSearchRadius should be within the image borders)
fArcDirectionRad	double		Input search starting direction in radian. Valid range is [-pi, +pi], where right is 0 rad, top is -pi/2 rad and bottom +pi/2 rad. The circular arc should at least be valid in range fArcDirectionRad +- 4 * fAngleStepRad)
eEdgeDirection	ETcVnEdgeDirection [▶ 1615]		Specification of the edge direction to search for
fMinStrength	float		Specification of the minimum strength (intensity difference) of the edge to search for
fAngleStepRad	double	0.1	Search step in rad (should be chosen so that about 20 - 60 steps are available for the whole arc. In most cases 0.1 rad (5.7 deg) is a good value)
nMaxThickness	ULONG	10	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels

Name	Type	Default	Description
bInvertSearchDirection	bool	false	If true, the search starts from outside the circular arc in direction of the center
nSubpixelsIterations	ULONG	10	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)
fApproxPrecision	float	0.001f	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▶ 1614]	EDA_INTERPOLATION	Selection of the edge detection algorithm
pipContourPoints	ITcVnContainer* [▶ 1760]*	nullptr	Returns the subpixel accurate contour (optional, set to 0 if not required; ContainerType_Vector_TcVnPoint2_REAL; Non-zero interface pointers are reused.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.20.4 LocateEdge

Locate the points of the first occuring edge inside a specified search window with subpixel accuracy.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT LocateEdge(
    HRESULT                      hrPrev,
    ITcVnImage*                  ipSrcImage,
    ITcVnContainer*&             ipEdgePoints,
    TcVnPoint2_REAL&             aStartPoint,
    TcVnPoint2_REAL&             aEndPoint,
    ETcVnEdgeDirection           eEdgeDirection,
    float                        fMinStrength,
    ULONG                        nSearchLines,
    float&                       fAvgStrength,
    float                        fSearchLineDist = 1,
    ULONG                        nMaxThickness = 10,
    ULONG                        nSubpixelsIterations = 10,
    float                        fApproxPrecision = 0.001f,
    ETcVnEdgeDetectionAlgorithm eAlgorithm = EDA_INTERPOLATION
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (1 channel)
ipEdgePoints	ITcVnContainer* [▶ 1760]&		Returns the detected edge points (ContainerType_Vector_TcVnPoint2_REAL)
aStartPoint	TcVnPoint2_REAL [▶ 1588]&		Position from which to start the search process (in the direction of aEndPoint)
aEndPoint	TcVnPoint2_REAL [▶ 1588]&		Position where the search process ends
eEdgeDirection	ETcVnEdgeDirection [▶ 1615]		Specification of the edge direction to search for
fMinStrength	float		Specification of the minimum strength (intensity difference) of the edge to search for
nSearchLines	ULONG		Width of the search window, i.e. the number of search lines (1, 3, 5, 7, ...), centered around the line specified by aStartPoint and aEndPoint
fAvgStrength	float&		Returns the average strength of the detected edge
fSearchLineDist	float	1	Distance between the search lines in pixels (> 0)
nMaxThickness	ULONG	10	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels
nSubpixelsIterations	ULONG	10	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)

Name	Type	Default	Description
fApproxPrecision	float	0.001f	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▶ 1614]	EDA_INTERPOLATION	Selection of the edge detection algorithm

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.20.5 LocateEdges

Locate the points of multiple occuring edges inside a specified search window with subpixel accuracy.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT LocateEdges(
    HRESULT                      hrPrev,
    ITcVnImage*                  ipSrcImage,
    ITcVnContainer*&             ipEdgePoints,
    TcVnPoint2_REAL&             aStartPoint,
    TcVnPoint2_REAL&             aEndPoint,
    ULONG                        nNumEdges,
    ETcVnEdgeDirection           eEdgeDirection,
    bool                         bAlternateDirection,
    float                        fMinStrength,
    ULONG                        nSearchLines,
    float&                       fAvgStrength,
    float                        fSearchLineDist = 1,
    ULONG                        nMaxThickness = 10,
    ULONG                        nSubpixelsIterations = 10,
    float                        fApproxPrecision = 0.001f,
    ETcVnEdgeDetectionAlgorithm eAlgorithm = EDA_INTERPOLATION
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (1 channel)
ipEdgePoints	ITcVnContainer* [▶ 1760]&		Returns the detected edge points (ContainerType_Vector_Vector_TcVnPoint2_REAL)
aStartPoint	TcVnPoint2_REAL [▶ 1588]&		Position from which to start the search process (in the direction of aEndPoint)
aEndPoint	TcVnPoint2_REAL [▶ 1588]&		Position where the search process ends
nNumEdges	ULONG		The (maximum) number of edges to search for
eEdgeDirection	ETcVnEdgeDirection [▶ 1615]		Specification of the edge direction to search for
bAlternateDirection	bool		If true, eEdgeDirection is alternated after each detected edge. Else, only edges with eEdgeDirection are searched for.
fMinStrength	float		Specification of the minimum strength (intensity difference) of the edge to search for
nSearchLines	ULONG		Width of the search window, i.e. the number of search lines (1, 3, 5, 7, ...), centered around the line specified by aStartPoint and aEndPoint
fAvgStrength	float&		Returns the average strength of the detected edges
fSearchLineDist	float	1	Distance between the search lines in pixels (> 0)
nMaxThickness	ULONG	10	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels

Name	Type	Default	Description
nSubpixelsIterations	ULONG	10	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)
fApproxPrecision	float	0.001f	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▸ 1614]	EDA_INTERPOLATION	Selection of the edge detection algorithm

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.20.6 LocateEllipse

Locate an ellipse.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT LocateEllipse(
    HRESULT                       hrPrev,
    ITcVnImage*                   ipSrcImage,
    TcVnRotatedRectangle&         stEllipse,
    TcVnPoint2_REAL&              aCenter,
    float                         fSearchRadius,
    ETcVnEdgeDirection            eEdgeDirection,
    float                         fMinStrength,
    ULONG                         nMaxThickness = 10,
    bool                          bInvertSearchDirection = false,
    float                         fMinSearchRadius = 0,
    ULONG                         nSubpixelsIterations = 10,
    ULONG                         nSearchLines = 92,
    float                         fApproxPrecision = 0.001f,
    ETcVnEdgeDetectionAlgorithm eAlgorithm = EDA_INTERPOLATION,
    ITcVnContainer**              pipContourPoints = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (1 channel)
stEllipse	TcVnRotatedRectangle [▶ 1658]&		Returns the detected ellipse
aCenter	TcVnPoint2_REAL [▶ 1588]&		The expected ellipse center
fSearchRadius	float		Radius around aCenter to search for edges
eEdgeDirection	ETcVnEdgeDirection [▶ 1615]		Specification of the edge direction to search for (from center to outside ellipse or other way round, if bInvertSearchDirection is true)
fMinStrength	float		Specification of the minimum strength (intensity difference) of the edge to search for
nMaxThickness	ULONG	10	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels
bInvertSearchDirection	bool	false	If true, the search starts from outside the ellipse in direction of the center
fMinSearchRadius	float	0	Radius around aCenter to skip before starting to search for edges (e.g. to save time or if the center contains edges that should be ignored)
nSubpixelsIterations	ULONG	10	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)

Name	Type	Default	Description
nSearchLines	ULONG	92	Specifies the amount of search lines, which are equally distributed in all directions (must be >= 8 and a multiple of 4)
fApproxPrecision	float	0.001f	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▶ 1614]	EDA_INTERPOLATION	Selection of the edge detection algorithm
pipContourPoints	ITcVnContainer* [▶ 1760]*	nullptr	Returns the subpixel accurate ellipse contour (optional, set to 0 if not required; ContainerType_Vector_TcVnPoint2_REAL; Non-zero interface pointers are reused.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.20.7 MeasureAngleBetweenEdges

Measure the angle between 2 edges.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT MeasureAngleBetweenEdges(
    HRESULT                     hrPrev,
    ITcVnImage*                 ipSrcImage,
    float&                      fAngle,
    TcVnPoint2_REAL&            aInnerPoint,
    TcVnPoint2_REAL&            aOuterPoint1,
    TcVnPoint2_REAL&            aOuterPoint2,
    ETcVnEdgeDirection          eEdgeDirection,
    float                       fMinStrength,
    ULONG                       nSearchLines,
    float                       fSearchLineDist = 1,
    ULONG                       nMaxThickness = 10,
    bool                        bInvertSearchDirection = false,
    ULONG                       nSubpixelsIterations = 10,
```

```
    bool                          bAngleInDegrees = false,
    float                         fApproxPrecision = 0.001f,
    ETcVnEdgeDetectionAlgorithm eAlgorithm = EDA_INTERPOLATION,
    ITcVnContainer**              pipEdgePoints1 = nullptr,
    ITcVnContainer**              pipEdgePoints2 = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (1 channel)
fAngle	float&		Returns the angle between the detected edges (radians or degrees, depending on bAngleInDegrees)
aInnerPoint	TcVnPoint2_REAL [▶ 1588]&		Position inside the angle to measure, from which to start the edge search process (in the direction of aOuterPoint1 and aOuterPoint2)
aOuterPoint1	TcVnPoint2_REAL [▶ 1588]&		Position where the search process for edge 1 ends
aOuterPoint2	TcVnPoint2_REAL [▶ 1588]&		Position where the search process for edge 2 ends
eEdgeDirection	ETcVnEdgeDirection [▶ 1615]		Specification of the edge direction from aInnerPoint to aOuterPoint1 and aOuterPoint2 to search for
fMinStrength	float		Specification of the minimum strength (intensity difference) of the edge to search for
nSearchLines	ULONG		Width of the search window, i.e. the number of search lines (3, 5, 7, ...), centered around the line specified by aInnerPoint and aOuterPoint1, aOuterPoint2
fSearchLineDist	float	1	Distance between the search lines in pixels (> 0)
nMaxThickness	ULONG	10	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels

Name	Type	Default	Description
bInvertSearchDirection	bool	false	If true, the search starts from each aOuterPoint in direction of aInnerPoint
nSubpixelsIterations	ULONG	10	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)
bAngleInDegrees	bool	false	fAngle is in degrees, if true
fApproxPrecision	float	0.001f	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▶ 1614]	EDA_INTERPOLATION	Selection of the edge detection algorithm
pipEdgePoints1	ITcVnContainer* [▶ 1760]*	nullptr	Returns the detected edge points between aInnerPoint and aOuterPoint1 (ContainerType_Vector_TcVnPoint2_REAL)
pipEdgePoints2	ITcVnContainer* [▶ 1760]*	nullptr	Returns the detected edge points between aInnerPoint and aOuterPoint2 (ContainerType_Vector_TcVnPoint2_REAL)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.20.8 MeasureEdgeDistance

Measure the distance between 2 parallel edges.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT MeasureEdgeDistance(
    HRESULT                      hrPrev,
    ITcVnImage*                  ipSrcImage,
    float&                       fAvgDistance,
    TcVnPoint2_REAL&             aStartPoint,
    TcVnPoint2_REAL&             aEndPoint,
    ETcVnEdgeDirection           eEdgeDirection,
    float                        fMinStrength,
    ULONG                        nSearchLines,
    float                        fSearchLineDist = 1,
    ULONG                        nMaxThickness = 10,
    bool                         bInvertSearchDirection = false,
    float                        fSearchGap = 0,
    ULONG                        nSubpixelsIterations = 10,
    float                        fApproxPrecision = 0.001f,
    ETcVnEdgeDetectionAlgorithm eAlgorithm = EDA_INTERPOLATION,
    ITcVnContainer**             pipEdgePoints1 = nullptr,
    ITcVnContainer**             pipEdgePoints2 = nullptr,
    ITcVnContainer**             pipDistances = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (1 channel)
fAvgDistance	float&		Returns the average distance between the detected edges
aStartPoint	TcVnPoint2_REAL [▶ 1588]&		Position from which to start the search process (in the direction of aEndPoint)
aEndPoint	TcVnPoint2_REAL [▶ 1588]&		Position where the search process ends
eEdgeDirection	ETcVnEdgeDirection [▶ 1615]		Specification of the edge direction from aStartPoint to aEndPoint to search for
fMinStrength	float		Specification of the minimum strength (intensity difference) of the edge to search for
nSearchLines	ULONG		Width of the search window, i.e. the number of search lines (1, 3, 5, 7, ...), centered around the line specified by aStartPoint and aEndPoint
fSearchLineDist	float	1	Distance between the search lines in pixels (> 0)
nMaxThickness	ULONG	10	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels
bInvertSearchDirection	bool	false	If true, the search starts from the center point between aStartPoint and aEndPoint in both directions

Name	Type	Default	Description
fSearchGap	float	0	Optional width of a gap (>= 0, centered between aStartPoint and aEndPoint), that is neglected for searching edges (can reduce execution time). The 2 edges to search for must be on different sides of the gap.
nSubpixelsIterations	ULONG	10	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)
fApproxPrecision	float	0.001f	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▸ 1614]	EDA_INTERPOLATION	Selection of the edge detection algorithm
pipEdgePoints1	ITcVnContainer* [▸ 1760]*	nullptr	Returns the detected edge points of the edge near aStartPoint (optional, set to 0 if not required; ContainerType_Vector_TcVnPoint2_REAL)
pipEdgePoints2	ITcVnContainer* [▸ 1760]*	nullptr	Returns the detected edge points of the edge near aEndPoint (optional, set to 0 if not required; ContainerType_Vector_TcVnPoint2_REAL)
pipDistances	ITcVnContainer* [▸ 1760]*	nullptr	Returns the distances between the detected edge points (optional, set to 0 if not required; ContainerType_Vector_REAL)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.20.9 MeasureMinEdgeDistance

Measure the minimum distance within the specified search window between 2 edges.
Can use available TwinCAT Job Tasks for executing parallel code regions.
Can return partial results when canceled by Watchdog.

Syntax

Definition:

```
HRESULT MeasureMinEdgeDistance(
    HRESULT                      hrPrev,
    ITcVnImage*                  ipSrcImage,
    float&                       fMinDistance,
    TcVnPoint2_REAL&             aStartPoint,
    TcVnPoint2_REAL&             aEndPoint,
    ETcVnEdgeDirection           eEdgeDirection,
    float                        fMinStrength,
    ULONG                        nSearchLines,
    TcVnPoint2_REAL&             aPoint1,
    TcVnPoint2_REAL&             aPoint2,
    float                        fSearchLineDist = 1,
    ULONG                        nMaxThickness = 10,
    bool                         bInvertSearchDirection = false,
    float                        fSearchGap = 0,
    ULONG                        nSubpixelsIterations = 10,
    float                        fApproxPrecision = 0.001f,
    ETcVnEdgeDetectionAlgorithm  eAlgorithm = EDA_INTERPOLATION,
    ITcVnContainer**             pipEdgePoints1 = nullptr,
    ITcVnContainer**             pipEdgePoints2 = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (1 channel)
fMinDistance	float&		Returns the minimum distance between the detected edges
aStartPoint	TcVnPoint2_REAL [▶ 1588]&		Position from which to start the search process (in the direction of aEndPoint)
aEndPoint	TcVnPoint2_REAL [▶ 1588]&		Position where the search process ends
eEdgeDirection	ETcVnEdgeDirection [▶ 1615]		Specification of the edge direction from aStartPoint to aEndPoint to search for
fMinStrength	float		Specification of the minimum strength (intensity difference) of the edge to search for
nSearchLines	ULONG		Width of the search window, i.e. the number of search lines (1, 3, 5, 7, ...), centered around the line specified by aStartPoint and aEndPoint
aPoint1	TcVnPoint2_REAL [▶ 1588]&		Returns the point on the edge near aStartPoint, for which the minimum distance is achieved
aPoint2	TcVnPoint2_REAL [▶ 1588]&		Returns the point on the edge near aEndPoint, for which the minimum distance is achieved
fSearchLineDist	float	1	Distance between the search lines in pixels (> 0)
nMaxThickness	ULONG	10	Specification of the maximum thickness of the edge to search for, which means fMinStrength must be reached within nMaxThickness pixels

Name	Type	Default	Description
bInvertSearchDirection	bool	false	If true, the search starts from the center point between aStartPoint and aEndPoint in both directions
fSearchGap	float	0	Optional width of a gap (>= 0, centered between aStartPoint and aEndPoint), that is neglected for searching edges (can reduce execution time). The 2 edges to search for must be on different sides of the gap.
nSubpixelsIterations	ULONG	10	Specifies the number of subpixels (for INTERPOLATION, 10 - 20 usually is sufficient) or maximum number of iterations for optimizing the parameters (for APPROX_ERF and APPROX_GAUSSIAN, 50 - 100 usually is sufficient)
fApproxPrecision	float	0.001f	Specifies the approximation precision for APPROX_ERF and APPROX_GAUSSIAN (0.001 usually is sufficient, unused for INTERPOLATION)
eAlgorithm	ETcVnEdgeDetectionAlgorithm [▸ 1614]	EDA_INTERPOLATION	Selection of the edge detection algorithm
pipEdgePoints1	ITcVnContainer* [▸ 1760]*	nullptr	Returns the detected edge points of the edge near aStartPoint (optional, set to 0 if not required; ContainerType_Vector_TcVnPoint2_REAL)
pipEdgePoints2	ITcVnContainer* [▸ 1760]*	nullptr	Returns the detected edge points of the edge near aEndPoint (optional, set to 0 if not required; ContainerType_Vector_TcVnPoint2_REAL)

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Metrology 2D

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21 Miscellaneous

该组包含可跨组使用的其他函数。

函数

几何辅助函数

- ClipLineToBoundary [▶ 2529]
- ClipLineToBoundary_ITcVnImage [▶ 2530]
- ClipLineToBoundary_TcVnRectangle_DINT [▶ 2531]
- LineIntersectionPoint [▶ 2538]
- LineIntersectionPointAndAngle [▶ 2539]
- RotatedRectangleCorners [▶ 2541]
- RotatedRectangleIntersection [▶ 2542]

时间戳

- GetTimestamp [▶ 2535]
- UpdateTimestamp [▶ 2543]

矩阵辅助函数

- InitMatrixStruct [▶ 2537]
- InvertMatrix3x3 [▶ 2538]
- MultiplyMatrices [▶ 2540]

转换接口

- ConvertITcUnknownToITcVnBitmapExport [▶ 2532]
- ConvertITcUnknownToITcVnContainer [▶ 2533]
- ConvertITcUnknownToITcVnImage [▶ 2533]
- ConvertITcUnknownToITcVnMlModel [▶ 2534]

初始化函数

- CheckFunctionInitialization [▶ 2528]
- DeinitializeFunction [▶ 2535]

其它

- HuMomentInvariants [▶ 2536]
- SetRngSeed [▶ 2542]

6.2.3.21.1 CheckFunctionInitialization

Check if a function is initialized with the specified options.

Syntax

Definition:

```
HRESULT CheckFunctionInitialization(
    HRESULT                     hrPrev,
    ETcVnInitializableFunction eFunction,
    ULONGLONG                   nOptions
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
eFunction	ETcVnInitializableFunction [▶ 1620]	Initializable function
nOptions	ULONGLONG	Initialization options for the function

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.2 ClipLineToBoundary

Clips a line to a rectangular boundary.

Syntax

Definition:

```
HRESULT ClipLineToBoundary(
    HRESULT             hrPrev,
    TcVnVector4_LREAL&  aLine,
    LONG                nX1,
    LONG                nY1,
    LONG                nX2,
    LONG                nY2,
    TcVnPoint2_REAL&    aStartPoint,
    TcVnPoint2_REAL&    aEndPoint
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aLine	TcVnVector4_LREAL [▶ 1590]&	The first and second element describe the x and y component of the direction vector. The third and fourth element describe the x and y component of the position vector.
nX1	LONG	x of top left rectangle point
nY1	LONG	y of top left rectangle point
nX2	LONG	x of bottom right rectangle point
nY2	LONG	Y of bottom right rectangle point
aStartPoint	TcVnPoint2_REAL [▶ 1588]&	Returns the clipped starting point of the line.
aEndPoint	TcVnPoint2_REAL [▶ 1588]&	Returns the clipped end point of the line.

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.3 ClipLineToBoundary (ITcVnImage*)

Clips a line to the boundary of an image.

Syntax

Definition:

```
HRESULT ClipLineToBoundary(
    HRESULT             hrPrev,
    TcVnVector4_LREAL& aLine,
    ITcVnImage*         ipImage,
    TcVnPoint2_REAL&    aStartPoint,
    TcVnPoint2_REAL&    aEndPoint
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aLine	TcVnVector4_LREAL [▶ 1590]&	The first and second element describe the x and y component of the direction vector. The third and fourth element describe the x and y component of the position vector.
ipImage	ITcVnImage* [▶ 1797]	Image, from which to derive the boundary
aStartPoint	TcVnPoint2_REAL [▶ 1588]&	Returns the clipped starting point of the line.
aEndPoint	TcVnPoint2_REAL [▶ 1588]&	Returns the clipped end point of the line.

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.4 ClipLineToBoundary (TcVnRectangle_DINT)

Clips a line to a rectangular boundary.

Syntax

Definition:

```
HRESULT ClipLineToBoundary(
    HRESULT              hrPrev,
    TcVnVector4_LREAL&   aLine,
    TcVnRectangle_DINT&  stRectangle,
    TcVnPoint2_REAL&     aStartPoint,
    TcVnPoint2_REAL&     aEndPoint
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aLine	TcVnVector4_LREAL [▶ 1590]&	The first and second element describe the x and y component of the direction vector. The third and fourth element describe the x and y component of the position vector.
stRectangle	TcVnRectangle_DINT [▶ 1657]&	Rectangular boundary
aStartPoint	TcVnPoint2_REAL [▶ 1588]&	Returns the clipped starting point of the line.
aEndPoint	TcVnPoint2_REAL [▶ 1588]&	Returns the clipped end point of the line.

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.5 ConvertITcUnknownToITcVnBitmapExport

Convert an ITcUnknown interface pointer to an ITcVnBitmapExport interface pointer. The destination pointer will be released (if existing) and set to the source pointer. The source pointer will be set to 0.

Syntax

Definition:

```
HRESULT ConvertITcUnknownToITcVnBitmapExport(
    HRESULT              hrPrev,
    ITcUnknown*&         ipSrc,
    ITcVnBitmapExport*&  ipDest
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrc	ITcUnknown* [▶ 1810]&	Source pointer
ipDest	ITcVnBitmapExport* [▶ 1782]&	Destination pointer

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.6 ConvertITcUnknownToITcVnContainer

Convert an ITcUnknown interface pointer to an ITcVnContainer interface pointer. The destination pointer will be released (if existing) and set to the source pointer. The source pointer will be set to 0.

Syntax

Definition:

```
HRESULT ConvertITcUnknownToITcVnContainer(
    HRESULT          hrPrev,
    ITcUnknown*&     ipSrc,
    ITcVnContainer*& ipDest
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrc	ITcUnknown* [▶ 1810]&	Source pointer
ipDest	ITcVnContainer* [▶ 1760]&	Destination pointer

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.7 ConvertITcUnknownToITcVnImage

Convert an ITcUnknown interface pointer to an ITcVnImage interface pointer. The destination pointer will be released (if existing) and set to the source pointer. The source pointer will be set to 0.

Syntax

Definition:

```
HRESULT ConvertITcUnknownToITcVnImage(
    HRESULT      hrPrev,
    ITcUnknown*& ipSrc,
    ITcVnImage*& ipDest
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrc	ITcUnknown* [▶ 1810]&	Source pointer
ipDest	ITcVnImage* [▶ 1797]&	Destination pointer

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.8 ConvertITcUnknownToITcVnMlModel

Convert an ITcUnknown interface pointer to an ITcVnMlModel interface pointer. The destination pointer will be released (if existing) and set to the source pointer. The source pointer will be set to 0.

Syntax

Definition:

```
HRESULT ConvertITcUnknownToITcVnMlModel(
    HRESULT        hrPrev,
    ITcUnknown*&   ipSrc,
    ITcVnMlModel*& ipDest
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrc	ITcUnknown* [▶ 1810]&	Source pointer
ipDest	ITcVnMlModel* [▶ 1804]&	Destination pointer

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.9 DeinitializeFunction

Deinitialize the specified options for the function.

Syntax

Definition:

```
HRESULT DeinitializeFunction(
    HRESULT                      hrPrev,
    ETcVnInitializableFunction eFunction,
    ULONGLONG                    nOptions
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
eFunction	ETcVnInitializableFunction [▶ 1620]	Initializable function
nOptions	ULONGLONG	Initialization options for the function

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.10 GetTimestamp

Get the timestamp (actual DC time) from the last change of an object (e.g. ITcVnImage or ITcVnContainer)

Syntax

Definition:

```
HRESULT GetTimestamp(
    HRESULT      hrPrev,
    ITcUnknown* ipUnknown,
    LONGLONG&    nTimestamp
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipUnknown	ITcUnknown* [▶ 1810]	An object implementing ITcUnknown and ITcVnTimestamp (e.g. ITcVnImage or ITcVnContainer)
nTimestamp	LONGLONG&	Returns the timestamp

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.11 HuMomentInvariants

Computes the Hu moment invariants.

Syntax

Definition:

```
HRESULT HuMomentInvariants(
    HRESULT           hrPrev,
    TcVnMoments&      stMoments,
    TcVnArray7_LREAL& aHuInvariants
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stMoments	TcVnMoments [▶ 1644]&	Moments from which to compute the Hu invariants
aHuInvariants	TcVnArray7_LREAL [▶ 1590]&	Returns the Hu moment invariants

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.12 InitMatrixStruct

Initialize a struct of the type TcVnMatrix extending a buffer with metainformation so that it can be used as a matrix.

Syntax

Definition:

```
HRESULT InitMatrixStruct(
    HRESULT          hrPrev,
    PVOID            pSrcBuffer,
    TcVnMatrix&      stDestMatrix,
    ULONG            nRows,
    ULONG            nCols,
    ETcVnElementType eElementType
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
pSrcBuffer	PVOID	Source buffer
stDestMatrix	TcVnMatrix [▶ 1644]&	Returns completed matrix struct
nRows	ULONG	Matrix rows
nCols	ULONG	Matrix columns
eElementType	ETcVnElementType [▶ 1615]	Type of the matrix elements

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.13 InvertMatrix3x3

Invert a 3x3 matrix.

Syntax

Definition:

```
HRESULT InvertMatrix3x3(
    HRESULT              hrPrev,
    TcVnMatrix3x3_LREAL& aSrcMatrix,
    TcVnMatrix3x3_LREAL& aInvertedMatrix
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aSrcMatrix	TcVnMatrix3x3_LREAL [▸ 1590]&	Source matrix
aInvertedMatrix	TcVnMatrix3x3_LREAL [▸ 1590]&	Returns inverted matrix

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.14 LineIntersectionPoint

Computes the intersection point between two lines (returns S_FALSE if the provided lines are parallel).

Syntax

Definition:

```
HRESULT LineIntersectionPoint(
    HRESULT             hrPrev,
    TcVnVector4_LREAL& aLine1,
    TcVnVector4_LREAL& aLine2,
    TcVnPoint2_REAL&    aIntersecPoint
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aLine1	TcVnVector4_LREAL [▶ 1590]&	First line. The first and second element describe the x and y component of a vector collinear to the line. The third and fourth element describe the x and y component of a point on the line.
aLine2	TcVnVector4_LREAL [▶ 1590]&	Second line. The first and second element describe the x and y component of a vector collinear to the line. The third and fourth element describe the x and y component of a point on the line.
aIntersecPoint	TcVnPoint2_REAL [▶ 1588]&	Returns the intersection point of aLine1 and aLine2.

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.15 LineIntersectionPointAndAngle

Computes the intersection point and angle between two lines (returns S_FALSE if the provided lines are parallel).

Syntax

Definition:

```
HRESULT LineIntersectionPointAndAngle(
    HRESULT            hrPrev,
    TcVnVector4_LREAL& aLine1,
    TcVnVector4_LREAL& aLine2,
    TcVnPoint2_REAL&   aIntersecPoint,
    float&             fAngle,
    bool               bAngleInDegrees
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
aLine1	TcVnVector4_LREAL [▸ 1590]&	First line. The first and second element describe the x and y component of a vector collinear to the line. The third and fourth element describe the x and y component of a point on the line.
aLine2	TcVnVector4_LREAL [▸ 1590]&	Second line. The first and second element describe the x and y component of a vector collinear to the line. The third and fourth element describe the x and y component of a point on the line.
aIntersecPoint	TcVnPoint2_REAL [▸ 1588]&	Returns the intersection point of aLine1 and aLine2.
fAngle	float&	Returns the intersection angle of aLine1 and aLine2.
bAngleInDegrees	bool	If TRUE, fAngle is in degrees. If FALSE, fAngle is in radians.

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.16 MultiplyMatrices

Matrix multiplication of two matrices (A * B = C). All matrices are represented by structs holding a pointer to an array of the appropriate size.

Syntax

Definition:

```
HRESULT MultiplyMatrices(
    HRESULT     hrPrev,
    TcVnMatrix& stSrcMatrix1,
    TcVnMatrix& stSrcMatrix2,
    TcVnMatrix& stDestMatrix
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stSrcMatrix1	TcVnMatrix [▶ 1644]&	First source matrix (A)
stSrcMatrix2	TcVnMatrix [▶ 1644]&	Second source matrix (B)
stDestMatrix	TcVnMatrix [▶ 1644]&	Destination matrix (C) (The destination matrix is filled by this function, but the required memory needs to be provided.)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.17 RotatedRectangleCorners

Computes the 4 corner points of a rotated rectangle.

Syntax

Definition:

```
HRESULT RotatedRectangleCorners(
    HRESULT                 hrPrev,
    TcVnRotatedRectangle&   stRect,
    TcVnArray4_Point2_REAL& aCorners
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stRect	TcVnRotatedRectangle [▶ 1658]&	Rotated rectangle
aCorners	TcVnArray4_Point2_REAL [▶ 1590]&	Returns the 4 corner points

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.18 RotatedRectangleIntersection

Finds the intersection of 2 rotated rectangles.

Syntax

Definition:

```
HRESULT RotatedRectangleIntersection(
    HRESULT                       hrPrev,
    TcVnRotatedRectangle&         stRect1,
    TcVnRotatedRectangle&         stRect2,
    ITcVnContainer*&              ipIntersection,
    ETcVnRectangleIntersection&   eIntersection
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
stRect1	TcVnRotatedRectangle [▶ 1658]&	Rotated rectangle 1
stRect2	TcVnRotatedRectangle [▶ 1658]&	Rotated rectangle 2
ipIntersection	ITcVnContainer* [▶ 1760]&	Returns the intersection points (ContainerType_Vector_TcVnPoint2_REAL, non-zero interface pointers are reused)
eIntersection	ETcVnRectangleIntersection [▶ 1629]&	Returns the intersection type

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.19 SetRngSeed

Sets the internal pseudo random number generator seed (intended for testing purposes only).

Syntax

Definition:

```
HRESULT SetRngSeed(
    HRESULT hrPrev,
    LONG    nSeed
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
nSeed	LONG	seed (0 sets the generator back to its initial state)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.21.20 UpdateTimestamp

Update the timestamp of an object to the actual DC time (e.g. ITcVnImage or ITcVnContainer)

Syntax

Definition:

```
HRESULT UpdateTimestamp(
    HRESULT      hrPrev,
    ITcUnknown* ipUnknown
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipUnknown	ITcUnknown* [▶ 1810]	An object implementing ITcUnknown and ITcVnTimestamp (e.g. ITcVnImage or ITcVnContainer)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.22 OCR

该组包含**光学**字符**识别**函数。

OCR 函数 [▶ 2546]识别图像中的字符，并将识别出的字符作为字符串返回。分类基于经典机器学习模型。这些模型以训练形式提供，因此可以直接使用函数。无需额外设置或训练。支持的字符和字体取决于训练中使用的数据。因此，有多种模型，其函数范围以及一般前提条件和要求如下所述。

对于图像中识别出的所有字符，模型会将每个字符分配到匹配度最高的已知类别。因此，结果总是包含所使用模型的已知字符。不包括拒绝或删除未知字符。

函数仅支持单行字符串；必须将多行字符串分为单独的 ROI，并通过多次函数调用读取。

识别字符的一般要求

- 字符高度最小为 20 像素
- 线宽最小为 3 像素
- 点最小为 3 x 3 像素
- 线最小为 3 x 6 像素
- 字符间距/分隔最小为 4 像素
- 字符不得重叠
- 仅水平对齐/字符排列最大 ± 6° 偏差
- 字符线/轮廓不得中断

图像的一般要求：

- ROI 只包含待识别字符和字符周围的无干扰区
- 在 ROI 内，字符不得被其他轮廓（如矩形）包围
- 字符与背景要对比鲜明
- 背景均匀、无噪音或干扰、不透明

对字体的要求

- 仅使用字宽相等的比例字体
- 较大的间隙始终会被识别为单个空格
- 仅使用无衬线字体，如 Arial、Tahoma、Courier、Univers、Frutiger、Verdana、OCR-B 等
- 无混合字体
- 无圆点打印或斜体字体

模型：

枚举 ETcVnOcrModelType [▶ 1625] 允许访问以下模型：

- OMT_NUMBERS
  - 支持数字分类
  - 包含 0-9 的字符
- OMT_NUMBERS_SC
  - 支持数字和特殊字符分类
  - 包含 0-9 的字符和五个特殊字符 . / - : =
- OMT_UCLETTERS
  - 支持大写字母分类
  - 包含 A-Z 的字符

- OMT_NUMBERS_SC_UCLETTERS
    - 支持数字、特殊字符和大写字母分类
    - 包含 0-9 的字符、五个特殊字符 . / - : = 和 A-Z

在使用组合模型（OMT_NUMBERS_SC_UCLETTERS）时，由于某些字符非常相似，可能会出现混淆。例如 0 和 O、S 和 5 以及 B 和 8。

如果数字或字母的位置已知，则应使用可选格式规格（sPattern）和单独模型的组合作为替代。另外，也可以单独分析结果字符串（ipCharacters）的字符，这样就可以单独决定是否接受 0 而非 O。

初始化函数

为了能够将 OCR 函数与一个或多个模型一起使用，必须首先使用功能块 InitializeFunction [▶ 2563] 初始化该函数以及相应的模型。

加载模型所需的时间取决于所使用 IPC 的性能和模型大小，可能需要几百毫秒甚至几秒钟。因此，在任务循环时间较短的情况下，预计会出现循环时间超限。

如果要使用组合模型 OMT_NUMBERS_SC_UCLETTERS，由于文件大小的原因，路由器内存必须至少设置为 512 MB。

具体大小取决于路由器内存的其他用途和任何现有分段储存。例如，如果功能块 InitializeFunction 返回返回代码 0x80004005，则表示路由器内存不足。因此，应首先检查路由器内存，必要时增加内存。加载模型后，大部分内存会重新释放，供应用使用。

HRESULT 的解释

- 如果识别到图像上的字符，标准函数返回 S_OK。使用可选参数时，这取决于是否指定了 sPattern。如果没有定义格式规格，则返回对应于标准函数。如果已传递格式规格，只会在识别字符与模板规格相匹配的情况下才会返回 S_OK。
- 如果标准函数执行成功，但未找到任何字符，或者找到的字符与可选模板规格不匹配，则返回 S_FALSE。
- 如果可选 sPattern 规格缺少传输模型，则返回 HRESULT = ADSERR_DEVICE_NOTINIT。
- 如果向标准函数传递了多个模型，则始终返回 HRESULT = ADSERR_DEVICE_INVALIDPARM，因为该函数仅支持每次调用一个模型。使用可选参数时，这取决于是否指定了 sPattern。如果 sPattern 为空，则只使用 TCVN_OMT_NUMBERS_SC_UCLETTERS 模型。如果没有传递该模型，则返回 HRESULT = ADSERR_DEVICE_INVALIDPARM。

使用 sPattern 的示例

在使用可选参数 sPattern 和 eOcrOptions 时，还可能出现其他情况。根据输入图像上的字符组合、来自 sPattern 和 eOcrOptions 的信息以及识别的字符，如果出现不匹配，则返回 S_FALSE。然后，还可以对结果字符串（ipCharacters）中的字符进行单独分析，例如，如果长度不同，也会返回 S_FALSE。这些示例涉及 OMT_NUMBERS_SC_UCLETTERS 模型的使用。

ipSrcImage 上的字符	sPattern	eOcrOptions	ipCharacters	HRESULT
12/34	dd.dd		12/34	S_OK
12534	dd.dd		12534	S_OK
12/34	dd!dd		1234	S_OK
12 34	dd!d		124	S_OK
12534	dd!dd		1234	S_OK
12 34	dd_dd		12 34	S_OK
AB12/	uudd#		AB12/	S_OK
12 34	dddd	WITHBLANKS	12 34	S_OK
12/4	dd#d	WITHBLANKS	12/4	S_OK
12 34	dd_dd	WITHBLANKS	12 34	S_OK
12 34 56	dd_dddd	WITHBLANKS	12 34 56	S_OK
12 3 4	dd.dd	WITHBLANKS	12 3 4	S_FALSE
12 34	dd!dd		124	S_FALSE
12 34	dd.dd		1234	S_FALSE
1234	!dddd		234	S_FALSE
12 34	dd_dd		12.34	S_FALSE
12 3	dd_dd		123	S_FALSE
12 3 4	dd_d_d		123 4	S_FALSE
AB12/	uddd#		AB12/	S_FALSE
AB12/	uuddd		AB12/	S_FALSE

6.2.3.22.1 OCR

Detects and recognizes characters in a binary image (white characters on black background).

Syntax

Definition:

```
HRESULT OCR(
    HRESULT           hrPrev,
    ITcVnImage*       ipSrcImage,
    ULONGLONG         eModel,
    ITcVnContainer*&  ipCharacters,
    float&            fMinConfidence,
    PCCH              sPattern = "",
    ULONG             eOcrOptions = OO_NONE,
    ITcVnContainer**  pipBoundingBoxes = nullptr,
    ITcVnContainer**  pipConfidences = nullptr
)
```

Parameters

Name	Type	Default	Description
hrPrev	HRESULT [▶ 122]		HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipSrcImage	ITcVnImage* [▶ 1797]		Source image (USINT, 1 channel, binary)
eModel	ULONGLONG		Selection of OCR model types (ETcVnOcrModelType)
ipCharacters	ITcVnContainer* [▶ 1760]&		Returns the recognized characters (ContainerType_Vector_String_SINT)
fMinConfidence	float&		Returns the minimum value of the confidences
sPattern	PCCH	""	String pattern containing the format in which characters are presented
eOcrOptions	ULONG	OO_NONE	Specifies which options must be applied to the function (ETcVnOcrOptions)
pipBoundingBoxes	ITcVnContainer* [▶ 1760]*	nullptr	Returns the corresponding bounding boxes of the recognized characters (ContainerType_Vector_TcVnRectangle_DINT, optional, set to 0 if not required)
pipConfidences	ITcVnContainer* [▶ 1760]*	nullptr	Returns the corresponding classification confidences of the recognized characters (ContainerType_Vector_REAL, optional, set to 0 if not required)

Return value

HRESULT [▶ 122]

Required License

TC3 Vision OCR

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.23 Texture Analysis

该组包含用于分析图像纹理的函数。

函数

粒度属性

- Granulometry [▶ 2548]

Haralick 特征

- HaralickFeatures [▶ 2549]

6.2.3.23.1 Granulometry

Computes the granulometry of a textured image using morphological operations.

Syntax

Definition:

```
HRESULT Granulometry(
    HRESULT                     hrPrev,
    ITcVnImage*                 ipImage,
    ITcVnContainer*&            ipIntensityDiffs,
    ETcVnMorphologicalOperator  eMorphType,
    ULONG                       nMinSize,
    ULONG                       nMaxSize,
    ULONG                       nStep
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▶ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▶ 1797]	Source image (1 channel)
ipIntensityDiffs	ITcVnContainer* [▶ 1760]&	Returns the intensity differences between successive structuring element sizes (CTcVnContainer_Vector_LREAL), which can be used as features for texture description.
eMorphType	ETcVnMorphologicalOperator [▶ 1623]	Only opening or closing are supported.
nMinSize	ULONG	Min structuring element size (odd, >= 3)
nMaxSize	ULONG	Max structuring element size (>= nMinSize)
nStep	ULONG	Step, by which nMinSize is incremented until nMaxSize is exceeded (even, >= 2).

Return value

HRESULT [▶ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.3.23.2 HaralickFeatures

Computes the Haralick features of a graylevel image, which describe the texture.

Syntax

Definition:

```
HRESULT HaralickFeatures(
    HRESULT           hrPrev,
    ITcVnImage*       ipImage,
    ITcVnContainer*& ipFeatures,
    ULONG             nDist
)
```

Parameters

Name	Type	Description
hrPrev	HRESULT [▸ 122]	HRESULT indicating the result of previous operations (If SUCCEEDED(hrPrev) equals false, no operation is executed.)
ipImage	ITcVnImage* [▸ 1797]	Source image (1 channel, USINT)
ipFeatures	ITcVnContainer* [▸ 1760]&	Returns the computed Haralick features (CTcVnContainer_Vector_LREAL). The returned feature set contains the mean and standard deviation for each of the first 13 Haralick features, computed for 0, 45, 90, 135 degrees. So there will be 26 features in total (f1 mean, f1 stdDev, f2 mean, f2 stdDev, ..., f13 mean, f13 stdDev).
nDist	ULONG	Distance

Return value

HRESULT [▸ 122]

Required License

TC3 Vision Machine Learning

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4 Function Blocks

与 GigE Vision 相机或文件系统的通信以及复杂图像处理函数通过功能块进行映射。

功能块分组

TwinCAT Vision 库中的所有功能块按主题分为以下几组：

- 相机寄存器访问 [▸ 2551]
- 文件访问 [▸ 2561]
- 图像采集 [▸ 2575]
- 图像处理 [▸ 2594]

6.2.4.1 License Overview

以下列表显示了 TwinCAT Vision 功能块与可用许可证的分配情况（见许可证模型 [▸ 11]）。它显示了某些功能块的应用需要哪些许可证。

> 始终需要基本许可证 TC3 Vision Base（TF7100）。

TC3 Vision Base

FileSourceControl [▸ 2577]

GevCameraControl [▸ 2580]

InitializeFunction [▸ 2563]

ReadCalibrationPattern [▸ 2564]

ReadCalibrationResult [▸ 2565]

ReadContainer [▸ 2567]

ReadImage [▸ 2568]

ReadMemory [▸ 2552]

ReadMlModel [▸ 2569]

ReadRegister_REAL [▸ 2553]

ReadRegister_UDINT [▸ 2554]

ReadRegister_ULINT [▸ 2555]

SimpleCameraControl [▸ 2591]

WriteCalibrationResult [▸ 2570]

WriteContainer [▸ 2572]

WriteImage [▸ 2573]

WriteMemory [▸ 2556]

WriteMlModel [▸ 2574]

WriteRegister_REAL [▸ 2558]

WriteRegister_UDINT [▸ 2559]

WriteRegister_ULINT [▸ 2560]

TC3 Vision Matching

GeneralizedHoughBallard [▸ 2594]

SSIM [▸ 2597]

6.2.4.2 Camera Register Access

该组包含用于读写相机寄存器或内存区域的功能块。

功能块

UDINT 寄存器（4 字节）

- ReadRegister_UDINT [▶ 2554]
- WriteRegister_UDINT [▶ 2559]

ULINT 寄存器（8 字节）

- ReadRegister_ULINT [▶ 2555]
- WriteRegister_ULINT [▶ 2560]

REAL 寄存器（4 字节）

- ReadRegister_REAL [▶ 2553]
- WriteRegister_REAL [▶ 2558]

更大的内存区域

- ReadMemory [▶ 2552]
- WriteMemory [▶ 2556]

参数

以下参数与本组中所有功能块的工作方式相似：

地址

地址 nAddress 指定要读取或保存的相机寄存器。GVCP_REGISTER_ADDRESS 类型只是 UDINT 的别名；因此，地址长度必须为 4 字节（或 32 位）。对于相机的较大内存区域（请参见 ReadMemory [▶ 2552]），地址指向要读取或保存区域的起始位置。

执行函数

写入和读取命令的触发器分别是 bWrite 和 bRead。它们由一个上升边缘触发（从 false 更改为 true）。相关输入参数在内部保存，因此在执行过程中参数的改变不会产生任何影响。触发器的下降边缘也同样没有影响。

> **调用功能块**
> 请注意，功能块的参数描述仅在实际调用功能块时才会生效。

超时

为了在出现未知错误时不必长时间等待功能块的执行，可以通过 nTimeout 设置允许的最大执行时间。如果在触发命令触发器后超过这一时间仍无成功结果，执行将被中止。然后，超时错误可由两个输出 bError 和 nErrorId 识别。

工作状态

输出 bBusy 指示功能块是否繁忙。如果为 bBusy == true，则不能触发新命令。

通过适当的 if 查询，可以确定功能块的执行是否完成：

```
if (!fbReadOrWrite.bBusy)
{
    // Function block execution is complete or error occurred.
    // Next write-command can be issued.
}
```

错误状态

如果适用，错误状态通过 bError 以二进制方式表示错误，而 nErrorId 输出错误代码（ADS 返回代码 [▶ 2753]）。一旦有新命令被触发，错误状态就会被重置。如果功能块繁忙（bBusy == true），错误代码为 PENDING (0x71E)。

错误代码的十六进制值可以如下提取：

```
if (fbReadOrWrite.bError)
{
    // Show relevant error code for debugging purposes:
    nErrorCode = fbReadOrWrite.nErrorId & 0xFFF;
}
```

6.2.4.2.1 ReadMemory

This FB reads consecutive memory locations from the camera. Requires an open control channel (e.g. by calling FB_VN_GevCameraControl.OpenCamera() before)

Syntax

Constructor:

```
ReadMemory(
    OTCID oidITcVnGevImageProvider
)
```

Execute main:

```
FB_Main()

FB_Main(
    GVCP_REGISTER_ADDRESS nAddress,
    USHORT                nLength,
    PVOID                 pBuffer,
    bool                  bRead
)

FB_Main(
    bool bRead
)
```

Inputs

Name	Type	Default	Description
nAddress	GVCP_REGISTER_ADDRESS [▶ 1588]		The address of the first byte to read from the camera (must be 32-bit aligned)
nLength	USHORT		The number of bytes to read, starting with nAddress (must be a multiple of 4, not more than 536)
pBuffer	PVOID		Pointer to the buffer where the read memory content is written to (The buffer must have a size of at least nLength bytes!)
bRead	bool		Reading the memory is triggered by a rising edge at this input.
nTimeout	unsigned int	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	bool	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	bool	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	ULONG	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.2.2 ReadRegister_REAL

This FB reads a REAL register from the camera. Requires an open control channel (e.g. by calling FB_VN_GevCameraControl.OpenCamera() before)

Syntax

Constructor:

```
ReadRegister_REAL(
    OTCID oidITcVnGevImageProvider
)
```

Execute main:

```
FB_Main()

FB_Main(
    GVCP_REGISTER_ADDRESS nAddress,
    bool                  bRead
)

FB_Main(
    bool bRead
)
```

Inputs

Name	Type	Default	Description
nAddress	GVCP_REGISTER_ADDRESS [▶ 1588]		The address of the register that should be read
nEndian	USHORT		OPTIONAL: The endianness of the register. 0 = Big, 1 = Little. Default: 0
bRead	bool		Reading the register is triggered by a rising edge at this input.
nTimeout	unsigned int	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
fValue	float	The read value
bBusy	bool	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	bool	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	ULONG	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.2.3 ReadRegister_UDINT

This FB reads a DINT register from the camera. Requires an open control channel (e.g. by calling FB_VN_GevCameraControl.OpenCamera() before)

Syntax

Constructor:

```
ReadRegister_UDINT(
    OTCID oidITcVnGevImageProvider
)
```

Execute main:

```
FB_Main()

FB_Main(
    GVCP_REGISTER_ADDRESS nAddress,
    bool                  bRead
)
```

```
FB_Main(
    bool bRead
)
```

Inputs

Name	Type	Default	Description
nAddress	GVCP_REGISTER_ADDRESS [▶ 1588]		The address of the register that should be read
nEndian	USHORT		OPTIONAL: The endianness of the register. 0 = Big, 1 = Little. Default: 0
bRead	bool		Reading the register is triggered by a rising edge at this input.
nTimeout	unsigned int	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
nValue	ULONG	The read value
bBusy	bool	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	bool	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	ULONG	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.2.4 ReadRegister_ULINT

This FB reads a ULINT register from the camera. Requires an open control channel (e.g. by calling FB_VN_GevCameraControl.OpenCamera() before)

Syntax

Constructor:

```
ReadRegister_ULINT(
    OTCID oidITcVnGevImageProvider
)
```

Execute main:

```
FB_Main()

FB_Main(
    GVCP_REGISTER_ADDRESS nAddress,
    bool                  bRead
)

FB_Main(
    bool bRead
)
```

Inputs

Name	Type	Default	Description
nAddress	GVCP_REGISTER_ADDRESS [▶ 1588]		The address of the register that should be read
nEndian	USHORT		OPTIONAL: The endianness of the register. 0 = Big, 1 = Little. Default: 0
bRead	bool		Reading the register is triggered by a rising edge at this input.
nTimeout	unsigned int	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
nValue	ULONGLONG	The read value
bBusy	bool	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	bool	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	ULONG	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.2.5 WriteMemory

This FB writes data to consecutive memory locations on the camera. Requires an open control channel (e.g. by calling FB_VN_GevCameraControl.OpenCamera() before)

Syntax

Constructor:

```
WriteMemory(
    OTCID oidITcVnGevImageProvider
)
```

Execute main:

```
FB_Main()

FB_Main(
    GVCP_REGISTER_ADDRESS nAddress,
    USHORT                nLength,
    PVOID                 pData,
    bool                  bWrite
)

FB_Main(
    bool bWrite
)
```

Inputs

Name	Type	Default	Description
nAddress	GVCP_REGISTER_ADDRESS [▶ 1588]		The address of the first byte to write to the camera (must be 32-bit aligned)
nLength	USHORT		The number of bytes to write, starting at nAddress (must be a multiple of 4, not more than 536)
pData	PVOID		Pointer to the data that should be written to the camera (must have a size of at least nLength bytes!)
bWrite	bool		Writing the memory is triggered by a rising edge at this input.
nTimeout	unsigned int	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	bool	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	bool	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	ULONG	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.
nBytesWritten	USHORT	The number of bytes that have been successfully written to the camera.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.2.6 WriteRegister_REAL

This FB writes a REAL value into a register on the camera. Requires an open control channel (e.g. by calling FB_VN_GevCameraControl.OpenCamera() before)

Syntax

Constructor:

```
WriteRegister_REAL(
    OTCID oidITcVnGevImageProvider
)
```

Execute main:

```
FB_Main()

FB_Main(
    GVCP_REGISTER_ADDRESS nAddress,
    float                 fValue,
    bool                  bWrite
)

FB_Main(
    bool bWrite
)
```

Inputs

Name	Type	Default	Description
nAddress	GVCP_REGISTER_ADDRESS [▶ 1588]		The address of the register that should be written
fValue	float		The value to write
nEndian	USHORT		OPTIONAL: The endianness of the register. 0 = Big, 1 = Little. Default: 0
bWrite	bool		Writing the register is triggered by a rising edge at this input.
nTimeout	unsigned int	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	bool	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	bool	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	ULONG	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.2.7 WriteRegister_UDINT

This FB writes a DINT value into a register on the camera. Requires an open control channel (e.g. by calling FB_VN_GevCameraControl.OpenCamera() before)

Syntax

Constructor:

```
WriteRegister_UDINT(
    OTCID oidITcVnGevImageProvider
)
```

Execute main:

```
FB_Main()

FB_Main(
    GVCP_REGISTER_ADDRESS nAddress,
    ULONG                 nValue,
    bool                  bWrite
)

FB_Main(
    bool bWrite
)
```

Inputs

Name	Type	Default	Description
nAddress	GVCP_REGISTER_ADDRESS [▶ 1588]		The address of the register that should be written
nValue	ULONG		The value to write
nEndian	USHORT		OPTIONAL: The endianness of the register. 0 = Big, 1 = Little. Default: 0
bWrite	bool		Writing the register is triggered by a rising edge at this input.
nTimeout	unsigned int	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	bool	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	bool	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	ULONG	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.2.8 WriteRegister_ULINT

This FB writes a ULINT value into a register on the camera. Requires an open control channel (e.g. by calling FB_VN_GevCameraControl.OpenCamera() before)

Syntax

Constructor:

```
WriteRegister_ULINT(
    OTCID oidITcVnGevImageProvider
)
```

Execute main:

```
FB_Main()

FB_Main(
    GVCP_REGISTER_ADDRESS nAddress,
    ULONGLONG             nValue,
    bool                  bWrite
```

```
)

FB_Main(
    bool bWrite
)
```

Inputs

Name	Type	Default	Description
nAddress	GVCP_REGISTER_ADDRESS [▶ 1588]		The address of the register that should be written
nValue	ULONGLONG		The value to write
nEndian	USHORT		OPTIONAL: The endianness of the register. 0 = Big, 1 = Little. Default: 0
bWrite	bool		Writing the register is triggered by a rising edge at this input.
nTimeout	unsigned int	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	bool	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	bool	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	ULONG	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.3 File Access

该组包含用于在文件系统上读写数据的功能块。

功能块

图像

- ReadImage [▶ 2568]
- WriteImage [▶ 2573]

容器

- ReadContainer [▸ 2567]
- WriteContainer [▸ 2572]

机器学习模型

- ReadMlModel [▸ 2569]
- WriteMlModel [▸ 2574]

标定模板

- ReadCalibrationPattern [▸ 2564]
- ReadCalibrationResult [▸ 2565]
- WriteCalibrationResult [▸ 2570]

函数初始化

- InitializeFunction [▸ 2563]

通用

这些功能块异步工作；从第一次执行到所需结果的时间跨越几个循环。每次读/写操作的实际持续时间取决于多个因素，如图像的大小和操作系统的行为。以下参数对所有功能块均一样，因此在此只集中描述一次。

参数

文件路径

文件路径 sFilePath 指定保存文件的位置或加载哪个文件。可以使用绝对和相对路径。在相对路径的情况下，服务配置 [▸ 59]中设置的默认路径被作为基础。

文件路径在写入命令 bWrite 的上升边缘上进行内部存储，因此，在以后的循环中更改路径没有影响。

执行函数

写入和读取命令的触发器分别是 bWrite 和 bRead。它们由一个上升边缘触发（从 false 更改为 true）。相关输入参数在内部保存，因此在执行过程中参数的改变不会产生任何影响。触发器的下降边缘也同样没有影响。

调用功能块

请注意，功能块的参数描述仅在实际调用功能块时才会生效。

超时

为了在出现未知错误时不必长时间等待功能块的执行，可以通过 nTimeout 设置允许的最大执行时间。如果在触发命令触发器后超过这一时间仍无成功结果，执行将被中止。然后，超时错误可由两个输出 bError 和 nErrorId 识别。

工作状态

输出 bBusy 指示功能块是否繁忙。如果为 bBusy == true，则不能触发新命令。

通过适当的 if 查询，可以确定功能块的执行是否完成：

```
if (!fbReadOrWrite.bBusy)
{
    // Function block execution is complete or error occurred.
    // Next write-command can be issued.
}
```

错误状态

如果适用，错误状态通过 bError 以二进制方式表示错误，而 nErrorId 输出错误代码（ADS 返回代码 [▸ 2753]）。一旦有新命令被触发，错误状态就会被重置。如果功能块繁忙（bBusy == true），错误代码为 PENDING (0x71E)。

错误代码的十六进制值可以如下提取：

```
if (fbReadOrWrite.bError)
{
    // Show relevant error code for debugging purposes:
    nErrorCode = fbReadOrWrite.nErrorId & 0xFFF;
}
```

最常见的错误代码包括：

Hex	Dec	名称	含义
0x006	6	TARGETPORTNOTFOUND	ADS 服务器无法访问。基本上是在 TwinCAT Vision Service 没有启动时发生。请参见 视觉服务不启动 [▶ 2739]。
0x700	1792	ERROR	保存或加载失败时出现一般错误。这通常表明指定的文件格式不受支持或者文件路径无法访问。 该错误通常发生在 TwinCAT Vision Service [▶ 59] 中。
0x70B	1803	INVALIDPARM	无效参数。这通常表明接口指针 [▶ 119]无效。
0x719	1817	TIMEOUT	超时。加载或保存已超过最大执行时间 nTimeout。一般来说，主要发生在 ADS 负载较重的情况下。
0x71E	1822	PENDING	并非真正的错误，因此这里没有设置输出 bError。该代码发出信号，表明功能块处于繁忙状态。如果在功能块繁忙时记录到新上升边缘，其将与输出 bBusy 同时发生。

6.2.4.3.1 InitializeFunction

This FB initializes selected functions. Before, it needs to initialize itself, i.e. call the FB until the output bInitialized is true before using it for function initialization.

Syntax

Constructor:

```
InitializeFunction()
```

Execute main:

```
FB_Main()

FB_Main(
    ETcVnInitializableFunction eFunction,
    ULONGLONG                  nOptions,
    bool                       bStart
)

FB_Main(
    bool bStart
)
```

Inputs

Name	Type	Default	Description
eFunction	ETcVnInitializableFunction [▶ 1620]		Selects the function to initialize
nOptions	ULONGLONG	0	Selects initialization options for the function
bStart	bool		Function initialization is triggered by a rising edge at this input.
nTimeout	unsigned int	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	bool	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	bool	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	ULONG	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.3.2 ReadCalibrationPattern

This FB reads calibration pattern reference points from an xml file on the targte pc.

Syntax

Constructor:

```
ReadCalibrationPattern()
```

Execute main:

```
FB_Main()

FB_Main(
    PCCH              sFilePath,
    ITcVnContainer** pipDestContainer,
    bool              bRead
)

FB_Main(
```

```
    bool bRead
)
```

Inputs

Name	Type	Default	Description
sFilePath	PCCH		Full path of the file or relative path to the default directory on the target pc
pipDestContainer	ITcVnContainer* [▶ 1760]*		Returns a container with the pattern points (ContainerType_Vector_TcVnPoint3_REAL)
bRead	bool		Reading the file is triggered by a rising edge at this input.
nTimeout	unsigned int	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	bool	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	bool	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	ULONG	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.3.3 ReadCalibrationResult

This FB reads calibration results from a json file on the targte pc.

Syntax

Constructor:

```
ReadCalibrationResult()
```

Execute main:

```
FB_Main()

FB_Main(
```

```
    PCCH                  sFilePath,
    TcVnMatrix3x3_LREAL* paCameraMatrix,
    TcVnArray8_LREAL*    paDistortionCoefficients,
    TcVnMatrix3x3_LREAL* paRotationMatrix,
    TcVnVector3_LREAL*   paTranslationVector,
    bool                 bRead
)

FB_Main(
    bool bRead
)
```

Inputs

Name	Type	Default	Description
sFilePath	PCCH		Full path of the file or relative path to the default directory on the target pc
paCameraMatrix	TcVnMatrix3x3_LREAL [▸ 1590]*		Returns the camera matrix
paDistortionCoefficients	TcVnArray8_LREAL [▸ 1590]*		Returns the distortion coefficients
paRotationMatrix	TcVnMatrix3x3_LREAL [▸ 1590]*		Returns the rotation matrix
paTranslationVector	TcVnVector3_LREAL [▸ 1590]*		Returns the translation vector
bRead	bool		Reading the file is triggered by a rising edge at this input.
nTimeout	unsigned int	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	bool	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	bool	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	ULONG	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.3.4 ReadContainer

This FB reads a container from an xml file on the target pc.

Syntax

Constructor:

```
ReadContainer()
```

Execute main:

```
FB_Main()

FB_Main(
    PCCH                sFilePath,
    ITcVnContainer** pipDestContainer,
    bool                bRead
)

FB_Main(
    PCCH                sFilePath,
    ITcVnContainer** pipDestContainer,
    GUID                nDestTypeGuid,
    bool                bRead
)

FB_Main(
    bool bRead
)
```

Inputs

Name	Type	Default	Description
sFilePath	PCCH		Full path of the file or relative path to the default directory on the target pc
pipDestContainer	ITcVnContainer* [▶ 1760]*		Returns a container with the loaded content
nDestTypeGuid	GUID		The type id of the container to return. If this parameter is not set, a suitable container type will be determined automatically (except for csv files).
bRead	bool		Reading the file is triggered by a rising edge at this input.
nTimeout	unsigned int	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	bool	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	bool	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	ULONG	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.3.5 ReadImage

This FB reads an image from a file on the target pc.

Syntax

Constructor:

```
ReadImage()
```

Execute main:

```
FB_Main()

FB_Main(
    PCCH          sFilePath,
    ITcVnImage** pipDestImage,
    bool          bRead
)

FB_Main(
    bool bRead
)
```

Inputs

Name	Type	Default	Description
sFilePath	PCCH		Full path of the file or relative path to the default image directory on the target pc
pipDestImage	ITcVnImage* [▶ 1797]*		Returns an image with the loaded content
bRead	bool		Reading the file is triggered by a rising edge at this input.
nTimeout	unsigned int	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	bool	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	bool	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId' . If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	ULONG	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.3.6 ReadMlModel

This FB reads a machine learning model from a file on the target pc.

Syntax

Constructor:

```
ReadMlModel()
```

Execute main:

```
FB_Main()

FB_Main(
    PCCH           sFilePath,
    ITcVnMlModel** pipDestMlModel,
    bool           bRead
)

FB_Main(
```

```
    bool bRead
)
```

Inputs

Name	Type	Default	Description
sFilePath	PCCH		Full path of the file or relative path to the default directory on the target pc
pipDestMlModel	ITcVnMlModel* [▶ 1804]*		Returns the loaded machine learning model
bRead	bool		Reading the file is triggered by a rising edge at this input.
nTimeout	unsigned int	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	bool	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	bool	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	ULONG	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.3.7 WriteCalibrationResult

This FB writes calibration results as a json file to the hard drive.

Syntax

Constructor:

```
WriteCalibrationResult()
```

Execute main:

```
FB_Main()

FB_Main(
    PCCH                 sFilePath,
    TcVnMatrix3x3_LREAL* paCameraMatrix,
    TcVnArray8_LREAL*    paDistortionCoefficients,
```

```
    TcVnMatrix3x3_LREAL* paRotationMatrix,
    TcVnVector3_LREAL*   paTranslationVector,
    bool                 bWrite
)

FB_Main(
    bool bWrite
)
```

Inputs

Name	Type	Default	Description
sFilePath	PCCH		Full path of the file or relative path to the default directory on the target pc (e.g. only a file name). If the string is empty, a file name will be generated (containing the current time).
paCameraMatrix	TcVnMatrix3x3_LREAL [▶ 1590]*		Returns the camera matrix
paDistortionCoefficients	TcVnArray8_LREAL [▶ 1590]*		Returns the distortion coefficients
paRotationMatrix	TcVnMatrix3x3_LREAL [▶ 1590]*		Returns the rotation matrix
paTranslationVector	TcVnVector3_LREAL [▶ 1590]*		Returns the translation vector
bWrite	bool		Writing the container is triggered by a rising edge at this input.
nTimeout	unsigned int	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	bool	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	bool	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	ULONG	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.3.8 WriteContainer

This FB writes a container as an xml file to the hard drive.

Syntax

Constructor:

```
WriteContainer()
```

Execute main:

```
FB_Main()

FB_Main(
    ITcVnContainer**            pipContainer,
    PCCH                        sFilePath,
    ETcVnContainerExportFormat eExportFormat,
    bool                        bWrite
)

FB_Main(
    bool bWrite
)
```

Inputs

Name	Type	Default	Description
pipContainer	ITcVnContainer* [▶ 1760]*		The container to write
sFilePath	PCCH		Full path of the file or relative path to the default directory on the target pc (e.g. only a file name). If the string is empty, a file name will be generated (containing the current time).
eExportFormat	ETcVnContainerExportFormat [▶ 1610]	CEF_XML	Container export format (default is human readable xml)
bWrite	bool		Writing the container is triggered by a rising edge at this input.
nTimeout	unsigned int	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	bool	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	bool	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	ULONG	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.3.9 WriteImage

This FB writes an image to the hard drive.

Syntax

Constructor:

```
WriteImage()
```

Execute main:

```
FB_Main()

FB_Main(
    ITcVnImage** pipImage,
    PCCH         sFilePath,
    bool         bWrite
)

FB_Main(
    bool bWrite
)
```

Inputs

Name	Type	Default	Description
pipImage	ITcVnImage* [▸ 1797]*		The image to write
sFilePath	PCCH		Full path of the file or relative path to the default directory on the target pc (e.g. only a file name). If the string is empty, a file name will be generated (containing the current time).
bWrite	bool		Writing the image is triggered by a rising edge at this input.
nTimeout	unsigned int	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	bool	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	bool	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	ULONG	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.3.10 WriteMlModel

This FB writes a machine learning model to the hard drive.

Syntax

Constructor:

```
WriteMlModel()
```

Execute main:

```
FB_Main()

FB_Main(
    ITcVnMlModel** pipMlModel,
    PCCH           sFilePath,
    bool           bWrite
)

FB_Main(
    bool bWrite
)
```

Inputs

Name	Type	Default	Description
pipMlModel	ITcVnMlModel* [▶ 1804]*		The machine learning model to write
sFilePath	PCCH		Full path of the file or relative path to the default directory on the target pc (e.g. only a file name). If the string is empty, a file name will be generated (containing the current time).
bWrite	bool		Writing the machine learning model is triggered by a rising edge at this input.
nTimeout	unsigned int	VISION_ADS_TIMEOUT	Indicates the time before the function is cancelled.

Outputs

Name	Type	Description
bBusy	bool	This output remains TRUE until the function block has executed a command, but at the longest for the duration supplied to the 'nTimeout' input. While bBusy = TRUE, no new command will be accepted at the inputs.
bError	bool	This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the function block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.
nErrorId	ULONG	Contains the ADS error code or the command-specific error code of the last executed command. Is reset to 0 by the execution of a command at the inputs.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.4 Image Acquisition

该组包含用于通过 Vision 设备（GigE Vision 相机 [▶ 63]和文件源 [▶ 110]对象）采集图像的功能块

功能块

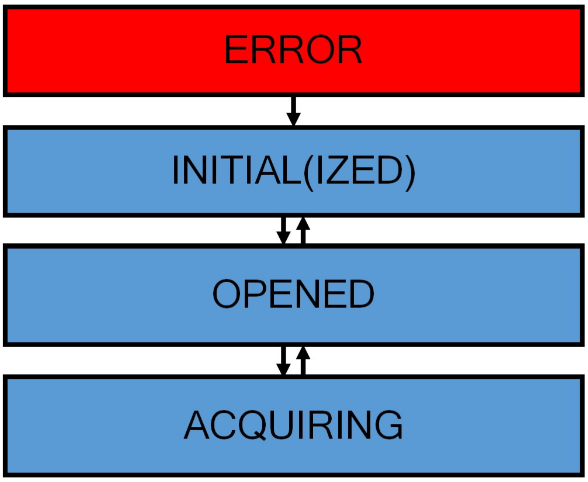
- SimpleCameraControl [▶ 2591]，用于相机和文件源
- GevCameraControl [▶ 2580]，用于 GigE Vision 相机
- FileSourceControl [▶ 2577]，用于文件源

通用

该组中的功能块代表图像提供程序 TcCOM 对象的代理。因此，它们不需要循环调用。相反，TcCOM 对象通过功能块进行控制。

状态机

所有图像采集功能块均包括一个状态机，用于管理所代表设备的状态。基本上，功能块必须置于 ACQUIRING（采集中）状态，才能连续接收图像。INITIAL（初始）状态代表重启或复位后的起始点，可触发 INITIALIZING（初始化）。如为 OPENED（已打开）状态，与设备的通信通道会被打开，以读取或写入寄存器值。



所有状态都由枚举 ETcVnCameraState [▸ 1595] 表示，并在下文中加以说明：

主要状态

主要的状态是静止状态。它们基本上表明是否有与 Vision 设备的连接（OPENED），该设备是否正在捕捉图像（ACQUIRING）以及与该设备的连接是否有错误（ERROR）。你可以通过 FB 的方法在这些状态之间切换。一般来说，两个主要状态之间的转换不会立即发生，而是至少经过一个周期。在这个中间时间，视觉设备处于中间状态 [▸ 2576]。

状态	描述
ERROR	错误状态，在这个状态下，每个设备只能通过调用方法Reset重新激活。
INITIAL	每个设备的初始状态。
INITIALIZED	与 INITIAL 类似，手动初始化相机（例如使用 Force-IP 和初始化命令）
OPENED	存在与相机的连接，可以触发单个图像，且可以读写相机寄存器。 **这个状态可以通过 StartAcquisition 方法跳过。**
ACQUIRING	设备处于采集状态，且正在发送图像，具体取决于设置。

中间状态

中间状态十分必要，因为大多数行动不能立即完成（如图像采集）。如果功能块处于这些中间状态之一，必须再次调用相应的方法，以完成向相应的主状态的过渡。

状态	描述
INITIALIZING	INITIAL > INITIALIZED
OPENING	INITIAL > OPENED 且 INITIALIZED > OPENED
STARTACQUISITION	OPENED > ACQUIRING.
STOPACQUISITION	ACQUIRING > OPENED
RESETTINGFEATURES	OPENED > OPENED
TRIGGERING	OPENED > OPENED 且 ACQUIRING > ACQUIRING
CLOSING	OPENED > INITIALIZED

必须在中间状态下调用方法！

在中间状态中也必须调用各自的状态转换方法。在相应的方法调用后使用GetState方法检测到静止状态时，状态转换才算完成，且不再需要调用该方法。触发一次方法，而随后的等待并不会导致所需的状态转换。

然后，C++ 中的状态机既可以用 switch 结构实现，也可以用 if … else if … 结构实现。

6.2.4.4.1 FileSourceControl

This FB provides access to the images send by a File Source instance.

Syntax

Constructor:

```
FileSourceControl(
    OTCID oidITcVnFileImageProvider
)
```

Methods

Name	Description
GetCurrentImage [▶ 2577]	Gets the current available image (if any).
GetCurrentImageAndFileName [▶ 2578]	Gets the current available image (if any) and the corresponding file name.
StartAcquisition [▶ 2578]	Activates the File Source. If not in trigger mode, it will start sending images.
StopAcquisition [▶ 2578]	Deactivates the File Source. It will stop sending images.
Reset [▶ 2579]	Reset the controller to initial state (might require multiple calls depending on current state, until S_OK is returned)
GetState [▶ 2579]	Gets the current state of the internal camera control state machine.
TriggerImage [▶ 2579]	If in trigger mode, this function triggers the next image in the list.
TriggerImageExp [▶ 2579]	If in trigger mode, this function triggers a specific image in the list.
TriggerImageByName [▶ 2580]	If in trigger mode, this function triggers a specific image in the list by name.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.4.1.1 GetCurrentImage

Gets the current available image (if any).

Syntax

Definition:

```
HRESULT GetCurrentImage(
    ITcVnImage*& ipImage
)
```

Parameters

Name	Type	Description
ipImage	ITcVnImage* [▶ 1797]&	Returns the image

Return value

HRESULT [▶ 122]

6.2.4.4.1.2 GetCurrentImageAndFileName

Gets the current available image (if any) and the corresponding file name.

Syntax

Definition:

```
HRESULT GetCurrentImageAndFileName(
    ITcVnImage*& ipImage,
    PCHAR        sFileName,
    USHORT       nMaxLen
)
```

Parameters

Name	Type	Description
ipImage	ITcVnImage* [▶ 1797]&	Returns the image
sFileName	PCHAR	Returns the file name
nMaxLen	USHORT	Define the maximum allowed length for the file name

Return value

HRESULT [▶ 122]

6.2.4.4.1.3 StartAcquisition

Activates the File Source. If not in trigger mode, it will start sending images.

Syntax

Definition:

```
HRESULT StartAcquisition()
```

Return value

HRESULT [▶ 122]

6.2.4.4.1.4 StopAcquisition

Deactivates the File Source. It will stop sending images.

Syntax

Definition:

```
HRESULT StopAcquisition()
```

Return value

HRESULT [▶ 122]

6.2.4.4.1.5 Reset

Reset the controller to initial state (might require multiple calls depending on current state, until S_OK is returned)

Syntax

Definition:

```
HRESULT Reset()
```

Return value

HRESULT [▸ 122]

6.2.4.4.1.6 GetState

Gets the current state of the internal camera control state machine.

Syntax

Definition:

```
ETcVnCameraState GetState()
```

Return value

ETcVnCameraState [▸ 1595]

6.2.4.4.1.7 TriggerImage

If in trigger mode, this function triggers the next image in the list.

Syntax

Definition:

```
HRESULT TriggerImage()
```

Return value

HRESULT [▸ 122]

6.2.4.4.1.8 TriggerImageExp

If in trigger mode, this function triggers a specific image in the list.

Syntax

Definition:

```
HRESULT TriggerImageExp(
    LONG nSkipImages
)
```

Parameters

Name	Type	Description
nSkipImages	LONG	Amount of images to skip, relative to current

Return value

HRESULT [▶ 122]

6.2.4.4.1.9 TriggerImageByName

If in trigger mode, this function triggers a specific image in the list by name.

Syntax

Definition:

```
HRESULT TriggerImageByName(
    PCCH sImageName
)
```

Parameters

Name	Type	Description
sImageName	PCCH	Name of the image to trigger

Return value

HRESULT [▶ 122]

6.2.4.4.2 GevCameraControl

This FB provides the basic functionality to control a GigE Vision camera and access its calibration data.

Syntax

Constructor:

```
GevCameraControl(
    OTCID oidITcVnGevImageProvider
)
```

Methods

Name	Description
ClearImageQueue [▶ 2583]	Delete all images contained in the TcVnGevImageProvider TcCOM module receive queue and reset the corresponding omitted images counter.
CloseCamera [▶ 2583]	Close the control channel to the camera.
GetCalibPatternRef [▶ 2583]	Gets the reference calibration pattern points from the TcVnGevImageProvider TcCOM module (Can be set from the calibration assistant)
GetCameraMatrix [▶ 2584]	Gets the camera matrix from the TcVnGevImageProvider TcCOM module (requires intrinsic calibration parameters, e.g. set by the calibration assistant)
GetCurrentImage [▶ 2584]	Gets the current available image (first in receive queue).
GetCurrentImageUndistorted [▶ 2585]	Gets the current available image (first in receive queue) with undistortion applied. Incompatible with pixel formats that contain an encoding, e.g. a Bayer pattern. Requires intrinsic calibration parameters in the TcVnGevImageProvider TcCOM module, e.g. set by the calibration assistant.
GetCurrentImageWithGvspInfo [▶ 2585]	Gets the current available image (first in receive queue) and its corresponding GVSP info.
GetCurrentImageWithGvspInfoUndistorted [▶ 2585]	Gets the current available image (first in receive queue) with its corresponding GVSP info and undistortion applied. Incompatible with pixel formats that contain an encoding, e.g. a Bayer pattern. Requires intrinsic calibration parameters in the TcVnGevImageProvider TcCOM module, e.g. set by the calibration assistant.
GetCurrentImageWithTimestamps [▶ 2586]	Gets the current available image (first in receive queue) and corresponding timestamps.
GetDistortionCoefficients [▶ 2586]	Gets the distortion coefficients from the TcVnGevImageProvider TcCOM module (requires intrinsic calibration parameters, e.g. set by the calibration assistant)
GetLastImageFromQueue [▶ 2587]	Gets the last received image from the queue.
GetOmittedImagesNum [▶ 2587]	Gets the number of omitted images since the last call of ClearImageQueue(). If the image receive queue in the TcVnGevImageProvider TcCOM module is full and a new image arrives, the first one in the queue will be deleted and the omitted counter is increased.
GetRotationMatrix [▶ 2587]	Gets the rotation matrix from the TcVnGevImageProvider TcCOM module (requires extrinsic calibration parameters, e.g. set by the calibration assistant)
GetState [▶ 2588]	Gets the current state of the internal camera control state machine.
GetTranslationVector [▶ 2588]	Gets the translation vector from the TcVnGevImageProvider TcCOM module (requires extrinsic calibration parameters, e.g. set by the calibration assistant)
InitializeCamera [▶ 2588]	Initialize the camera to the intended state (includes sending the 'InitCameraCommands' defined in the GevImageAcquisition TcCOM module to the camera).
OpenCamera [▶ 2589]	Open a control channel to the camera.
Reset [▶ 2589]	Reset the camera controller to initial state (might require multiple calls depending on current state, until S_OK is returned)
ResetCameraFeatures [▶ 2589]	Reset the camera features to initial state (sends the 'InitCameraCommands' defined in the GevImageAcquisition TcCOM module to the camera).
SetCameraMatrix [▶ 2589]	Sets the camera matrix to the TcVnGevImageProvider TcCOM module

Name	Description
SetDistortionCoefficients [▶ 2590]	Sets the distortion coefficients to the TcVnGevImageProvider TcCOM module
SetRotationMatrix [▶ 2590]	Sets the rotation matrix to the TcVnGevImageProvider TcCOM module
SetTranslationVector [▶ 2590]	Sets the translation vector to the TcVnGevImageProvider TcCOM module
StartAcquisition [▶ 2591]	Send the 'StartAcquisitionCommands' defined in the TcVnGevImageProvider TcCOM module to the camera.
StopAcquisition [▶ 2591]	Send the 'StopAcquisitionCommands' defined in the TcVnGevImageProvider TcCOM module to the camera.
TriggerImage [▶ 2591]	Send the 'SoftwareTriggerCommands' defined in the TcVnGevImageProvider TcCOM module to the camera.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.4.2.1 ClearImageQueue

Delete all images contained in the TcVnGevImageProvider TcCOM module receive queue and reset the corresponding omitted images counter.

Syntax

Definition:

```
HRESULT ClearImageQueue()
```

Return value

HRESULT [▶ 122]

6.2.4.4.2.2 CloseCamera

Close the control channel to the camera.

Syntax

Definition:

```
HRESULT CloseCamera()
```

Return value

HRESULT [▶ 122]

6.2.4.4.2.3 GetCalibPatternRef

Gets the reference calibration pattern points from the TcVnGevImageProvider TcCOM module (Can be set from the calibration assistant)

Syntax

Definition:

```
HRESULT GetCalibPatternRef(
    ITcVnContainer*& ipCalibPatternRef
)
```

Parameters

Name	Type	Description
ipCalibPatternRef	ITcVnContainer* [▶ 1760]&	Returns the reference calibration pattern points (ContainerType_Vector_TcVnPoint3_REAL)

Return value

HRESULT [▶ 122]

6.2.4.4.2.4 GetCameraMatrix

Gets the camera matrix from the TcVnGevImageProvider TcCOM module (requires intrinsic calibration parameters, e.g. set by the calibration assistant)

Syntax

Definition:

```
HRESULT GetCameraMatrix(
    TcVnMatrix3x3_LREAL& aCameraMatrix
)
```

Parameters

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	Returns the camera matrix

Return value

HRESULT [▶ 122]

6.2.4.4.2.5 GetCurrentImage

Gets the current available image (first in receive queue).

Syntax

Definition:

```
HRESULT GetCurrentImage(
    ITcVnImage*& ipImage
)
```

Parameters

Name	Type	Description
ipImage	ITcVnImage* [▶ 1797]&	Returns the image

Return value

HRESULT [▶ 122]

6.2.4.4.2.6 GetCurrentImageUndistorted

Gets the current available image (first in receive queue) with undistortion applied. Incompatible with pixel formats that contain an encoding, e.g. a Bayer pattern. Requires intrinsic calibration parameters in the TcVnGevImageProvider TcCOM module, e.g. set by the calibration assistant.

Syntax

Definition:

```
HRESULT GetCurrentImageUndistorted(
    ITcVnImage*& ipImage
)
```

Parameters

Name	Type	Description
ipImage	ITcVnImage* [▸ 1797]&	Returns the undistorted image

Return value

HRESULT [▸ 122]

6.2.4.4.2.7 GetCurrentImageWithGvspInfo

Gets the current available image (first in receive queue) and its corresponding GVSP info.

Syntax

Definition:

```
HRESULT GetCurrentImageWithGvspInfo(
    ITcVnImage*&     ipImage,
    GVSP_IMAGE_INFO& stGvspInfo
)
```

Parameters

Name	Type	Description
ipImage	ITcVnImage* [▸ 1797]&	Returns the image
stGvspInfo	GVSP_IMAGE_INFO [▸ 1637]&	Returns the GVSP info

Return value

HRESULT [▸ 122]

6.2.4.4.2.8 GetCurrentImageWithGvspInfoUndistorted

Gets the current available image (first in receive queue) with its corresponding GVSP info and undistortion applied. Incompatible with pixel formats that contain an encoding, e.g. a Bayer pattern. Requires intrinsic calibration parameters in the TcVnGevImageProvider TcCOM module, e.g. set by the calibration assistant.

Syntax

Definition:

```
HRESULT GetCurrentImageWithGvspInfoUndistorted(
    ITcVnImage*&     ipImage,
    GVSP_IMAGE_INFO& stGvspInfo
)
```

Parameters

Name	Type	Description
ipImage	ITcVnImage* [▶ 1797]&	Returns the undistorted image
stGvspInfo	GVSP_IMAGE_INFO [▶ 1637]&	Returns the GVSP info

Return value

HRESULT [▶ 122]

6.2.4.4.2.9 GetCurrentImageWithTimestamps

Gets the current available image (first in receive queue) and corresponding timestamps.

Syntax

Definition:

```
HRESULT GetCurrentImageWithTimestamps(
    ITcVnImage*&     ipImage,
    ULONGLONG        nTimestamps,
    LONGLONG*        pTimestamps,
    GVSP_IMAGE_INFO& stGvspInfo
)
```

Parameters

Name	Type	Description
ipImage	ITcVnImage* [▶ 1797]&	Returns the image
nTimestamps	ULONGLONG	Select timestamps to return (ETcVnTimestamp)
pTimestamps	LONGLONG*	Returns the requested timestamps (make sure to provide an array of sufficient size).
stGvspInfo	GVSP_IMAGE_INFO [▶ 1637]&	Returns the GVSP info

Return value

HRESULT [▶ 122]

6.2.4.4.2.10 GetDistortionCoefficients

Gets the distortion coefficients from the TcVnGevImageProvider TcCOM module (requires intrinsic calibration parameters, e.g. set by the calibration assistant)

Syntax

Definition:

```
HRESULT GetDistortionCoefficients(
    TcVnArray8_LREAL& aDistortionCoefficients
)
```

Parameters

Name	Type	Description
aDistortionCoefficients	TcVnArray8_LREAL [▶ 1590]&	Returns the distortion coefficients [k1, k2, p1, p2, k3, k4, k5, k6]

Return value

HRESULT [▶ 122]

6.2.4.4.2.11 GetLastImageFromQueue

Gets the last received image from the queue.

Syntax

Definition:

```
HRESULT GetLastImageFromQueue(
    ITcVnImage*& ipImage
)
```

Parameters

Name	Type	Description
ipImage	ITcVnImage* [▶ 1797]&	Returns the image

Return value

HRESULT [▶ 122]

6.2.4.4.2.12 GetOmittedImagesNum

Gets the number of omitted images since the last call of ClearImageQueue(). If the image receive queue in the TcVnGevImageProvider TcCOM module is full and a new image arrives, the first one in the queue will be deleted and the omitted counter is increased.

Syntax

Definition:

```
HRESULT GetOmittedImagesNum(
    ULONGLONG& nOmitted
)
```

Parameters

Name	Type	Description
nOmitted	ULONGLONG&	Returns the number of omitted images

Return value

HRESULT [▶ 122]

6.2.4.4.2.13 GetRotationMatrix

Gets the rotation matrix from the TcVnGevImageProvider TcCOM module (requires extrinsic calibration parameters, e.g. set by the calibration assistant)

Syntax

Definition:

```
HRESULT GetRotationMatrix(
    TcVnMatrix3x3_LREAL& aRotationMatrix
)
```

Parameters

Name	Type	Description
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	Returns the rotation matrix

Return value

HRESULT [▶ 122]

6.2.4.4.2.14 GetState

Gets the current state of the internal camera control state machine.

Syntax

Definition:

```
ETcVnCameraState GetState()
```

Return value

ETcVnCameraState [▶ 1595]

6.2.4.4.2.15 GetTranslationVector

Gets the translation vector from the TcVnGevImageProvider TcCOM module (requires extrinsic calibration parameters, e.g. set by the calibration assistant)

Syntax

Definition:

```
HRESULT GetTranslationVector(
    TcVnVector3_LREAL& aTranslationVector
)
```

Parameters

Name	Type	Description
aTranslationVector	TcVnVector3_LREAL [▶ 1590]&	Returns the translation vector

Return value

HRESULT [▶ 122]

6.2.4.4.2.16 InitializeCamera

Initialize the camera to the intended state (includes sending the 'InitCameraCommands' defined in the GevImageAcquisition TcCOM module to the camera).

Syntax

Definition:

```
HRESULT InitializeCamera()
```

Return value

HRESULT [▶ 122]

6.2.4.4.2.17 OpenCamera

Open a control channel to the camera.

Syntax

Definition:

```
HRESULT OpenCamera()
```

Return value

HRESULT [▶ 122]

6.2.4.4.2.18 Reset

Reset the camera controller to initial state (might require multiple calls depending on current state, until S_OK is returned)

Syntax

Definition:

```
HRESULT Reset()
```

Return value

HRESULT [▶ 122]

6.2.4.4.2.19 ResetCameraFeatures

Reset the camera features to initial state (sends the 'InitCameraCommands' defined in the GevImageAcquisition TcCOM module to the camera).

Syntax

Definition:

```
HRESULT ResetCameraFeatures()
```

Return value

HRESULT [▶ 122]

6.2.4.4.2.20 SetCameraMatrix

Sets the camera matrix to the TcVnGevImageProvider TcCOM module

Syntax

Definition:

```
HRESULT SetCameraMatrix(
    TcVnMatrix3x3_LREAL& aCameraMatrix
)
```

Parameters

Name	Type	Description
aCameraMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	The camera matrix to be copied to the TcCOM module

Return value

HRESULT [▶ 122]

6.2.4.4.2.21 SetDistortionCoefficients

Sets the distortion coefficients to the TcVnGevImageProvider TcCOM module

Syntax

Definition:

```
HRESULT SetDistortionCoefficients(
    TcVnArray8_LREAL& aDistortionCoefficients
)
```

Parameters

Name	Type	Description
aDistortionCoefficients	TcVnArray8_LREAL [▶ 1590]&	The distortion coefficients to be copied to the TcCOM module

Return value

HRESULT [▶ 122]

6.2.4.4.2.22 SetRotationMatrix

Sets the rotation matrix to the TcVnGevImageProvider TcCOM module

Syntax

Definition:

```
HRESULT SetRotationMatrix(
    TcVnMatrix3x3_LREAL& aRotationMatrix
)
```

Parameters

Name	Type	Description
aRotationMatrix	TcVnMatrix3x3_LREAL [▶ 1590]&	The rotation matrix to be copied to the TcCOM module

Return value

HRESULT [▶ 122]

6.2.4.4.2.23 SetTranslationVector

Sets the translation vector to the TcVnGevImageProvider TcCOM module

Syntax

Definition:

```
HRESULT SetTranslationVector(
    TcVnVector3_LREAL& aTranslationVector
)
```

Parameters

Name	Type	Description
aTranslationVector	TcVnVector3_LREAL [▸ 1590]&	The translation vector to be copied to the TcCOM module

Return value

HRESULT [▸ 122]

6.2.4.4.2.24 StartAcquisition

Send the 'StartAcquisitionCommands' defined in the TcVnGevImageProvider TcCOM module to the camera.

Syntax

Definition:

```
HRESULT StartAcquisition()
```

Return value

HRESULT [▸ 122]

6.2.4.4.2.25 StopAcquisition

Send the 'StopAcquisitionCommands' defined in the TcVnGevImageProvider TcCOM module to the camera.

Syntax

Definition:

```
HRESULT StopAcquisition()
```

Return value

HRESULT [▸ 122]

6.2.4.4.2.26 TriggerImage

Send the 'SoftwareTriggerCommands' defined in the TcVnGevImageProvider TcCOM module to the camera.

Syntax

Definition:

```
HRESULT TriggerImage()
```

Return value

HRESULT [▸ 122]

6.2.4.4.3 SimpleCameraControl

This FB provides the basic functionality to control a camera or a FileSource.

Syntax

Constructor:

```
SimpleCameraControl(
    OTCID oidITcVnImageProvider
)
```

Methods

Name	Description
GetCurrentImage [▸ 2592]	Get the current available image (if any).
GetState [▸ 2592]	Get the current state of the internal camera control state machine.
Reset [▸ 2593]	Reset the camera to initial state (might require multiple calls depending on current state, until S_OK is returned)
StartAcquisition [▸ 2593]	Start the image acquisition.
StopAcquisition [▸ 2593]	Stop the image acquisition.
TriggerImage [▸ 2593]	Trigger the next image.

Required License

TC3 Vision Base

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.4.3.1 GetCurrentImage

Get the current available image (if any).

Syntax

Definition:

```
HRESULT GetCurrentImage(
    ITcVnImage*& ipImage
)
```

Parameters

Name	Type	Description
ipImage	ITcVnImage* [▸ 1797]&	Returns the image

Return value

HRESULT [▸ 122]

6.2.4.4.3.2 GetState

Get the current state of the internal camera control state machine.

Syntax

Definition:

```
ETcVnCameraState GetState()
```

Return value

ETcVnCameraState [▶ 1595]

6.2.4.4.3.3 Reset

Reset the camera to initial state (might require multiple calls depending on current state, until S_OK is returned)

Syntax

Definition:

```
HRESULT Reset()
```

Return value

HRESULT [▶ 122]

6.2.4.4.3.4 StartAcquisition

Start the image acquisition.

Syntax

Definition:

```
HRESULT StartAcquisition()
```

Return value

HRESULT [▶ 122]

6.2.4.4.3.5 StopAcquisition

Stop the image acquisition.

Syntax

Definition:

```
HRESULT StopAcquisition()
```

Return value

HRESULT [▶ 122]

6.2.4.4.3.6 TriggerImage

Trigger the next image.

Syntax

Definition:

```
HRESULT TriggerImage()
```

Return value

HRESULT [▶ 122]

6.2.4.5 Image Processing

该组包含用于图像处理的功能块。在这些综合功能块中，设定值在执行后会被保留。此外，功能块还可以包含多个方法。

函数 [▶ 1812]章节中包含的函数则没有内部状态信息，也不保留变量值。

功能块

- GeneralizedHoughBallard [▶ 2594]
- SSIM [▶ 2597]

6.2.4.5.1 GeneralizedHoughBallard

This FB provides the Generalized Hough Ballard functionality. First, set the parameters via the corresponding methods (optional, defaults are used otherwise). Then, set the template (required). Afterwards, the Detect method can be executed.

Syntax

Constructor:

```
GeneralizedHoughBallard()
```

Methods

Name	Description
Detect [▶ 2594]	Detect the template in an image (SetTemplate must have been called before).
SetCannyThreshold [▶ 2595]	Sets the canny edge detection thresholds.
SetInvAccuRatio [▶ 2595]	Sets the inverted ratio of the accumulator size in relation to the source image's size (e.g. a value of 2 means that the size is halved in both directions.)
SetLevels [▶ 2596]	Sets the number of R-table levels.
SetMinDist [▶ 2596]	Sets the minimum distance between the centers of different objects.
SetTemplate [▶ 2596]	Sets the template image to search for in the Detect method.
SetVotesThreshold [▶ 2597]	Sets the accumulator threshold, i.e. the number of votes required to detect a match (too small values lead to false detections).

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.5.1.1 Detect

Detect the template in an image (SetTemplate must have been called before).

Syntax

Definition:

```
HRESULT Detect(
    ITcVnImage*       ipImage,
    ITcVnContainer*& ipPositions
)
```

Parameters

Name	Type	Description
ipImage	ITcVnImage* [▶ 1797]	Source image (1 channel, USINT)
ipPositions	ITcVnContainer* [▶ 1760]&	Returns the centers of the detected template positions (ContainerType_Vector_TcVnPoint2_REAL; Non-zero interface pointers are reused.)

Return value

HRESULT [▶ 122]

6.2.4.5.1.2 SetCannyThreshold

Sets the canny edge detection thresholds.

Syntax

Definition:

```
HRESULT SetCannyThreshold(
    LONG nLow = 50,
    LONG nHigh = 100
)
```

Parameters

Name	Type	Description
nLow	LONG	Low threshold (> 0)
nHigh	LONG	High threshold (> nLow, usually 2 to 3 * nLow)

Return value

HRESULT [▶ 122]

6.2.4.5.1.3 SetInvAccuRatio

Sets the inverted ratio of the accumulator size in relation to the source image's size (e.g. a value of 2 means that the size is halved in both directions.)

Syntax

Definition:

```
HRESULT SetInvAccuRatio(
    double fInvAccuRatio = 1.0
)
```

Parameters

Name	Type	Description
fInvAccuRatio	double	Inverted accumulator ratio (> 0)

Return value

HRESULT [▶ 122]

6.2.4.5.1.4 SetLevels

Sets the number of R-table levels.

Syntax

Definition:

```
HRESULT SetLevels(
    LONG nLevels = 360
)
```

Parameters

Name	Type	Description
nLevels	LONG	Number of R-table levels (> 0)

Return value

HRESULT [▶ 122]

6.2.4.5.1.5 SetMinDist

Sets the minimum distance between the centers of different objects.

Syntax

Definition:

```
HRESULT SetMinDist(
    double fMinDist = 1.0
)
```

Parameters

Name	Type	Description
fMinDist	double	Minimum distance (> 0)

Return value

HRESULT [▶ 122]

6.2.4.5.1.6 SetTemplate

Sets the template image to search for in the Detect method.

Syntax

Definition:

```
HRESULT SetTemplate(
    ITcVnImage* ipImage
)
```

Parameters

Name	Type	Description
ipImage	ITcVnImage* [▶ 1797]	Template image (1 channel, USINT)

Return value

HRESULT [▶ 122]

6.2.4.5.1.7 SetVotesThreshold

Sets the accumulator threshold, i.e. the number of votes required to detect a match (too small values lead to false detections).

Syntax

Definition:

```
HRESULT SetVotesThreshold(
    LONG nVotes = 100
)
```

Parameters

Name	Type	Description
nVotes	LONG	Number of votes required to detect a match (> 0)

Return value

HRESULT [▶ 122]

6.2.4.5.2 SSIM

This FB provides the structural similarity (SSIM) functionality. First, set the reference image. Afterwards, the Compute method can be executed.

Syntax

Constructor:

```
SSIM()
```

Methods

Name	Description
Compute [▶ 2598]	Compute the structural similarity (SetReference must have been called before).
SetReference [▶ 2598]	Sets the reference image, used by the SSIM method.

Required License

TC3 Vision Matching

System Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT V3.1.4024.54 or later	PC or CX (x64) with PL50, e.g. Intel 4-core Atom CPU	Tc3_Vision

6.2.4.5.2.1 Compute

Compute the structural similarity (SetReference must have been called before).

Syntax

Definition:

```
HRESULT Compute(
    ITcVnImage*  ipSrcImage,
    ITcVnImage*& ipSSIM
)
```

Parameters

Name	Type	Description
ipSrcImage	ITcVnImage* [▶ 1797]	Source image (USINT)
ipSSIM	ITcVnImage* [▶ 1797]&	Returns the SSIM (REAL, values from 0 to 1)

Return value

HRESULT [▶ 122]

6.2.4.5.2.2 SetReference

Sets the reference image, used by the SSIM method.

Syntax

Definition:

```
HRESULT SetReference(
    ITcVnImage* ipImage
)
```

Parameters

Name	Type	Description
ipImage	ITcVnImage* [▶ 1797]	Reference image (USINT)

Return value

HRESULT [▶ 122]

7 TwinCAT HMI 软件包

本章涉及 TwinCAT Vision 与 TwinCAT HMI 的交互。这包括一个服务器扩展 [▸ 2599]，通过服务器扩展可以在 TwinCAT HMI 中显示来自 PLC 的图像 [▸ 130]，以及一个控制包 [▸ 2601]，用于图像和颜色的可视化和互动。

软件包的集成

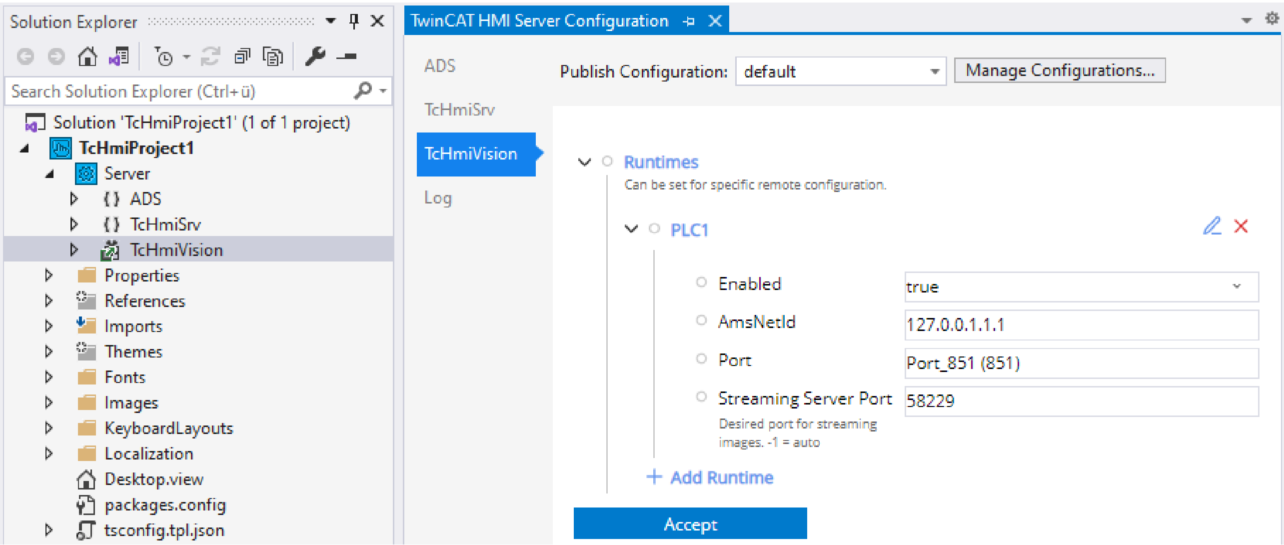
在 TwinCAT HMI 1.12 以上版本中，软件包的集成通过软件包管理NuGet进行。在 TwinCAT HMI 项目中，右键单击**参考**，并选择**管理 NuGet 软件包...**。然后点击**浏览**，并选择“TwinCAT HMI Official”作为**软件包源**。现在将看到正式可用的 TwinCAT HMI 软件包列表，从中可以选择**Beckhoff.TwinCAT.HMI.Vision**等。关于详细信息，请参考TwinCAT HMI 文档。

7.1 服务器扩展

TwinCAT HMI NuGet 包 Beckhoff.TwinCAT.HMI.Vision 包含 HMI 服务器扩展 TcHmiVision，通过该扩展可在 TwinCAT HMI 中显示来自 PLC 的图像。为此，来自配置运行时的所有 ITcVnDisplayableImage [▸ 390] 类型的 ADS 符号均会列出，并可以各种格式与 HMI 符号进行链接。有关所需的 .NET 版本，请参见 系统要求 [▸ 13]。

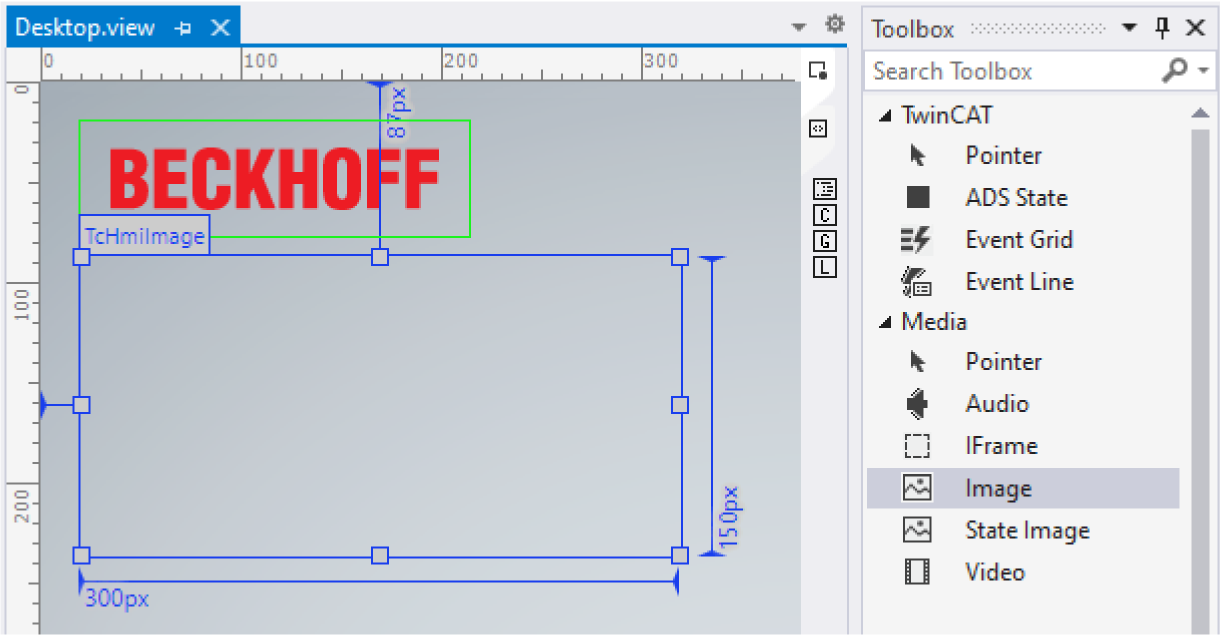
配置

包成功集成到 HMI 项目后，请在 **Server（服务器）** > **TcHmiVision** 下输入目标 PLC：

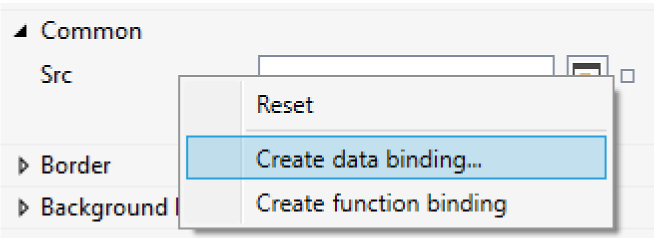


应用

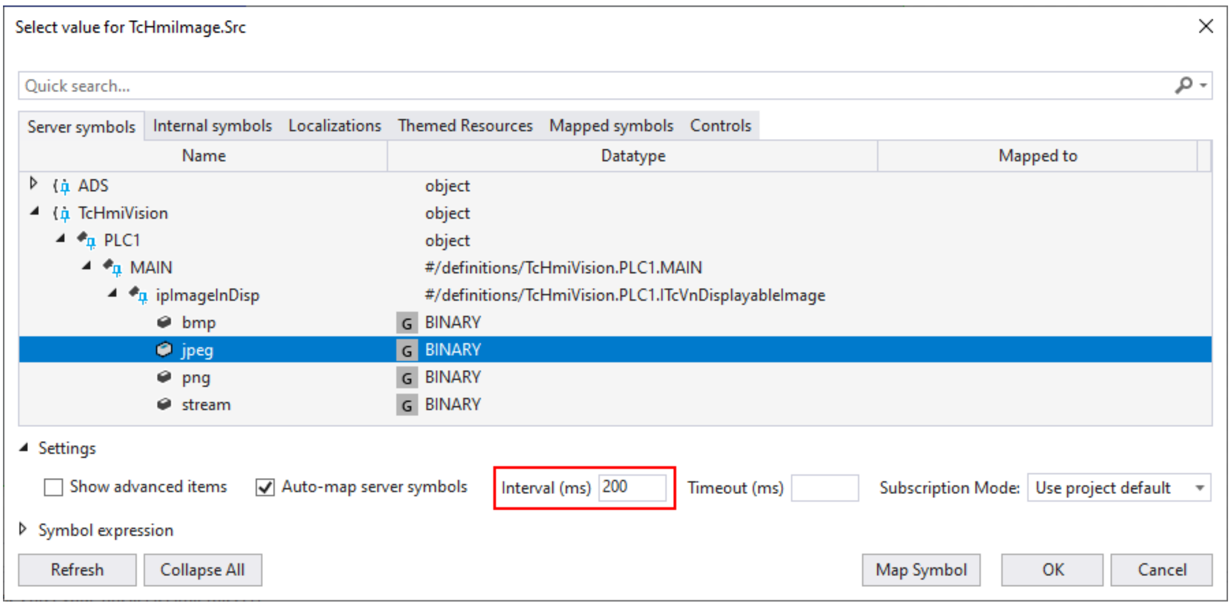
1. 导航至 HMI 视图并添加**图像**控制器。



2. 打开控制元件的属性并点击参数 Src... 下的 Create data binding… （创建数据绑定……）



3. 在以下位置将图像控制器的 Src 参数链接到相应的图像符号：Server symbols（服务器符号） > TcHmiVision > PLC > MAIN > ipImage > <image format>（<图像格式>）



链接图像

只有当配置目标系统的过程图象包含相应图像变量时才能链接图像。

间隔时间

从 PLC 检索图像的速度取决于系统、图像大小和工作量。因此，我们建议在 Settings（设置）的 Binding（绑定）对话框中指定一个固定的间隔时间，以防止系统超载。

如果一个页面上要显示多个图像，也建议使用不同的时间间隔，如 200 ms 和 201 ms。这样，图像之间的检索分开进行，这对于数据量较大的情况有利。

传输格式

服务器扩展包含来自 PLC 的 BMP 格式图像数据，并将其作为 Base64 字符串转发至 TwinCAT HMI 客户端。为了减少传输到客户端的数据量，可以在服务器扩展中对图像进行压缩。提供下列图像格式：

- BMP：未压缩
- JPEG：有损压缩，适用于自然图像
- PNG：压缩（无损），适用于人工创建图像
- 流：适用于连续图像流（例如实时相机图像，不推荐！）。

要以数据流形式传输，必须以管理员身份启动 HMI 服务器。否则无法传输图像。

7.2 控制

NuGet 软件包Beckhoff.TwinCAT.HMI.Vision.Controls提供了一些用于显示和操作视觉数据的控件。本文档介绍了 TwinCAT HMI 概念的基本知识，参见TE2000。关于控制使用的详细描述，也可以参见此处。

控制

该软件包包含以下控件，可以通过 HMI 工具箱添加到 HMI 页面：

控制	类别	描述
图像查看 [▶ 2602]	视觉	• 显示图像 • 缩放和拖动选项 • 提取图像信息 • 通过不同形状标记图像区域 • 多图像选择 • 叠加图形 • 具有操作功能的工具条 • 用于显示图像和形状信息的信息栏
颜色 [▶ 2623]	视觉	显示和设置颜色
矩形（交互式） [▶ 2633]	形状	交互式显示和绘制一个矩形。
多边形（交互式） [▶ 2634]	形状	交互式显示和绘制一个多边形。

形状控制

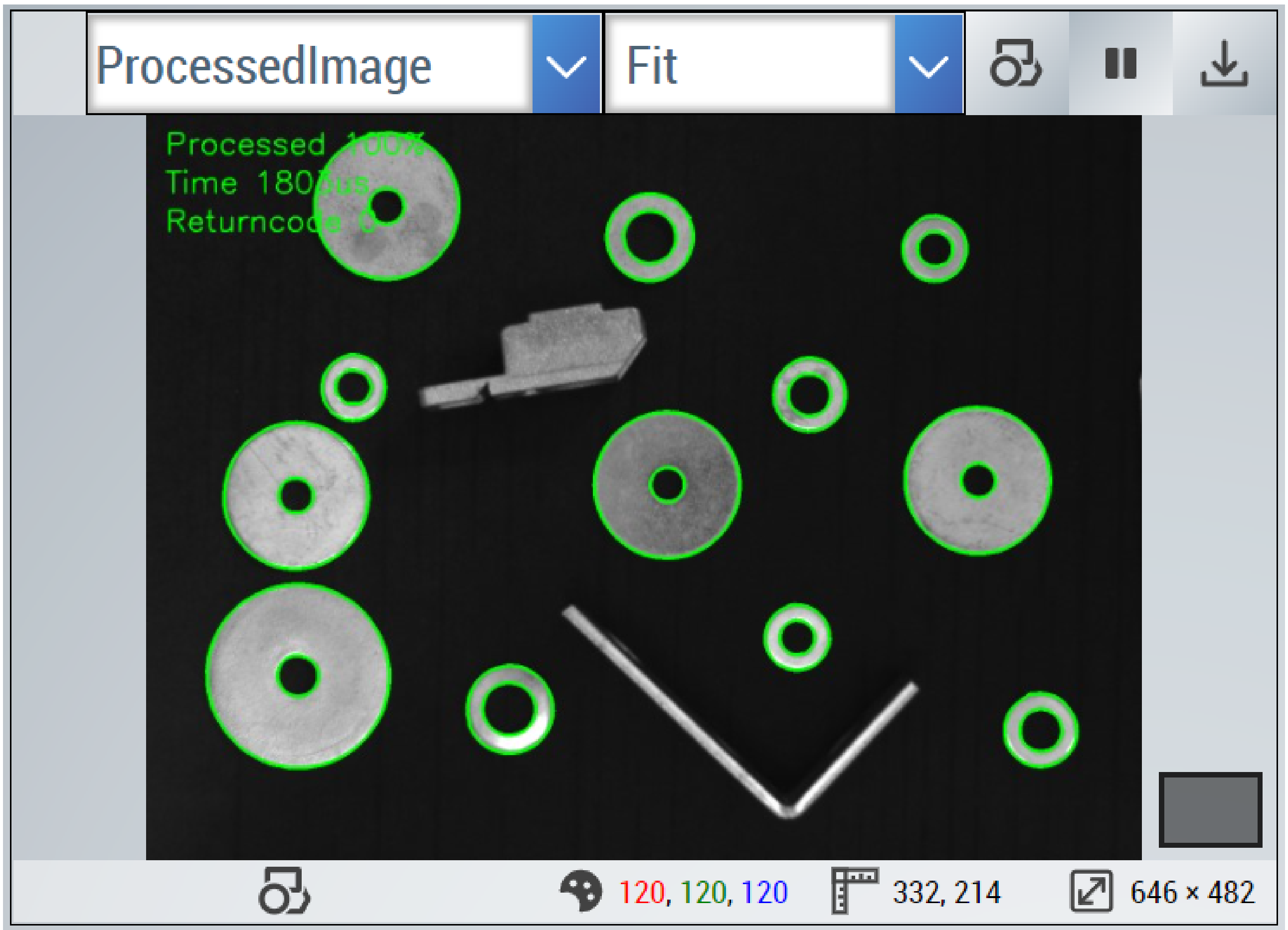
矩形（交互式）和多边形（交互式）控件主要是辅助性控件，在图像查看控件中被重复使用。如有必要，也可以单独使用。此外，通过其属性，控件还可用于派生形状，如点、线、圆、椭圆或方形。

此外，该软件包还包含以下帮助功能：

函数	类别	描述
BoxColorConverter [▸ 2635]	视觉	用作颜色 [▸ 2623]控件的BoxColorConversion属性的默认值。
ConvertColor [▸ 2635]	视觉	在灰色、RGB、HSV 和 HSL 格式之间转换颜色值。
PixelColorFormatter [▸ 2636]	格式	将图像查看 [▸ 2602]控件PixelColor属性形成文本，以显示在工具栏 [▸ 2613]中。
ShapeValueFormatter [▸ 2636]	格式	将图像查看 [▸ 2602]控件ShapeValue属性形成文本，以显示在工具栏 [▸ 2613]中。
ToRotatedRectangle [▸ 2636]	数据转换	将类型为UprightRectangle [▸ 2638]的矩形转换为类型RotatedRectangle [▸ 2638]。
ToUprightRectangle [▸ 2637]	数据转换	将一个类型为RotatedRectangle [▸ 2638]的矩形转换为类型UprightRectangle [▸ 2638]。

7.2.1 图像查看

除图像显示外，**Image Watch** Control 还提供许多其他工具和信息显示。有关各区域和组件的概述，请参见控制结构 [▸ 2608]分章。每个配置选项的说明可在本页的以下类别中找到。每个类别都有分章，提供更多信息、细节和应用示例。为了从 PLC 直接显示 TwinCAT Vision 图像，还需要适用于 HMI 服务器的 Vision 扩展 [▸ 2599]。



对于以下所有属性，控制器有一个 getter 和一个 setter 方法。例如，属性 BarColor 存在 getBarColor(): SolidColor 和setBarColor(value: SolidColor): void 两个方法。属性的使用将在链接分章中进行更详细的说明。

基类**控制器**所提供的所有其他属性、事件和权限均载于 TwinCAT HMI 文档。

类别：颜色

属性	类型	描述
BarColor	SolidColor	工具栏和信息栏颜色（请参见**工具栏和信息栏**类别）。
ShapeStrokeColor	SolidColor	可在图像上绘制的形状轮廓的颜色（请参见**形状**类别）。
ShapeHandleColor	SolidColor	可在图片上绘制的形状手柄颜色（请参见**形状**类别）。

类别：通用

以下属性指定了要显示图像的加载资源 [▶ 2609]。

属性	类型	描述
Image	String	图像显示资源。可以是图像文件的路径，也可以是数据 URI。
ImageList	ImageList	已命名图像资源的列表。如果列表不为空，Image 的绑定将被 ImageIndex 所选的资源替换。
ImageIndex	Number	如果 ImageList 不为空，选择将显示的列表图像资源。
Alt	String	如果当前图像资源不可用，将显示替代文本。

函数	描述
downloadImage(fileName: string): void	触发以原始尺寸对当前显示图像进行浏览器特定下载。 也可通过工具栏 [▶ 2613]中的用户界面元素触发。

类别：视图

以下属性和函数决定了显示图像的部分 [▶ 2611]以及如何对其进行更改。

属性	类型	描述
ImageFreeze	Boolean	如果激活，当前显示的图像将固定，直到 ImageFreeze 禁用后才会更新。
Offset	[Number, Number]	图像左上角在控制器坐标系中的位置。
Scale	Number (%)	根据 ScaleReference 按百分比缩放图像。
ScaleReference	ScaleReference	如为 ImageSize，100% 对应原始图像大小。 如为 ControlWidth，100% 对应控制器的宽度。 如为 ControlHeight，100% 对应控制器的高度。 如为 ControlSize，100% 要么对应控制器的宽度，要么对应控制器的高度，具体取决于哪个先限制图像的大小。
Draggable	Boolean	如果激活，则可以使用鼠标或触控移动图像。在此过程中，Offset 会自动调整。
Scalable	Boolean	如果激活，则可以使用鼠标滚轮或多点触控缩放图像。在此过程中，Offset 和 Scale 会自动进行调整。
ScaleMin	Number	通过鼠标滚轮或多点触控缩放时的最小 Scale。
ScaleMax	Number	通过鼠标滚轮或多点触控缩放时的最大 Scale。
ScalingCenter	ScalingCenter	使用鼠标滚轮缩放时保持固定的点。 如为 TopLeftCorner，图像的左上角保持固定。 如为 ImageCenter，图像的中心点保持固定。 如为 ControlCenter，控制器的中心保持固定。
View （只读）	Number[4]	以感兴趣区域的形式指定图像的显示部分。格式为：[X, Y, Breite, Höhe]。

函数	描述
centerImageToControl(): void	调整 Offset 和 Scale，使当前图像在控制器中居中。
fitImageToControl(center: boolean): void	调整 Offset 和 Scale，使当前图像在控制器中居中并填充控制器。
setView(view: number[]): void	更改 Offset 和 Scale，以便显示图像的指定部分。如果剪切的长宽比与控制器不匹配，则不一定要与视图匹配。

类别：工具栏

以下属性可用于自定义控制器的工具栏 [▶ 2613]。

属性	类型	描述
ToolbarVisible	Boolean	工具栏显示。
ToolbarItems	ToolbarItems	选择要在工具栏中显示的元素。
ScaleOptions	Number[]	除“Fit（适合）”选项外，还可在 ScaleSelection 组合框中显示和选择缩放值。
ToolbarHeight	Number (px)	工具栏的高度。
ImageSelectionWidth	Number (px)	ImageSelection 组合框的宽度。
ScaleSelectionWidth	Number (px)	ScaleSelection 组合框的宽度。
ScaleSelectionEditable	Boolean	如果激活，可将值手动输入 ScaleSelection。如果未启用，则只能从提供的 ScaleOptions 中进行选择。

类别：信息栏

以下属性可用于自定义控制器的信息栏 [▶ 2614]。

属性	类型	描述
InfobarVisible	Boolean	显示信息栏。
InfobarItems	InfobarItems	选择要在信息栏中显示的元素。
PixelColorFormat	Function	根据 PixelColor 属性的值为信息栏生成文本的函数。
ShapeValueFormat	Function	根据 ShapeValue 属性的值为信息栏生成文本的函数。
InfobarHeight	Number (px)	信息栏的高度。

类别：形状

以下属性有助于在图像上绘制形状 [▶ 2617]。

属性	类型	描述
ShapeSelectionActive	Boolean	如果激活，则可以在图像上绘制、显示和编辑形状（基于 ShapeType）。
ShapeType	ShapeType	形状类型。可能为： • 点状 • 线状 • 矩形/正方形 • 椭圆/圆 • 多边形
ShapeIsRotatable	Boolean	如果激活，可以使用用户界面中的手柄旋转合适的图形。（直线、矩形、正方形、椭圆圈、多边形）
ShapeData	ShapeData	形状的几何描述。可用于查询绘制形状的数据或显示特定的预定义形状。
ShapeAutoClear	Boolean	如果启用，ShapeData 将在 ShapeSelectionActive 被停用时重置为相应 ShapeType 的基准值。
ShapeStrokeThickness	Number (px)	显示形状线的粗细。
ShapeHandleSize	Number (px)	显示形状手柄的大小。
ShapeClickableSize	Number (px)	显示形状手柄可点击区域的大小。可用于简化在触控面板上编辑形状的过程，而不会显示不必要的大手柄。
ShapeAngleInterval	Number (°)	可以改变形状旋转的步长。例如，当值为 90 时，只有四种可能的排列方式。如为 0，任意旋转。
ShapeIsInImageArea （只读）	Boolean	表示显示形状是否完全位于图像区域内。可用于检查 ShapeData 对某些 PLC 函数的有效性。

函数	描述
clearShape(): void	删除显示的形状，同时将 ShapeData 重置为相应 ShapeType 的基准值。

类别：缩略图

以下属性决定缩略图的显示。缩略图 [▶ 2620]显示图像可见部分的概况。

属性	类型	描述
ThumbnailVisible	Boolean	显示缩略图。
ThumbnailPosition	ThumbnailPosition	缩略图的位置。
ThumbnailSize	Number (px)	缩略图的大小。确定缩略图的宽度；高度根据长宽比自动计算。

类别：叠加

以下属性允许在图像上叠加对象 [▶ 2621]（例如作为定位或标记辅助）。

属性	类型	描述
OverlayVisible	Boolean	显示叠加。
OverlayImage	String	叠加在显示图像上的图像。
OverlayElements	OverlayElements	在显示的图像上叠加了几何元素。

类别：信息

以下属性可用于获取有关图像或图像特定像素的信息 [▶ 2622]。

属性	类型	描述
PixelInfoUpdate	PixelInfoUpdate	决定何时更新 PixelColor 和 PixelPosition。 • 如为 Never，它们从未更新。 • 如为 AtClick，它们会在您点击图像的特定部分时更新。 • 如为 AtCursor，它们会在每次在图像上移动光标时更新。此操作会始终显示当前光标位置的值。
PixelColor （只读）	[Number, Number, Number, Number]	所选图像像素的颜色。
PixelPosition （只读）	[Number, Number]	所选图像像素在图像坐标中的位置。
ImageSize （只读）	[Number, Number]	显示图像的原始尺寸。
ImageDisplaySize （只读）	[Number, Number]	显示图像的显示尺寸。

事件

所有事件均列在“Vision”类别中。

事件	描述
.onPixelInfo	如果 PixelInfoUpdate <> Never 且用户点击图像中的某个位置或将鼠标指针移至该位置以选择某个像素，则会触发该事件。 在内部，事件会更新属性 PixelColor 和 PixelPosition 的值。
.onShapeConfirmed	当设置、清除或修改形状从而使 ShapeData 属性发生变化时，会触发该事件。
.onImageReceived	控制器接收到图像时会触发该事件。

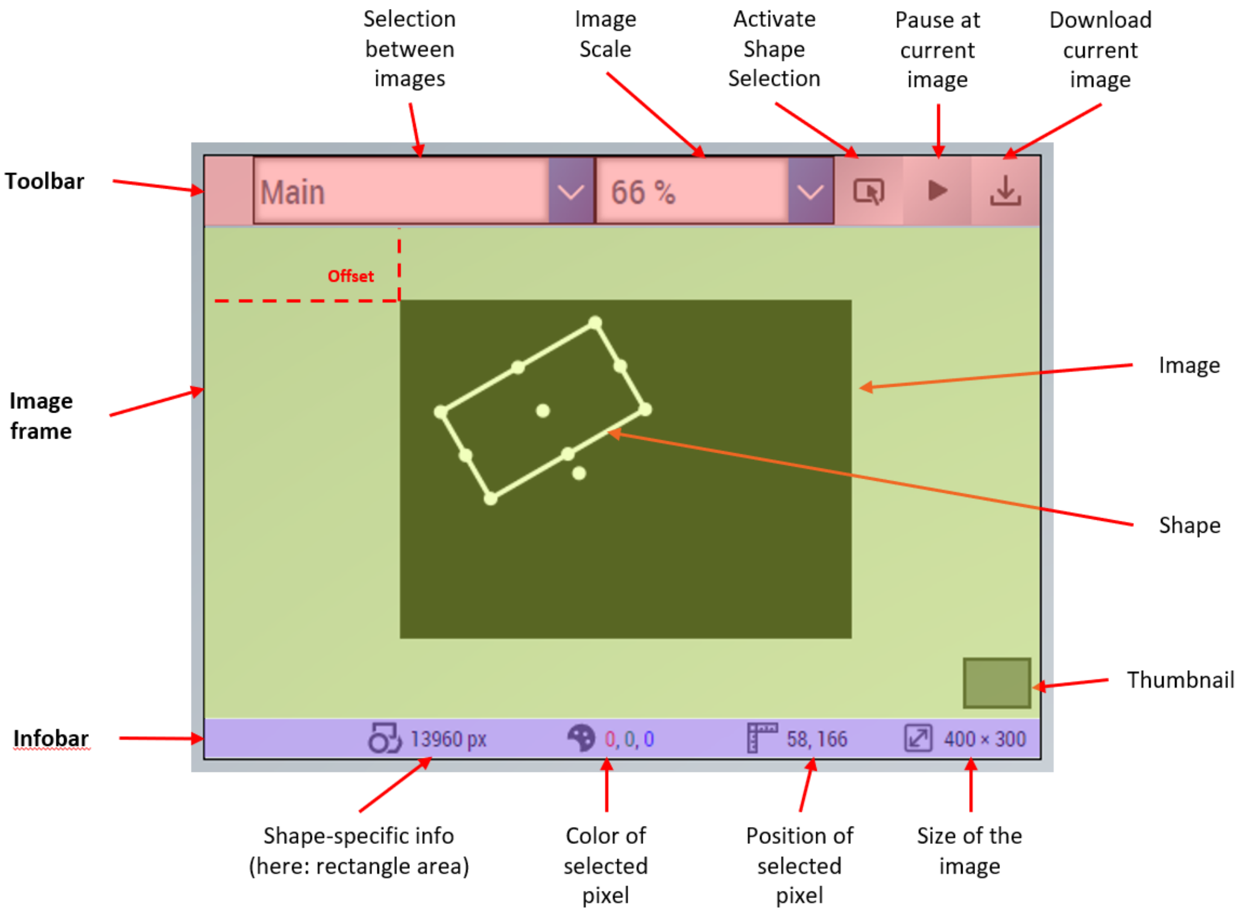
主题资源

如为主题设置，可以使用以下属性更改子控制器（工具栏中的组合框）的外观。关于各个属性的详细说明，请参见标准 HMI 文档。

属性	类型
ImageSelection_ContentPadding	Padding
ImageSelection_DataHeight	DataHeight
ImageSelection_DropDownBackgroundColor	Color
ImageSelection_DropDownFontFamily	FontFamily
ImageSelection_DropDownFontSize	MeasurementValue
ImageSelection_DropDownFontStyle	FontStyle
ImageSelection_DropDownFontWeight	FontWeight
ImageSelection_DropDownHorizontalAlignment	HorizontalAlignment
ImageSelection_DropDownTextColor	SolidColor
ImageSelection_DropDownVerticalAlignment	VerticalAlignment
ImageSelection_DropDownStyle	TcHmiCombobox.DropDownStyle
ImageSelection_FontFamily	FontFamily
ImageSelection_FontSize	Number (px)
ImageSelection_FontStyle	FontStyle
ImageSelection_FontWeight	FontWeight
ImageSelection_HorizontalAlignment	HorizontalAlignment
ImageSelection_MaxListHeight	Number (px)
ImageSelection_TextColor	SolidColor
ScaleSelection_ContentPadding	Padding
ScaleSelection_DataHeight	DataHeight
ScaleSelection_DropDownBackgroundColor	Color
ScaleSelection_DropDownFontFamily	FontFamily
ScaleSelection_DropDownFontSize	MeasurementValue
ScaleSelection_DropDownFontStyle	FontStyle
ScaleSelection_DropDownFontWeight	FontWeight
ScaleSelection_DropDownHorizontalAlignment	HorizontalAlignment
ScaleSelection_DropDownStyle	TcHmiCombobox.DropDownStyle
ScaleSelection_DropDownTextColor	SolidColor
ScaleSelection_DropDownVerticalAlignment	VerticalAlignment
ScaleSelection_FontFamily	FontFamily
ScaleSelection_FontSize	Number (px)
ScaleSelection_FontStyle	FontStyle
ScaleSelection_FontWeight	FontWeight
ScaleSelection_HorizontalAlignment	HorizontalAlignment
ScaleSelection_MaxListHeight	Number (px)
ScaleSelection_TextColor	SolidColor

7.2.1.1 控件结构

Image Watch Control 由显示工具和信息的区域（工具栏 [▶ 2613]和信息栏 [▶ 2614]）以及显示图像的主区域组成。在此区域内，可以通过编程、鼠标或触控来移动和缩放图像。缩略图或形状选择等其他特征可根据具体应用使用。



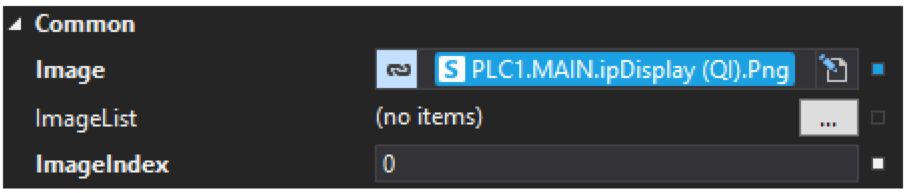
7.2.1.2 图片源

本页解释的属性和函数载于**通用**类别中。

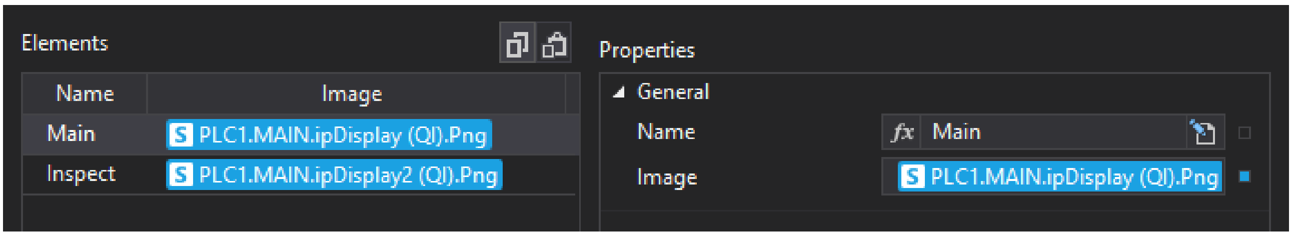
图像资源

图像资源定义将在图像查看器中显示的图像。预期用途是使用 HMI.Vision 扩展 [▶ 2599]将 ITcVnDisplayableImage 符号作为图像资源进行链接。不过，一般来说，图像资源只需要是一个有效的图像 URL。这可以是项目中的直接路径、外部服务器的 URL 或数据 URL。其作用类似于标准图像控制器的 Src 属性。

最简单的情况是，通过 Image 属性指定图像资源：



此外，还可以在 ImageList 属性中存储多个图像资源：



ImageIndex 属性可用于选择实际使用的资源。

例如，这可用于轻松切换来自 PLC 的各图像符号。也可以通过工具栏 [▶ 2613]中的ImageSelection组合框以图表形式进行切换。为此，在 ImageList 中不仅可以分配相应的图像资源，还可以分配相关图像名称。

同时使用图像和图像列表互不兼容

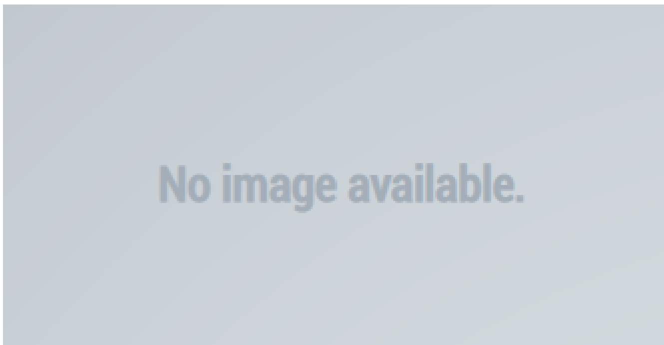
在内部，使用 ImageList 属性会将所选 ImageList 元素的相应图像符号绑定到 Image 符号。因此，使用 Image 和ImageList 互不兼容；如果已经使用 ImageList，则不应将图像符号与 Image 属性绑定。此外，符号必须在 ImageList 中链接；不可能将原子字符串指定为图像路径。

接收事件

每次控制器接收新图像时，都会触发事件 onImageReceived。

替代文本

如果未指定图像资源或无法从指定资源加载有效图像，则会显示替代文本。这可通过 Alt 属性进行更改，默认情况下为 "无可用图像"。



图片下载

当前显示的图像可通过浏览器特定下载本地保存到客户端设备上。下载可通过 downloadImage 函数或工具栏 [▶ 2613]上的下载按钮触发。

图像以原始大小保存。

文件名可在程序调用中作为参数指定。或者，根据链接符号和时间戳计算：<symbol name>_<date>_<time>-<MS>.png。

在控制器内无法选择保存路径，但它对应于所使用浏览器的下载路径设置。通常的下载路径是 C:\User\<User>\Downloads\。

图像文件格式由文件名中指定的文件扩展名决定。有效类型为：

- .png
- .jpg
- .bmp
- .tif

除非指定了文件名，否则该方法会尝试从链接符号的格式中推导出适当的文件类型。如果无法实现，则使用 PNG 格式。

如果显示的图像以不同的格式保存，则只会转换为相应的类型，而不会进行额外的压缩。因此，保存图像的文件大小取决于TcHmiVision服务器扩展名 [▶ 2599]中选择的符号图像格式。

图像下载要求和限制

由于图像下载通过浏览器特定功能完成，因此可能性、设置和功能取决于 TwinCAT HMI 客户端所使用的网络浏览器。

方法 downloadImage 需要对 TwinCAT HMI 用户管理进行授权**操作**。

这种方法不适用于作为数据流链接的图像，也不适用于 HMI 工程的实时视图。

7.2.1.3 图像视图

此处解释的属性和函数载于**视图**类别。

在内部，显示图像的大小和位置由属性 Offset、Scale 和ScaleReference 精确定义。不过，改变大小和位置的典型用例使用触摸控和鼠标手势等交互特征。因此，这些手势会自动在内部更改上述属性。

属性：Offset

Offset 属性指定图像相对于控制器本身的位置。左上角被定义为原点，X 轴向左，Y 轴向下。图像偏移也是从图像的左上角开始测量。偏移坐标以像素为单位指定。

如前所述，Offset 属性由控制器本身更改，例如，当用户使用触控或鼠标手势与图像交互的情况。

默认情况下，偏移属性与 TwoWay 绑定相关联。

属性：Scale **和** ScaleReference

属性 Scale 以百分比为单位指定缩放因子，用于计算图像在屏幕上显示的像素大小。根据属性 ScaleReference 的值，大小总是根据原始图像尺寸或控制器尺寸计算。

ScaleReference 可以是以下值之一：

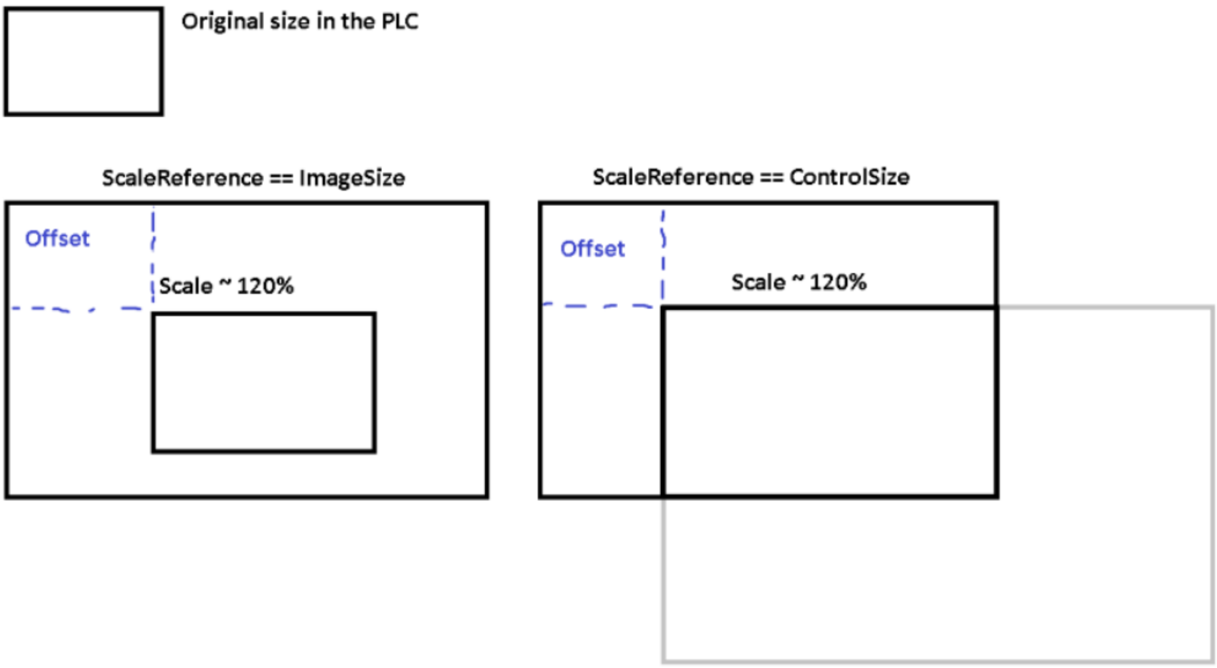
图像大小	显示图像的大小等于原始图像大小乘以属性 Scale 的值。 例如：图像原始大小为 300x200，属性 Scale 为 150%。最终显示图像大小为 450x300。
控制尺寸宽度	显示图像的大小等于控制器的宽度（或更精确地说，控制器内的图像框架）乘以 Scale 属性的值。
控制尺寸高度	显示图像的大小等于控制器的高度（更精确地说，控制器内图像框架的高度）乘以 Scale 属性的值。
控制尺寸	显示图像的大小等于控制器的大小（或更精确地说，控制器内的图像框架）乘以 Scale 属性的值。如果控制器的长宽比与图像长宽比不匹配，则会使用高度或宽度作为参考尺寸，具体取决于哪个尺寸相差更多。

ImageSize 设置的优势在于显示尺寸与图像原始尺寸之间的直接关系。PLC 中尺寸不同的两个图像在 HMI 中总是有不同的尺寸。ControlSize 设置的优势在于根据控制器自动调整图像。无论图像在 PLC 中的实际多大如何，其大小始终是一样的。

长宽比

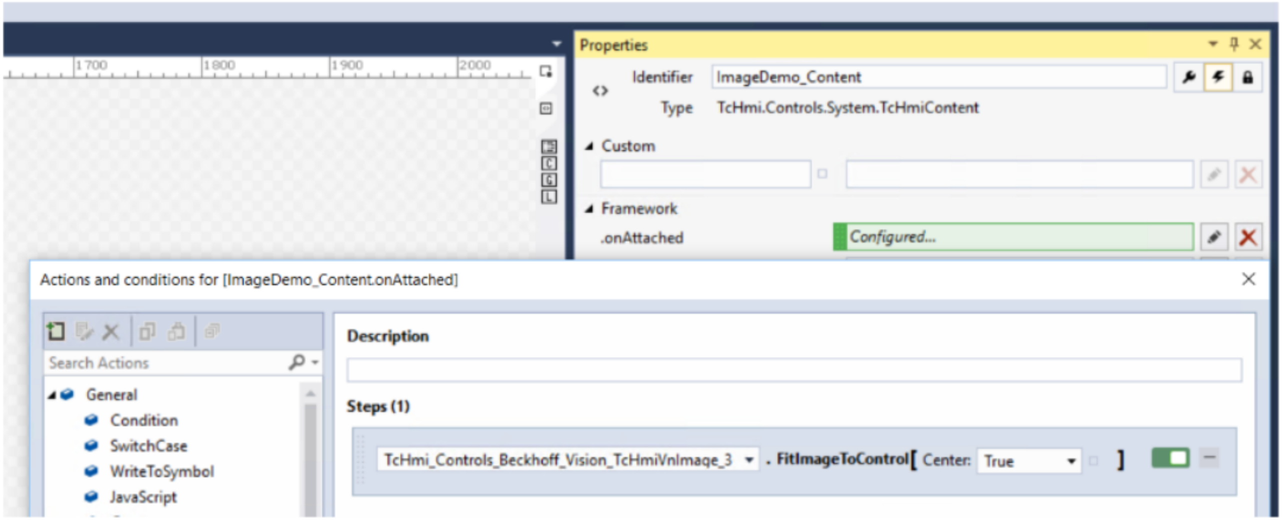
始终计算图像的高度和宽度以保持原始图像的长宽比。

示例：Offset、Scale **和** ScaleReference



函数：FitImageToControl

要在图像查看器中将当前显示图像最大化并居中，可以调用 FitImageToControl 函数。如果要从一开始就这样做，则应将该函数与 .onAttached 事件相链接。一旦激活该函数，所有后续图像也将居中并最大化，直到视图在其他地方被修改，从而取消该模式。



属性：ScaleCenter

ScaleCenter 提供调整缩放行为的可能性。Scale 属性的单独变化始终会导致图像右下角发生变化，而左上角则保持不变。要改变这一点并固定其他点，可以使用属性 ScaleCenter。在这种情况下，属性 Offset 会自动进行调整。提供以下选项：

ScaleCenter 值	描述
TopLeftCorner	缩放时图像左上角保持居中。
ImageCenter	缩放时图像中心保持居中。
ControlCenter	缩放时控制器中心保持居中。

手势

用户可以通过以下手势/交互来改变图像的大小和位置：

- 在图像框内任意位置按住鼠标左键，拖动鼠标移动图像，从而改变属性 Offset。属性 Scale 不受影响。

- 用一根手指触摸图像框，然后移动手指以移动图像，从而更改 Offset 属性。属性 Scale 不受影响。
- 滚动鼠标滚轮以改变图像大小，从而改变属性 Scale。缩放的中心将始终是鼠标指针的位置。因此，Offset 属性也会相应改变。滚动时，鼠标指针必须位于图像控制器上方。
- 在移动和挤压图框时，用两根手指触摸图框即可改变图片的位置和大小。Offset 和 Scale 这两个属性进行了调整。

为防止这些用户交互，可将 Draggable 和 Scalable 这两个属性设置为 false，以防止用户引起的位置或尺改变。始终有可能更改值（例如写入符号），这不受两个属性的影响。

属性：限制

控制定位和缩放系统应用限制规则，以防止用户创建混乱或失向图像视图。

ScaleMin 和 ScaleMax 属性表示对 Scale 属性设置的用户交互限制。请注意，这些限制只影响用户交互。从控制器外部或通过工具栏中的 ScaleSelection 组合框（请参见工具栏 [▶ 2613]类别中的 ScaleSelectionVisible 属性）设置 Scale 属性不受这些限制的限制。

此外，控制器会自动将图像限制在具有一定外填充（目前为 10px）的边界内。该边界以外的位置可以直接通过缩放和偏移属性进行配置，但不能通过用户交互（如拖动或鼠标滚轮）进行配置。

只读：View

Offset 和 Scale 属性有利于图像视图的内部处理，但对用户来说不一定直观（例如定义图像区域）。人们通常倾向于从感兴趣区域或类似表征的角度来思考问题。因此，在改变图像的大小或位置后，会计算并通过只读属性**视图**提供图像的当前可见区域。

内部计算总是将坐标原点设置为图像的左上角，并由此计算左上角位置以及图像框的宽度和高度。

View 属性以 [X, Y, Breite, Höhe] 形式的数组提供。

要使用视图描述设置位置和大小，请参见下一节中的方法 SetView。

方法：SetView

SetView 方法允许用户显示图像的特定区域。调用此方法时，由参数指定的图像区域将被最大化，并在图像框内居中。

区域必须在原始图像坐标中指定为 [x, y, Breite, Höhe]。

> **不同视图**
>
> 根据指定区域和图像框的长宽比，请求的视图可能与实际视图不同。这反过来又可以通过只读属性 View 进行检索。

> **单独行动**
>
> 视图功能不是通过属性访问，而是通过一个只读属性和一个 setter 方法访问。其中的原因是上面提到的有序视图和实际视图之间的区别，而且在实现哪个视图的问题上，**视图**属性会与 Offset 和 Scale 属性相冲突。由于视图 setter 是一次性操作，而不是绑定属性，因此无法做到这一点。

停止图像流

如果属性 ImageFreeze 为真，即使收到新图像，当前显示图像也不会改变。预期用例是冻结当前图像，以供进一步审查，而无需在 PLC 中添加额外的逻辑。

如果属性 ImageFreeze 从true 变为 false，控制器会立即再次显示最后一个图像，并忽略中间图像。

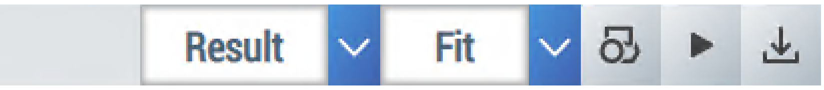
属性 ImageFreeze 可由工具栏中的 Freeze（冻结）按钮控制（请参见工具栏 [▶ 2613]类别中的 FreezeButtonVisible 属性）。

7.2.1.4 工具栏

关于此处解释的属性和功能，可查看**工具栏**类别。

工具栏中的控件可以用来交互式地执行各种其他情况下仅可通过属性访问的操作。每个元素均可显示或隐藏。此外，整个工具栏可以调整高度或完全隐藏。

工具栏中的元素位置固定不变，无法定制。如果需要调整，可以使用ToolbarItems **属性**隐藏相应的元素，并可通过自己的控件自行执行，同时将其放在图像控制工具栏上。可用的工具栏元素从左到右如下：



ImageSelection	用于从图像列表中选择待显示图像的组合框（见类别**通用的图像列表**）。仅在图像列表至少包含一个元素时才显示。在图像之间切换时，预计加载时间会很短。
ScaleSelection	用于选择特定缩放值的组合框，以将当前图像调整到控件的尺寸。可选择缩放值可通过额外的属性进行调整。 如果在设置用户定义的缩放时ScaleSelectionEditable停用，组合框将不再显示数值。这是组合框控件的默认行为。
ShapeButton	用于设置**ActivateShape**属性值的按钮。如果激活，将显示一个现有的形状，并可以进行编辑。如果没有显示形状，可以绘制新的形状。
FreezeButton	用于设置FreezeCurrent属性值的按钮。如果激活，当前显示的图像固定不变，且不会更新。
DownloadButton	通过默认名称下载当前图像的按钮（见方法downloadImage）。如需分配不同的文件名，该按钮可以由客户执行，然后通过自我定义的名称调用这个方法。

也可通过程序访问的功能

对于任何可以通过工具栏界面元素访问的操作，也可以通过控件的属性以程序方式访问。

大小设置

本工具栏为工具栏本身的高度（属性ToolbarHeight）和两个组合框的宽度（属性ImageSelectionWidth和ScaleSelectionWidth）提供了自定义选项。三个附加按钮的宽度根据工具栏的高度自动调整，因此为方形。

组合框下拉按钮的宽度无法调整，因为标准组合框控件中未提供此功能。

在内部，属性ToolbarHeight最大限制为 150 像素。

背景颜色

工具栏的背景颜色默认为半透明白色 "rgba(255, 255, 255, 127)"，且可以通过**颜色**类别中的BarColor属性自定义。

高级设置

更多图形设置（如组合框字体）可以通过主题资源进行更改。基本上，提供了组合框的属性。主题资源可以在<u>主题编辑器</u>的TcHmiVnImage >Themed Resources处进行自定义。然后这些设置将适用于图像浏览器控件的所有实例。

7.2.1.5 信息栏

关于此处解释的属性和功能，可查看**信息栏**类别。

信息栏中的显示元素可用于显示有关形状值、像素值或图像尺寸的各种信息。每个元素均可显示或隐藏。此外，整个信息栏可以调整高度或完全隐藏。

信息栏中的元素位置固定不变，无法定制。如果需要调整，可以使用**InfobarItems**属性隐藏相应的元素，并可通过自己的控件自行执行，同时将其放在图像控制信息栏上。可用的信息栏元素从左到右如下：

ShapeValue	当前显示形状的计算值。这个值通过绑定到ShapeValueFormat属性的功能计算。 标准格式化功能 [▶ 2636]根据所选形状类型返回一个特定的值： • 点的坐标 • 线的长度 • 方形、矩形、圆或椭圆的面积 • 多边形的点数 这个示例 [▶ 2615]描述了自定义格式化功能的实现和使用。
PixelColor	所选像素的颜色值。这个值通过绑定到PixelColorFormat属性的功能计算。 标准格式化功能 [▶ 2636]根据选定的颜色空间返回一个特定值： • 灰色 • RGB • HSV • HSL 这个示例 [▶ 2616]描述了自定义格式化功能的实现和使用。
PixelPosition	所选像素的位置。
ImageSize	当前显示的图像的原始尺寸。

大小设置

工具栏提供了工具栏高度的定制选项。

在内部，属性InfobarHeight最大限制为150 Pixel。

背景颜色

信息栏的背景颜色默认为半透明白色“rgba(255, 255, 255, 127)”，且可以使用**颜色**类别中的BarColor属性进行自定义。

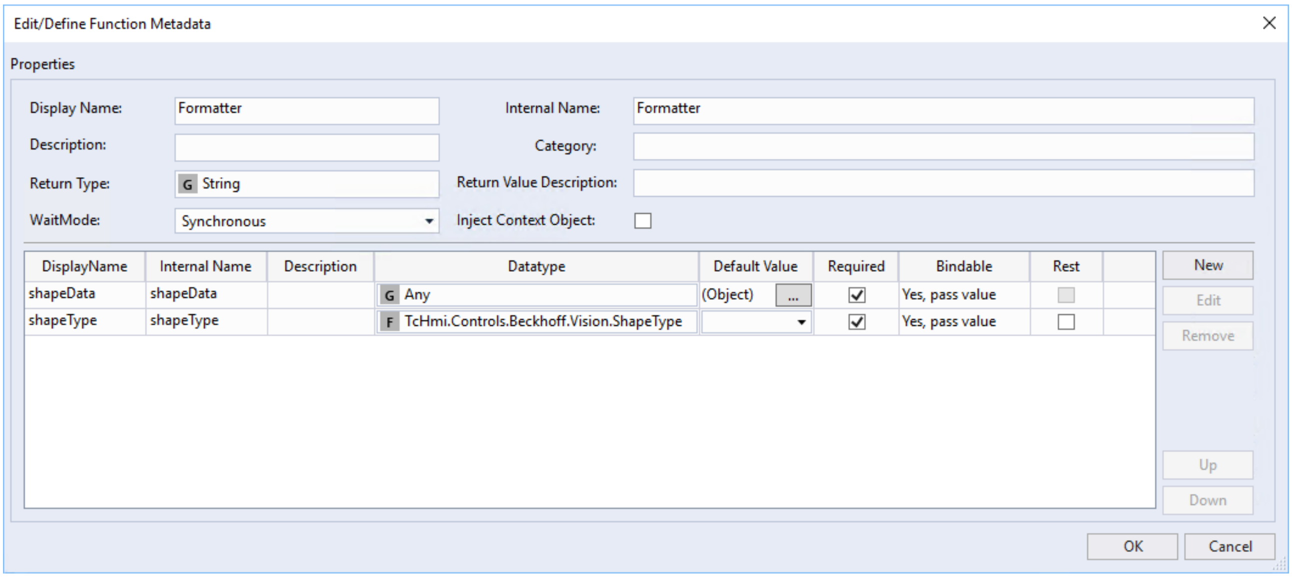
7.2.1.5.1 ShapeValueFormatting 函数

作为ShapeValueFormatter的替代方案，可以创建一个自定义格式化功能来显示信息栏中的其他形状信息。该功能接收完整的形状数据以及类型，并返回一个字符串。下面的例子显示了如何创建这样的功能以及它必须具备哪些接口。

关于功能的一般使用，可参见HMI 文件。

创建 TypeScript 函数后，必须创建两个参数。第一个参数可以传递的形状描述，类似 ShapeData 属性，因为它可以从控件中检索到。该参数必须创建为类型Any。第二个参数传递形状类型，类似于 ShapeType 属性。该参数必须被指定为类型TcHmi.Controls.Beckhoff.Vision.ShapeType。应该注意的是，由于时间原因，这些传递的参数可能与功能执行时相应控制属性的值不同。

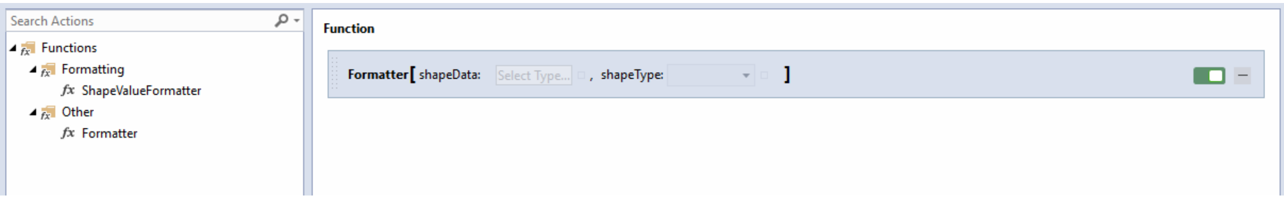
该功能必须返回一个类型为String的值，然后显示在信息栏中。每当形状描述发生变化，从而需要更新信息栏中的字符串时，就会调用该功能。



在下面的代码示例中，如果对象被包含在形状数据中，就会返回宽度和高度：

```
module TcHmi {
    export module Functions {
        export module TcHmiProject {
            export function Formatter(shapeData: any, shapeType: any) {
                if (shapeData.stSize != null && shapeData.stSize.fWidth != null &&
shapeData.stSize.fHeight != null) {
                    return 'w: ' + shapeData.stSize.fWidth.toFixed(0) + ' h: ' +
shapeData.stSize.fHeight.toFixed(0);
                }
                else {
                    return '';
                }
            }
        }
        registerFunctionEx(‘Formatter’, ‘TcHmi.Functions.TcHmiProject’, TcHmiProject.Formatter);
    }
}
```

如需使用创建的格式化功能，请点击ShapeValueFormat属性处的选择按钮，并在功能编辑器中选择刚刚创建的功能。



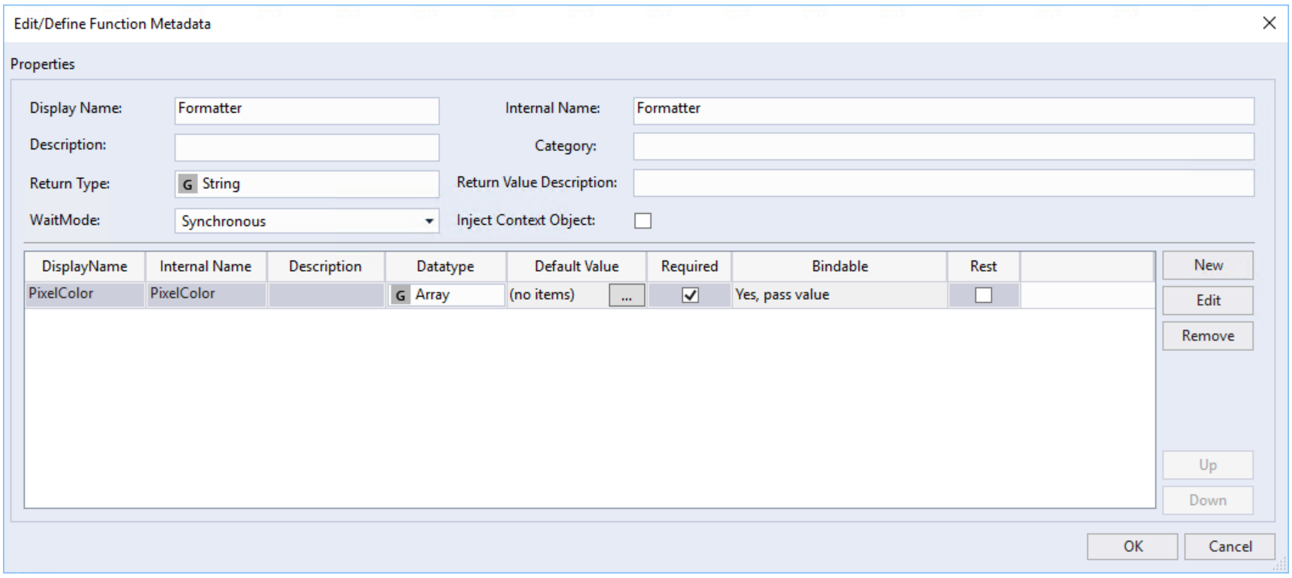
7.2.1.5.2 PixelColorFormatting 函数

作为PixelColorFormatter功能的替代方案，可以创建一个用户定义的格式化功能，以不同的方式格式化所选像素的颜色。该功能获取像素的颜色值，并返回一个字符串，显示在信息栏中。下面的例子显示了如何创建这样的功能以及它必须具备哪些接口。

> 关于功能的一般使用，可参见HMI 文件。

创建 TypeScript 函数后，必须创建一个参数。该参数类似于控件的 PixelColor 属性，以数组的形式传递颜色值，且因此必须以Array类型创建。应该注意的是，由于时间上的原因，传递参数可能与功能执行时相应控制属性的值不同。

该功能必须返回一个类型为String的值，然后显示在信息栏中。每当PixelColor属性的值发生变化，从而必须更新信息栏中的字符串时，就会调用该功能。

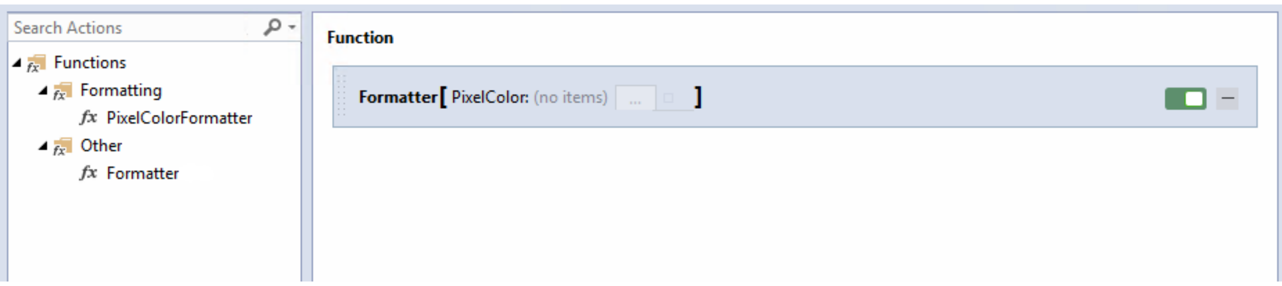


在下面的代码示例中，只要 PixelColor 中至少有 3 个元素，就会返回红、绿、蓝三种颜色的值：

```
module TcHmi {
    export module Functions {
        export module TcHmiProject {
            export function Formatter(PixelColor: number[]) {
                if (PixelColor.length >= 3) {
                    return `<span style="color:red;">${PixelColor[0]}</span>, ` +
                        `<span style="color:green;">${PixelColor[1]}</span>, ` +
                        `<span style="color:blue;">${PixelColor[2]}</span>`;
                }
                return '';
            }
        }
        registerFunctionEx('Formatter', 'TcHmi.Functions.TcHmiProject', TcHmiProject.Formatter);
    }
}
```

另外，只有 1 个通道可以用 color:black 返回，作为灰度值显示。为了转换为 16 位数值，各个 PixelColor 元素必须乘以 256。

如需使用创建的格式化功能，请点击PixelColorFormat属性处的选择按钮，并在功能编辑器中选择刚刚创建的功能。



7.2.1.6 形状

本页解释的属性和功能可以在**形状**类别中查看。

常见用例是用户选择某些图像区域。这可以是一个感兴趣区域或一个物体的位置。另一个相关的用例是用户定义的对图像内的距离或区域的测量。

为了适应所有这些用例，**形状**类别中的属性提供了灵活的功能，用于在图像上绘制/编辑各种形状并检索它们各自的几何描述。

属性：ShapeSelectionActive

形状选择可以通过属性ShapeSelectionActive或通过工具栏上**形状**按钮激活（见类别工具栏 [▶ 2613]）。只要这个属性处于激活状态，就可以通过点击图像表面绘制形状，并通过拖放角点或形状本身进行编辑。如果这个属性被设置为false，且ShapeAutoClear属性处于活动状态，那么当前的形状将被重置并丢失，因此必须先提取任何需要的数据。

绘制形状的几何描述可以通过属性ShapeData检索。根据属性ShapeType（见下文），形状数据有不同的格式。你可以期待交互式控件本身使用的格式完全相同（例如TcHmiVnPolygon和TcHmiVnRectangle），因为符号直接与内部使用的形状控件绑定。应该注意的是，这些形状不限于图像区域，且因此也可以采取负值。这可以用功能shapeIsInImageArea 查询。

属性：ShapeType 和 ShapeData

根据两种不同的数据类型，提供不同类型的形状以供选择。通过使用ShapeType属性，可选择以下形状之一：

形状类型	形状数据	PLC 声明
点	[[x, y]]	ARRAY[0..0] OF TcVnPoint2_REAL 或 ARRAY[0..0] OF TcVnPoint2_LREAL
线	[[x1, y1], [x2, y2]]	ARRAY[0..1] OF TcVnPoint2_REAL 或 ARRAY[0..1] OF TcVnPoint2_LREAL

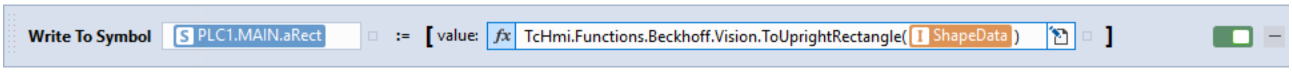
方形，
矩形，
圆形，
椭圆
	{
	fAngle：角度，
	aCenter：[cx, cy]，
	stSize: {
	fWidth：宽度，
	fHeight：高度，
	},
	}
		TcVnRotatedRectangle

多边形	[[x1, y1], [x2, y2], ..., ..., [xn, yn]]	ARRAY[0.n] OF TcVnPoint2_REAL 或 ARRAY[0.n] OF TcVnPoint2_LREAL

ShapeData 中的数值指的是原始图像的图像坐标。

> **i** 由于根据定义形状类型方形和圆形始终具有相同的高度和宽度，所以如果长宽比不一样，在点击第二个位置时它们不会准确地画到这个端点。在这种情况下，可以通过移动相应的角来达到理想位置。

如需将类型TcVnRotatedRectangle [▶ 2638]的形状类型方形和矩形的数据直接传递给类型为TcVnRectangle_UDINT [▶ 2638]的 PLC 变量并将其作为无旋转角度的直立边界矩形使用，可使用转换函数ToUprightRectangle [▶ 2637]。对于相反的情况，可使用ToRotatedRectangle [▶ 2636]函数。



在程序上，可以通过 Java 脚本改变ShapeData属性，如下所示：

```
let control = TcHmi.Controls.get("TcHmiVnImage")
// Polygon with 4 Points
control.setShapeData([[10, 10], [10, 100], [150, 150], [100, 10]])

// Rectangle
control.setShapeData({"aCenter":[200,200],"fAngle":15,"stSize":{"fWidth":150,"fHeight":150}})
```

在第一个例子中，写了一个有 4 个点的数组的多边形。如需写入一个点，只需要第一个元素，而要写一条线，则需要前两个元素。在第二个例子中，写的是一个矩形。由于形状类型方形、圆形和椭圆是基于相同的数据类型，所以这些类型也可以使用相同的符号。

数据也可以通过 WriteToSymbol 动作写入ShapeData。为此，复制setShapeData()功能的圆括号内的部分，并将其作为对象值粘贴。

属性：形状自动清除

如果ShapeAutoClear属性处于活动状态，并且在ShapeSelectionActive为true时改变了ShapeType，那么当前的形状就会被重置，并且可以绘制一个选定形状类型的新形状。如果ShapeAutoClear未处于启用状态，则在切换ShapeType时，数据会被保留，且仅在点击或重新绘制形状时才会更新。在同一基础控件的形状类型之间切换时，会直接显示新的形状，否则什么也不显示。点、线和多边形属于一组，方形、矩形、圆和椭圆属于另一组。

属性：外观

形状和控制点的外观可以通过调整颜色和轮廓的大小/厚度来改变。

- 通过ShapeStrokeThickness，可以设置线条的厚度，单位：像素。
- 通过ShapeHandleSize，可以设置显示形状的控制点的大小，单位：像素
- 通过ShapeClickableSize ，可以设置控制点可点击区域的大小，单位：像素。这在触摸操作中的好处在于，操作区域可以大于形状的显示点。
- 通过ShapeStrokeColor和ShapeHandleColor，线条的颜色和控制点的颜色可以独立设置。

属性：可旋转

如果ShapeIsRotatable处于启用状态，对于合适的形状（直线、矩形、方形、椭圆、多边形）会显示一个额外的控键，通过控键可以旋转形状。通过ShapeAngleInterval属性，可以设置改变形状旋转的步长，单位：度。在默认值为 0° 的情况下，形状可以根据需要进行旋转。例如，如果该形状只能放置在 4 个位置，可以输入 90° 为数值。

方法：clearShape()

通过调用方法clearShape，可以编程方式删除绘制的形状。

显示元素

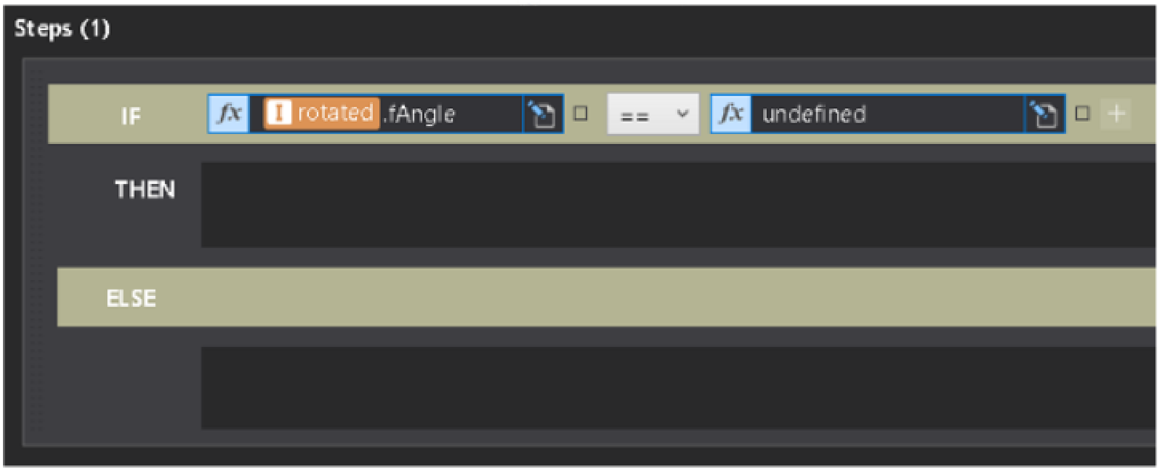
为了支持在图像中测量距离或面积的用例，**信息栏**（见类别信息栏 [▶ 2614]）提供了 UI 元素，来显示代表当前绘制形状的文本。默认情况下，“点”形状类型显示坐标，“线”显示线长，“多边形”显示点的数量，“方形”、“矩形”、“圆形”和“椭圆”形状类型显示面积。

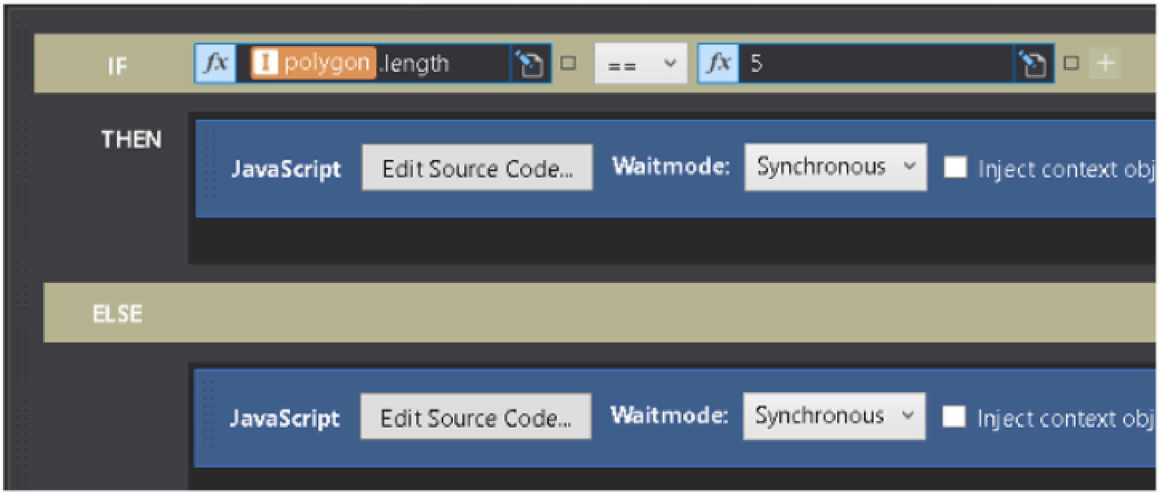
查询形状类型和数据值

根据选择的形状类型，ShapeData中的数据可以有不同的格式。如需确定哪种格式存在，可以查询当前ShapeType，或者也可以通过数据值确定，例如 fAngle 属性是否已定义。如果是这种情况，就会出现一个TcVnRotatedRectangle 类型，否则就是一个点的数组。如需查询数组中包含多少个点，可以查询其长度。然后可以用这个数字来确定是否为一个点、一条线或一个有多少个点的多边形。

此外，还可以通过查询各个数值来检查是否存在有效数据。在删除或重置形状数据时，各个数值会被写入 0，这代表无效数值。

这些查询可以用来确保各种形状类型都被写入指定的相同类型的 PLC 变量中，并且只有有效的数据可以被传输到控制程序中。





属性： shapeIsInImageArea

通常需要使形状完全在图像内，例如以便能够设置感兴趣区域。通过 ReadOnly 属性shapeIsInImageArea可以对此进行查询。仅在当前有效ShapeData可用且该形状所有顶点都在图像区域内时，才会返回true。为了能够区分不存在的形状和不在图像中的形状，你还应该查询ShapeSelectionActive和有效ShapeData。

形状的高级使用

形状功能通过集成到图像查看控件中的单独控件实现。这些形状控件有一些 ImageWatch 控件不能直接使用的属性。其中包括矩形的精确长宽比。这些属性可以通过编程方式或单独使用形状控件访问。

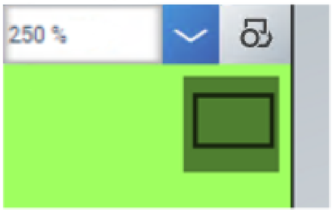
7.2.1.7 缩略图

本页解释的属性和函数可以在**缩略图**类别中查看。

缩略图指示图像的哪个部分在控件中可见。它与图像的**视图**密切相关（见类别**视图**中的只读属性**视图**），但在某些方面有区别。

缩略图是否可见是由ThumbnailVisbile属性决定。它的一般位置和大小是由属性ThumbnailPosition和ThumbnailSize决定，且总是方形。如果图像的长宽比不是 1:1，缩略图的大小就是图像的全宽，而高度则根据长宽比调整。

当整个图像框架被图像覆盖时，缩略图的显示与只读属性View几乎相同。否则，它的不同之处在于，图像以外的区域被包括在View内，但不包括在缩略图中。



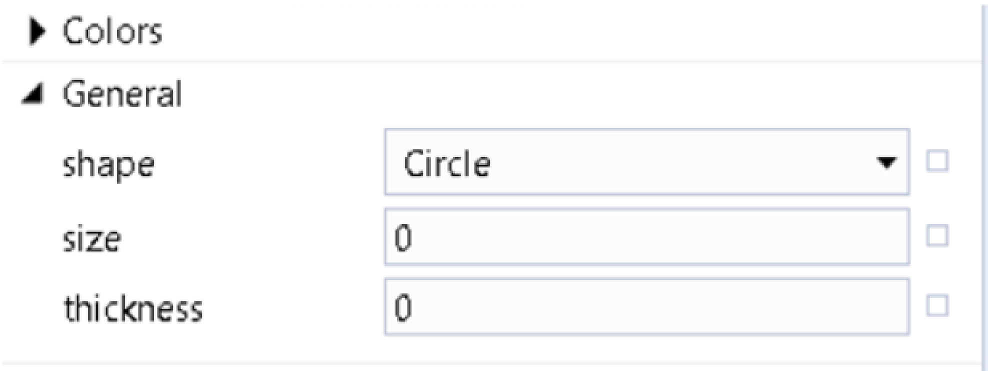
7.2.1.8 覆盖物

本页解释的属性和函数载于**叠加**类别。

例如，叠加可用于帮助设置相机或定位图像中的对象。

预定义几何形状

控制器提供可以在图像中心显示的预定义几何对象（如圆圈、矩形、十字形、框架）。例如，这些形状可用作十字或图像中特定位置的框架和标记。还可以同时显示多个对象（例如一个圆和一个十字光标）。这通过数组属性 OverlayElements 实现。对象配置如下所示：



对象在图像中始终居中。它们具有由**尺寸**定义的固定比例，不会随图像缩放。

叠加图像

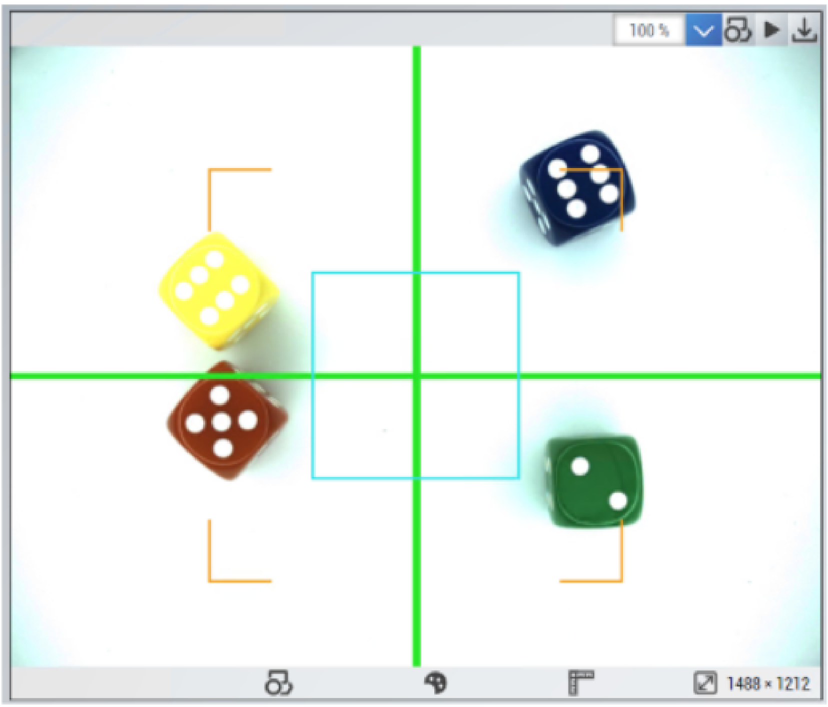
如果这种功能对于特定应用来说不够灵活，可以使用 OverlaySource 属性指定叠加图像。显示的位置和大小与主图像完全相同，也就是说，叠加图像会被拉伸/缩放以适应主图像。因此，我们建议叠加图像的长宽比至少与主图像相同，大小最好也与主图像相同，并且除相关部分外均透明。

例如，您可以在透明图像上绘制一个带有附加附件的用户定义十字光标，并将其用作叠加。要创建透明图像，必须在背景颜色的第四个通道处创建颜色值为 0 的 4 通道图像。然后，要画入的光标线在第四通道处获得颜色值 255，这样这些元素就可以覆盖背景图像。

视图

例如，叠加了矩形和文本的图像可能会产生以下结果：

在下方示例中，来自 PLC 的叠加图像（绿线）与控制器中定义的两个对象相结合：



7.2.1.9 信息

本页解释的属性和功能可以在**信息**类别中查看。

通过以下属性，可查询所显示图像或图像中特定像素的信息。

ImageSize	显示图像的原始尺寸。这与 PLC 中的图像尺寸相对应。
ImageDisplaySize	显示的图像尺寸，单位：像素。
PixelColor	特定选定像素的颜色（详见下文）。颜色以四个元素组成的数组提供，其中元素代表颜色通道 [R、G、B、A]。
PixelPosition	特定选定像素的位置（详见下文）。位置以包括两个元素[X, Y]的数组形式指定，其中元素为原始图像坐标，而不是显示坐标。

PixelColor和PixelPosition针对特定像素计算。这个像素的选择通过点击、触摸或光标在显示图像上的位置完成。

属性PixelInfoUpdate决定是否进行计算或由哪个事件触发。

从不	PixelColor和PixelPosition完全不更新。
点击时	PixelColor和PixelPosition在用户点击或触摸特定像素时更新。当用户点击或触摸当前图像外图像框架上的一个点时，这些值会被重置。
光标处	当用户在图像上移动鼠标指针时，PixelColor和PixelPosition会持续更新。当鼠标指针不在图像上时，这些值会被重置。

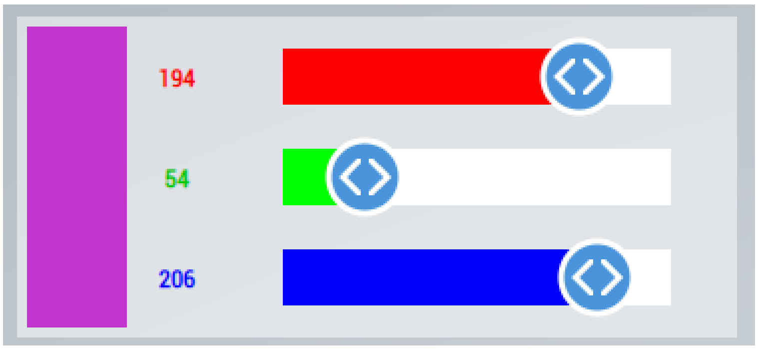
每当像素的颜色和位置更新时，事件onPixelInfo被触发。

显示

PixelColor、PixelPosition 和 ImageSize 的当前信息可以显示在信息栏中（见类别**信息栏**中的属性 InfobarElements）。

7.2.2 颜色

“颜色控制”显示颜色，并在必要时对颜色进行调整。它包含各种表示颜色值的显示元素，根据配置的不同，其也可用于设置颜色值。有关组件和结构的概述，请参见 显示元素 [▶ 2624] 分章。每个配置选项的说明可在本页的以下类别中找到。每个类别都有分章，提供更多信息、细节和应用示例。



对于以下所有属性，控制器有一个 getter 和一个 setter 方法。例如，属性 Orientation 存在 getOrientation 和setOrientation 两个方法。属性的使用将在链接分章中进行更详细的说明。

基类**控制器**所提供的所有其他属性、事件和权限均载于 TwinCAT HMI 文档。

类别：通用

以下属性表示控制器颜色值的数据接口 [▶ 2627]。它一方面用于设置控制器的显示颜色，另一方面用于检索用户输入的颜色值。

属性	类型	描述
ColorSymbol	颜色符号 [▶ 2637]	包含当前颜色值的 HMI 符号。
ColorValue （只读）	颜色值 [▶ 2637]	颜色值。这可以在没有符号被链接为 ColorSymbol 的情况下检索。

函数	描述
setColorValue(colorValue: 颜色值 [▶ 2637]): void	设置颜色值。可以在没有符号被链接为 ColorSymbol 的情况下调用。

类别：外观

以下属性决定了控制器的外观 [▶ 2624]以及对不同颜色格式 [▶ 2629]的处理。

属性	类型	描述
ColorChannels	颜色通道 [▶ 2637]	定义颜色的格式。这包括： • 通道数量 • 通道的最小最大值 • 整数或浮点数 • 显示颜色通道
Orientation	定向 [▶ 2638]	控制器的定向；类似于 LinearGauge 控制器的定向。“水平”或“垂直”。
BoxUsage	使用模式 [▶ 2638]	确定是否显示颜色框以及其是否可以编辑。
ValuesUsage	使用模式 [▶ 2638]	确定是否显示单个颜色值以及是否可对它们进行编辑。
SliderUsage	使用模式 [▶ 2638]	确定是否以滑块的形式显示单个颜色值以及是否可对它们进行编辑。
BoxColorConversion	Function	函数可将相应的颜色值转换为可由浏览器解释的字符串，反之亦然。如果每个颜色通道与默认的 RGB 格式不同，则需要告诉控制器每个颜色通道的含义。
BoxSize	Number (px)	根据定向（定向值）确定色框的宽度或高度。
ValuesSize	Number (px)	根据定向（定向值）确定显示颜色值的宽度或高度。

主题资源

如为主题设置，可以使用以下属性来更改子控制器（颜色值的文本框）的外观。关于各个属性的详细说明，请参见标准 HMI 文档。

属性	类型
TextFontFamily	FontFamily
TextFontSize	MeasurementValue
TextFontSizeUnit	MeasurementValue
TextFontStyle	FontStyle
TextFontWeight	FontWeight
TextPadding	Padding

7.2.2.1 显示元素

元素

颜色控制器中可以显示以下元素。在每种情况下，相应的使用属性（BoxUsage, ValuesUsage und SliderUsage）决定元素是隐藏（Hide）还是显示（ViewOnly）以及用户是否可以借助元素（Editable）改变颜色。

名称	位置	描述
颜色框	左侧或顶部	这是一个显示当前颜色的矩形区域。如果 BoxUsage 属性设置为“可编辑”，则点击 ColorBox 会打开用于选择颜色的浏览器特定对话框。根据浏览器的不同，这可能看起来像一个单独的窗口或叠加，也可能什么都不显示。 由于每个浏览器都控制着对话框本身的外观和功能，因此我们对此没有任何影响。 例如，颜色吸管会对屏幕进行截图，因此它无法处理不断变化的图像。更多信息载于 不同浏览器中的颜色框 [▸ 2625] 章节。
颜色值	中心	该区域包括用于查看和编辑控制器中存储颜色值的文本字段。值元素的数量及其文本颜色取自属性 ColorChannels。
颜色滑块	右侧或底部	该区域包括与 ColorValues 相对应的线性滑块。

排列

显示元素的排列取决于 Orientation 属性。“水平”排列的元素彼此相邻，“垂直”排列的元素彼此相叠。根据定向的不同，属性 BoxSize 和ValuesSize 定义相应元素的宽度或高度。滑块的宽度或高度会自动与可用空间的其余部分相对应。

注意
小尺寸 如果所选定向上的控制器非常小，则可能无法完整且正确地显示所有元素。这主要影响滑块。超过一定尺寸时，JavaScript 控制台中会显示警告，说明尺寸可能会影响控制器的功能。

数量

显示颜色值和滑块的数量始终与设定颜色格式 [▸ 2629]的通道数量对应。颜色格式通过 ColorChannels 属性设置。

文本

可以使用主题编辑器中的主题资源通过各种字体属性自定义颜色值的文本框。

7.2.2.1.1 不同浏览器中的颜色框

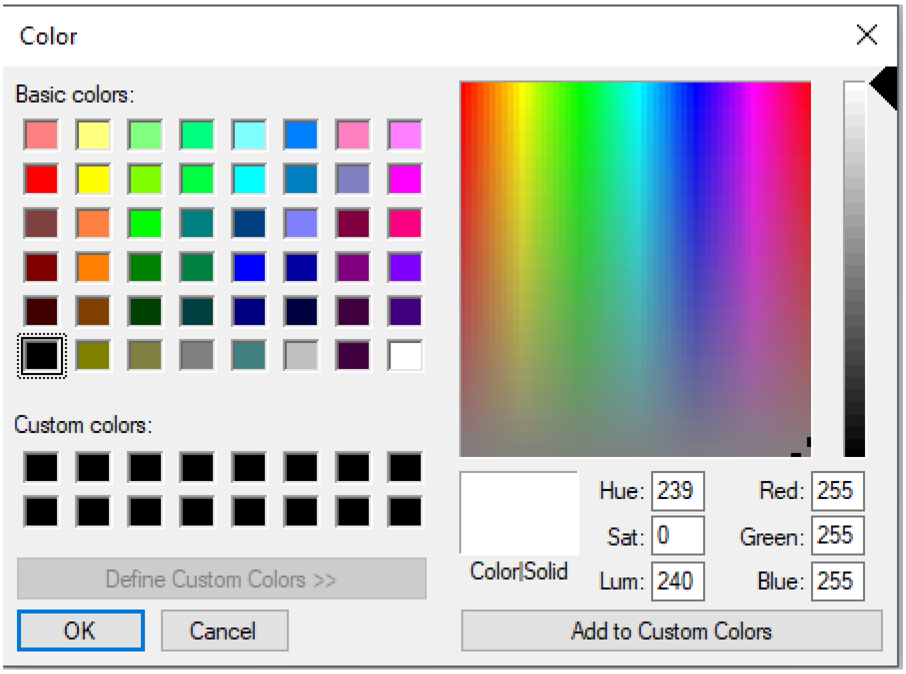
如果 BoxUsage 属性设置为 Editable，则可以通过点击颜色框打开另一个颜色选择对话框。此对话框取决于浏览器。以下是一些最常用浏览器的对话框：

Google Chrome 和 Microsoft Edge

吸管

基于 Chromium 的浏览器中的吸管功能会在选择工具时对浏览器窗口进行截图。然后，所选颜色就来自该截图。这意味着当图像发生变化时，所选颜色将不是点击时显示的颜色。因此，建议在使用吸管功能前通过 ImageFreeze 属性停止滚动。

Mozilla Firefox



7.2.2.2 颜色数据

ColorSymbol属性可以绑定不同符号。为了避免输出的颜色值与符号的数据类型不一致，颜色控件会读取符号的类型，并试图自动推导出正确的返回类型。

例如，如果需要显示三个颜色通道，但在 PLC 中链接了一个类型为TcVnVector4_LREAL的阵列，这些类型就不匹配。颜色控件会自动识别这一点，并将一个有四个值而不是三个值的数组写入符号中，这样就不会发生错误。其余通道部分留空。

只读属性ColorValue用于使颜色能够以编程方式访问，而无需将符号绑定到属性ColorSymbol。除此以外，没有任何区别。

与属性ColorValue相类似，方法SetColorValue提供了一种设置该控件颜色值的方法，而不需要将符号与属性ColorSymbol链接。

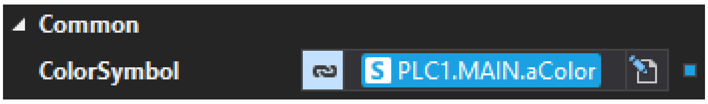
在下面的示例中解释了访问颜色值的两种方式：

- 与 PLC 连接 [▶ 2627]
- 编程使用 [▶ 2627]

这个示例转换为 HMI 颜色对象 [▶ 2628]显示了颜色数据如何也可以用于其他使用情况。

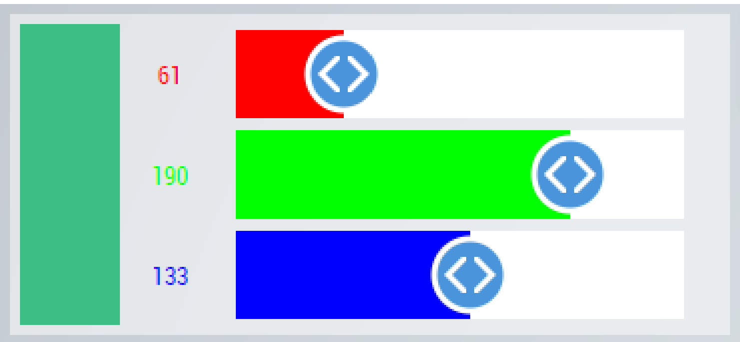
7.2.2.2.1 与 PLC 连接

在最简单的情况下，颜色控制可用来表示来自 PLC 的颜色变量。例如，为此可将 TcVnVector4_LREAL 类型的 PLC 变量与 ColorSymbol 属性相链接：



现在，PLC 变量和控制器彼此同步：

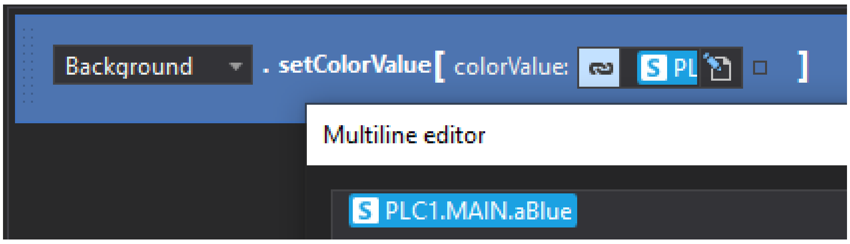
aColor	TcVnVector4_LREAL	
aColor[0]	LREAL	61
aColor[1]	LREAL	190
aColor[2]	LREAL	133
aColor[3]	LREAL	0



PLC 数组的第四个元素会被忽略，因为默认情况下只有三个通道在控制器中显示为 RGB。这可以通过颜色格式 [▶ 2629]进行设置。

7.2.2.2.2 编程使用

如果不将颜色控制的颜色值直接链接到符号，则可以通过编程方式访问。在本示例中，我们考虑使用 ID 为“Background”的颜色控制。要在控制器中设置颜色，可通过 JavaScript 代码或 Actions and conditions （操作和条件）对话框使用 setColorValue 函数：




```
TcHmi.Controls.get("Background").setColorValue([0, 0, 255])
```

要查询控制器的颜色，请使用只读属性 ColorValue：

在 JavaScript 中，必须使用 getColorValue 方法来检索属性：

```
const color = TcHmi.Controls.get("Background").getColorValue()
```

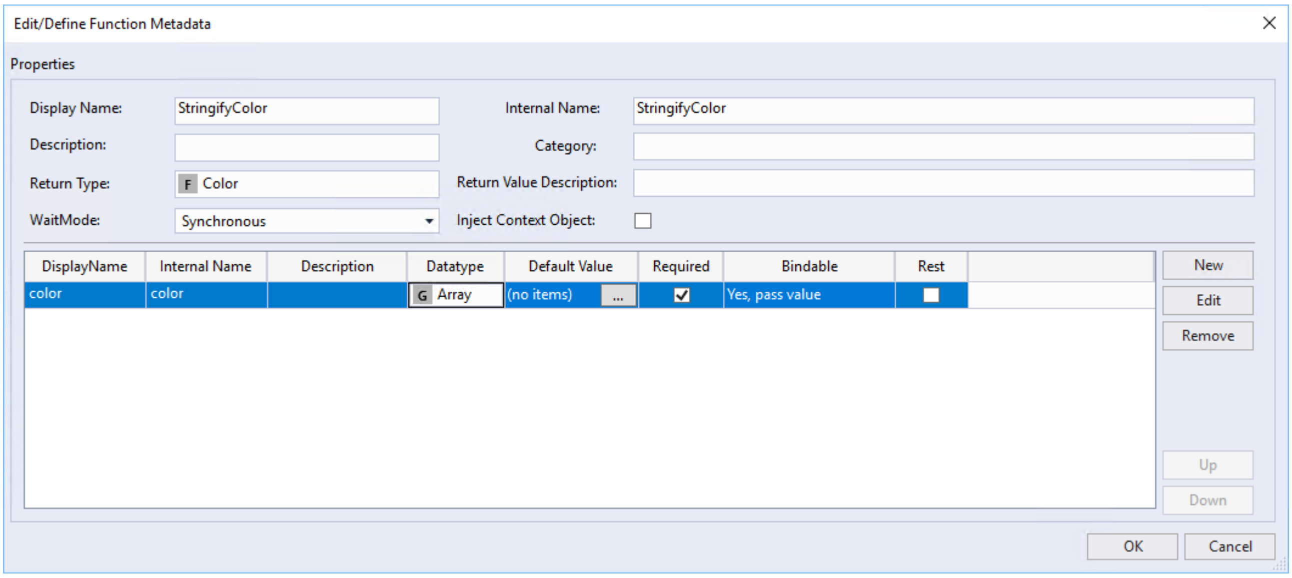
7.2.2.2.3 转换为 HMI 颜色对象

要在 TwinCAT HMI 中使用一种颜色，例如作为背景或用于文本或框架等其他属性，必须将其转换为 TcHmi.Color 类型的对象。下方示例展示了如何创建这样一个函数，以及它必须具备哪些接口。

各函数的一般使用方法在 HMI 文档中有说明。

创建 TypeScript 函数后，必须创建一个参数。例如，这会从图像控制的 PixelColor 属性或颜色控制的 ColorValue 中获取颜色值，因此必须创建 Array 类型。需要注意的是，由于时序原因，传递的参数可能与函数执行时相应控制属性的值不同。

函数必须返回一个 Color 类型的值，该值然后可用于 HMI 中的其他属性。



以下代码示例显示 RGB 颜色值的转换：

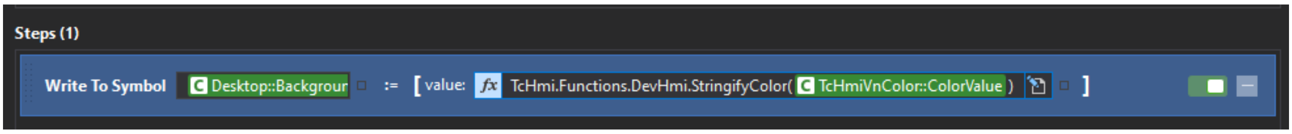
```
module TcHmi {
    export module Functions {
        export module Custom {
            export function StringifyColor(color: number[]): TcHmi.Color {
                if (color.length >= 3) {
                    return { color: `rgb(${color[0]}, ${color[1]}, ${color[2]})` };
                }
                throw new Error("Could not stringify color.");
            }
        }
```

```
        registerFunctionEx('StringifyColor', 'TcHmi.Functions.DevHmi', DevHmi.StringifyColor);
    }
}
```

返回值必须是 Color 类型且结构如下，才能作为颜色对象被接受：

```
{
    color: ‘rgb(<red>, <green>, <blue>)’
}
```

例如，可以使用这样的函数转换 TcHmiVnColor 控制器的颜色值，然后将其设置为另一个用户界面元素的背景颜色。



7.2.2.3 颜色格式

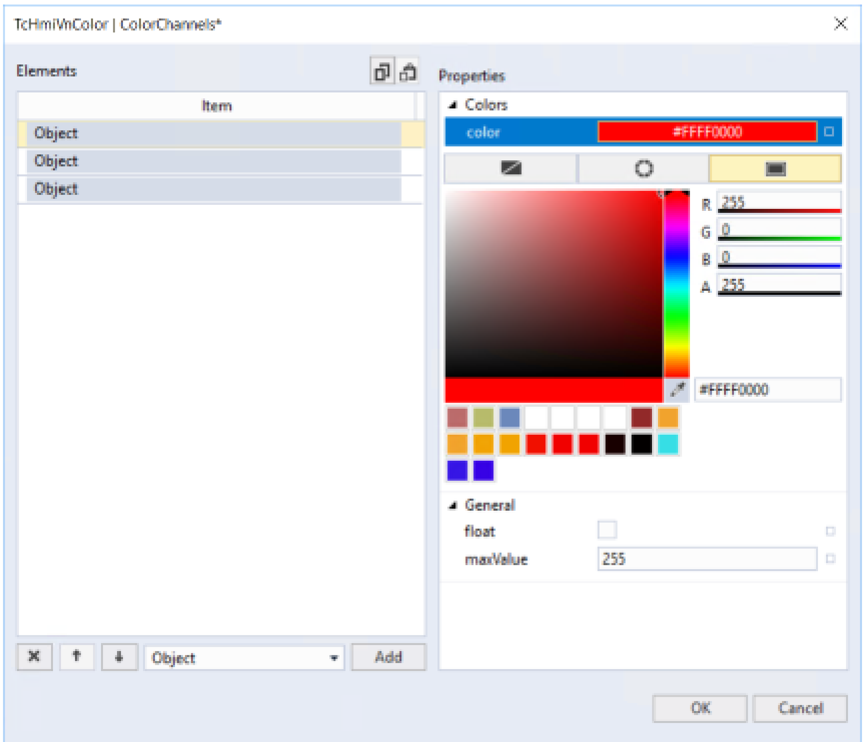
颜色表示方法有很多种。诸如单色（1 通道灰度值）、RGB 颜色空间（3 通道颜色值）、HSV / HLS 颜色空间（3 通道颜色值）等最常见格式将在下文详细介绍。不过，该控制器提供了许多配置选项，因此也可以设置和使用特殊颜色格式。

设置颜色格式

为了使控制器适应任何颜色格式，可以详细设置一般条件：

- 通道数量（对象）
- 每个通道的最大值
- 每个通道的数据类型（浮点数或整数）
- 显示每个通道的颜色。这将用作相应滑块的颜色和显示的颜色值。

可通过属性 ColorChannels 的对话框进行调整。RGB 格式为默认预设值。例如，要设置单色 8 格式，必须先删除现有三个通道（对象）中的两个。之后，可以将 R、G 和 B 颜色通道中的颜色表示调整为 128 这样的平均灰度值。



针对以下颜色格式提供配置示例：

- RGB（标准） [▶ 2630]
- HSV / HLS（值调整） [▶ 2631]

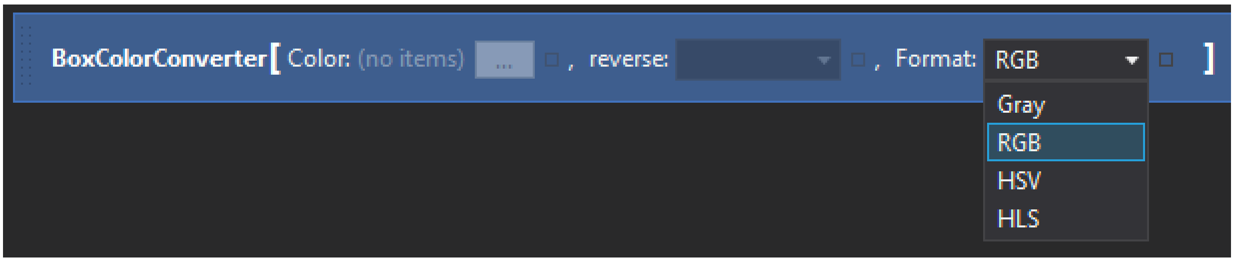
- 百分比 RGB（自定义格式） [▸ 2631]

颜色框转换

本节仅适用于使用颜色框且颜色不符合标准 RGB 格式的情况。

通过数字和滑块显示颜色值时，不需要对颜色格式有更深入的了解，因为只显示数值。不过，为了在颜色框中显示颜色，必须定义如何解释单个颜色通道及其值。例如，RGB 值的显示方式必须与 HSV 值不同。

由于浏览器特定颜色输入元素默认处理 RGB 颜色，因此所有其他颜色格式都必须首先转换为 RGB，才能在颜色框中显示。这种转换可通过属性 BoxColorConversion 进行设置。“灰色”、“HSV”和“HLS”格式的转换函数已经存在。



通用颜色转换器

除了这种针对颜色框的特殊转换函数外，还有一个通用颜色转换器，其具有同一套色彩格式，适用于 HMI 中的一般用途，也以 ConvertColor [▸ 2635] 的形式提供。

对于完全不同的颜色格式或具有不同缩放比例的颜色格式，可以创建用户定义函数 [▸ 2631]。

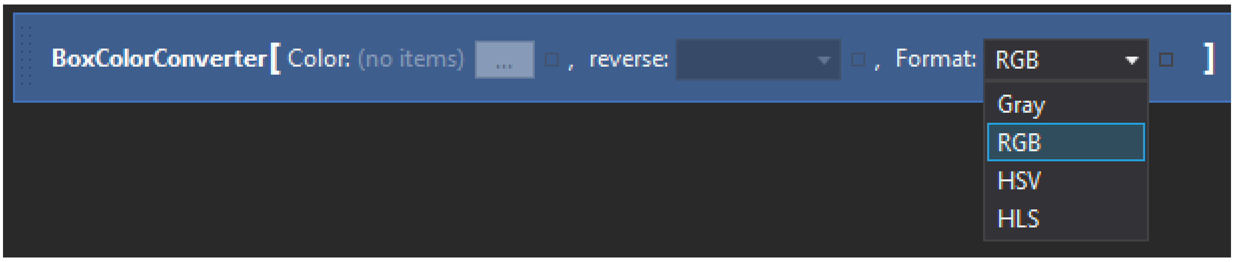
7.2.2.3.1 标准的 RGB 颜色格式

默认情况下，定义三个通道，所有通道均配置为最大值为 255 的整数，且显示颜色对应于 RGB 格式。

ColorChannels 属性的值可通过脚本如下写入：

```
[
  {
    "color": {
      "color": "rgba(255, 0, 0, 1)"
    },
    "float": false,
    "maxValue": 255.0
  },
  {
    "color": {
      "color": "rgba(0, 255, 0, 1)"
    },
    "float": false,
    "maxValue": 255.0
  },
  {
    "color": {
      "color": "rgba(0, 0, 255, 1)"
    },
    "float": false,
    "maxValue": 255.0
  }
]
```

BoxColorConversion 属性设置为供给函数 BoxColorConverter，配置格式为“RGB”。



7.2.2.3.2 HSV/HLS 颜色格式

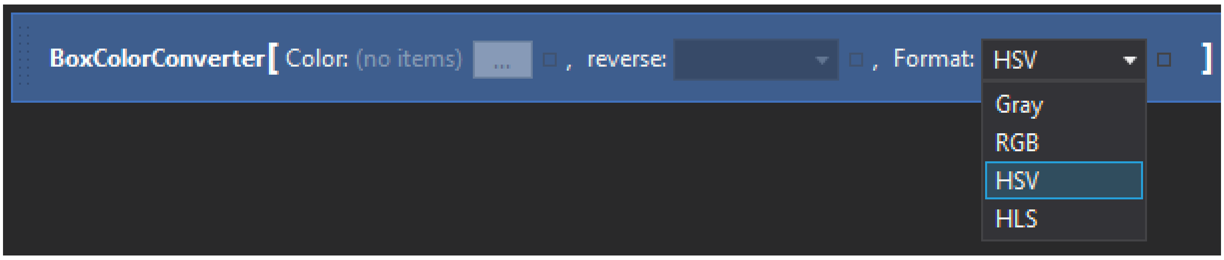
如果需要通过颜色控件以 HSV 格式显示一种颜色，必须进行一些调整。“色调”通道的数值范围仅定义在 0 到 180 之间，因此第一个通道的maxValue必须减少到 180。由于 HSV 格式的各个通道不能被指定为特定的颜色，用于显示滑块和数值的颜色应该被设置为统一的灰度值，这样就不会产生混淆。

同样的设置和数值范围可以用于 HLS 颜色空间，因为只有 L 和 S 通道的计算不同。

ColorChannels属性的数值可以通过脚本编写，如下所示：

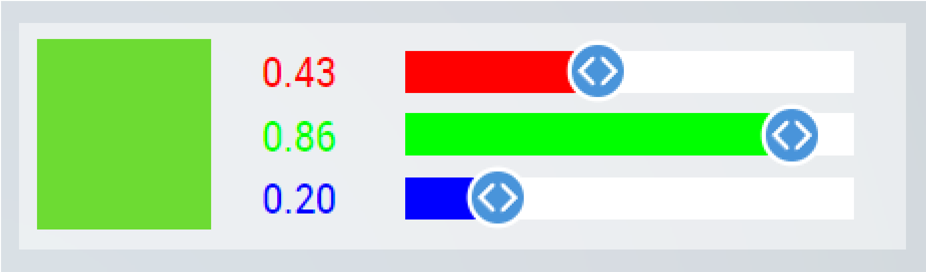
```
[
  {
    "color": {
      "color": "rgba(90, 90, 90, 1)"
    },
    "float": false,
    "maxValue": 180.0
  },
  {
    "color": {
      "color": "rgba(90, 90, 90, 1)"
    },
    "float": false,
    "maxValue": 255.0
  },
  {
    "color": {
      "color": "rgba(90, 90, 90, 1)"
    },
    "float": false,
    "maxValue": 255.0
  }
]
```

最后，必须调整BoxColorConversion属性中颜色框的转换。为此，必须在BoxColorConverter功能中选择相应的颜色格式：



7.2.2.3.3 用户定义的颜色格式

作为颜色控制配置的另一个示例，考虑使用值范围在 0 和 1 之间的 RGB 颜色格式。要在“颜色控制”中这样显示，必须在 ColorChannel 属性中将通道设置为 float，且 maxValue 为1。



ColorChannels 属性的值可通过脚本如下写入：

```
[
  {
    "color": {
      "color": "rgba(255, 0, 0, 1)"
    },
    "float": true,
    "maxValue": 1.0
  },
  {
    "color": {
      "color": "rgba(0, 255, 0, 1)"
    },
    "float": true,
    "maxValue": 1.0
  },
  {
    "color": {
      "color": "rgba(0, 0, 255, 1)"
    },
    "float": true,
    "maxValue": 1.0
  }
]
```

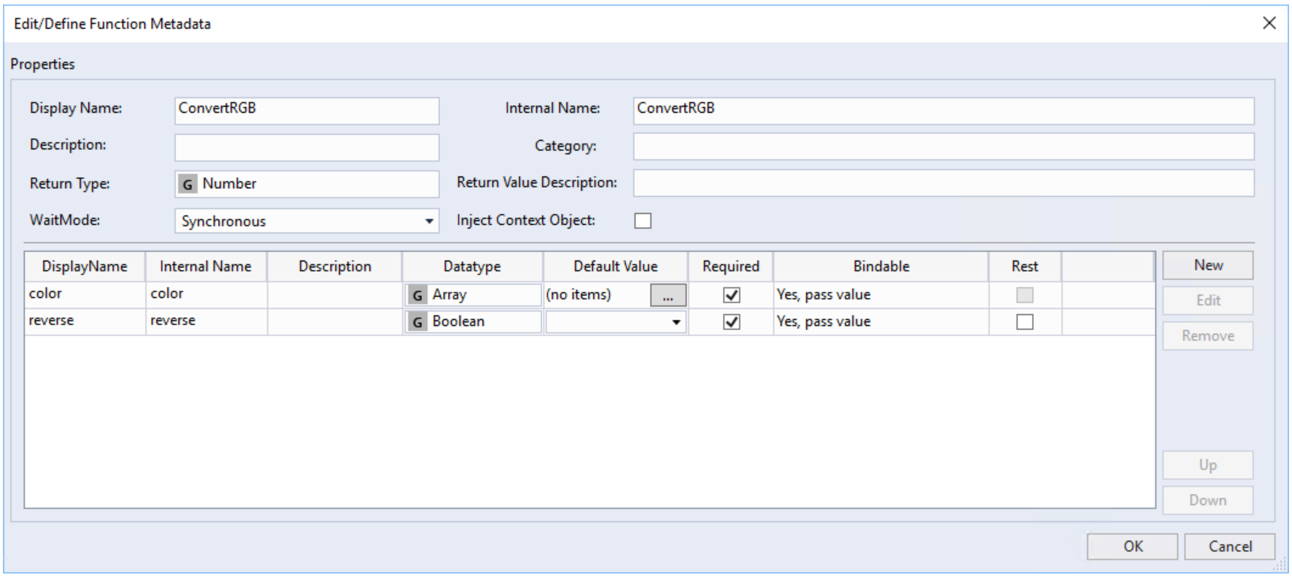
ColorBoxFormatting 函数

为了使颜色框在值范围改变后仍能显示正确的颜色，必须调整 BoxColorConversion 属性。与标准转换不同，没有适用于此的现成函数。相反，有必要创建一个单独的函数。下方样本展示了如何为上述用例创建这样一个函数，以及它必须具备哪些接口。

> **i** 各函数的一般使用方法在 HMI 文档中有说明。

创建 TypeScript 函数后，必须创建两个参数。第一个参数获取要转换的颜色值，因此必须以 Array 类型创建。第二个参数指定转换方向，必须创建为 Boolean。如为 True，值转换为 RGB；如为 False，则从 RGB 转换回用户定义格式。需要注意的是，由于时序原因，这些传递的参数可能与函数执行时相应控制属性的值不同。

函数必须返回一个 Number 类型的值，该值然后显示在颜色框中。



在下方代码样本中，颜色值的缩放根据传递的反向值进行调整：

```
module TcHmi {
  export module Functions {
    export module TcHmiProject {
      export function ConvertRGB(color: number[], reverse: boolean) {
        if (!reverse) {
          return color.map((value: number) => value * 255)
        }
        else {
          return color.map((value: number) => value / 255)
        }
      }
    }
    registerFunctionEx('ConvertRGB', 'TcHmi.Functions.TcHmiProject, TcHmiProject.ConvertRGB);
  }
}
```

这可以确保颜色框显示正确的颜色，并且手动更改颜色框中的颜色可以获得正确的颜色值。

7.2.3 矩形（交互式）

矩形（交互式）控件提供了通过点击显示或绘制矩形的环境。控件主要在图像查看 [▶ 2602]中使用；但是，也可以独立使用。因此，可以选择图像区域以标记物体或定义 ROI。

对于以下所有属性，控件都有一个 gette r和一个 setter 方法。关于基类**控制**所提供的所有其他属性、事件和权限，可查看TwinCAT HMI 文档。

类别：颜色

属性	类型	描述
StrokeColor	SolidColor	所绘制矩形的轮廓颜色。
HandleColor	SolidColor	所绘制矩形的控键颜色。

类别：通用

属性	类型	描述
ShapeData	RectangleData	包含矩形的描述。既可以用来检索一个已画好的矩形，也可以用来设置一个特定的矩形。
AspectRatio	Number	指定绘制矩形时强制执行的长宽比。0值并不强制要求一个特定的长宽比。

函数	描述
clear	从ShapeData属性中删除绘制的矩形和描述。

类别：使用方法

属性	类型	描述
InitialSelection	"OnlyViaClicks" | "OnlyViaDrag" | "ViaBoth"	指定是否可以用两次点击来绘制矩形，或用扫动方式来绘制，或同时使用这两个选项。
CenterHandle	Boolean	在矩形的中心显示一个元素来进行移动。
CornerHandles	Boolean	在矩形的四角显示元素以改变其大小。
EdgeHandles	Boolean	在矩形的边缘显示元素以调整其大小。
RotationHandle	Boolean	在矩形的外侧显示元素，以进行旋转。
ShapeDragging	Boolean	确定矩形是否可以在边缘移动。
AngleInterval	Number	指定矩形可以旋转的步骤。在 0 处，矩形可以根据需要进行旋转。

类别：外观

属性	类型	描述
StrokeThickness	MeasurementValue	矩形的线条厚度。
HandleSize	MeasurementValue	用于移动和改变矩形的元素的尺寸。
StrokeShape	"Rectangle" | "Ellipse"	指定绘制矩形或椭圆。两者的描述相同。如需得到方形或圆形，AspectRatio 可以设置为 1。
ClickableSize	MeasurementValue	上述移动和更改矩形的元素周围可点击区域的大小。这可以简化触摸屏的使用，而不改变外观。

7.2.4 多边形（交互式）

多边形（交互式）控件提供了通过点击显示或绘制多边形的环境。该控件主要在图像查看 [▸ 2602]控件中使用，但也可以独立使用。除了多边形外，还可以显示或绘制线条或单个点等。为此，NumPoints属性必须相应地设置为 1 或 2。

对于以下所有属性，控件都有一个 gette r和一个 setter 方法。关于基类**控制**所提供的所有其他属性、事件和权限，可查看TwinCAT HMI 文档。

类别：颜色

属性	类型	描述
StrokeColor	SolidColor	所绘制多边形的轮廓颜色。
HandleColor	SolidColor	所绘制多边形的控键颜色。

类别：通用

属性	类型	描述
ShapeData	PolygonData	包含多边形的描述。既可用于检索绘制的多边形，也可用于设置一个特定的多边形。
NumPoints	Number	指定完整的多边形应该有多少个点。在 0 时，任何数量的点都有可能。这个规范对于能够处理整个多边形很重要。
Closed	Boolean	指示多边形是否封闭，或者是否多段线。

函数	描述
clear	从ShapeData属性中删除绘制的多边形和描述。

类别：使用方法

属性	类型	描述
RotationHandle	Boolean	显示多边形外部的元素，以进行旋转。
AngleInterval	Number	指定多边形可以旋转的步骤。在 0 处，多边形可以根据需要进行旋转。

类别：外观

属性	类型	描述
StrokeThickness	MeasurementValue	多边形的线条厚度。
HandleSize	MeasurementValue	用于移动和改变多边形的元素的尺寸。
ClickableSize	MeasurementValue	上述移动和更改多边形的元素周围可点击区域的大小。这可以简化触摸屏的使用，而不改变外观。

7.2.5 帮助功能

以下帮助函数被某些控件用作默认行为，或用于确保与 PLC 数据类型的互操作性。

框颜色转换器

BoxColorConverter函数列在Vision类别中。它被用作颜色控制 [▶ 2623]BoxColorConversion属性的默认值。它在 RGB 和format参数指定的颜色格式之间进行转换。

参数	数据类型	描述
color	Number[]	待转换的颜色值。
reverse	Boolean	转换方向。 false：RGB 到format true：format 到 RGB
format	ColorSpace	转换格式： "Gray" | "RGB" | "HSV" | "HLS"

该功能以Number[]返回转换后的颜色。

转换颜色

ConvertColor函数列在Vision类别中。它在灰色、RGB、HSV 和 HLS 格式之间转换颜色数组。

参数	数据类型	描述
color	Number[]	待转换的颜色值。
srcFormat	ColorSpace	指定颜色值的格式： "Gray" | "RGB" | "HSV" | "HLS"
destFormat	ColorSpace	待转换为格式： "Gray" | "RGB" | "HSV" | "HLS"
rounded	Boolean	将所有输出元素四舍五入为整数值。

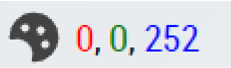
该功能以Number[]返回转换后的颜色。

例如，如需将 RGB 格式的红色转换为 HLS 格式，可以使用以下函数调用：

```
let rgb = [255, 0, 0];
let hls = TcHmi.Functions.Beckhoff.Vision.ConvertColor(rgb, "RGB", "HLS", true);
console.log(hls);

//console output
//[0, 127.5, 255]
```

PixelColorFormatter

函数 PixelColorFormatter 列于 Formatting 类别中。它是 Image Watch Control [▶ 2602] PixelColorFormat 属性的默认值。它将 PixelColor 值转换为由 HTLM 表示的字符串。具体来说，单个通道的颜色值被附加并用颜色突出显示：



参数	数据类型	描述
PixelColor	Number[]	要转换的颜色值。

该函数返回一个 HTML 编码的 String。

形状值格式

ShapeValueFormatter函数列在Formatting类别中。它是图像查看控制 [▶ 2602]的ShapeValueFormat属性的默认值。它将ShapeValue值转换为可以通过HTML显示的字符串。例如，对于一个矩形，面积以像素计算，并以字符串形式返回。

参数	数据类型	描述
shapeData	Any	待转换的形状数据。
shapeType	形状类型 [▶ 2638]	所传递的形状类型。

该函数返回一个 HTML 编码的String。

旋转矩形

ToRotatedRectangle函数列在数据转换类别中。它将一个UprightRectangle转换为一个RotatedRectangle，角度为0。

参数	数据类型	描述
rectangle	TcVnRectangle [▶ 2638]	待转换矩形。

该函数以TcVnRotatedRectangle [▶ 2638]的形式返回转换后的矩形，并可按如下方式应用：

```
let upright = {
    nWidth: 100,
    nHeight: 60,
    nX: 200,
    nY: 100,
```

```
}
let rotated = TcHmi.Functions.Beckhoff.Vision.ToRotatedRectangle(upright);
console.log(rotated);

//console output
// {
//     aCenter: [250, 130],
//     fAngle: 0,
//     stSize: { fWidth: 100, fHeight: 60 },
// }
```

直立矩形

ToUprightRectangle函数列在数据转换类别中。它将一个RotatedRectangle转换为一个UprightRectangle，放弃了角度。

参数	数据类型	描述
rectangle	TcVnRotatedRectangle [▶ 2638]	待转换矩形。

该函数以TcVnRectangle [▶ 2638]的形式返回转换后的矩形，并可按如下方式应用：

```
let rotated = {
    aCenter: [250, 130],
    fAngle: 0,
    stSize: { fWidth: 100, fHeight: 60 },
}
let upright = TcHmi.Functions.Beckhoff.Vision.ToUprightRectangle(rotated);
console.log(upright);

//console output
// {
//     nWidth: 100,
//     nHeight: 60,
//     nX: 200,
//     nY: 100,
// }
```

7.2.6 数据类型

颜色通道

定义一个颜色格式的通道。是一个具有以下属性的对象数组，且每个对象描述一个颜色通道：

属性	数据类型	描述
color	SolidColor	通道的显示颜色。
maxValue	Number	允许值范围的上限。下限始终为0。
float	Boolean	定义通道通过浮点数或通过整数描述。

例如，在颜色控制 [▶ 2629]中使用。

颜色符号

定义一个具有 ColorValue 兼容数据类型的符号。

颜色值

定义了一个可以作为颜色进行解释的值。是一个描述灰度值的数字，或者是一个通过相应通道数描述颜色的 3 或 4 元素数组。

例如，有效值为：

- 0
- 128
- [0, 0, 255]
- [0, 0, 255, 255]

方向

定义了显示元素的方向。可能的值为“水平”和“垂直”。

TcVnRotatedRectangle

代表可以旋转的矩形。是一个具有以下属性的对象：

属性	数据类型	描述
aCenter	TcVnPoint2_REAL	矩形中心。
fAngle	DINT	旋转矩形的角度。
stSize	TcVnSize2_REAL	围绕中心点的矩形尺寸。

形状类型

在图像查看控件 [▸ 2602]中定义了可选择的形状类型。是一个枚举函数，附带以下值：

- 点
- 线
- 方形
- 矩形
- 圆形
- 椭圆
- 多边形

TcVnRectangle

代表一个轴对齐的矩形，且是一个具有以下属性的对象：

属性	数据类型	描述
nHeight	DINT	矩形高度。
nWidth	DINT	矩形宽度。
nX	DINT	右上角的 X 坐标。
nY	DINT	右上角的 Y 坐标。

使用模式

定义了显示元素的显示和可编辑性。可能的值：

数值	描述
隐藏	元素未显示。
仅查看	元素已显示，但不能编辑。
可编辑	元素已显示且可编辑。

主要用于颜色控制 [▸ 2624]。

8 样本

在本章中，您可以找到 PLC 中许多函数和功能块以及向导和相机配置的应用示例。在开始查看 PLC 例程之前，请访问 第一步 [▸ 23]。

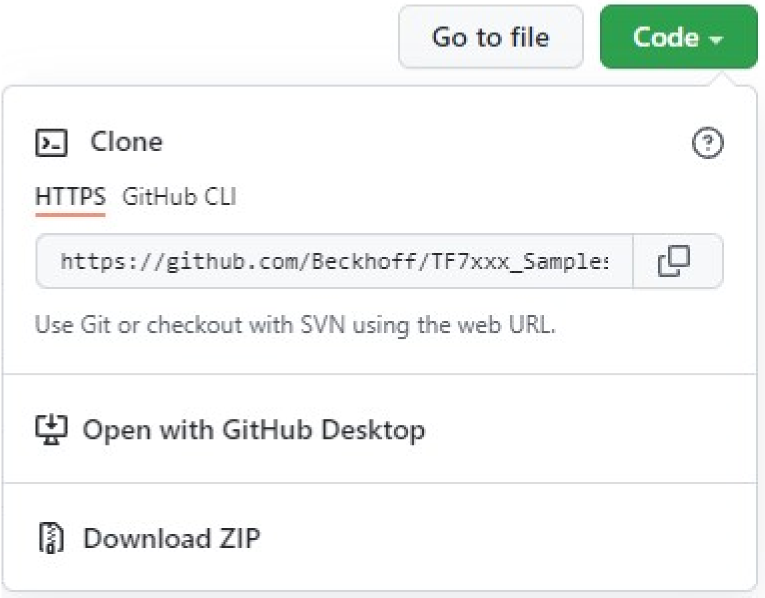
例程相互独立。每个例程都有一个重点，在下方会对其进行分组。以函数或功能块为重点的例程根据 API 结构列出。

下载

例程可从 GitHub 上的存储库获取：https://github.com/Beckhoff/TF7xxx_Samples。您可以选择克隆存储库或下载包含所有例程的 ZIP 文件。

必须单独下载例程图像

GitHub 存储库只包含带有相应代码的 TwinCAT 项目。必须从倍福网站单独下载图像数据“TF7xxx_Sample_Images.zip”。链接和更多信息也可在 GitHub 存储库的 Readme 文件中找到。



项目资源

每个例程都按照大纲顺序和章节标题名称放在一个单独文件夹中。其中有一个 TwinCAT 项目和相应的 PLC 代码。

例如，包含例程图像的独立 zip 文件夹可以解压到存储库中的图像文件夹中。文件夹名称以及例程图像的存储结构与项目结构相同。因此，只需通过具有相同文件夹名称的路径即可分配图像。

注意通往存储库或 TwinCAT 项目的短路径，例如

由于操作系统可能将路径长度限制为 255 个字符，如果文件夹路径过长，在执行 TwinCAT 项目时可能会出现问题。为了便于理解，带有章节标题的文件夹结构已经很长，因此应为整个存储库或单个 TwinCAT 项目选择一个较短的路径，例如 C:\TF7xxx_Samples/。

在执行项目之前，请根据目标系统调整系统配置 [▸ 27]，尤其要注意核、路由器内存和 PLC 循环时间的配置。此外，请按照 版本概览 [▸ 13] 中的说明检查相机或文件源的 TcCOM 对象以及 PLC 库的版本，必要时调整版本。

在 GitHub 存储库中可以找到一个 PowerShell 脚本 (SetTcCOMModuleVersion.ps1)，其可用于调整 PLC 文件夹中所有项目的版本。为此，请通过 PowerShell 控制台运行脚本，如下指定所需版本：

```
PS C:\TF7xxx_Samples> .\SetTcCOMModuleVersion.ps1 4.0.4.8
```

如果需要图像或数据流，可通过 文件源控制 [▸ 111] 或相机对象的 记录/回放 [▸ 89] 选项卡对它们进行整合。

8.1 函数样本

以下所有样本均针对单个函数，并按照 TwinCAT Vision API 参考文献进行排序。对于每个样本，都有相应的图像，可通过文件源加载到 PLC。此外，各自的样本描述显示，为了看到某些效果，必须改变哪些函数参数。

启动函数样本的一般步骤：

1. 在文件源控件中添加所附的样本图像。
   如果使用自己的图像，请注意预期的图像格式，其可能因示例不同而不同。
2. 激活配置，并将 TwinCAT 和 PLC 切换到运行模式。

⇨ 现在，通过**查看** >**其他窗口** > **ADS 图像查看**，可以查看来自 PLC 的图像。
- ipImageInDisp 始终显示输入图像
-ipImageResDisp 始终显示结果图像
（当然，这两个图像的名称对你来说可以是不同的，我们在示例中总是将变量命名为相同，这样可以很容易识别，你可以在任何项目中快速找到你的方式）
- 如果需要，其他图像显示中间步骤的结果或进一步的信息。
关于详细信息，可查看各自的样本描述。

在样本描述中，你还会发现在登录的运行模式下，通过**预备值**，然后点击 **写入值**，来写出函数参数。然后可以在 Ads 图像查看中显示的图像中再次看到这些效果。

样本

- 基本图像操作 [▶ 2642]
- 代码读取 [▶ 2645]
- 轮廓分析 [▶ 2655]
- 图像分析 [▶ 2664]
- 图像颜色和对比度处理 [▶ 2684]
- 图像分割 [▶ 2693]
- 测量 [▶ 2695]

编写自己函数的样本

如果在应用中重复使用 TwinCAT Vision API 函数的某些序列，最好将它们封装在一个函数中。下面的样本解释了在操作时应该注意的事项：

示例：自行编写的函数 [▶ 2712]

8.1.1 容器基本操作

8.1.1.1 容器元素选择

在这个样本中，容器元素在用户定义条件的帮助下进行过滤。函数 F_VN_CopyContainerElementsConditional_ITcVnContainer [▶ 722]用于此目的，因为容器元素反过来是容器。如果容器元素是迭代器，函数F_VN_CopyContainerElementsConditional_ITcVnForwardIterator [▶ 724] 也可以用于此。

说明

如果需要对多维容器的元素进行过滤，由于访问子元素，这很快就会造成大量的编程工作。为了应对这种情况，TwinCAT Vision 正是为此提供了一个函数，使用方法如下：

创建了一个功能块，实现了接口ITcVnCustomElementCondition_ITcVnContainer。这个接口拥有Condition方法，它接受一个类型为ITcVnContainer的元素。待应用于元素的过滤条件在这个方法的实现中定义。具有过滤条件的功能块然后被转移到功能F_VN_CopyContainerElementsConditional_ITcVnContainer，以便对一个容器的所有元素进行过滤。在这个样本中，元素根据是否拥有超过五个子元素进行过滤。

应用

举例来说，一个容器的子容器根据它们各自拥有的元素数量进行过滤。具体来说，只有那些有五个以上元素的子容器才会被接受到一个新的容器中。容器的结构可以通过阵列aContainerStructure更改。这些元素表示每个子容器有多少个子元素。如果子容器本身的数量也同样变化，常数cNumberOfSubContainers必须与之相匹配。

程序

主

首先，定义了一个常数整数，以便能够简单地改变待过滤的子容器：

```
VAR CONSTANT
    cNumberOfSubContainers   :   INT := 5;
END_VAR
```

然后声明所有必要的变量：

```
VAR
    aContainerStructure      :   ARRAY [0..(cNumberOfSubContainers-1)] OF INT := [2, 7, 3, 5, 12];
    ipHelper                 :   ITcVnContainer;
    ipContainerBase          :   ITcVnContainer;
    ipContainerFiltered      :   ITcVnContainer;
    fbCondition              :   FB_ConditionMoreThanFive;
    nSelectedContainers      :   ULINT;
    hr                       :   HRESULT;
    i                        :   INT;
END_VAR
```

aContainerStructure	容器结构的定义。例如，一个长度为5初始化值为[2, 7, 3, 5, 12]的数组意味着，将在样本中创建一个有5个子容器的容器。第一个子容器将拥有 2 个元素，第二个子容器将拥有 7 个元素，以此类推。 数组长度由常数cNumberOfSubContainers定义。
ipHelper	辅助接口指针，用于通过子容器填充主容器ipContainer。
ipContainerBase	容器，其中创建在aNumberOfElement中定义的子容器。
fbCondition	包含过滤条件的功能块。
ipContainerFiltered	容器，在最后将仅包含来自ipContainer且满足过滤条件的子容器。
nSelectedContainers	整数，表示有多少个子容器实际满足过滤条件。
hr	类型为HRESULT [▶ 122]的状态变量。
i	辅助变量

在程序部分，创建了一个容器，其中充满了上面定义的子容器。调用F_VN_ReserveContainerMemory [▶ 742]是可选的，只是为了实现更好的性能。

```
hr := F_VN_CreateContainer(ipContainerBase, ContainerType_Vector_Vector_REAL, 0, hr);
hr := F_VN_ReserveContainerMemory(ipContainerBase, cNumberOfSubContainers, hr);
FOR i:=0 TO (cNumberOfSubContainers-1) DO
    hr := F_VN_CreateContainer(ipHelper, ContainerType_Vector_REAL, aContainerStructure[i], hr);
    hr := F_VN_AppendToContainer_ITcVnContainer(ipHelper, ipContainerBase, hr);
END_FOR
```

然后对刚刚创建的容器进行过滤。具有自写条件的功能块被指定为过滤条件。

```
hr := F_VN_CopyContainerElementsConditional_ITcVnContainer(
    ipSrcContainer       :=  ipContainerBase,
    ipDestContainer      :=  ipContainerSelection,
    ipConditionFB        :=  fbCondition,
    hr);
```

为了理解函数的功能，对满足所述条件的元素进行计数。变量nSelectedContainers现在根据子容器和过滤条件的不同而变化。

```
hr := F_VN_GetNumberOfElements(ipContainerSelection, nSelectedContainers, hr);
```

在这种情况下，这个数字表示有多少个子容器有五个以上的子元素。

具有过滤条件的功能块

通过方法Condition，过滤条件作为一个功能块被转移。为了使自己编写的功能块能够被转移到功能中，它必须实现接口 ITcVnCustomElementCondition_ITcVnContainer [▶ 228]：

```
FUNCTION_BLOCK FB_ConditionMoreThanFive IMPLEMENTS ITcVnCustomElementCondition_ITcVnContainer
```

Condition方法定义了过滤条件。对于每个子容器，都单独调用方法。根据返回值是TRUE还是FALSE，决定相应子容器是否被接受到新容器中。

```
METHOD Condition : BOOL
```

在每次调用该方法时，当前的子容器被作为输入参数传输：

```
VAR_INPUT
    ipElement           :   ITcVnContainer;
END_VAR
```

在这种情况下，如果一个元素有五个以上的子元素，那么它将被接受。为此，将计算子元素的数量，并根据结果返回数值TRUE或FALSE。

```
hr := F_VN_GetNumberOfElements(ipElement, nNumberOfElements, hr);
IF FAILED(hr) OR nNumberOfElements <= 5 THEN
    Condition := FALSE;
ELSE
    Condition := TRUE;
END_IF
```

所需变量的声明如下：

```
VAR
    nNumberOfElements   :   ULINT;
    hr                  :   HRESULT;
END_VAR
```

功能块本身和Condition方法都必须包含编译器属性{attribute 'c++_compatible'}。

8.1.2 基本图像操作

8.1.2.1 融合多个图像

在本例程中，使用 F_VN_FuseImagesArray [▶ 766] 函数将五个图像连接起来。该功能在与线扫描相机和连续物料流的无缝记录相结合时尤为实用。

说明

F_VN_FuseImagesArray [▶ 766] 函数接受一个图像数组（最大为十），并将它们连接成一个新图像。在下方样本程序中，最初创建了 5 个不同颜色、高度为 10px 的图像。数量可以通过变量 *nImages* 更改，高度可以通过 *nHeight* 更改。请确保该数字仅在 1-10 范围内变化。

使用参数 *nFirstLine* 和 *nNumLines* 指定图像线，即可确定结果图像的大小。这样就可以裁剪第一和最后一个图像。

如果只需要连接两个图像，也可以使用函数 F_VN_FuseImages [▶ 765]。

程序

声明了以下变量：

```
VAR
    hr                  :   HRESULT;
    aImages             :   ARRAY[0..9] OF ITcVnImage;
    ipImageFused        :   ITcVnImage;
    ipImageFusedDisp    :   ITcVnDisplayableImage;
    nImages             :   INT := 5;
    nHeight             :   UDINT := 10;
    aColor              :   ARRAY[0..9] OF TcVnVector4_LREAL := [[255,0,0], [200,50,0], [150,100,0],
[100,150,0], [50,200,0], [0,250,0], [0,200,50], [0,150,100], [0,100,150], [0,50,200]];
    i                   :   INT;
VAR
```

hr	类型 HRESULT [▶ 122] 的状态变量。
aImages	包含多个要连接在一起的图像之数组。
ipImageFused	将来自 aImages 的多个图像连接在一起的结果图像。
ipImageFusedDisp	可显示的结果图像。
nImages	要连接的图像数量。
aColor	十种不同的颜色，以便能够区分 aImages 中的图像。
i	任意变量。

在程序中，首先对 aImages 数组中的图像进行初始化和着色。为了能够直观地区分图像，使用十种不同的 aColor 颜色。

```
FOR i := 0 TO nImages-1 DO
     hr := F_VN_CreateImageAndSetPixels(aImages[i], 100, nHeight, ETcVnElementType.TCVN_ET_USINT, 3,
aColor[i], hr);
END_FOR
```

然后，将来自 aImages 的定义数量 nImages 图像垂直连接到图像ipImageFused 中：

```
hr := F_VN_FuseImagesArray(aImages, nImages, ipImageFused, 0, nImages * nHeight, hr);
```

最后显示图像：

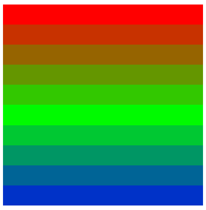
```
hr := F_VN_TransformIntoDisplayableImage(ipImageFused, ipImageFusedDisp, hr);
```

结果

nImages = 2：

nImages = 5：

nImages = 10：



类似例程

复制图像区域 [▶ 2643]

8.1.2.2 复制图像区域

在本例程中，使用不同方法将两个图像复制到了另一图像：

- 使用函数 F_VN_CopyImageRegionToRegion [▶ 752]，
- 选择 ROI 并复制图像，然后
- 使用函数 F_VN_AddImages [▶ 478] 选择一个 ROI 并添加两个图像。

说明

下图所示方法可用于在图像之间交换矩形图像区域。如果需要，也可以借助算术图像运算以部分透明的方式来实现。

函数 F_VN_CopyImageRegionToRegion [▸ 752] 接收两个图像和两个图像区域，并将其中一个图像区域的内容复制到另一图像区域。两个图像区域的大小相同，并且必须完全呈现在图像中。示例中使用该函数将红色图像复制到混合图像。不过，要复制蓝色图像，需要使用具有相同功能的程序化版本。该版本的优点是不需要复制也可以将蓝色和混合图像混合。

在该样本中，可以使用变量 aPositionRed 和aPositionBlue 改变两个复制图像的位置。参数 bCopyBlue 决定是复制还是混合蓝色图像。

变量

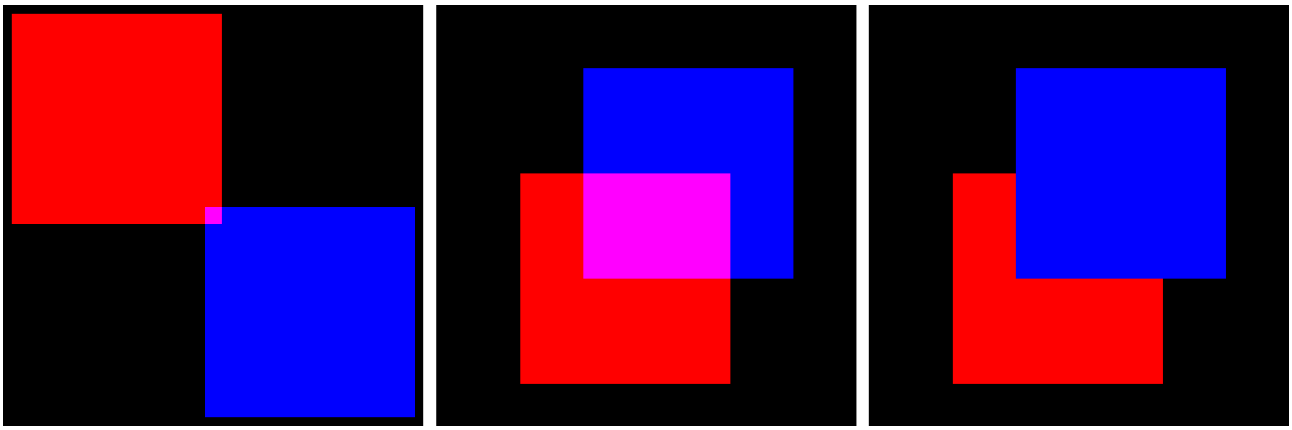
```
hr                  :   HRESULT;

ipImageMerge        :   ITcVnImage;
ipImageRed          :   ITcVnImage;
ipImageBlue         :   ITcVnImage;
ipImageMergeDisp    :   ITcVnDisplayableImage;

aBlack              :   TcVnVector4_LREAL := [0, 0, 0];
aRed                :   TcVnVector4_LREAL := [255, 0, 0];
aBlue               :   TcVnVector4_LREAL := [0, 0, 255];

aPositionRed        :   TcVnPoint := [20, 20];
aPositionBlue       :   TcVnPoint := [480, 480];
bCopyBlue           :   BOOL := FALSE;
```

程序

```
// create images & set colors
hr := F_VN_CreateImageAndSetPixels(ipImageMerge, 1000, 1000, TCVN_ET_USINT, 3, aBlack, hr);
hr := F_VN_CreateImageAndSetPixels(ipImageRed, 500, 500, TCVN_ET_USINT, 3, aRed, hr);
hr := F_VN_CreateImageAndSetPixels(ipImageBlue, 500, 500, TCVN_ET_USINT, 3, aBlue, hr);

// Copy red image to selected region in merge-image
hr := F_VN_CopyImageRegionToRegion(
    ipSrcImage  :=  ipImageRed,
    nXSrc       :=  0,      nYSrc   :=  0,
    nWidth      :=  500,    nHeight :=  500,
    ipDestImage :=  ipImageMerge,
    nXDest      :=  aPositionRed[0],
    nYDest      :=  aPositionRed[1],
    hrPrev      :=  hr
);

// Copy or mix blue image to selected region in merge-image
hr := F_VN_SetRoi(aPositionBlue[0], aPositionBlue[1], 500, 500, ipImageMerge, hr);
IF bCopyBlue THEN
    hr := F_VN_CopyImage(ipImageBlue, ipImageMerge, hr);
ELSE
    hr := F_VN_AddImages(ipImageBlue, ipImageMerge, ipImageMerge, hr);
END_IF
hr := F_VN_ResetRoi(ipImageMerge, hr);

hr := F_VN_TransformIntoDisplayableImage(ipImageMerge, ipImageMergeDisp, hr);
```

结果

以下是程序结果例程。在左边和中间的图像中，添加了蓝色方块，在右边图像中，复制了蓝色方块。此外，方块的目标位置也不同。



类似例程

融合多个图像 [▶ 2642]

8.1.3 代码读取

8.1.3.1 EAN-13 条形码读取

该样本说明了以下内容

- 使用函数F_VN_ReadBarcodeExp [▶ 799]，每个图像读取一个 EAN13 代码
- 函数的执行时间通过看门狗 [▶ 127]进行监控。

说明

- 函数F_VN_ReadBarcode [▶ 797]/F_VN_ReadBarcodeExp [▶ 799]可以搜索和读取图像中不同类型的条形码。如需加快处理时间，可以指定条形码类型（eBarcodeType）。在这个例子中，这些为 TCVN_BT_EAN13。此外，处理时间可以通过看门狗进行监控。样本带看门狗监控的斑点检测 [▶ 2664]提供了关于看门狗的更全面的信息。
- 与F_VN_ReadBarcode [▶ 799]相比，本示例中使用的函数F_VN_ReadBarcodeExp [▶ 797]具有以下特点：
  - 除了条形码类型外，还可以指定搜索方向（eSearchDirection）：
    – TCVN_BSD_ANY 首先在水平方向搜索，然后在垂直方向搜索
    – TCVN_BSD_HORIZONTAL 只在水平方向搜索
    – TCVN_BSD_VERTICAL 只在垂直方向搜索
  - 可以返回进行读取的代码部分。

变量

```
hr                          :   HRESULT;

ipImageIn                   :   ITcVnImage;
ipImageInDsply              :   ITcVnDisplayableImage;

ipImageRes                  :   ITcVnImage;
ipImageResDsply             :   ITcVnDisplayableImage;

// Barcode
ipCodeDecodedList           :   ITcVnContainer;
ipCodeContourList           :   ITcVnContainer;
sCodeAsString               :   STRING(255);
eBarcodeSearchDirection     :   ETcVnBarcodeSearchDirection := TCVN_BSD_ANY;
eBarcodeType                :   ETcVnBarcodeType := TCVN_BT_EAN13;

// Watchdog
hrWD                        :   HRESULT;
tStop                       :   DINT := 50000;
tRest                       :   DINT;

// Output
sText                       :   STRING;

// Color
aColorRed                   :   TcVnVector4_LREAL := [255, 0, 0];
```

代码

```
// Execute the Barcode Reading Function with EAN13 selected monitored by the Watchdog-Function
// ------------------------------------------------------------------
hrWD := F_VN_StartRelWatchdog(tStop, S_OK);
    hr := F_VN_ReadBarcodeExp(
               ipSrcImage          :=   ipImageIn,
               ipDecodedData       :=   ipCodeDecodedList,
               ipContours          :=   ipCodeContourList,
               eBarcodeType        :=   eBarcodeType,
               nCodeNumber         :=   1,
               eSearchDirection    :=   eBarcodeSearchDirection,
               hrPrev              :=   hr);
hrWD := F_VN_StopWatchdog(hrWD, tRest => tRest);
```

```
// Check if the function was executed successfully
IF hr = S_OK THEN
    // Export Code into String
    hr := F_VN_ExportSubContainer_String(ipCodeDecodedList, 0, sCodeAsString, 255, hr);

    // Write Code into Result Image
    hr := F_VN_PutTextExp(sCodeAsString, ipImageRes, 50, 100, ETcVnFontType.TCVN_FT_HERSHEY_PLAIN,
5, aColorRed,3, TCVN_LT_4_CONNECTED, FALSE, hr);

    // Draw Code Contour into Result Image
    hr := F_VN_DrawContours(ipCodeContourList, 0, ipImageRes, aColorRed, 3, hr);
ELSE
    // Write HRESULT into Result Image
    sText := CONCAT('Returncode ', DINT_TO_STRING(hr));
    hr := F_VN_PutTextExp(sText, ipImageRes, 50, 100, ETcVnFontType.TCVN_FT_HERSHEY_PLAIN, 5,
aColorRed,3, TCVN_LT_4_CONNECTED, FALSE, hr);
END_IF

// Write Code Reading proceeded time into Result Image
sText := CONCAT(CONCAT('Time: ', DINT_TO_STRING(tStop - tRest)), 'us');
hr := F_VN_PutTextExp(sText, ipImageRes, 50, 200, ETcVnFontType.TCVN_FT_HERSHEY_PLAIN, 5,
aColorRed,3, TCVN_LT_4_CONNECTED, FALSE, hr);
```

结果

如果选择 TCVN_BSD_ANY 为搜索方向，水平和垂直 EAN13 代码都将被检测到。然而，当比较看门狗的时间时，就会发现水平 EAN13 码（第一张图片）的识别速度比垂直 EAN13 码（第二张图片）快得多，因为代码搜索最初在水平方向进行。

如果搜索方向设置为 TCVN_BSD_VERTICAL，则垂直 EAN13 码的识别速度会快得多（第三张图片）。然而，在这种情况下，没有搜索水平码。

类似样本

- 数据矩阵码读取 [▶ 2648]
- 药品编码读取 [▶ 2649]
- QR 码读取 [▶ 2651]

- 读码过程中的结果评估 [▶ 2653]

8.1.3.2 数据矩阵码读取

该样本说明了以下内容

- 对于每张图片，使用函数F_VN_ReadDataMatrixCodeExp [▶ 805]读取数据矩阵码。
- 函数的执行时间通过看门狗 [▶ 127]进行监控。

说明

函数F_VN_ReadDataMatrixCodeExp [▶ 805]在指定的图像中搜索数据矩阵码并进行读取。与F_VN_ReadDataMatrixCode [▶ 803]函数相比，可以指定每个图像需要搜索的代码数量。在这个样本中，每个图像一个代码。此外，还会返回找到的代码轮廓。TCVN_CSS_ONLY_NOT_FLIPPED被选为搜索策略，因为该图像没有被镜像。

如果图像中存在与代码相似的区域，或者图像中根本没有代码，这对执行时间会产生负面影响。为了防止周期时间超限，在这个样本中，函数F_VN_ReadDataMatrixCodeExp [▶ 805]通过看门狗 [▶ 127]进行监控（见带看门狗监控的斑点检测 [▶ 2664]上的样本）。

变量

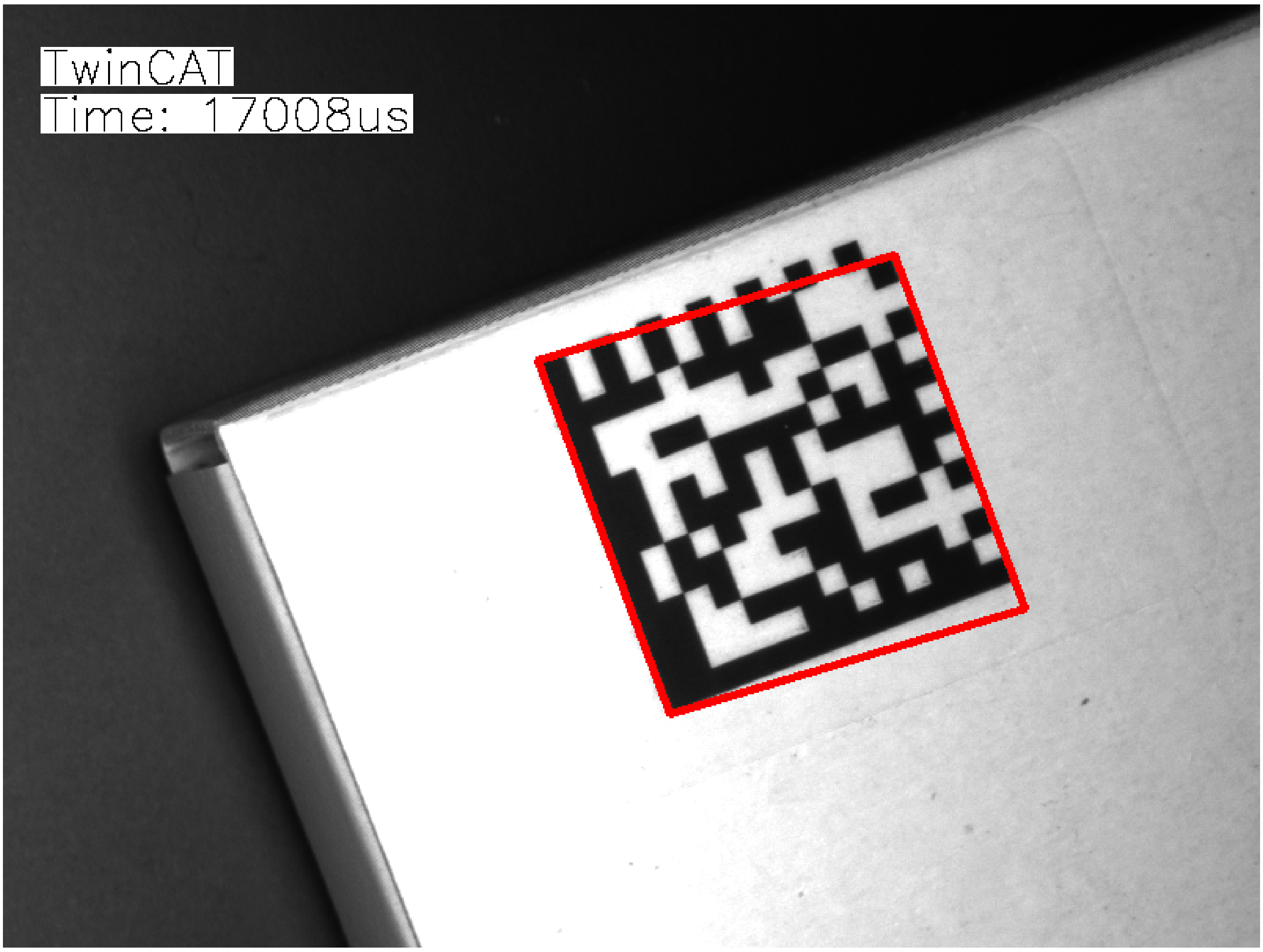
```
hr                   :    HRESULT;

ipImageIn            :    ITcVnImage;
ipImageInDisp        :    ITcVnDisplayableImage;

ipImageRes           :    ITcVnImage;
ipImageResDisp       :    ITcVnDisplayableImage;

// Data Matrix Code
ipCodeDecodedList    :    ITcVnContainer;
ipCodeContourList    :    ITcVnContainer;
sCodeAsString        :    STRING(255);

// Watchdog
hrWD                 :    HRESULT;
tStop                :    DINT := 20000;
tRest                :    DINT;

// Output
sText                :    STRING;

// Color
aColorRed            :    TcVnVector4_LREAL := [255, 0, 0];
```

代码

```
// Execute the DMC Code Reading Function monitored by the Watchdog function
// ------------------------------------------------------------------
hrWD := F_VN_StartRelWatchdog(tStop, S_OK);
    hr := F_VN_ReadDataMatrixCodeExp(
        ipSrcImage       :=  ipImageIn,
        ipDecodedData    :=  ipCodeDecodedList,
        ipContours       :=  ipCodeContourList,
        nCodeNumber      :=  1,
        eSearchStrategy  :=  TCVN_CSS_ONLY_NOT_FLIPPED,
        hrPrev           :=  hr
    );
hrWD := F_VN_StopWatchdog(hrWD, tRest => tRest);

// Check if the function was executed successfully
IF hr = S_OK THEN
    // Export Code into String
    hr := F_VN_ExportSubContainer_String(ipCodeDecodedList, 0, sCodeAsString, 255, hr);

    // Write Code into Result Image
    hr := F_VN_PutText(sCodeAsString, ipImageRes, 25, 50, ETcVnFontType.TCVN_FT_HERSHEY_PLAIN, 2,
aColorRed, hr);

    // Draw Code Contour into Result Image
    hr := F_VN_DrawContours(ipCodeContourList, 0, ipImageRes, aColorRed, 3, hr);
ELSE
    // Write HRESULT into Result Image
```

```
    sText := CONCAT('Returncode ', DINT_TO_STRING(hr));
    hr := F_VN_PutLabelExp(sText, ipImageRes, 25, 50, 2, 2, ETcVnFontType.TCVN_FT_HERSHEY_PLAIN,
aColorRed, aColorWhite, ETcVnLineType.TCVN_LT_8_CONNECTED, hr);
END_IF

// Write Code Reading proceeded time into Result Image
sText := CONCAT(CONCAT('Time: ', DINT_TO_STRING(tStop - tRest)), 'us');
hr := F_VN_PutText(sText, ipImageRes, 25, 80, ETcVnFontType.TCVN_FT_HERSHEY_PLAIN, 2, aColorRed,
hr);
```

结果



类似样本

- EAN-13 条形码读取 [▶ 2645]
- 药品编码读取 [▶ 2649]
- QR 码读取 [▶ 2651]
- 读码过程中的结果评估 [▶ 2653]

8.1.3.3 药品编码读取

该样本说明了以下内容

- 通过函数F_VN_ReadPharmaCodeExp [▶ 810]，每个图像读取一个药品代码。
- 函数的执行时间通过看门狗 [▶ 127]进行监控。

说明

- 函数F_VN_ReadPharmaCodeExp [▶ 810]在指定的图像中搜索药品代码并进行读取。与函数F_VN_ReadPharmaCode [▶ 808]相比，还会返回代码位置。

- 类似代码的结构对执行时间有负面的影响。为了防止周期时间超限，在样本中，函数 F_VN_ReadPharmaCodeExp [▶ 810]通过看门狗 [▶ 127]进行监控。样本带看门狗监控的斑点检测 [▶ 2664]解释了如何使用看门狗 [▶ 127]。

变量

```
hr                  :    HRESULT;

ipImageIn           :    ITcVnImage;
ipImageInDisp       :    ITcVnDisplayableImage;

ipImageRes          :    ITcVnImage;
ipImageResDisp      :    ITcVnDisplayableImage;

// Pharma-Code
ipCodeDecodedList   :    ITcVnContainer;
ipCodeContourList   :    ITcVnContainer;
sCodeAsString       :    STRING(255);
aCodeContour        :    ARRAY[0..3] OF TcVnPoint2_DINT;

// Watchdog
hrWD                :    HRESULT;
tStop               :    DINT := 5000;
tRest               :    DINT;

// Output
sText               :    STRING;

// Color
aColorRed           :    TcVnVector4_LREAL := [255, 0, 0];
```

代码

```
// Execute the Pharma-Code Reading Function monitored by the Watchdog-Function
// ----------------------------------------------------------------
hrWD := F_VN_StartRelWatchdog(tStop, S_OK);
    hr := F_VN_ReadPharmaCodeExp(
                    ipSrcImage       := ipImageIn,
                    ipDecodedData    := ipCodeDecodedList,
                    ipContours       := ipCodeContourList,
                    nCodeNumber      := 1,
                    nMinBarNumber    := 4,
                    hrPrev           := hr);
hrWD := F_VN_StopWatchdog(hrWD, tRest => tRest);

// Check if the function was executed successfully
IF hr = S_OK THEN
    // Export Code into String
    hr := F_VN_ExportSubContainer_String(ipCodeDecodedList, 0, sCodeAsString, 255, hr);

    // Write Code into Result Image
    sText := CONCAT('Code: ', sCodeAsString);
    hr := F_VN_PutTextExp(sText, ipImageRes, 25, 50, ETcVnFontType.TCVN_FT_HERSHEY_PLAIN, 3,
aColorRed, 2, TCVN_LT_4_CONNECTED, FALSE, hr);

    // Draw Code Contour into Result Image
    hr := F_VN_DrawContours(ipCodeContourList, 0, ipImageRes, aColorRed, 3, hr);
ELSE
    // Write HRESULT into Result Image
    sText := CONCAT('Returncode ', DINT_TO_STRING(hr));
    hr := F_VN_PutTextExp(sText, ipImageRes, 25, 50, ETcVnFontType.TCVN_FT_HERSHEY_PLAIN, 3,
aColorRed,2, TCVN_LT_4_CONNECTED, FALSE, hr);
END_IF

// Write Code Reading proceeded time into Result Image
sText := CONCAT(CONCAT('Time: ', DINT_TO_STRING(tStop - tRest)), 'us');
hr := F_VN_PutTextExp(sText, ipImageRes, 25, 100, ETcVnFontType.TCVN_FT_HERSHEY_PLAIN, 3,
aColorRed,2, TCVN_LT_4_CONNECTED, FALSE, hr);
```

结果

类似样本

- 数据矩阵码读取 [▸ 2648]
- EAN-13 条形码读取 [▸ 2645]
- QR 码读取 [▸ 2651]
- 读码过程中的结果评估 [▸ 2653]

8.1.3.4 QR 码读取

该样本说明了以下内容

- 通过函数F_VN_ReadBarcodeExp [▸ 815]，每个图像读取一个 QR 码。
- 函数的执行时间通过看门狗 [▸ 127]进行监控。

说明

函数F_VN_ReadPharmaCodeExp [▸ 815]在指定的图像中搜索 QR 码并进行读取。与函数F_VN_ReadQRCode [▸ 813]相比，还会返回找到代码的轮廓。

如果图像中存在与代码相似的区域，或者图像中根本没有代码，这对执行时间会产生负面影响。为了防止周期时间超限，在这个样本中，函数F_VN_ReadQRCodeExp [▸ 815]通过看门狗 [▸ 127]进行监控（见带看门狗监控的斑点检测 [▸ 2664]上的样本）。

变量

```
hr                  :    HRESULT;

ipImageIn           :    ITcVnImage;
ipImageInDisp       :    ITcVnDisplayableImage;
```

```
ipImageRes           :    ITcVnImage;
ipImageResDisp       :    ITcVnDisplayableImage;

// QR Code
ipCodeDecodedList    :    ITcVnContainer;
ipCodeContourList    :    ITcVnContainer;
sCodeAsString        :    STRING(255);

// Watchdog
hrWD                 :    HRESULT;
tStop                :    DINT := 20000;
tRest                :    DINT;

// Output
sText                :    STRING;

// Color
aColorRed            :    TcVnVector4_LREAL := [255, 0, 0];
```

代码

```
// Execute the QR Code Reading Function monitored by the Watchdog-Function
// ---------------------------------------------------------------
hrWD := F_VN_StartRelWatchdog(tStop, S_OK);
    hr := F_VN_ReadQRCodeExp(
            ipSrcImage       :=  ipImageIn,
            ipDecodedData    :=  ipCodeDecodedList,
            ipContours       :=  ipCodeContourList,
            nCodeNumber      :=  1,
            eSearchStrategy  :=  TCVN_CSS_ONLY_NOT_INVERTED + TCVN_CSS_ONLY_NOT_FLIPPED,
            hrPrev           :=  hr
    );
hrWD := F_VN_StopWatchdog(hrWD, tRest => tRest);

// Check if the function was executed successfully
IF hr = S_OK THEN
    // Export Code into String
    hr := F_VN_ExportSubContainer_String(ipCodeDecodedList, 0, sCodeAsString, 255, hr);

    // Write Code into Result Image
    hr := F_VN_PutTextExp(sCodeAsString, ipImageRes, 50, 100, ETcVnFontType.TCVN_FT_HERSHEY_PLAIN,
5, aColorRed,3, TCVN_LT_4_CONNECTED, FALSE, hr);

    // Draw Code Contour into Result Image
    hr := F_VN_DrawContours(ipCodeContourList, 0, ipImageRes, aColorRed, 3, hr);
ELSE
    // Write HRESULT into Result Image
    sText := CONCAT('Returncode ', DINT_TO_STRING(hr));
    hr := F_VN_PutTextExp(sText, ipImageRes, 50, 100, ETcVnFontType.TCVN_FT_HERSHEY_PLAIN, 5,
aColorRed,3, TCVN_LT_4_CONNECTED, FALSE, hr);
END_IF

// Write Code Reading proceeded time into Result Image
sText := CONCAT(CONCAT('Time: ', DINT_TO_STRING(tStop - tRest)), 'us');
hr := F_VN_PutTextExp(sText, ipImageRes, 50, 200, ETcVnFontType.TCVN_FT_HERSHEY_PLAIN, 5,
aColorRed,3, TCVN_LT_4_CONNECTED, FALSE, hr);
```

结果

类似样本

- 数据矩阵码读取 [▸ 2648]
- EAN-13 条形码读取 [▸ 2645]
- 药品编码读取 [▸ 2649]
- 读码过程中的结果评估 [▸ 2653]

8.1.3.5 读码过程中的结果评估

该样本说明了以下内容

- 使用函数F_VN_ReadBarcode [▸ 799]，每个图像读取一个 EAN13 代码。
- 函数的执行时间通过看门狗 [▸ 127]进行监控。
- 处理功能的不同返回代码。

说明

- 函数F_VN_ReadBarcode [▸ 797]可以搜索和读取图像中不同类型的条形码。关于该函数的使用细节，可以查看样本EAN-13 条形码读取 [▸ 2645]。
- 通过相应的查询，说明对函数返回代码的处理。以下案例之间有所区别：
    - 成功执行了该函数，发现并解码代码。
    - 成功执行函数，代码不存在于图像中或无法找到/解码。
    - 该函数本身没有错误，而是被看门狗中止了。
    - 函数存在其他错误。

关于读码函数的返回代码的详细解释，可查看概览章节Code Reading [▸ 796]。

变量

```
hr                          :   HRESULT;

// Images
ipImageIn                   :   ITcVnImage;
```

```
ipImageInDisp           :   ITcVnDisplayableImage;
ipImageRes              :   ITcVnImage;
ipImageResDisp          :   ITcVnDisplayableImage;

// Barcode
ipCodeDecodedList       :   ITcVnContainer;
ipCodeContourList       :   ITcVnContainer;
sCodeAsString           :   STRING(255);

// Watchdog
hrWD                    :   HRESULT;
tStop                   :   DINT := 50000;
tRest                   :   DINT;

// Color
aColorGreen             :   TcVnVector4_LREAL := [0, 255, 0];

// Return code
nReturnCode             :   DWORD;
sReturnCode             :   STRING;
sResultText             :   STRING;
```

代码

```
// Execute the Barcode Reading Function with EAN13 selected monitored by the Watchdog-Function
hrWD := F_VN_StartRelWatchdog(tStop, S_OK);
    hr := F_VN_ReadBarcode(
        ipSrcImage      :=  ipImageIn,
        ipDecodedData   :=  ipCodeDecodedList,
        eBarcodeType    :=  ETcVnBarcodeType.TCVN_BT_EAN13,
        hrPrev          :=  hr
    );
hrWD := F_VN_StopWatchdog(hrWD, tRest => tRest);

// Handle return-code
IF hr = S_OK THEN
    // Prepare code result in string
    hr := F_VN_ExportSubContainer_String(ipCodeDecodedList, 0, sCodeAsString, 255, hr);
    sResultText := CONCAT('Code: ', sCodeAsString);
ELSE
    // Check for succeeded return codes or add specific error handling
    CASE hr OF
        S_FALSE:
            sResultText := 'No code found...';

        S_WATCHDOG_TIMEOUT:
            sResultText := 'Cancelled by watchdog...';
    ELSE
        // Extract error-code from HRESULT & react accordingly
        nReturnCode := DINT_TO_DWORD(hr) AND 16#FFF;
        sReturnCode := DWORD_TO_HEXSTR(nReturnCode, 3, FALSE);
        sResultText := CONCAT('Returncode ', sReturnCode);
    END_CASE
END_IF

// Draw result image
hr := F_VN_PutTextExp(sResultText, ipImageRes, 50, 100, ETcVnFontType.TCVN_FT_HERSHEY_PLAIN, 4,
aColorGreen, 3, TCVN_LT_4_CONNECTED, FALSE, S_OK);
hr := F_VN_TransformIntoDisplayableImage(ipImageRes, ipImageResDisp, hr);
```

注意

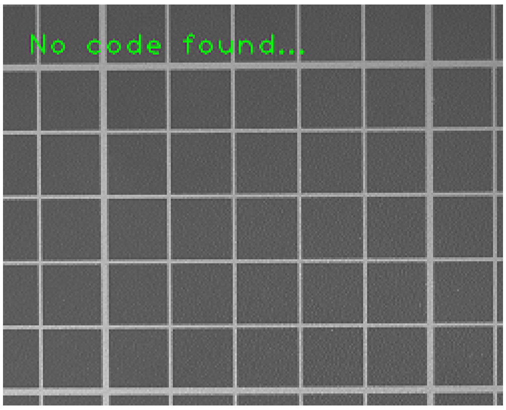
储备 hr

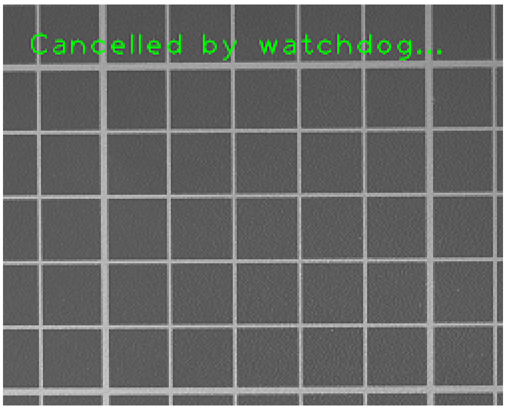
确保在调用读码函数和评估之间不要使用变量 hr 作为另一个函数的返回值。

结果

如果按照预期发现代码，则hr = S_OK，且可以输出结果。

如果无法找到预期的代码，返回代码是S_FALSE。如果执行被看门狗 [▶ 127]中止，则代码为16#256。可以输出相应的信息。





如果函数未正确执行，可能输出错误代码。这种情况在样本中没有出现，且只是由于不正确地使用读码函数而造成。

类似样本

- 数据矩阵码读取 [▶ 2648]
- EAN-13 条形码读取 [▶ 2645]
- 药品编码阅读 [▶ 2649]
- QR 码读取 [▶ 2651]

8.1.4 轮廓分析

8.1.4.1 匹配轮廓 1vsN（手动形状）

在这个样本中，使用函数F_VN_MatchContours1vsN [▶ 943]，将轮廓与参考轮廓进行比较。其中

- 参考轮廓是理想化的、手动创建的多边形

- 其他轮廓是从图像中提取出来的。

说明

函数F_VN_MatchContours [▶ 941]根据 Hu 矩比较样本中使用的两个轮廓。函数F_VN_MatchContours1vsN [▶ 943]代表一个扩展，可用于同时比较一个轮廓和其他几个轮廓。在这个样本中，这用于根据预定的参考形状对工件进行分类。

参考形状是理想化的、手动创建的多边形，包括等腰三角形、矩形/方形和梯形。由于只指定了多边形的角点，参考形状与测试轮廓在描述性轮廓点的数量上有很大不同。此外，它们在位置、方向、大小和轮廓内的干扰程度上也有不同。

函数F_VN_DetectBlobs [▶ 1139]用于搜索具有一定最小尺寸的轮廓，以便从图像中提取部件的形状。找到的轮廓然后用于描述的匹配。

输入参数

除了待比较轮廓外，只需向函数传递一个参数，描述用于确定 Hu 矩的异同的计算方法（eComparisonMethod，类型ETcVnContoursMatchComparisonMethod [▶ 172]）。方法TCVN_CMCM_CONTOURS_MATCH_I1计算各个特征的倒数值的差异之和，而方法TCVN_CMCM_CONTOURS_MATCH_I2计算纯特征的差异之和。与前两种方法不同，第三种方法TCVN_CMCM_CONTOURS_MATCH_I3只计算个体特征之间的最大差异。这三种方法中哪一种最适合于比较两个轮廓取决于使用情况。

变量

仅在本样本中的数据类型

CustomTcVnArray4_Point2_DINT 数据类型并不包含在 TwinCAT Vision 中，而是为该样本明确创建。执行方法可以在该样本的下载中找到。

```
// Indexes representing the different shapes
TRIANGLE                :   USINT := 0;
RECTANGLE               :   USINT := 1;
TRAPEZOID               :   USINT := 2;

// Images
ipImageIn               :   ITcVnImage;
ipImageInDisp           :   ITcVnDisplayableImage;
ipImageRes              :   ITcVnImage;
ipImageResDisp          :   ITcVnDisplayableImage;

// Contours
ipContourList           :   ITcVnContainer;
ipContour               :   ITcVnContainer;
ipIterator              :   ITcVnForwardIterator;
aShapeArrays            :   ARRAY [0..2] OF CustomTcVnArray4_Point2_DINT :=
                                [[[0,0],[40,0],[20, LREAL_TO_DINT(SQRT(1200))], [0,0]],  // Triangle
                                 [[0,0],[40,0],[40,40],[0,40]],                          // Rectangle
                                 [[0,0],[40,0],[60,22],[20,22]]];                        // Trapezoid;
aShapes                 :   ARRAY[0..2] OF ITcVnContainer;
ipShapes                :   ITcVnContainer;

// Matching
ipMatchIndexes          :   ITcVnContainer;
ipDissimilarities       :   ITcVnContainer;
aMatchIndexes           :   ARRAY [0..2] OF ULINT;
aDissimilarities        :   ARRAY [0..2] OF LREAL;

// Parameters
stBlobParams            :   TcVnParamsBlobDetection;
fThreshold              :   REAL := 170;
fMinArea                :   REAL := 10000;
fMaxDissimilarity       :   LREAL := 0.02;
eComparisonMethod       :   ETcVnContoursMatchComparisonMethod := TCVN_CMCM_CONTOURS_MATCH_I3;

// drawing
aColors                 :   ARRAY[0..2] OF TcVnVector4_LREAL :=
                                [[0, 175, 0, 0], [0, 0, 255, 0], [255, 0, 0, 0]]; // green, blue, red
aColorWhite             :   TcVnVector4_LREAL := [255, 255, 255];
aColorBlack             :   TcVnVector4_LREAL := [0, 0, 0];
aTexts                  :   ARRAY[0..2] OF STRING := ['Triangle', 'Rectangle', 'Trapezoid'];
sText                   :   STRING(255);
nTopLeftX               :   UDINT;
nTopLeftY               :   UDINT;
```

```
// Miscellaneous
aPixelValue           :   TcVnVector4_LREAL;
I                     :   USINT;
stBoundingRectangle   :   TcVnRectangle_UDINT;
aOffsets              :   ARRAY [0..2] OF TcVnPoint := [[20, 50], [20, 100], [20, 160]];
hr                    :   HRESULT;
```

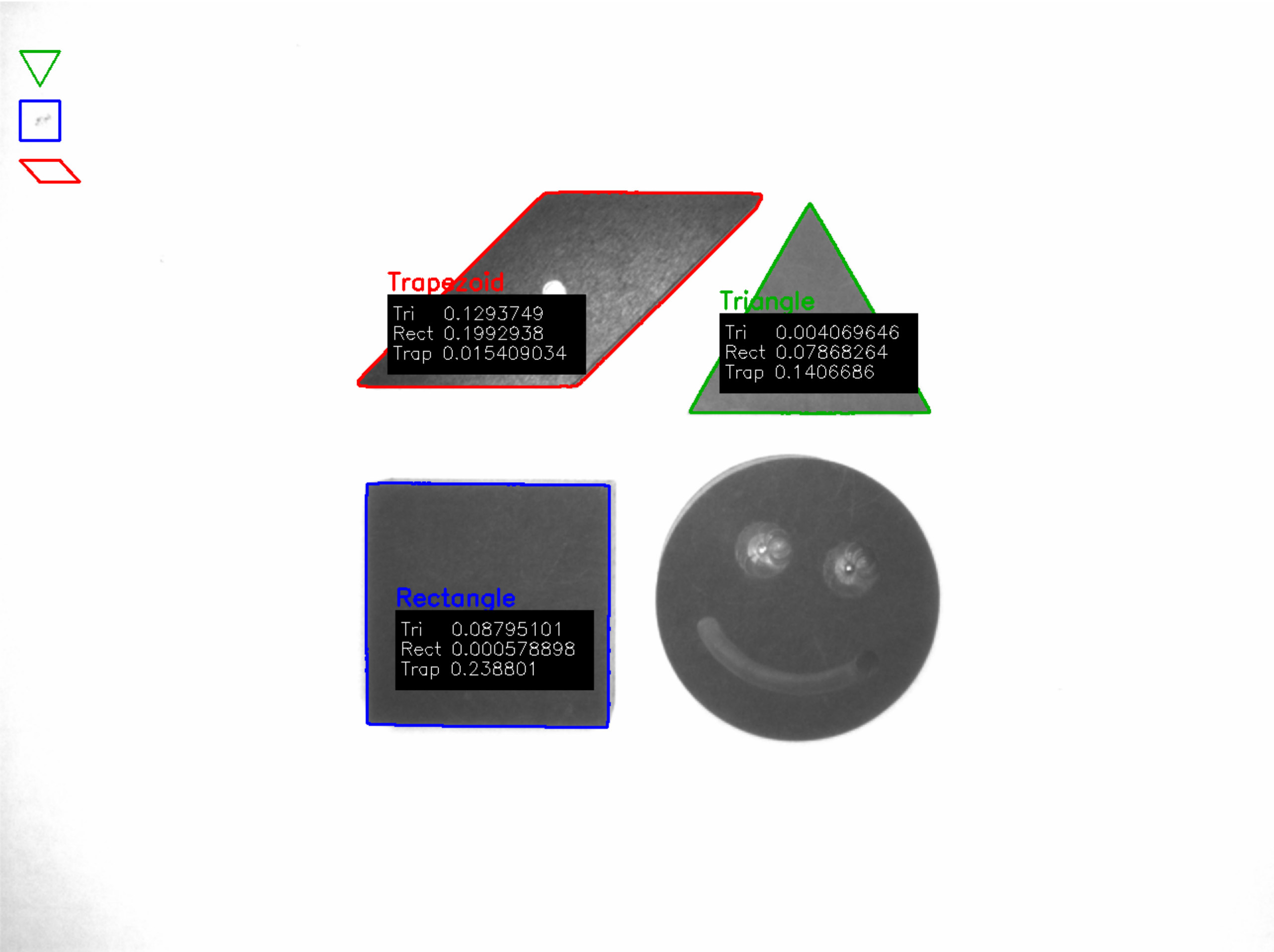
代码

```
// Fill manually defined shapes into one container
hr := F_VN_CreateContainer(ipShapes, ContainerType_Vector_Vector_TcVnPoint2_DINT, 0, hr);
FOR i:=TRIANGLE TO TRAPEZOID DO
    hr := F_VN_CreateContainerFromArray(ADR(aShapeArrays[i]), aShapes[i],
ContainerType_Vector_TcVnPoint2_DINT, 4, hr);
    hr := F_VN_InsertIntoContainer_ITcVnContainer(aShapes[i], ipShapes, i, hr);
END_FOR

// Prepare result image
hr := F_VN_ConvertColorSpace(ipImageIn, ipImageRes, TCVN_CST_GRAY_TO_RGB, hr);

// Check if background is light or dark
hr := F_VN_GetPixel(ipImageIn, aPixelValue, 50, 50, hr);
IF SUCCEEDED(hr) AND_THEN aPixelValue[0] < 128 THEN
    stBlobParams.eThresholdType := TCVN_TT_BINARY;
ELSE
    stBlobParams.eThresholdType := TCVN_TT_BINARY_INV;
END_IF

// Find contours in image
stBlobParams.bFilterByArea := TRUE;
stBlobParams.fMinArea := fMinArea;
stBlobParams.fMinThreshold := fThreshold;
hr := F_VN_DetectBlobs(ipImageIn, ipContourList, stBlobParams, hr);

// Iterate through all found contours
hr := F_VN_GetForwardIterator(ipContourList, ipIterator, hr);
WHILE hr = S_OK AND_THEN ipIterator.CheckIfEnd() <> S_OK DO
    hr := F_VN_GetContainer(ipIterator, ipContour, hr);
    hr := F_VN_IncrementIterator(ipIterator, hr);

    // Match the current contour with the reference shapes
    hr := F_VN_MatchContours1vsN(
        ipRefContour:= ipContour,
        ipContours:= ipShapes,
        ipMatchIndexes:= ipMatchIndexes,
        ipDissimilarities:= ipDissimilarities,
        fDissimilarityThreshold:= 100,
        eComparisonMethod:= eComparisonMethod,
        hrPrev:= hr
    );

    hr := F_VN_ExportContainer(ipMatchIndexes, ADR(aMatchIndexes), SIZEOF(aMatchIndexes), hr);
    hr := F_VN_ExportContainer(ipDissimilarities, ADR(aDissimilarities), SIZEOF(aDissimilarities),
hr);

    // Draw matching results
    IF aDissimilarities[0] < fMaxDissimilarity THEN

        // Calculate position of object
        hr := F_VN_UprightBoundingRectangle(ipContour, stBoundingRectangle, hr);
        nTopLeftX := LREAL_TO_UDINT(stBoundingRectangle.nX + 30);
        nTopLeftY := LREAL_TO_UDINT(stBoundingRectangle.nY + (stBoundingRectangle.nHeight / 2));

        // Draw matching result
        hr := F_VN_DrawContours(ipContour, -1, ipImageRes, aColors[aMatchIndexes[0]], 5, hr);
        hr := F_VN_PutTextExp(aTexts[aMatchIndexes[0]], ipImageRes, nTopLeftX, nTopLeftY,
TCVN_FT_HERSHEY_SIMPLEX, 0.8, aColors[aMatchIndexes[0]], 2, TCVN_LT_8_CONNECTED, FALSE, hr);
        hr := F_VN_DrawRectangle(nTopLeftX, nTopLeftY + 5, nTopLeftX + 200, nTopLeftY + 85,
ipImageRes, aColorBlack, -1, hr);
        // Draw all dissimilarity values
        FOR i:=0 TO 2 DO
            sText := CONCAT(LEFT(aTexts[aMatchIndexes[i]], 4), CONCAT(' ',
REAL_TO_STRING(LREAL_TO_REAL(aDissimilarities[i]))));
            hr := F_VN_PutTextExp(sText, ipImageRes, nTopLeftX + 5, nTopLeftY + 30 + 20*i,
TCVN_FT_HERSHEY_SIMPLEX, 0.6, aColorWhite, 1, TCVN_LT_8_CONNECTED, FALSE, hr);
        END_FOR

    END_IF
END_WHILE
```

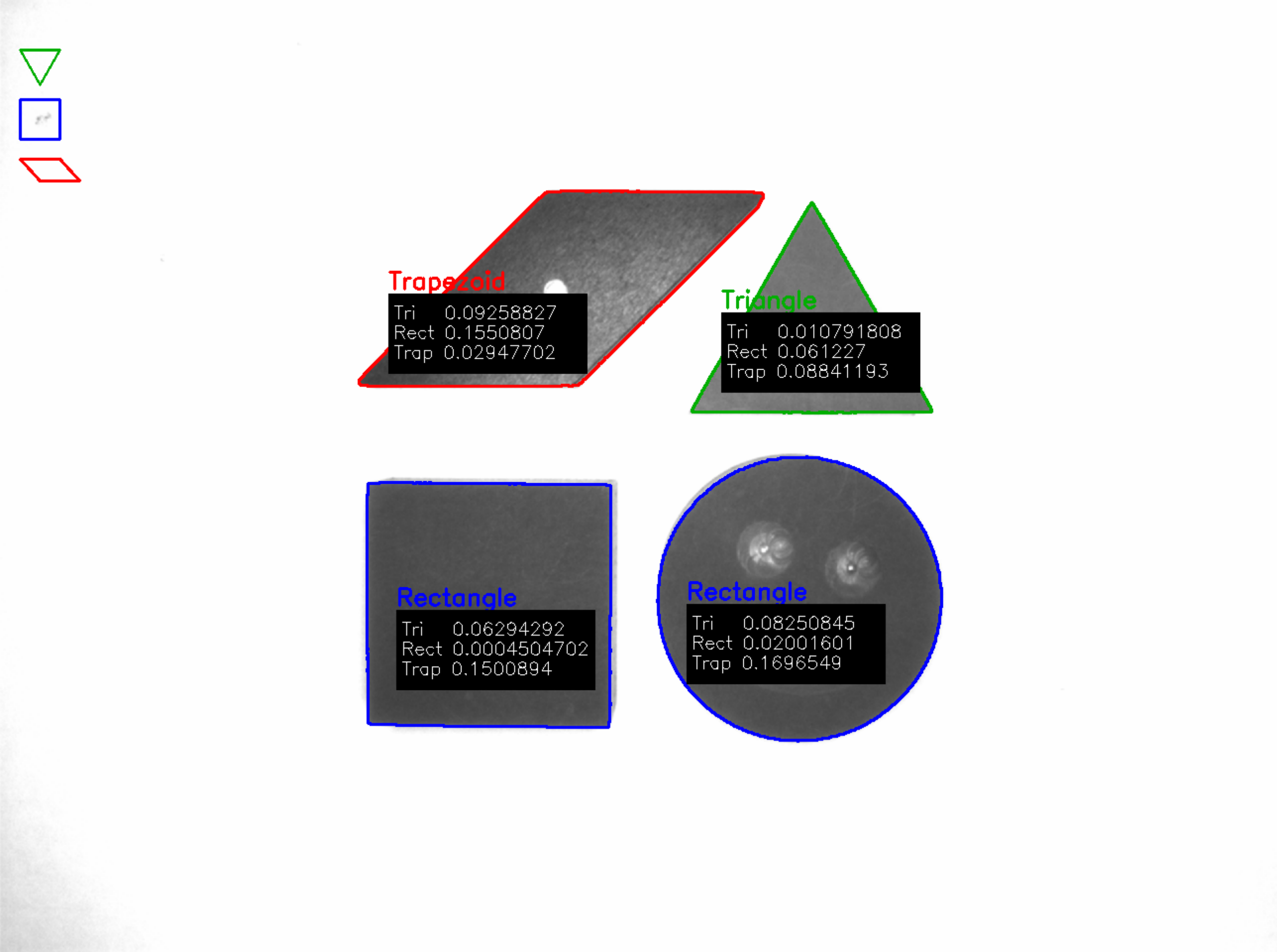
```
// Draw reference shapes
FOR i:=TRIANGLE TO TRAPEZOID DO
    hr := F_VN_DrawContoursExp(aShapes[i], -1, ipImageRes, aColors[i], 2, TCVN_LT_8_CONNECTED, 0, 0, 
aOffsets[i], hr);
END_FOR
hr := F_VN_TransformIntoDisplayableImage(ipImageRes, ipImageResDisp, hr);
hr := F_VN_TransformIntoDisplayableImage(ipImageIn, ipImageInDisp, hr);
```

结果

结果图像显示了参考形状（左边界）和找到的轮廓，这些轮廓与所有参考形状的相异度不超过一个最大值。此外，可视化显示了轮廓与哪个参考形状最相似。相应的描述文本（包括每个参考形状的相异度）被叠加在各自的轮廓上。对于本样本中使用的参数，MatchContours1.bmp 图像的结果如下：



在每种情况下，最低值表示正确的参考轮廓。如果参数 eComparisonMethod 从TCVN_CMCM_CONTOURS_MATCH_I3改为TCVN_CMCM_CONTOURS_MATCH_I2，结果会有很大变化。这也意味着必须调整最大接受相异度的参数：



新的数值允许对相应的参考形状进行更不明确的分配，且圆形组件被错误地分配给矩形参考形状。因此，这种计算方法显然不太适合这个样本。

类似样本

- 匹配轮廓（提取的形状） [▶ 2659]

8.1.4.2 匹配轮廓（提取的形状）

在这个样本中，使用函数F_VN_MatchContours [▶ 941]对两个轮廓进行比较。

说明

函数F_VN_MatchContours [▶ 941]根据 Hu 矩比较样本中使用的两个轮廓。在这个样本中，该功能用于在新的情况下识别一个特定的组件。为此，在参考图像中选择一个组件，保存其轮廓，然后与另一图像的轮廓进行比较。通过这种方式，与参考组件相比，低于一定相异度的组件被标记出来。

输入参数

除了待比较轮廓外，只需向函数传递一个参数，描述用于确定 Hu 矩的异同的计算方法（eComparisonMethod，类型ETcVnContoursMatchComparisonMethod [▶ 172]）。TCVN_CMCM_CONTOURS_MATCH_I1 方法计算各个特征的倒数值的差异之和，而TCVN_CMCM_CONTOURS_MATCH_I2 方法计算纯粹特征的差异之和。与前两种方法不同，第三种方法 TCVN_CMCM_CONTOURS_MATCH_I3 只计算各个特征之间的最大差异。这三种方法中哪一种最适合于比较两个轮廓取决于使用情况。

变量

```
hr                      :    HRESULT;

ipImageIn               :    ITcVnImage;
ipImageInDisp           :    ITcVnDisplayableImage;

ipImageRes              :    ITcVnImage;
```

```
ipImageResDisp          :   ITcVnDisplayableImage;
ipIterator              :   ITcVnForwardIterator;

// result
ipContourList           :   ITcVnContainer;
ipContourReference      :   ITcVnContainer;
ipContourCheck          :   ITcVnContainer;
fBestDissimilarity      :   LREAL;
fDissimilarity          :   LREAL;

// parameters
fThreshold              :   REAL := 170;
fMinArea                :   REAL := 10000;
fMaxDissimilarity       :   LREAL := 0.01;
eComparisonMethod       :   ETcVnContoursMatchComparisonMethod := TCVN_CMCM_CONTOURS_MATCH_I3;

// drawing
aColorGreen             :   TcVnVector4_LREAL := [0, 175, 0];
aColorBlue              :   TcVnVector4_LREAL := [0, 0, 255];
aColorRed               :   TcVnVector4_LREAL := [255, 0, 0];
aColorRes               :   TcVnVector4_LREAL;
sText                   :   STRING(255);
sTextReference          :   STRING(255) := 'Reference contour';
sTextCheck              :   STRING(255) := 'Check dissimilarity';

// other
bDarkBackground         :   BOOL;
nContours               :   ULINT;
nCounter                :   UINT := 0;
aPixelValue             :   TcVnVector4_LREAL;
stBoundingRectangle     :   TcVnRectangle_UDINT;
stParams                :   TcVnParamsBlobDetection;
```

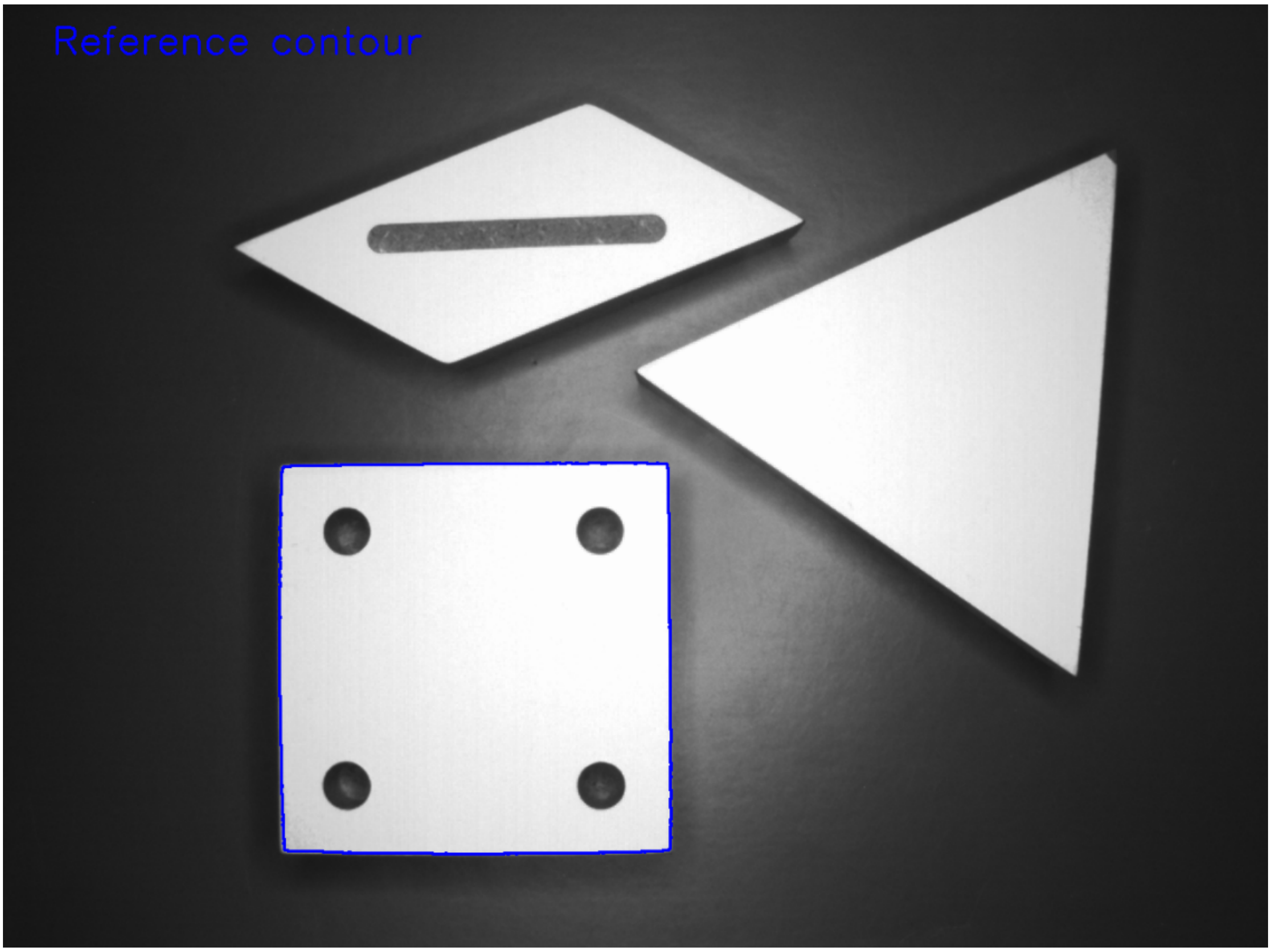
代码

```
// Prepare result image
hr := F_VN_ConvertColorSpace(ipImageIn, ipImageRes, TCVN_CST_GRAY_TO_RGB, hr);

// Check if background is dark in order to identify reference image
hr := F_VN_GetPixel(ipImageIn, aPixelValue, 50, 50, hr);
bDarkBackground := SUCCEEDED(hr) AND_THEN aPixelValue[0] < 128;

IF bDarkBackground THEN
    stParams.eThresholdType := TCVN_TT_BINARY;
ELSE
    stParams.eThresholdType := TCVN_TT_BINARY_INV;
END_IF

// Find contours in image
hr := F_VN_DetectBlobs(ipImageIn, ipContourList, stParams, hr);

// Distinguish reference and test image
IF bDarkBackground THEN

    // Select one of the found contours as reference for matching
    hr := F_VN_GetNumberOfElements(ipContourList, nContours, hr);
    hr := F_VN_GetAt_ITcVnContainer(ipContourList, ipContourReference, nCounter MOD nContours, hr);

    // Draw selected contour and text
    hr := F_VN_PutTextExp(sTextReference, ipImageRes, 50, 50, TCVN_FT_HERSHEY_SIMPLEX, 1.3,
aColorBlue, 2, TCVN_LT_8_CONNECTED, FALSE, hr);
    hr := F_VN_DrawContours(ipContourReference, -1, ipImageRes, aColorBlue, 5, hr);

    nCounter := nCounter + 1;

ELSE

    fBestDissimilarity := 10E300;

    // Iterate through all found contours
    hr := F_VN_GetForwardIterator(ipContourList, ipIterator, hr);
    WHILE hr = S_OK AND_THEN ipIterator.CheckIfEnd() <> S_OK DO
        hr := F_VN_GetContainer(ipIterator, ipContourCheck, hr);
        hr := F_VN_IncrementIterator(ipIterator, hr);

        // Match the current contour with the selected reference contour
        hr := F_VN_MatchContours(ipContourReference, ipContourCheck, eComparisonMethod,
fDissimilarity, hr);

        // Save best result
```

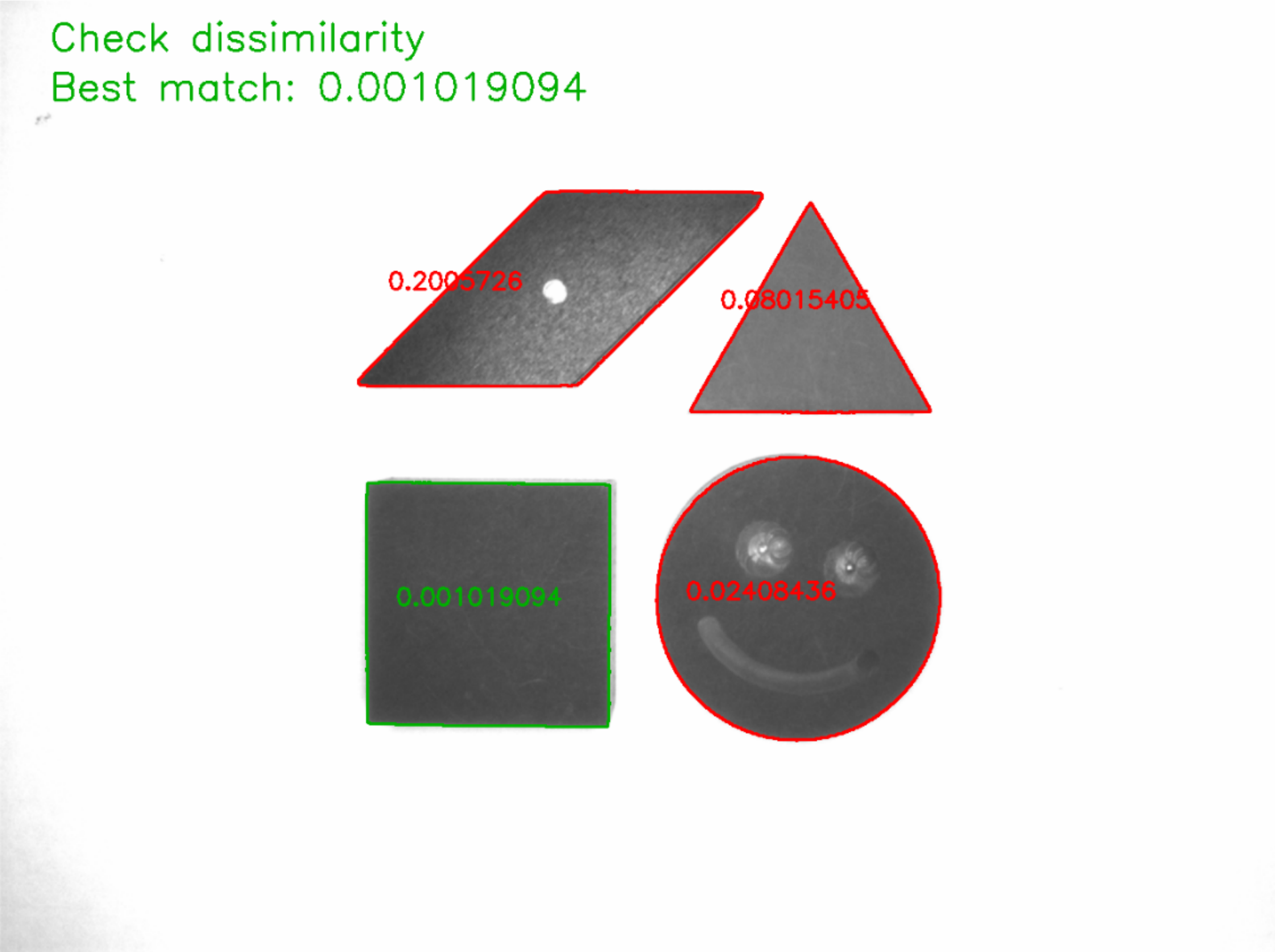
```
        IF fBestDissimilarity > fDissimilarity THEN
            fBestDissimilarity := fDissimilarity;
        END_IF

        // Choose action depending on the dissimilarity of both contours
        IF SUCCEEDED(hr) AND_THEN fDissimilarity < fMaxDissimilarity THEN
            aColorRes := aColorGreen;
        ELSE
            aColorRes := aColorRed;
        END_IF

        // Draw matching results
        sText := REAL_TO_STRING(LREAL_TO_REAL(fDissimilarity));
        hr := F_VN_UprightBoundingRectangle(ipContourCheck, stBoundingRectangle, hr);
        hr := F_VN_DrawContours(ipContourCheck, -1, ipImageRes, aColorRes, 5, hr);
        hr := F_VN_PutTextExp(sText, ipImageRes, LREAL_TO_UDINT(stBoundingRectangle.nX + 30),
LREAL_TO_UDINT(stBoundingRectangle.nY + (stBoundingRectangle.nHeight / 2)), TCVN_FT_HERSHEY_SIMPLEX,
0.8, aColorRes, 2, TCVN_LT_8_CONNECTED, FALSE, hr);

    END_WHILE

    // Draw text
    hr := F_VN_PutTextExp(sTextCheck, ipImageRes, 50, 50, TCVN_FT_HERSHEY_SIMPLEX, 1.3, aColorGreen,
2, TCVN_LT_8_CONNECTED, FALSE, hr);
    sText := CONCAT('Best match: ', REAL_TO_STRING(LREAL_TO_REAL(fBestDissimilarity)));
    hr := F_VN_PutTextExp(sText, ipImageRes, 50, 100, TCVN_FT_HERSHEY_SIMPLEX, 1.3, aColorGreen, 2,
TCVN_LT_8_CONNECTED, FALSE, hr);

END_IF

hr := F_VN_TransformIntoDisplayableImage(ipImageRes, ipImageResDisp, hr);
hr := F_VN_TransformIntoDisplayableImage(ipImageIn, ipImageInDisp, hr);
```

结果

当前选定的参考轮廓在参考图像上以蓝色标记：

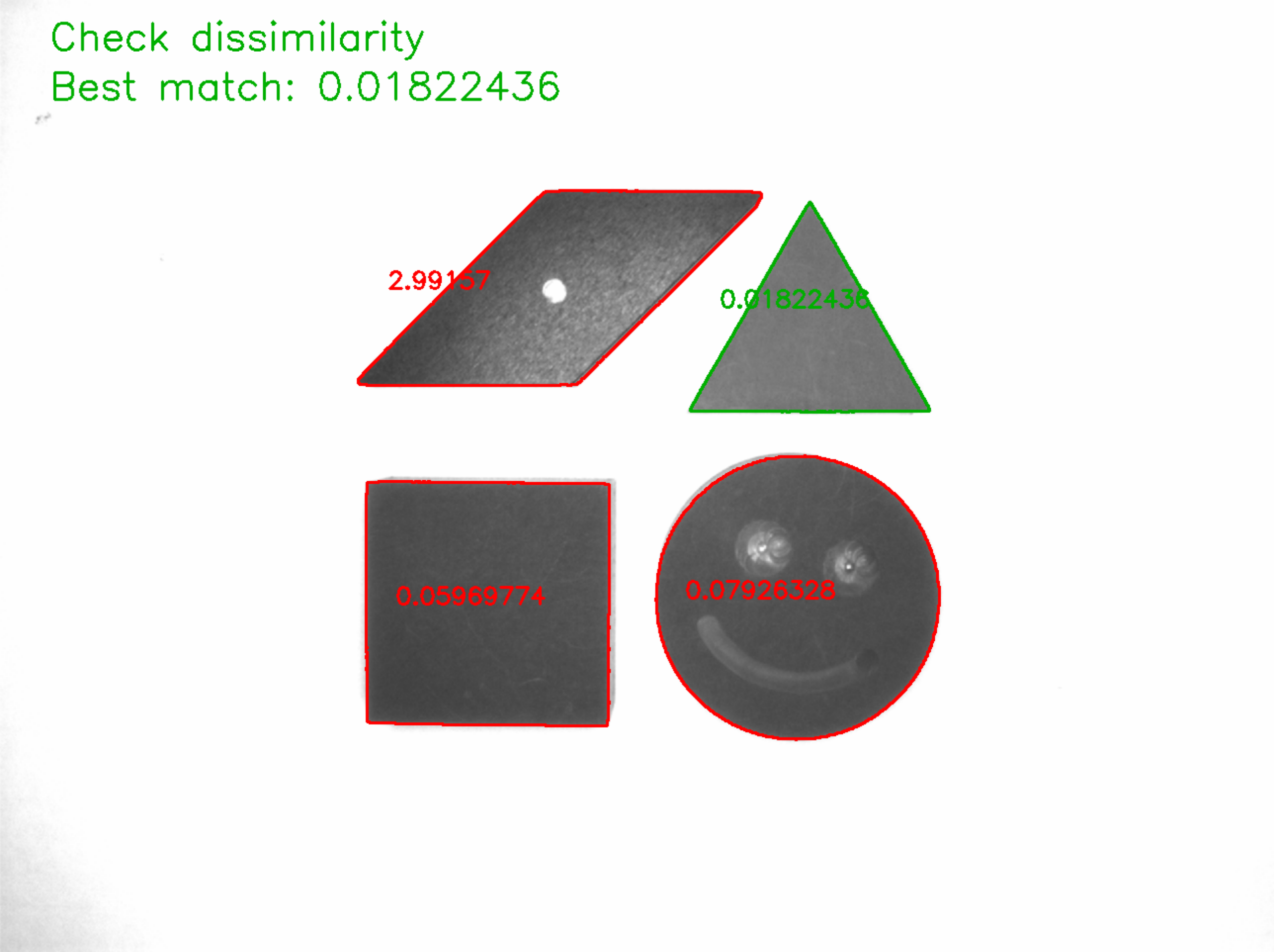


你可以看到轮廓因镜头变形而略有弯曲，并有一些小瑕疵。在测试图案中，相对于参考轮廓的相异度低于设定阈值的轮廓显示为绿色，其他轮廓显示为红色。此外，计算出的相异度被叠加在所有的轮廓上。下面的图示显示了当矩形被选为参考组件时*MatchContours1.bmp*输入图像的结果：



可以看出，到目前为止，与参考轮廓相比，矩形组件的相异度最小。如果参数 eComparisonMethod 从TCVN_CMCM_CONTOURS_MATCH_I3 改为TCVN_CMCM_CONTOURS_MATCH_I2，计算出的相异度也会相应改变：

在这种情况下，结果更加清晰，但这并不适用于所有形状。如果选择三角形作为参考组件，那么这个设置的结果就明显不如原始设置的清晰：



这一结果表明，不同的计算方法可以提供最好的结果，具体情况取决于应用。

类似样本

- 匹配轮廓 1vsN（手动形状） [▶ 2655]

8.1.5 图像分析

8.1.5.1 对象检测

8.1.5.1.1 带看门狗监控的斑点检测

本例程演示以下内容

- 函数 F_VN_DetectBlobs [▶ 1139] 用于查找图像中的圆圈结构。
- 函数的执行时间通过 Watchdog（看门狗）监控。

说明

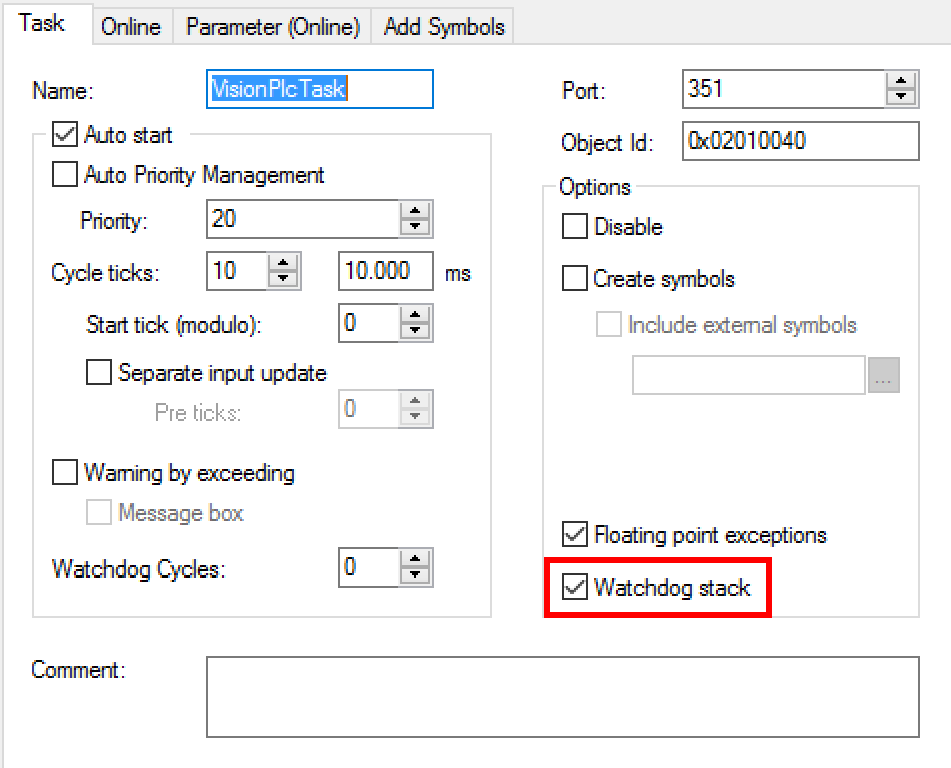
F_VN_DetectBlobs [▶ 1139] 函数便于在图像中搜索类似结构。它在一次调用中将设置阈值与轮廓搜索和滤波相结合。在本例程中，函数参数化为返回示例图像中的所有圆圈结构。

由于外部影响，实际图像内容可能与预期图像内容有很大偏差。对于执行时间取决于图像内容的图像处理算法来说，这会导致循环时间超限，从而产生不希望看到的副作用。为了防止出现这种情况，需要 Watchdog（看门狗）。Watchdog（看门狗）可以在指定时间后中止 API 中标有以下句子的函数：*“被 Watchdog（看门狗）取消时可返回部分结果”*。到此为止所获得的结果可以进一步使用。

F_VN_DetectBlobs 就是可以通过这种方式进行监控的函数之一。通过改变 Watchdog（看门狗）启动函数 F_VN_StartRelWatchdog [▸ 949] 中的允许执行时间 tStop，可以观察到函数终止后应用部分结果的情况，并与 F_VN_DetectBlobs 的完全执行情况进行比较。可以通过 F_VN_StopWatchdog [▸ 951] 停止 Watchdog（看门狗）。tRest 中的正返回值表示剩余时间，而 tRest 中的负返回值表示终止所需的额外时间。

任务配置

必须在执行的 PLC 任务时启用 Watchdog（看门狗）监控 [▸ 127]。否则，PLC 中的 Watchdog（看门狗）函数调用将被忽略，启动 Watchdog（看门狗）函数将返回返回值 16#71A: "No Interface"。



变量

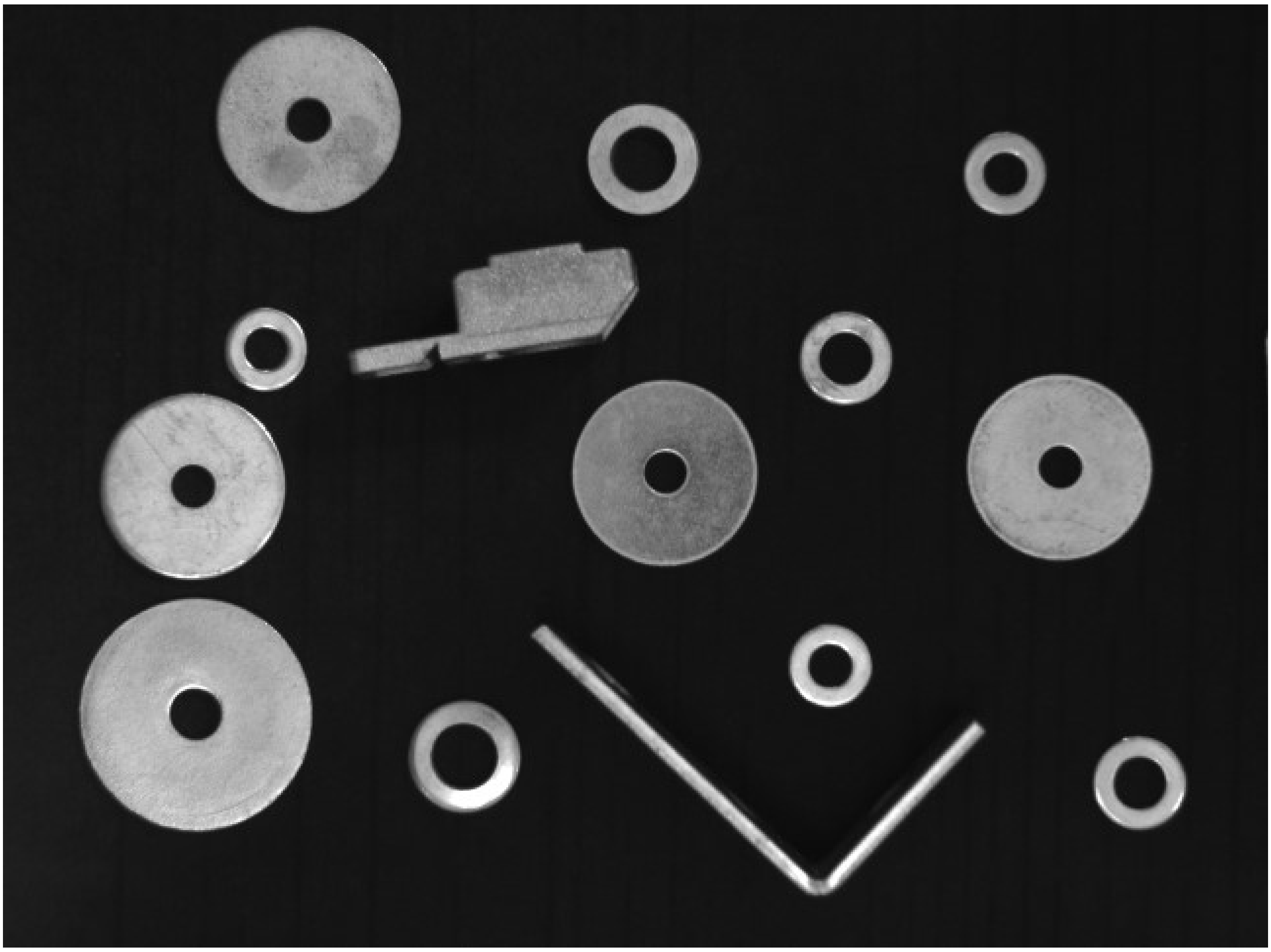
```
hr                  :   HRESULT;
hrFunc              :   HRESULT;

ipImageIn           :   ITcVnImage;
ipImageInDisp       :   ITcVnDisplayableImage;

ipImageRes          :   ITcVnImage;
ipImageResDisp      :   ITcVnDisplayableImage;

// Blob Detection
stBlobParams        :   TcVnParamsBlobDetection;
ipContourList       :   ITcVnContainer;

// Watchdog
hrWD                :   HRESULT;
tStop               :   DINT := 5000;
tRest               :   DINT;
nFraction           :   UDINT;

// Output
sText               :   STRING(255);

// Color
aColorGreen         :   TcVnVector4_LREAL := [0, 255, 0, 0];

// Image Infos
stPixelFormat       :   TcVnPixelFormat;
```

代码

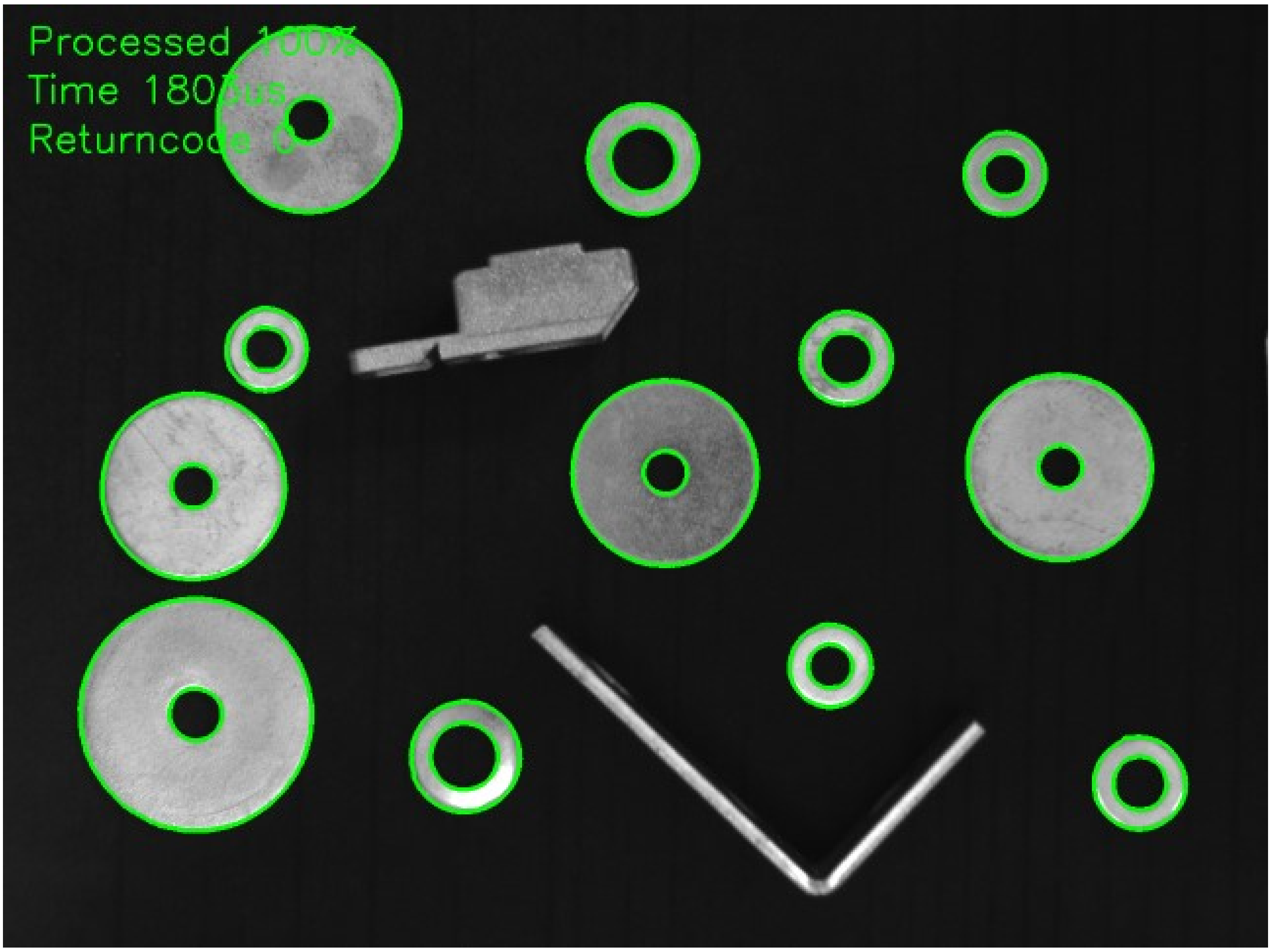
```
// Set Parameters
// ---------------------------------------------------------------
// - Set fMaxArea to 100000 detect all circles or set it to 10000
//   to detect only the small circles

     stBlobParams.bFilterByArea          := TRUE;
     stBlobParams.fMinArea               := 100;
     stBlobParams.fMaxArea               := 100000;

     stBlobParams.bFilterByCircularity   := TRUE;
     stBlobParams.fMinCircularity        := 0.80;

     stBlobParams.fMinThreshold          := 70;
     stBlobParams.fThresholdStep         := 0;

// Execute DetectBlobs-Function monitored by Watchdog-Function
// ---------------------------------------------------------------
// - Set the Watchdog stop time tStop from 5000us to 1000us to see
//   a Watchdog interrupt and that the interim results can be used

     hrWD  := F_VN_StartRelWatchdog(tStop, S_OK);
         hrFunc  := F_VN_DetectBlobs(
                                ipSrcImage      := ipImageIn,
                                ipBlobContours  := ipContourList,
                                stParams        := stBlobParams,
                                hrPrev          := hr);
     hrWD  := F_VN_StopWatchdog(hrWD, tRest => tRest, nFractionProcessed => nFraction);

// Draw Result Image
// ---------------------------------------------------------------
hr := F_VN_DrawContours(ipContourList, -1, ipImageRes, aColorGreen, 3, hr);

sText := CONCAT(CONCAT('Processed ', UDINT_TO_STRING(nFraction)), '%');
hr := F_VN_PutTextExp(sText, ipImageRes, 25, 50, TCVN_FT_HERSHEY_SIMPLEX, 1.3, aColorGreen, 2,
TCVN_LT_8_CONNECTED, FALSE, hr);

sText := CONCAT(CONCAT('Time ', DINT_TO_STRING(tStop - tRest)), 'us');
hr := F_VN_PutTextExp(sText, ipImageRes, 25, 100, TCVN_FT_HERSHEY_SIMPLEX, 1.3, aColorGreen, 2,
TCVN_LT_8_CONNECTED, FALSE,hr);

sText := CONCAT('Returncode ', DINT_TO_STRING(hrFunc));
hr := F_VN_PutTextExp(sText, ipImageRes, 25, 150, TCVN_FT_HERSHEY_SIMPLEX, 1.3, aColorGreen, 2,
TCVN_LT_8_CONNECTED, FALSE,hr);
```

结果

输入图像

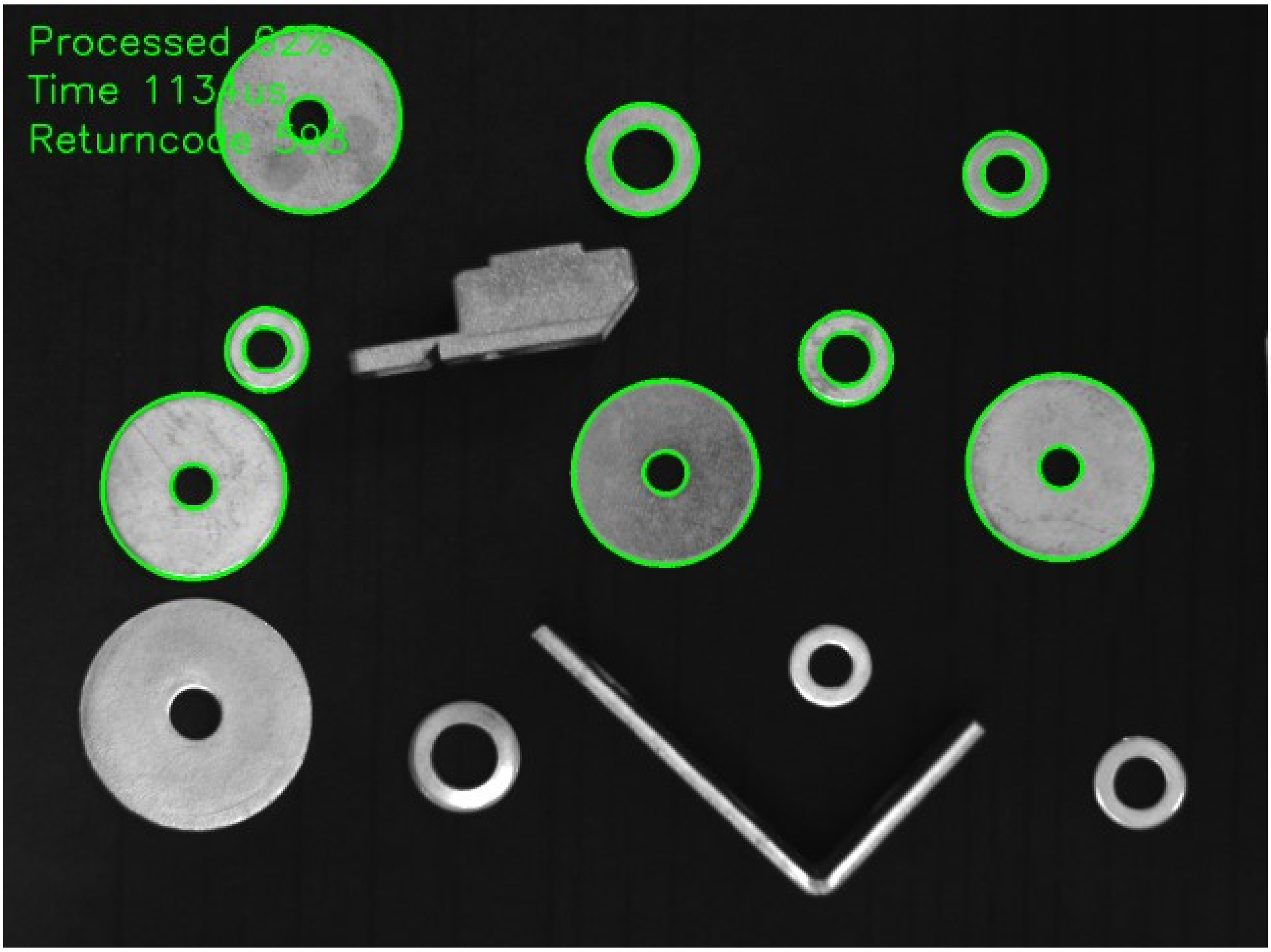


在图像中发现了所有圆圈，并对例程代码进行了参数设置。



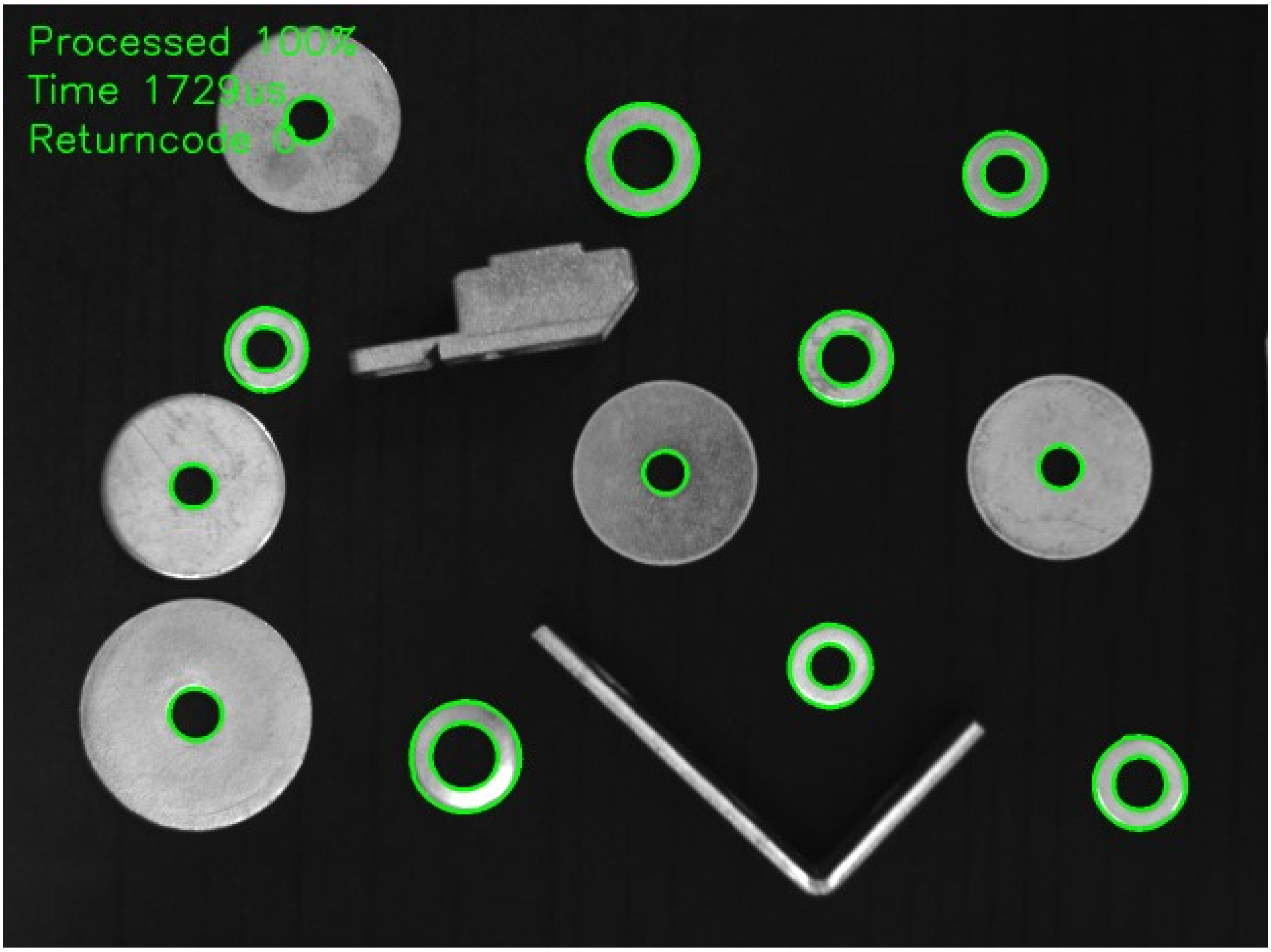
通过在 1 ms 后触发 Watchdog（看门狗），在图像中发现了一些圆圈。由于中止过程需要一些时间，因此实际所需时间大于 1 ms。

```
hrWD := F_VN_StartRelWatchdog(tStop:=1000, WATCHDOG_ACCUMULATION_TYPE_MEAN, S_OK);
```



通过缩小以像素为单位的最大封闭区域，只在图像中找到了小圆圈。

```
stBlobParams.fMinArea := 100;
stBlobParams.fMaxArea := 10000;
```



类似例程

- 查找轮廓而不是斑点检测 [▶ 2670]

8.1.5.1.2 查找轮廓而不是斑点检测

本例程展示了使用 F_VN_FindContoursExp [▶ 1149] 的斑点检测样本 [▶ 2664]的替代解决方案。

说明

函数 F_VN_DetectBlobs [▶ 1139] 将许多单独的图像处理步骤整合到一个函数调用中。不过，内部处理步骤也可以通过单独的函数调用来完成。这样可以更容易地设置参数，因为可以看到中间结果。因此，在该例程中，使用了与斑点检测样本 [▶ 2664]中相同的图像，并搜索了相同的圆圈结构。

变量

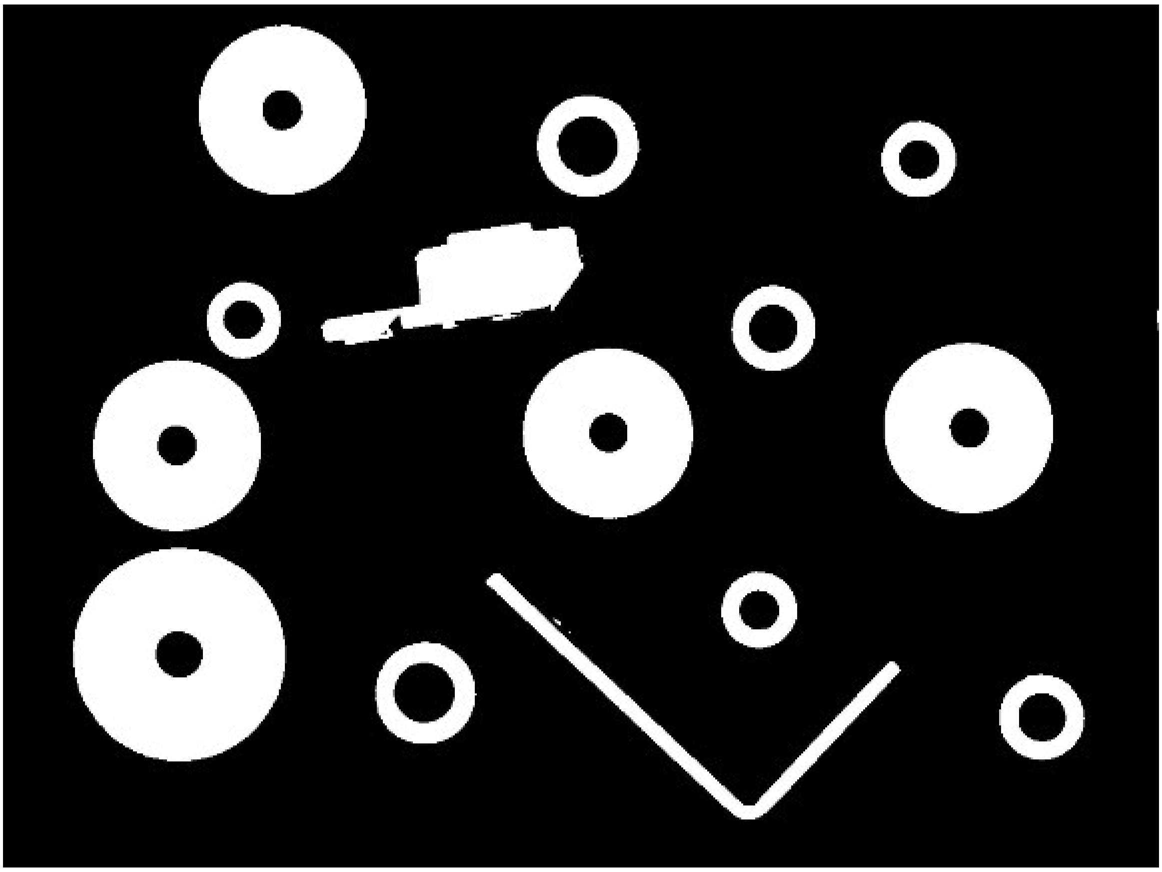
```
hr                      : HRESULT;

ipImageIn               : ITcVnImage;
ipImageInDisp           : ITcVnDisplayableImage;
ipImageWork             : ITcVnImage;
ipImageThresholdDisp    : ITcVnDisplayableImage;
ipImageRes              : ITcVnImage;
ipImageResDisp          : ITcVnDisplayableImage;

// Sample Specific Variables
ipContourList           : ITcVnContainer;
ipContourResultList     : ITcVnContainer;
ipContour               : ITcVnContainer;
ipIterator              : ITcVnForwardIterator;
aOffset                 : TcVnPoint;

fThreshold              : LREAL := 70;

fArea                   : LREAL;
```

```
fAreaMin                : LREAL := 100;
fAreaMax                : LREAL := 100000;

fCircularity            : LREAL;
fCircularityMin         : LREAL := 0.8;

// Image Infos
stPixelFormat           : TcVnPixelFormat;

// COLORS
aColorGreen             : TcVnVector4_LREAL := [0, 255, 0];
```

代码

图像分割预处理

```
// Image Segementation
hr := F_VN_Threshold(ipImageIn, ipImageWork, fThreshold, 255, TCVN_TT_Binary, hr);
```



轮廓搜索

```
// Find Contours
hr := F_VN_FindContoursExp(
ipSrcImage              := ipImageWork,
ipContours              := ipContourList,
eRetrievalMode          := ETcVnContourRetrievalmode.TCVN_CRM_LIST,
eApproximationMethod    := ETcVnContourApproximationMethod.TCVN_CAM_SIMPLE,
aOffset                 := aOffset,
hrPrev                  := hr);
```

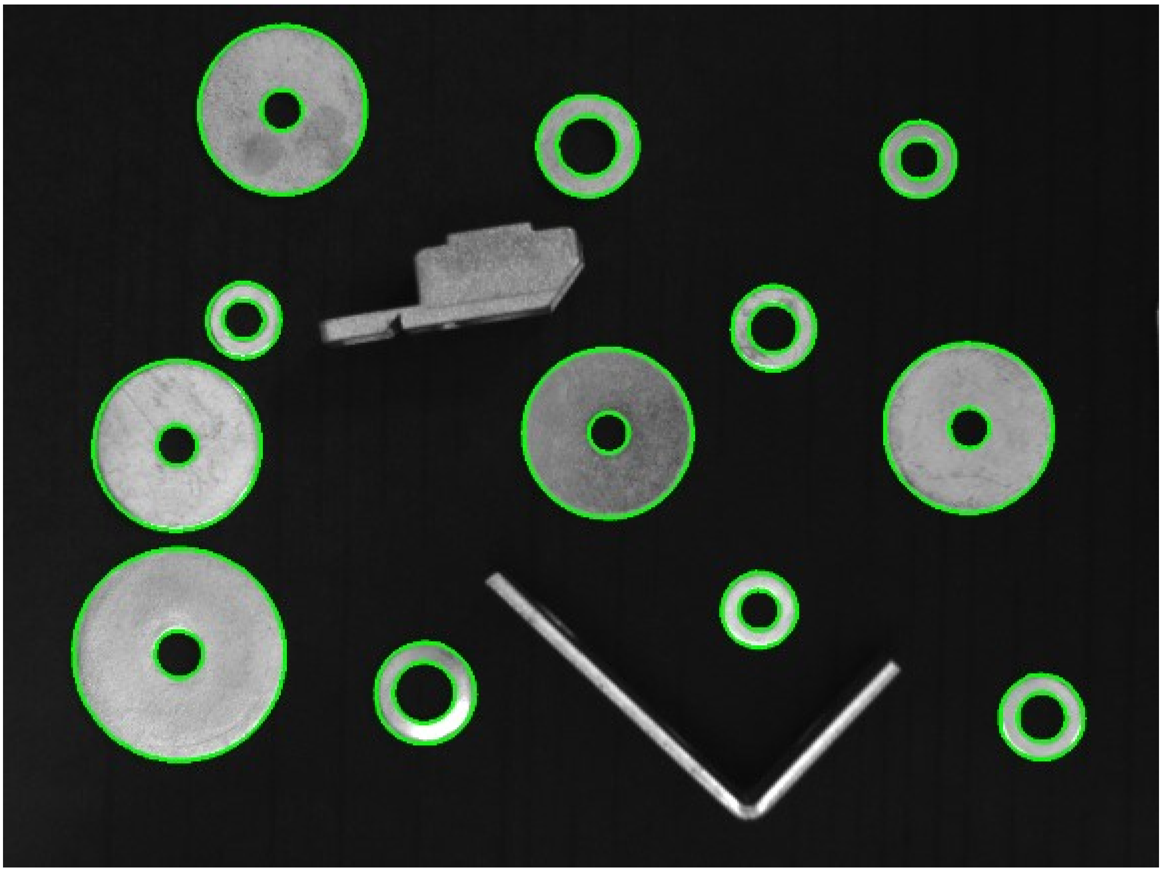
根据周边区域以像素和圆度对轮廓进行滤波

```
// Filter Contours
hr := F_VN_GetForwardIterator(ipContourList, ipIterator, hr);
hr := F_VN_CreateContainer(ipContourResultList, ContainerType_Vector_Vector_TcVnPoint2_DINT, 0, hr);

WHILE SUCCEEDED(hr) AND_THEN ipIterator.CheckIfEnd() <> S_OK DO

hr := F_VN_GetContainer(ipIterator, ipContour, hr);
hr := F_VN_IncrementIterator(ipIterator, hr);

// Filter by Area
hr := F_VN_ContourArea(ipContour, fArea, hr);
```

```
IF fArea > fAreaMin AND fArea < fAreaMax THEN

// Filter by Circularity
hr := F_VN_ContourCircularity(ipContour, fCircularity, hr);

IF fCircularity > fCircularityMin THEN
// Add contour to the result contour container
hr := F_VN_AppendToContainer_ITcVnContainer(ipContour, ipContourResultList, hr);

END_IF
END_IF
END_WHILE
```

结果输出

```
// Draw contours into the result image and display it
hr := F_VN_DrawContours(ipContourResultList, -1, ipImageRes, aColorGreen, 3, hr);
hr := F_VN_TransformIntoDisplayableImage(ipImageRes, ipImageResDisp, hr);
```



类似例程

- 带看门狗监控的斑点检测 [▶ 2664]

8.1.5.1.3 查找轮廓 - 近似法

这个样本涉及轮廓搜索的可能近似模式（eApproximationMethod，类型ETcVnContourApproximationMethod [▶ 167]）。

说明

函数F_VN_FindContours [▶ 1147]、F_VN_FindContoursExp [▶ 1149]和F_VN_FindContourByHierarchy [▶ 1143]用于检测图像中的轮廓。轮廓通过点进行描述。根据所选择的近似模式（eApproximationMethod，类型ETcVnContourApproximationMethod [▶ 167]），将返回每个轮廓的所有点或简化的点集：

- TCVN_CAM_NONE
  返回轮廓的所有点。
- TCVN_CAM_SIMPLE
  应用无损压缩。

- TCVN_CAM_TC89_L1 oder TCVN_CAM_TC89_KCOS
  这两个函数都应用 Teh-Chin 链近似算法的一个变体

变量

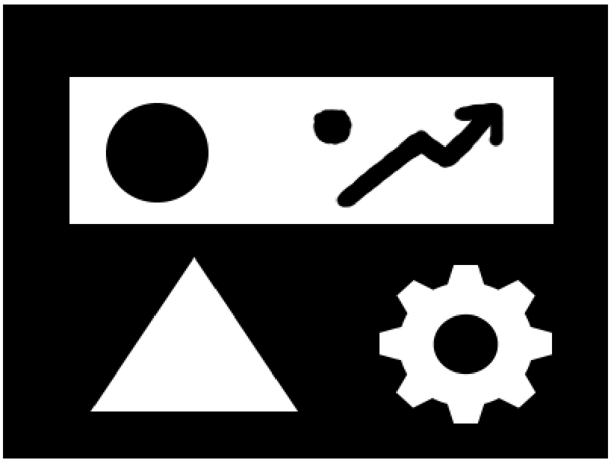
```
hr                       :   HRESULT;

ipImageIn                :   ITcVnImage;
ipImageInDisp            :   ITcVnDisplayableImage;

ipImageRes               :   ITcVnImage;
ipImageResDisp           :   ITcVnDisplayableImage;

// Sample Specific Variables
ipContourList            :   ITcVnContainer;
ipContour                :   ITcVnContainer;
ipIterator               :   ITcVnForwardIterator;
eRetrievalMode           :   ETcVnContourRetrievalMode := TCVN_CRM_LIST;
eApproximationMethod : ETcVnContourApproximationMethod := TCVN_CAM_SIMPLE;
aOffset                  :   TcVnPoint;

// Colors
aColorRed                :   TcVnVector4_LREAL := [255, 0, 0];
```

代码

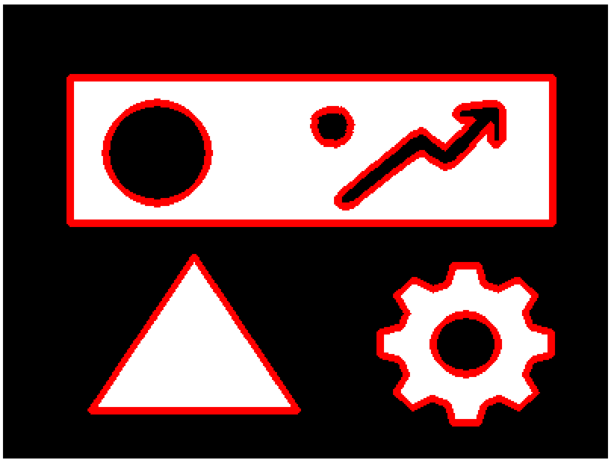
```
// Create Result Image
hr := F_VN_ConvertColorSpace(ipImageIn, ipImageRes, TCVN_CST_Gray_TO_RGB, hr);

// Find Contours and their Hierarchy
// -----------------------------------
hr := F_VN_FindContoursExp(
                 ipSrcImage              :=   ipImageIn,
                 ipContours              :=   ipContourList,
                 eRetrievalMode          :=   eRetrievalMode,
                 eApproximationMethod    :=   eApproximationMethod,
                 aOffset                 :=   aOffset,
                 hrPrev                  :=   hr);

// Draw the points of the contour
hr := F_VN_GetForwardIterator(ipContourList, ipIterator, hr);

WHILE SUCCEEDED(hr) AND_THEN ipIterator.CheckIfEnd() <> S_OK DO
   hr := F_VN_GetContainer(ipIterator, ipContour, hr);
   hr := F_VN_IncrementIterator(ipIterator, hr);
   hr := F_VN_DrawPoints(ipContour, ipImageRes, ETcVnDrawShape.TCVN_DS_CIRCLE, aColorRed, hr);
END_WHILE
```

结果

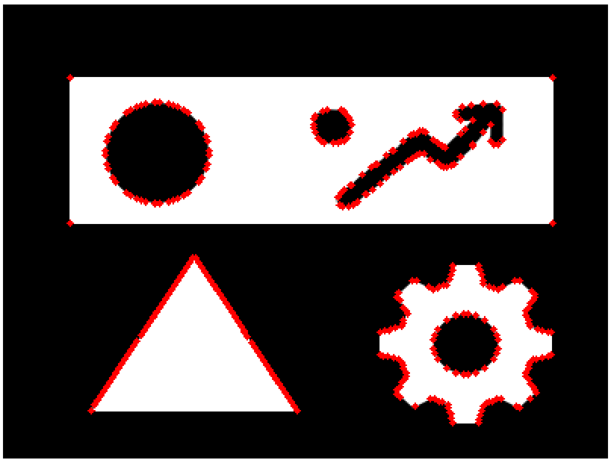
输入图像



根据ETcVnContourApproximationMethod [▸ 167].TCVN_CAM_NONE 绘制包括所有点的结果图像



根据ETcVnContourApproximationMethod [▶ 167].TCVN_CAM_TC89_L1 绘制包括所有点的结果图像



类似样本

- 查找轮廓 – 层次结构和检索模式 [▶ 2674]
- 查找轮廓而不是斑点检测 [▶ 2670]

8.1.5.1.4 查找轮廓 – 层次结构和检索模式

本样本显示了轮廓搜索（F_VN_FindContoursExp [▶ 1149]和F_VN_FindContourHierarchyExp [▶ 1143]）的检索模式（ETcVnContourRetrievalMode [▶ 170]）以及它如何影响轮廓结果和返回的层次结构。

说明

轮廓检索模式（eRetrievalMode，类型ETcVnContourRetrievalMode [▶ 170]）可用于指定在轮廓搜索期间如何考虑轮廓层次结构。该样本根据所选模式将结果可视化。

关于层次结构的详细解释，可查看F_VN_FindContourHierarchyExp [▶ 1143]。

变量

```
hr                     :   HRESULT;

ipImageIn              :   ITcVnImage;
ipImageInDisp          :   ITcVnDisplayableImage;

ipImageRes             :   ITcVnImage;
ipImageResDisp         :   ITcVnDisplayableImage;

ipImageHierarchy       :   ITcVnImage;
ipImageHierarchyDisp   :   ITcVnDisplayableImage;

// Sample Specific Variables
```

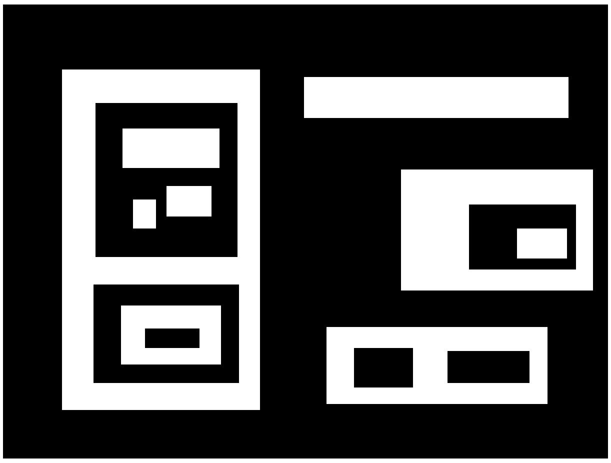
```
ipContourList         :   ITcVnContainer;
ipContour             :   ITcVnContainer;
ipHierarchyList       :   ITcVnContainer;
aHierarchy            :   TcVnVector4_DINT;
aHierarchyTree        :   TcVnVector4_DINT;
eRetrievalMode        :   ETcVnContourRetrievalMode := TCVN_CRM_LIST;
eApproximationMethod :   ETcVnContourApproximationMethod := TCVN_CAM_SIMPLE;
aOffset               :   TcVnPoint;

nNumOfElem            :   ULINT;
nParents              :   UINT;
i                     :   ULINT;
j                     :   ULINT;
stRect                :   TcVnRectangle_UDINT;
sText                 :   STRING(80);

// Colors
aColorWhite           :   TcVnVector4_LREAL := [255, 255, 255];
aColorBlack           :   TcVnVector4_LREAL := [0, 0, 0];
aColorList            :   ARRAY[0..4] OF TcVnVector4_LREAL := [ [255,0,0], [0,255,0], [0,0,255], [200
,200,0], [200,0,200]];
```

代码

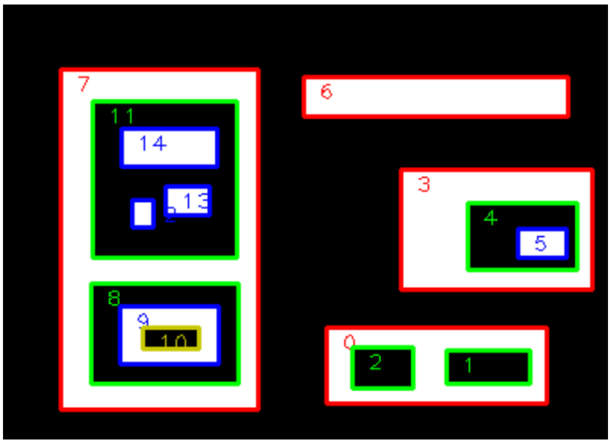
```
// Create Result Image
hr := F_VN_ConvertColorSpace(ipImageIn, ipImageRes, TCVN_CST_Gray_TO_RGB, hr);
hr := F_VN_CreateImage(ipImageHierarchy, 130, 200, EtcVnElementType.TCVN_ET_USINT, 3, hr);
hr := F_VN_SetPixels(ipImageHierarchy, aColorWhite, hr);

// Find Contours and their Hierarchy
// -----------------------------------
hr := F_VN_FindContourHierarchyExp(
               ipSrcImage               :=   ipImageIn,
               ipContours               :=   ipContourList,
               ipHierarchy              :=   ipHierarchyList,
               eRetrievalMode           :=   eRetrievalMode,
               eApproximationMethod     :=   eApproximationMethod,
               aOffset                  :=   aOffset,
               hrPrev                   :=   hr);


hr := F_VN_GetNumberOfElements(ipContourList, nNumOfElem, hr);

IF nNumOfElem > 0 THEN
   FOR i:= 0 TO (nNumOfElem-1) BY 1 DO

      hr := F_VN_GetAt_ITcVnContainer(ipContourList, ipContour, i, hr);
      hr := F_VN_GetAt_TcVnVector4_DINT(ipHierarchyList, aHierarchy, i, hr);

      // Count Parents and set aColor depending on parent number
      nParents := 0;
      aHierarchyTree := aHierarchy;
      WHILE SUCCEEDED(hr) AND_THEN aHierarchyTree[3] >= 0 DO
         nParents := nParents + 1;
         hr := F_VN_GetAt_TcVnVector4_DINT(ipHierarchyList, aHierarchyTree, aHierarchyTree[3], hr);
      END_WHILE

      // Draw contour and number into the result image
      sText := TO_STRING(i);
      hr := F_VN_UprightBoundingRectangle(ipContour, stRect, hr);
      hr := F_VN_PutText(sText, ipImageRes, stRect.nX +10, stRect.nY +15, TCVN_FT_HERSHEY_PLAIN, 1,
aColorList[(nParents) MOD 4], hr);
      hr := F_VN_DrawContours(ipContour, -1, ipImageRes, aColorList[(nParents) MOD 4], 2, hr);

      // Write Hierarchy
      hr := F_VN_PutText(sText, ipImageHierarchy, 10, TO_UDINT(10 + i*10), TCVN_FT_HERSHEY_PLAIN,
0.5, aColorList[(nParents) MOD 4], hr);
      FOR j := 0 TO 3 BY 1 DO
         sText := TO_STRING(aHierarchy[j]);
         hr := F_VN_PutText(sText, ipImageHierarchy, TO_UDINT(10 + (j+1)*22), TO_UDINT(10 + i*10),
TCVN_FT_HERSHEY_PLAIN, 0.5, aColorBlack, hr);
      END_FOR
   END_FOR
END_IF
```

结果

输入图像



通过 TCVN_CRM_TREE 发现的轮廓。颜色在同一水平上呈现轮廓



上述结果图像的层次结构。每一行指的是一个轮廓。这些栏目有以下含义：

- 轮廓指数（“彩色柱”）
- 同一层次结构上的下一个轮廓的索引
- 同一层次上的前一个轮廓的索引
- 第一个子轮廓的指数
- 父轮廓的指数

数字 -1 表示不存在相应的轮廓。

0	3	-1	1	-1
1	2	-1	-1	0
2	-1	1	-1	0
3	6	0	4	-1
4	-1	-1	5	3
5	-1	-1	-1	4
6	7	3	-1	-1
7	-1	6	8	-1
8	11	-1	9	7
9	-1	-1	10	8
10	-1	-1	-1	9
11	-1	8	12	7
12	13	-1	-1	11
13	14	12	-1	11
14	-1	13	-1	11

类似样本

- 查找轮廓 - 近似法 [▶ 2672]
- 查找轮廓而不是斑点检测 [▶ 2670]

8.1.5.1.5 模板匹配和评估

在这个样本中，将执行模板匹配，以便找到印刷电路板上的组件。为此，使用：

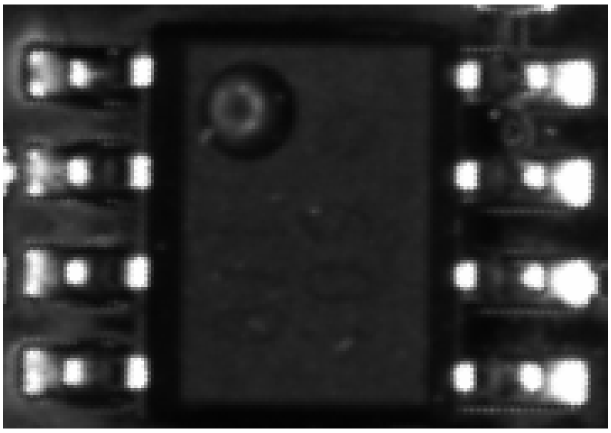
- F_VN_MatchTemplateAndEvaluate [▶ 1160]
- F_VN_DrawRectangle [▶ 996]

应用

记住把输入图像和模板图像都添加到各自的文件源中。通过阈值fMatchThreshold，可以设置一个限制，即为了被接受，匹配必须多好。

程序

在这个样本中重要的是，除了正常的输入图像，还需要一个模板图像，以便在输入图像中搜索模板的出现情况。



如果两个图像都可用，则使用函数F_VN_MatchTemplateAndEvaluate [▶ 1160]自动执行模板匹配。

```
hr := F_VN_MatchTemplateAndEvaluate(ipImageIn, ipImageTemplate, ipMatches, fMatchThreshold, hr);
```

找到的相关程度高于指定阈值fMatchThreshold的匹配物现在被定位为容器ipMatches中的点。每个点指定了ipImageTemplate中每次ipImageIn出现的左上角。

原则上，你现在已经完成了模板匹配的工作。举例来说，访问下面的单个匹配，以便将它们画到图像中。为此，使用访问容器元素的结构，如容器 [▶ 132]一节中所述。

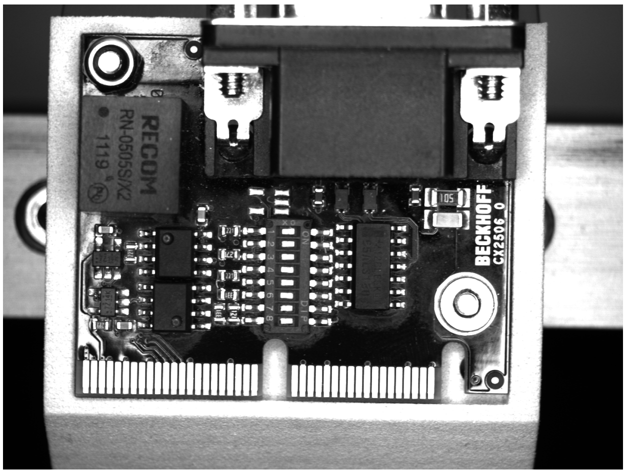
```
hr := F_VN_GetImageHeight(ipImageTemplate, nHeight, hr);
hr := F_VN_GetImageWidth(ipImageTemplate, nWidth, hr);
hr := F_VN_GetAt_TcVnPoint2_DINT(ipMatches, aPosition, 0, hr);
hr := F_VN_GetForwardIterator(ipMatches, ipIterator, hr);
IF SUCCEEDED(hr) AND ipIterator <> 0 THEN
    hr := ipIterator.TcQueryInterface(IID_ITcVnAccess_TcVnPoint2_DINT, ADR(ipAccess));
    IF SUCCEEDED(hr) AND ipAccess <> 0 THEN
        WHILE SUCCEEDED(hr) AND ipIterator.CheckIfEnd() <> S_OK DO
            hr := ipAccess.Get(aPosition);
            hr := F_VN_DrawRectangle(
                DINT_TO_UDINT(aPosition[0]),
                DINT_TO_UDINT(aPosition[1]),
                DINT_TO_UDINT(aPosition[0])+nWidth,
                DINT_TO_UDINT(aPosition[1])+nHeight,
                ipImageRes,
                aGreen,
                5,
                hr
            );
            hr := F_VN_IncrementIterator(ipIterator, hr);
        END_WHILE
    END_IF
END_IF
```

注意

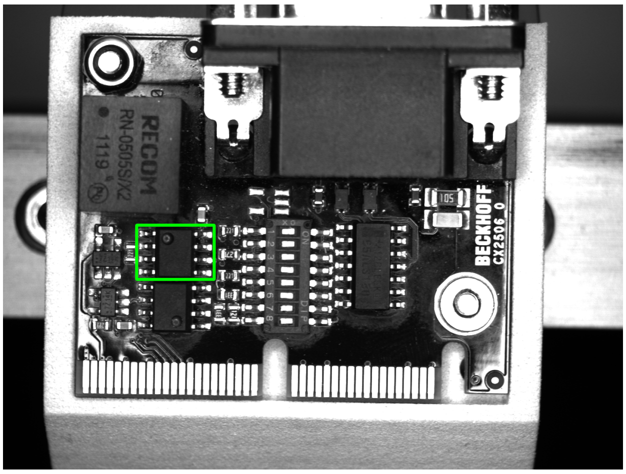
执行时间长

根据输入和模板图像的大小，与其他函数相比，执行模板匹配可能需要很长的时间。确保设置了相应的周期时间，且必要时使用看门狗！

原始图像



结果图像



8.1.5.2 任何类型的形状的平均强度

在本例程中，图像的平均强度在人工定义的图像区域内计算。为此，使用了一个圆形图像掩膜 [▸ 136]。

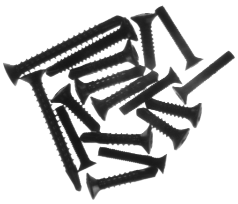
主要使用以下函数：

- F_VN_ImageAverageExp [▸ 1178]
- F_VN_DrawCircle [▸ 959]

说明

统计学图像特征 [▸ 2681] 例程显示如何计算统计图像特征（如平均值、最大值和最小值）。然而，这些特征往往不是针对整个图像计算，而只是针对特定图像区域计算。如果这些区域是矩形，设置 ROI [▸ 137] 会有所帮助。您必须使用图像掩膜 [▸ 136]才能将计算限制在任何形状的区域内。这些本质上是图像，其强度值描述了原始图像中哪些像素需要考虑。因此，掩膜可以假定为任何形状；本例程显示的是一个圆形掩膜。

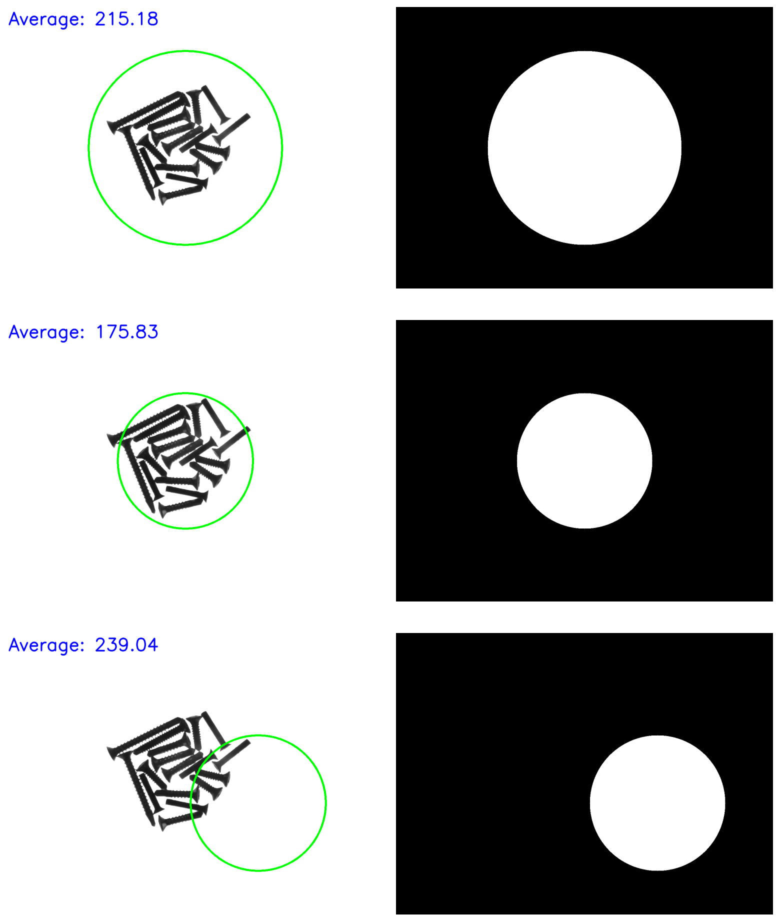
对于该例程，将观察到一个带有螺钉集合的拟二元图像。一个可以想象的用例是，例如，根据平均强度值大致确定螺钉的数量。



应用

通过改变参数 nCircleShrink 和 aCenterOffset，可以改变圆形掩膜的大小和位置。根据该掩膜，平均强度值会发生变化：如果掩膜变小，显示的白色背景减少，平均值会降低。如果掩膜移动并显示更多的白色背景，则平均值就会增加。

结果图像 掩膜图像



如有必要，还可以将其他图像加载到本例程中。请注意，随后可能需要调整掩膜参数。

代码

例程图像通过 FileSource 状态机加载为 ipImageWork，并通过以下代码进行处理。

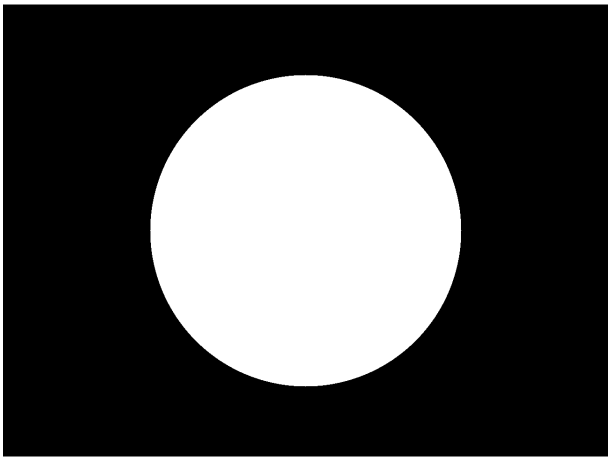
首先定义掩膜的属性。由于掩膜是圆形，它由中心点 aCenter 和半径 nRadius 描述。通过用户可调偏移 aCenterOffset 标定的图像中心点被用作圆圈的中心点。半径计算为从中心点到图像最近边界的距离减去值 nCircleShrink。

```
hr := F_VN_GetImageWidth(ipImageWork, nWidth, hr);
hr := F_VN_GetImageHeight(ipImageWork, nHeight, hr);
aCenter[0] := (nWidth / 2) + aCenterOffset[0];
aCenter[1] := (nHeight / 2) + aCenterOffset[1];
nRadius := (MIN(nWidth, nHeight) / 2) - nCircleShrink;
```

创建与输入图像 ipImageWork 大小相同的新图像作为掩膜，并在黑色背景上绘制白色圆圈。圆圈包含计算出的属性。选择 nThickness 值作为圆边界的宽度 -1，以便填满圆。

```
hr := F_VN_CreateImageAndSetPixels(ipImageMask, nWidth, nHeight, TCVN_ET_USINT, 1, aColorBlack, hr);
hr := F_VN_DrawCircle(
    nCenterX     :=  aCenter[0],
    nCenterY     :=  aCenter[1],
    nRadius      :=  nRadius,
    ipDestImage  :=  ipImageMask,
    aColor       :=  aColorWhite,
    nThickness   :=  -1,
    hrPrev       :=  hr
);
```

结果是现在可以得到一个具有所需圆形的掩膜图像 ipImageMask：



现在，输入图像 ipImageWork 在所定义圆形区域内的平均强度值将借助该掩膜 ipImageMask 计算。函数 F_VN_ImageAverageExp [▸ 1178] 返回TcVnVector4_LREAL 类型的数组 aAverage，以便覆盖最大图像通道数。由于本例程中的输入图像只有一个通道，因此第一个数组元素是所需的平均值，并单独保存。

```
hr := F_VN_ImageAverageExp(
    ipSrcImage  :=  ipImageWork,
    aAverage    :=  aAverage,
    ipMask      :=  ipImageMask,
    hrPrev      :=  hr
);
fAverageInMask := aAverage[0];
```

最后，在结果图像中绘制掩膜圆圈和计算出的平均值。

```
hr := F_VN_ConvertColorSpace(ipImageWork, ipImageRes, TCVN_CST_GRAY_TO_RGB, hr);
hr := F_VN_DrawCircle(aCenter[0], aCenter[1], nRadius, ipImageRes, aColorGreen, 5, hr);
sText := CONCAT('Average: ', LREAL_TO_FMTSTR(fAverageInMask, 2, TRUE));
hr := F_VN_PutTextExp(sText, ipImageRes, 40, 60, TCVN_FT_HERSHEY_SIMPLEX, 2, aColorBlue, 3,
TCVN_LT_ANTIALIASED, FALSE, hr);
```

函数 LREAL_TO_FMTSTR 需要 PLC 库 Tc2_Utilities。

8.1.5.3 统计学图像特征

在这个样本中，将计算图像的统计特征（最小、最大、中位数、平均值、标准差）。以下功能用于此目的：

- F_VN_MinPixelValue [▸ 1188]
- F_VN_MaxPixelValue [▸ 1186]
- F_VN_ImageMedian [▸ 1183]
- F_VN_ImageAverageStdDev [▸ 1179]

应用

这个项目包含一个文件源，任何 RGB 图像都可以载入其中。这些在程序中被转换为灰度图像，并对各种统计数据进行分析。结果值被画在图像中，且可以在ADS 图像查看 [▸ 115]的ipImageRes下查看。

程序

```
// Prepare images
hr := F_VN_ConvertColorSpace(ipImageIn, ipImageIn, TCVN_CST_RGB_TO_GRAY, hr);
hr := F_VN_CopyImage(ipImageIn, ipImageRes, hr);

// Calculate image statistics
hr := F_VN_MinPixelValue(
    ipImage     :=  ipImageIn,
    aMinValue   :=  aMin,
    aPosition   :=  aPos,
    hrPrev      :=  hr
);
hr := F_VN_MaxPixelValue(
    ipImage     :=  ipImageIn,
    aMaxValue   :=  aMax,
    aPosition   :=  aPos,
    hrPrev      :=  hr
```

```
);
hr := F_VN_ImageMedian(
    ipSrcImage  :=  ipImageIn,
    aMedian     :=  aMedian,
    hrPrev      :=  hr
);
hr := F_VN_ImageAverageStdDev(
    ipSrcImage  :=  ipImageIn,
    aAverage    :=  aAverage,
    aStdDev     :=  aStdDev,
    hrPrev      :=  hr
);

// Draw results
sText := CONCAT('Min: ', LREAL_TO_FMTSTR(aMin[0], 2, TRUE ));
hr := F_VN_PutLabel(sText, ipImageRes, nTextX, (nTextYBase + (1 * nTextYIncrement)), fFontScale, 
hr);
sText := CONCAT('Max: ', LREAL_TO_FMTSTR(aMax[0], 2, TRUE ));
hr := F_VN_PutLabel(sText, ipImageRes, nTextX, (nTextYBase + (2 * nTextYIncrement)), fFontScale, 
hr);
sText := CONCAT('Median: ', LREAL_TO_FMTSTR(aMedian[0], 2, TRUE ));
hr := F_VN_PutLabel(sText, ipImageRes, nTextX, (nTextYBase + (3 * nTextYIncrement)), fFontScale, 
hr);
sText := CONCAT('Average: ', LREAL_TO_FMTSTR(aAverage[0], 2, TRUE ));
hr := F_VN_PutLabel(sText, ipImageRes, nTextX, (nTextYBase + (4 * nTextYIncrement)), fFontScale, 
hr);
sText := CONCAT('StdDev: ', LREAL_TO_FMTSTR(aStdDev[0], 2, TRUE ));
hr := F_VN_PutLabel(sText, ipImageRes, nTextX, (nTextYBase + (5 * nTextYIncrement)), fFontScale, 
hr);

hr := F_VN_TransformIntoDisplayableImage(ipImageIn, ipImageInDisp, hr);
hr := F_VN_TransformIntoDisplayableImage(ipImageRes, ipImageResDisp, hr);
```

该函数LREAL_TO_FMTSTR需要 PLC 库Tc2_Utilities。

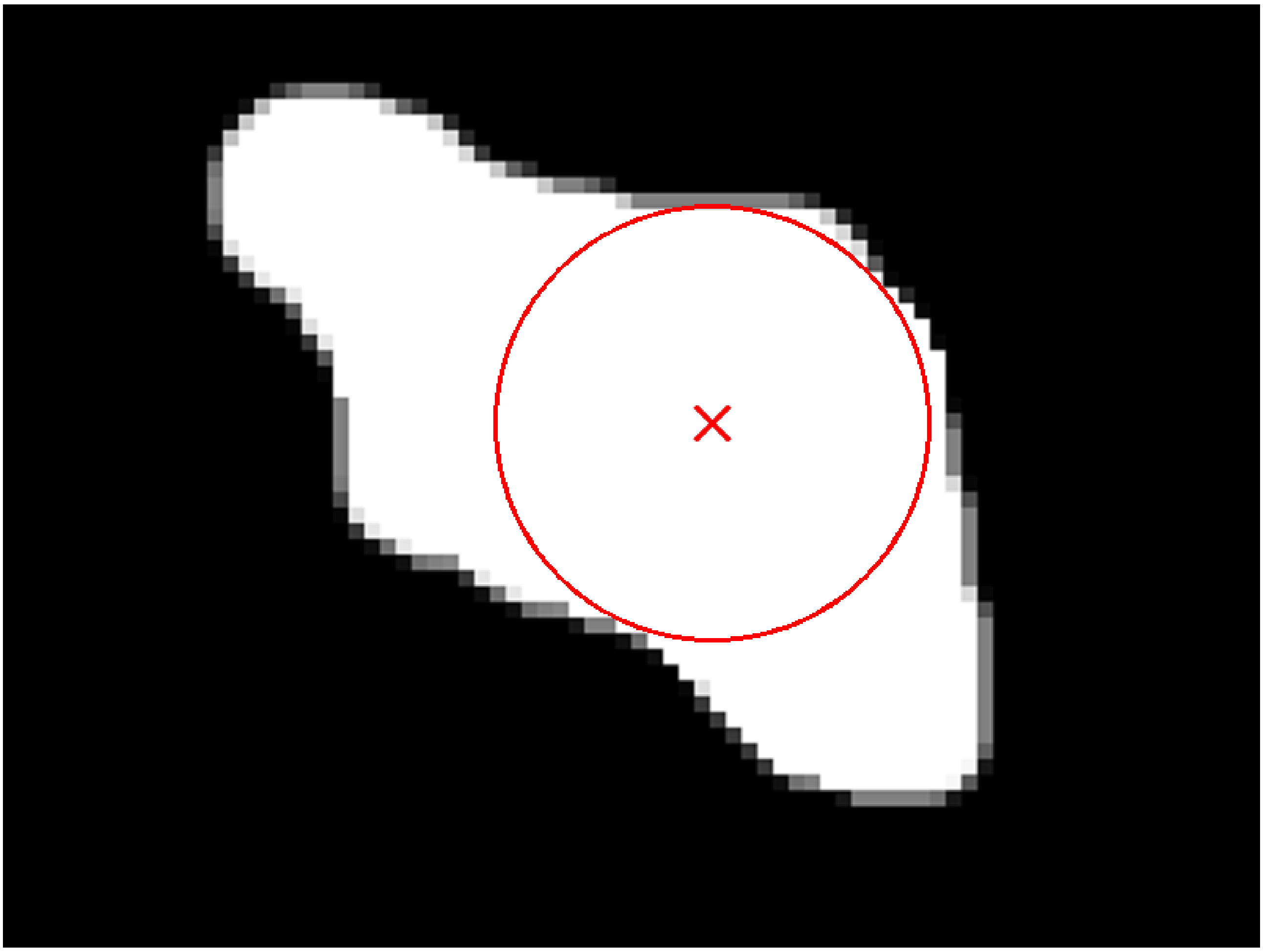
8.1.5.4 通过距离变换实现内圈

在这个样本中，你将通过距离变换的方法找到一个畸形物体的内圈。为此，基本上可以使用以下函数：

- F_VN_DistanceTransformation [▶ 1173]
- F_VN_DrawCircle [▶ 959]

说明

外部轮廓的近似值，如包围的矩形或凸包络，可以通过适当的 API 功能计算。反之，通过距离变换，可以找到轮廓内最大尺寸的内圈。



为此，通过距离变换确定与物体边缘存在最大欧氏距离的点。然后这个点就是物体内圈的中心点。

如果通过自己的图像使用这个样本，应该通过变量bInvert和fThreshold调整阈值和二值化。通过参数eDistanceType和eMaskSize，可以有目的地改变函数F_VN_DistanceTransformation [▶ 1173]的参数。

程序

第一步是对输入图像进行二值化。这实现了物体和背景之间的清晰区分。二值化对距离变换很重要，因为距离变换只对数值为 0 的像素有反应。如果物体是白色背景上的黑色，必须通过阈值类型BINARY_INV反转。

```
IF bInvert THEN
    eThresholdType := TCVN_TT_BINARY_INV;
ELSE
    eThresholdType := TCVN_TT_BINARY;
END_IF
hr := F_VN_Threshold(ipImageIn, ipImageWork, fThreshold, 255, eThresholdType, hr);
```

通过距离变换，你可以得到一个图像，其中每个像素的值表示与下一个值为 0 的像素的距离。因此，具有最高值的像素是距离最大的像素。如果使用 L2 准则（欧氏距离）作为距离类型eDistanceType，你可以把这个像素解释为一个尽可能填满物体的圆的中心点。因此，通过函数F_VN_MaxPixelValue [▶ 1186]确定具有最大值的像素，并将其位置作为圆的中心点，而其数值作为圆的半径。

```
hr := F_VN_DistanceTransformation(
    ipSrcImage       :=  ipImageWork,
    ipDestImage      :=  ipImageWork,
    eDistanceType    :=  TCVN_DT_L2,
    eMaskSize        :=  TCVN_DTM_5,
    hrPrev           :=  hr
);
hr := F_VN_MaxPixelValue(ipImageWork, aMax, aPos, hr);
fRadius := aMax[0];
aCenter := aPos;
```

掩码大小eMaskSize决定了距离计算的准确性，且对结果影响不大。

最后，在确定的中心点和半径的基础上，在结果图像中画出圆，以便能够验证执行情况。

```
hr := F_VN_DrawCircle(
    nCenterX    :=  TO_UDINT(aCenter[0]),
    nCenterY    :=  TO_UDINT(aCenter[1]),
    nRadius     :=  TO_UDINT(fRadius),
    ipDestImage :=  ipImageRes,
    aColor      :=  aRed,
    nThickness  :=  2,
    hrPrev      :=  hr
);
hr := F_VN_DrawPointExp(TO_UDINT(aCenter[0]), TO_UDINT(aCenter[1]), ipImageRes, TCVN_DS_X, aRed, 10,
2, TCVN_LT_ANTIALIASED, hr);
```

8.1.6 图像颜色和对比度处理

8.1.6.1 与 RGB 参考色的颜色相似度

在这个样本中，函数F_VN_ReferenceColorSimilarityExp_TcVnVector3_LREAL [▶ 1216]计算每个像素与 3 种参考颜色的相似度，并将它们作为灰度图像返回。然后，根据这些灰度图像，检测参考色的物体。

变量

```
hr                      :   HRESULT;

// Images
ipImageIn               :   ITcVnImage;
ipImageInDisp           :   ITcVnDisplayableImage;

ipImageThres            :   ITcVnImage;
ipImageThresDisp        :   ARRAY [0..2] OF ITcVnDisplayableImage;
ipImageWorkCol          :   ITcVnImage;
ipImageWorkColDisp      :   ARRAY [0..2] OF ITcVnDisplayableImage;

ipImageRes              :   ITcVnImage;
ipImageResDisp          :   ITcVnDisplayableImage;

// Color
iColor                  :   INT;
aColorTxt               :   ARRAY[0..2] OF STRING := [ 'RED', 'GREEN', 'BLUE' ];
aColor                  :   ARRAY[0..2] OF TcVnVector4_LREAL := [ [150, 0, 0], [0, 255, 0],
[0, 0, 255] ];
aColorRef               :   ARRAY[0..2] OF TcVnVector3_LREAL := [ [255, 75, 60], [40, 140, 95],
[40, 140, 190] ];

// Contours
ipContourList           :   ITcVnContainer;
ipIterator              :   ITcVnForwardIterator;
aOffset                 :   TcVnPoint;
ipContour               :   ITcVnContainer;
fArea                   :   LREAL;
aCenter                 :   TcVnPoint2_LREAL;
```

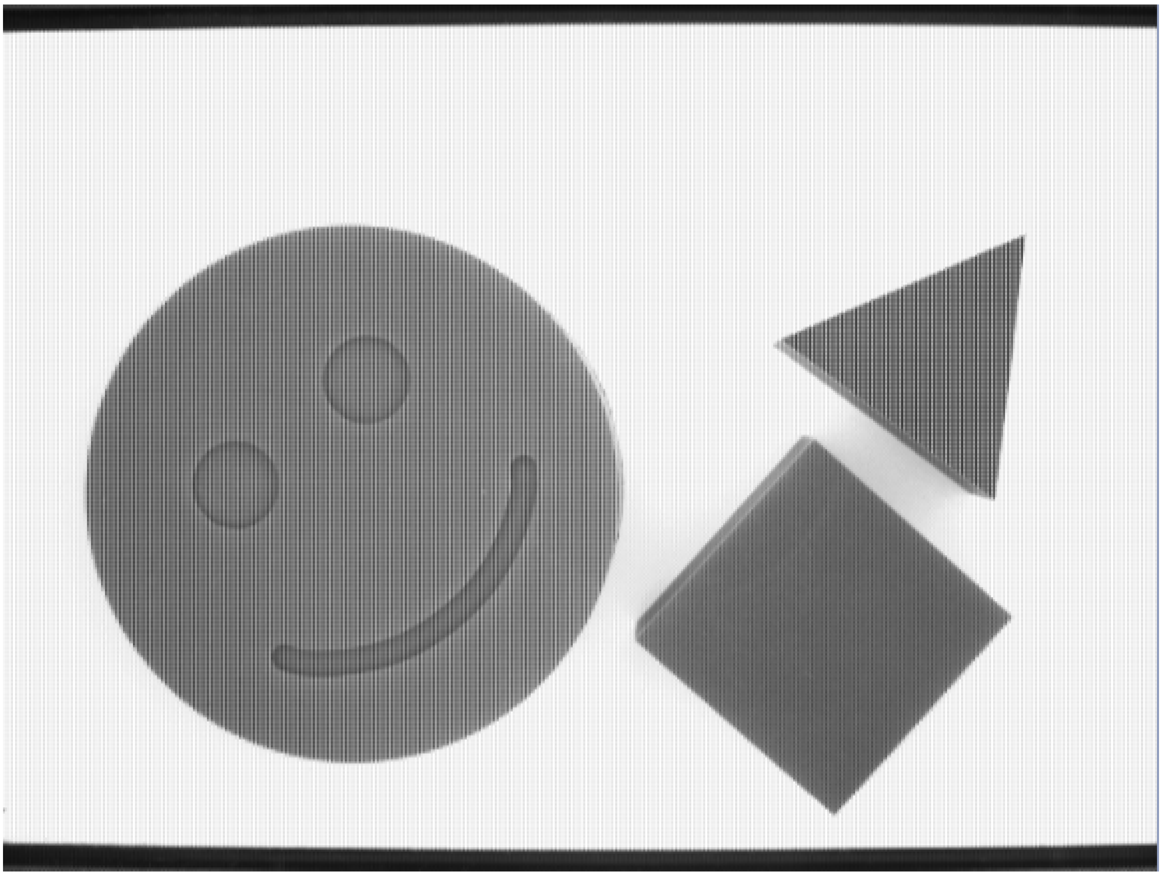
代码

```
// Attention: With other images another color space transformation could be necessary
hr := F_VN_ConvertColorSpace(ipImageIn, ipImageRes, TCVN_CST_Bayer_RG_TO_RGB, hr);

FOR iColor := 0 TO 2 DO

   // Compute the Color Similarity to a Reference Color
   hr := F_VN_ReferenceColorSimilarityExp_TcVnVector3_LREAL(
               ipSrcImage         := ipImageRes,
               ipDestImage        := ipImageWorkCol,
               aRefColor          := aColorRef[iColor],
               fVariance          := 0.1,
               fLuminanceWeight   := 0.2,
               hrPrev             := hr);

   hr := F_VN_Threshold(ipImageWorkCol, ipImageThres, 200, 255, TCVN_TT_Binary, hr);

   // Find all objects / contours in the black and white image
   hr := F_VN_FindContours(ipImageThres, ipContourList, hr);
   hr := F_VN_GetForwardIterator(ipContourList, ipIterator, hr);

   // Filter the objects by size and draw the contours
   WHILE SUCCEEDED(hr) AND_THEN ipIterator.CheckIfEnd() <> S_OK DO
      hr := F_VN_GetContainer(ipIterator, ipContour, hr);
```

```
      hr := F_VN_IncrementIterator(ipIterator, hr);

      // Filter contours by size
      hr := F_VN_ContourArea(ipContour, fArea, hr);
      IF fArea > 5000 THEN
         // Draw Results into an Image
         hr := F_VN_DrawContours(ipContour, -1, ipImageRes, aColor[iColor], 3, hr);
         hr := F_VN_ContourCenterOfMass(ipContour, aCenter, hr);
         hr := F_VN_PutText(aColorTxt[iColor], ipImageRes, LREAL_TO_UDINT(aCenter[0])-30, 
LREAL_TO_UDINT(aCenter[1])+10, TCVN_FT_HERSHEY_PLAIN, 2, aColor[iColor],hr);
      END_IF
   END_WHILE

   // Display effect of the Color-Similarity-Function
   hr := F_VN_TransformIntoDisplayableImage(ipImageThres, ipImageThresDisp[iColor], hr);
   hr := F_VN_TransformIntoDisplayableImage(ipImageWorkCol, ipImageWorkColDisp[iColor], hr);
END_FOR
```

结果

Bayer RG 格式的输入图像

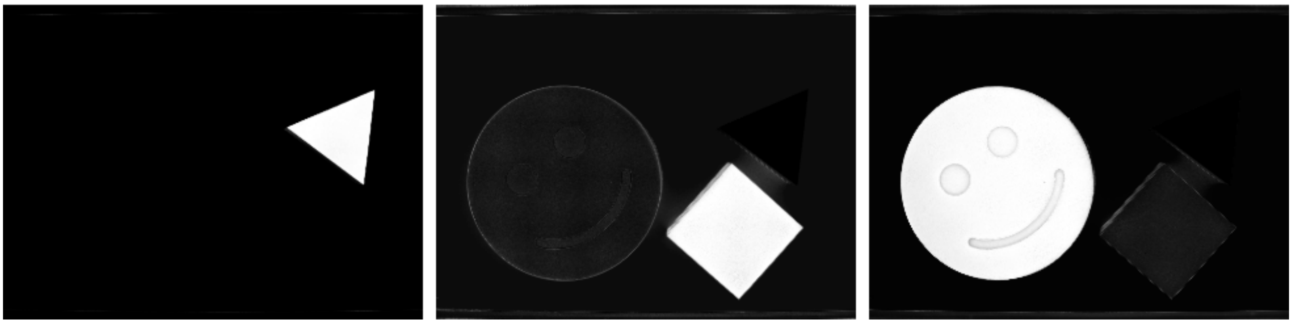


在通过函数F_VN_ConvertColorSpace [▶ 1197]将输入图像转换为 RGB 图像后，F_VN_ReferenceColorSimilarityExp_TcVnVector3_LREAL [▶ 1216]可用于确定每个像素与参考颜色的颜色相似度。以下内容适用于返回的结果图像：

- 像素值 255（白色）对应于与参考颜色的 100% 匹配
- 像素值 0（黑色）对应于与参考颜色的 0% 匹配
- 0 至 255 之间的像素值对应于与参考颜色部分相关。

```
// Compute the Color Similarity to a Reference Color
hr := F_VN_ReferenceColorSimilarityExp_TcVnVector3_LREAL(
               ipSrcImage          := ipImageRes,
               ipDestImage         := ipImageWorkCol,
               aRefColor           := aColorRef[iColor],
               fVariance           := 0.1,
               fLuminanceWeight    := 0.2,
               hrPrev              := hr);
```

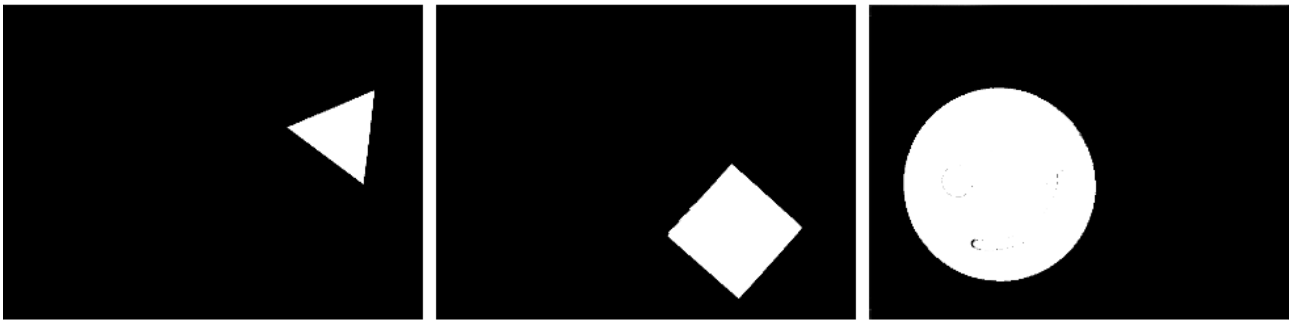
aColorRef := [255,75,60]; (RED) aColorRef := [40,140,95]; (GREEN) aColorRef := [40,140,190]; (BLUE)



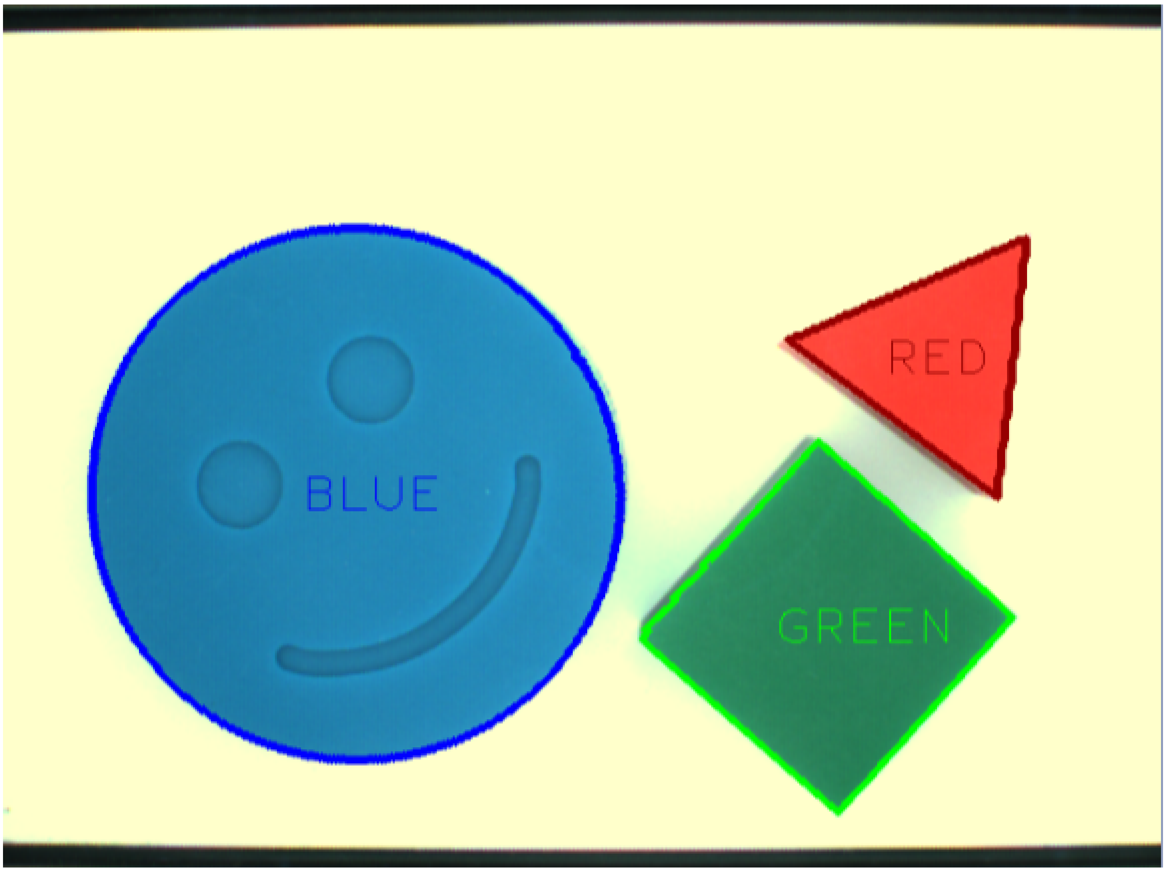
阈值（F_VN_Threshold [▶ 1287]）用于定义与参考颜色的匹配程度，以便将其分配给参考颜色。

```
hr := F_VN_Threshold(ipImageWorkCol, ipImageThres, 200, 255, TCVN_TT_Binary, hr);
```

aColorRef := [255,75,60]; (RED) aColorRef := [40,140,95]; (GREEN) aColorRef := [40,140,190]; (BLUE)



随后检测后的结果图像。



类似样本

- 通过 RGB 范围检查颜色范围 [▶ 2693]

8.1.6.2 通过颜色表显示

在这个样本中，会将颜色表应用于灰度图像。大体上可以使用以下功能：

- F_VN_GenerateColorMap [▶ 1198]
- F_VN_CreateContainerFromArray [▶ 726]
- F_VN_GenerateCustomColorMap [▶ 1200]
- F_VN_ApplyColorMap [▶ 1193]

说明

有几个原因可以说明为什么想在单色图像上应用颜色表。这些包括，例如更令人愉快的外观和对人眼来说更清晰的数值范围的显示。这首先与 16 位图像有关，因为如果数值频谱分布在三个颜色通道上，那么可识别性会大大增强。

根据使用情况，可以使用各种颜色的表格。TwinCAT Vision 通过枚举ETcVnColorMap [▶ 149]提供一些预定义的颜色梯度。此外，你还可以定义你自己的颜色梯度，并用它们创建颜色表。

在这个示例中，你将各种颜色表应用于以下灰度梯度（数值范围为 16 位）。

附图 26: 灰度梯度

当然，你也可以使用自己的图像。请注意，仅可使用单通道图像，否则必须事先使用F_VN_ConvertColorSpace [▶ 1197]进行适当的转换。

程序

在将颜色表应用于图像之前，必须首先创建颜色表。为此，必须知道颜色表用于 8 位还是 16 位图像，因为根据不同的情况，需要不同数量的条目（见颜色表）。因此，必须将类型为ETcVnColorMapSize [▶ 154]的参数eColorMapSize设置为256（如果是 8 位）和65536（如果是 16 位）以与输入图像ipImageIn的位深相匹配。通过F_VN_GetImageInfo [▶ 769]查询像素格式的元素大小来实现：

```
hr := F_VN_GetImageInfo(ipImageIn, stImageInfo, hr);
IF stImageInfo.stPixelFormat.nElementSize = 8 THEN
    eColorMapSize := TCVN_CMS_256;
ELSIF stImageInfo.stPixelFormat.nElementSize = 16 THEN
    eColorMapSize := TCVN_CMS_65536;
END_IF
```

根据颜色梯度，生成一个颜色表。可以手动定义颜色梯度或者使用模板中的颜色梯度。这个样本根据参数bUseCustomColors处理区分。

```
IF bUseCustomColors THEN
    // create color map from custom-defined colors
    <...>
ELSE
    // create color map from preset
    <...>
END_IF
```

通过创建包含多个条目的 TcVnVector3_REAL 类型容器，可以手动定义颜色梯度。每个条目代表一种颜色。在创建颜色表时，颜色被解释为均匀地分布在定义区域；所有其他数值都是对三个通道中的每一个进行线性插值。此处使用了来自阵列aColors的颜色梯度，包括红、绿、蓝三种颜色。然后，使用函数 F_VN_GenerateCustomColorMap 创建颜色表。

```
VAR
    aColors: ARRAY [0..(N_COLORS-1)] OF TcVnVector3_REAL := [
        [255, 0, 0],
        [0, 255, 0],
        [0, 0, 255]
    ];
```

```
END_VAR

hr := F_VN_CreateContainerFromArray(ADR(aColors), ipColors, ContainerType_Vector_TcVnVector3_REAL,
N_COLORS, hr);
hr := F_VN_GenerateCustomColorMap(
    ipColorMap      :=  ipColorMap,
    ipInitialColors :=  ipColors,
    eColorMapSize   :=  eColorMapSize,
    hrPrev          :=  hr
);
```

为了使用预定义的颜色梯度，请使用函数F_VN_GenerateColorMap [▶ 1198]，并从枚举ETcVnColorMap [▶ 149]将参数eColorMap设置为需要的颜色梯度。在这种情况下，最初设置为JET。

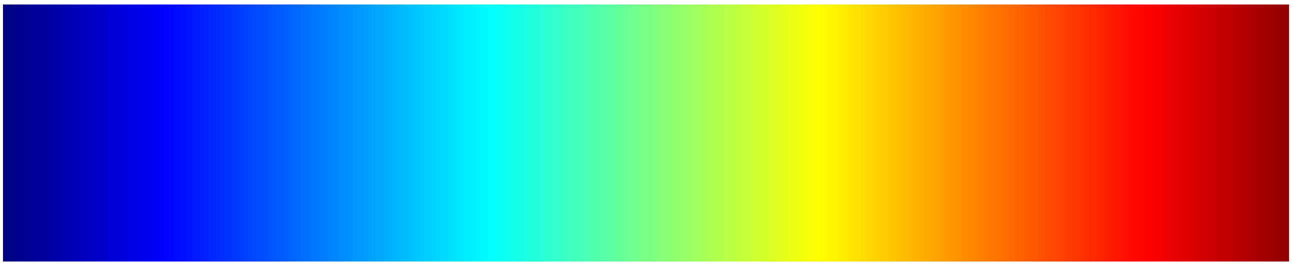
```
VAR
    eColorMap: ETcVnColorMap := TCVN_CM_JET;
END_VAR

hr := F_VN_GenerateColorMap(
    ipColorMap      :=  ipColorMap,
    eColorMap       :=  eColorMap,
    eColorMapSize   :=  eColorMapSize,
    hrPrev          :=  hr
);
```

创建的颜色表是一个容器 [▶ 132]，其中存储了输入图像中每个可能的值所要发出的颜色。因此，也可以通过定义所有 256 个或 65536 个条目，然后直接从这些条目中创建一个容器，自己创建一个颜色表。然而，在大多数情况下，使用 API 功能更为简单。

最后，将创建的颜色表应用于输入图像，以获得理想的结果。结果图像总是一个 3 通道的 8 位图像。

```
hr := F_VN_ApplyColorMap(
    ipSrcImage      :=  ipImage,
    ipDestImage     :=  ipImage,
    ipColorMap      :=  ipColorMap,
    hrPrev          :=  hr
);
```

如果将预定义的 JET 颜色表应用于上图所示的输入图像，会产生以下结果：



附图 27: 应用 JET 颜色表后的颜色梯度

8.1.7 图像过滤

8.1.7.1 模糊过滤器

在这个样本中，比较了以下模糊过滤器：

- 双边过滤器 [▶ 1232]
- 高斯过滤器 [▶ 1250]
- 中位数过滤器 [▶ 1261]
- 作为自定义过滤器 [▶ 1242]实现的平均数过滤器

说明

在某些情况下，图像需要平滑化或摆脱噪音。对于读码这样的任务，需要同时在图像上获得良好的对比度和良好的边缘，以便下游的算法能够成功地工作。

双边过滤器特别适用于将图像平滑与同时保留边缘相结合。对于每个像素的强度，它计算出周围像素的加权平均值。加权不仅取决于像素的距离有多远，还取决于强度的差异。因此，尖锐的边缘被保留下来。

高斯过滤器通过高斯钟形函数的二维离散近似值平滑图像。这样可以减少图像的噪音。较小的结构会丢失，但另一方面，较粗的结构会保留。因此，与下面两个过滤器相比，边缘的模糊程度较低，但较大的干扰没有被很好地去除。

中位数过滤器将指定掩码的所有像素值按升序排序，然后选择之后相应像素设置的中位数。这种过滤器很适合抑制小于所应用过滤器掩码的结构（干扰），并且对异常值非常稳健。因此，可以最好地去除小的椒盐干扰等，但这样做的缺点是，其他的详细信息也会丢失。

为了形成一个平均值，使用了**自定义过滤器**功能。在这里，每个像素的值都被内核像素的平均值所取代，这导致了模糊化。因此，干扰（噪音）会受到抑制，但随着内核数量的增加，细节和边缘逐渐变得模糊不清。由于自定义过滤器功能为应用自己的过滤器提供了许多不同的选择，为了实现过滤器内核的标准化，还需要F_VN_InitMatrixStruct功能。

变量

```
// Bilateral Filter
ipImageBilateral          :   ITcVnImage;
ipImageBilateralDisp      :   ITcVnDisplayableImage;
nBilateral_Diameter       :   DINT  := 7;
fBilateral_SigmaColor     :   LREAL := 100;
fBilateral_SigmaSpace     :   LREAL := 100;

// Gaussian Filter
ipImageGaussian           :   ITcVnImage;
ipImageGaussianDisp       :   ITcVnDisplayableImage;
nFilterWidth              :   UDINT := 7;
nFilterHeight             :   UDINT := 7;

// Median Filter
ipImageMedian             :   ITcVnImage;
ipImageMedianDisp         :   ITcVnDisplayableImage;
nMedian_FilterSize        :   UDINT := 7;

// Custom Filter e.g. Mean
ipImageCustom             :   ITcVnImage;
ipImageCustomDisp         :   ITcVnDisplayableImage;
stKernelMatrix            :   TcVnMatrix;
// 7x7 Mean Filter Kernel with weights of 1/49 ~ 0.0204081632653
aMatrixArray7x7           :   ARRAY [0..6,0..6] OF REAL := [49(0.0204081632653)];
```

代码

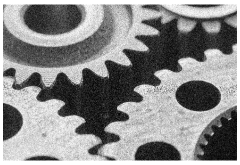
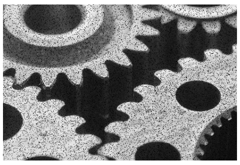
```
hr := F_VN_BilateralFilter(
        ipSrcImage      := ipImageIn,
        ipDestImage     := ipImageBilateral,
        nDiameter       := nBilateral_Diameter,
        fSigmaColor     := fBilateral_SigmaColor,
        fSigmaSpace     := fBilateral_SigmaSpace,
        hrPrev          := hr);

hr := F_VN_GaussianFilter(
        ipSrcImage      := ipImageIn,
        ipDestImage     := ipImageGaussian,
        nFilterWidth    := nFilterWidth,
        nFilterHeight   := nFilterHeight,
        hrPrev          := hr);

hr := F_VN_MedianFilter(
        ipSrcImage      := ipImageIn,
        ipDestImage     := ipImageMedian,
        nFilterSize     := nMedian_FilterSize,
        hrPrev          := hr);

// Mean-Filter with Custom Filter Function
hr := F_VN_InitMatrixStruct(
        pSrcBuffer      := ADR(aMatrixArray7x7),
        stDestMatrix    := stKernelMatrix,
        nRows           := 7,
        nCols           := 7,
        eElementType    := TCVN_ET_REAL,
        hrPrev          := hr);

hr := F_VN_CustomFilter(
        ipSrcImage      := ipImageIn,
        ipDestImage     := ipImageCustom,
```

```
        eDestDepth      := TCVN_ET_USINT,
        stKernel        := stKernelMatrix,
        hrPrev          := hr);
```

结果

未经处理的原始图像（第一行）在细节上已经显示出齿轮表面的精细结构。为了展示过滤器的效果，在原始图像中加入了高斯噪声（第二行）和椒盐噪声（第三行）等额外干扰。

为了使不同过滤器的效果尽可能具有可比性，所有的过滤器都使用了 7×7 的内核尺寸。如果使用其他参数值或内核大小以及功能的 EXP 变体的更多设置选项，可以具体地实现其他效果。由于过滤器的效果在很大程度上取决于输入图像的内容，建议通过不同的参数设置测试几个过滤器，以便直接比较并找到最适合使用情况的过滤器。下表包含了结果图像的概览：

原始图像	双边过滤器	高斯过滤器	中位数过滤器	平均数过滤器
				
				
				

注意

执行时间长或波动大

过滤器的执行时间基本上取决于过滤器的类型以及图像和内核的大小。因此，请注意，过滤器操作可能比其他功能需要更长的时间，所以应该相应地调整任务的周期时间。

如果过滤器的参数在运行期间更改，特别是内核大小，建议使用看门狗 [▶ 127]。对于中位数过滤器而言，由于原理的原因，运行时间总是会有一些波动。

此外，工作任务 [▶ 58]可导致总计算时间的减少，这应根据系统的情况进行测试。

8.1.7.2 边缘检测过滤器

在这个样本中，对以下边缘检测过滤器进行了比较：

- 索贝尔过滤器 [▶ 1276]
- 夏尔过滤器 [▶ 1268]
- 拉普拉斯过滤器 [▶ 1254]

说明

索贝尔过滤器是一个简单的线性过滤器，用于标记图像中的梯度。夏尔过滤器与此非常相似；只是过滤器内核的特性有些不同。夏尔过滤器具有更好的各向同性，即边缘的检测与方向无关。

拉普拉斯过滤器根据图像的离散二次推导的零点来检测边缘。相比之下，这个变体很容易受到噪音的影响。

变量

```
// Sobel
ipImageSobel          :   ITcVnImage;
ipImageSobelDisp      :   ITcVnDisplayableImage;
```

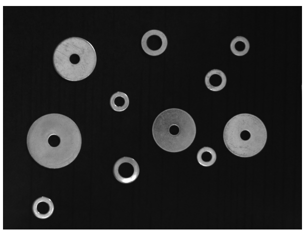
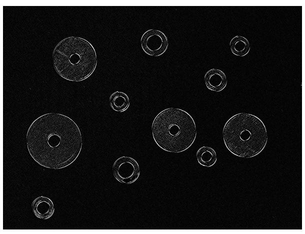
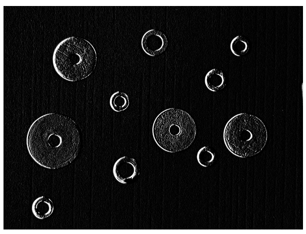
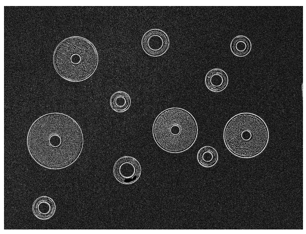
```
eSobel_DestDepth      :   ETcVnElementType := TCVN_ET_USINT;
aSobel_DerivOrder     :   ARRAY [0..1] OF UDINT := [1, 1];
nSobel_KernelSize     :   UDINT := 3;
fSobel_Scale          :   LREAL := 10;
fSobel_Delta          :   LREAL := 0;
eSobel_BorderExtra    :   ETcVnBorderInterpolationMethod := TCVN_BIM_DEFAULT;

// Scharr
ipImageScharr         :   ITcVnImage;
ipImageScharrDisp     :   ITcVnDisplayableImage;
eScharr_DestDepth     :   ETcVnElementType := TCVN_ET_USINT;
eScharr_FilterDir     :   ETcVnFilterDirection := TCVN_FD_X;
fScharr_Scale         :   LREAL := 1;
fScharr_Delta         :   LREAL := 0;
eScharr_BorderExtra   :   ETcVnBorderInterpolationMethod := TCVN_BIM_DEFAULT;

// Laplace
ipImageLaplace        :   ITcVnImage;
ipImageLaplaceDisp    :   ITcVnDisplayableImage;
eLaplace_DestDepth    :   ETcVnElementType := TCVN_ET_USINT;
nLaplace_KernelSize   :   UDINT := 3;
fLaplace_Scale        :   LREAL := 10;
fLaplace_Delta        :   LREAL := 0;
eLaplace_BorderExtra :    ETcVnBorderInterpolationMethod := TCVN_BIM_DEFAULT;
```

代码

```
// Execute the Sobel Filter
hr := F_VN_SobelFilterExp(
         ipSrcImage    := ipImageIn
         ipDestImage   := ipImageSobel,
         eDestDepth    := eSobel_DestDepth,
         nXOrder       := aSobel_DerivOrder[0],
         nYOrder       := aSobel_DerivOrder[1],
         nKernelSize   := nSobel_KernelSize,
         fScale        := fSobel_Scale,
         fDelta        := fSobel_Delta,
         eBorderType   := eSobel_BorderExtra,
         hrPrev        := hr);

// Execute the Scharr Filter
hr := F_VN_ScharrFilterExp(
         ipSrcImage       := ipImageIn,
         ipDestImage      := ipImageScharr,
         eDestDepth       := eScharr_DestDepth,
         eFilterDirection:= eScharr_FilterDir,
         fScale           := fScharr_Scale,
         fDelta           := fScharr_Delta,
         eBorderType      := eScharr_BorderExtra,
         hrPrev           := hr);

// Execute the Laplacian Filter
hr := F_VN_LaplacianFilterExp(
         ipSrcImage    := ipImageIn,
         ipDestImage   := ipImageLaplace,
         eDestDepth    := eLaplace_DestDepth,
         nKernelSize   := nLaplace_KernelSize,
         fScale        := fLaplace_Scale,
         fDelta        := fLaplace_Delta,
         eBorderType   := eLaplace_BorderExtra,
         hrPrev        := hr);
```

结果

该样本项目包含三个过滤器的函数调用。参数配置及其对样本图像的影响可以在这个样本中测试。

原始图像	索贝尔过滤器	夏尔过滤器	拉普拉斯过滤器
			

8.1.7.3 形态学算子

在这个样本中，会将形态学算子应用于图像。为此，使用：

- F_VN_CreateStructuringElement [▶ 1240]
- F_VN_MorphologicalOperator [▶ 1264]

说明

图像中的物体可以根据一定的规则用形态学算子进行更改。形态学算子大多应用于二进制图像；然而，一般也可以应用于灰度图像。通过诸如OPENING和CLOSING的组合运算，例如，有可能补充物体中缺失的部分，并且可以去除小的异常值。

变量

```
ipElement    :  ITcVnImage;
eSEShape     :  ETcVnStructuringElementShape := TCVN_SES_RECTANGLE;
aSESize      :  ARRAY [0..1] OF UDINT := [29, 29];
eOperator    :  ETcVnMorphologicalOperator := TCVN_MO_OPENING;
```

程序

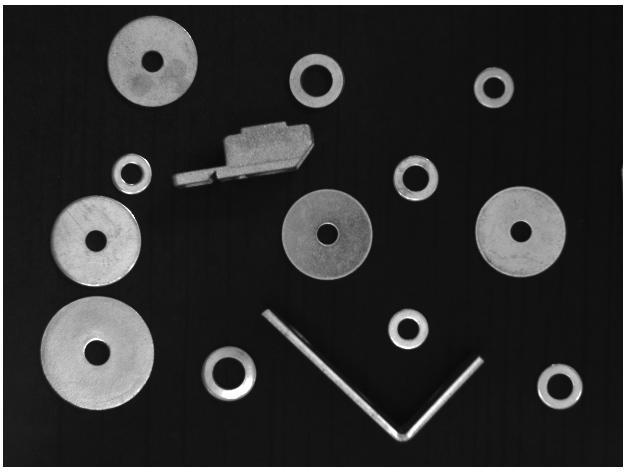
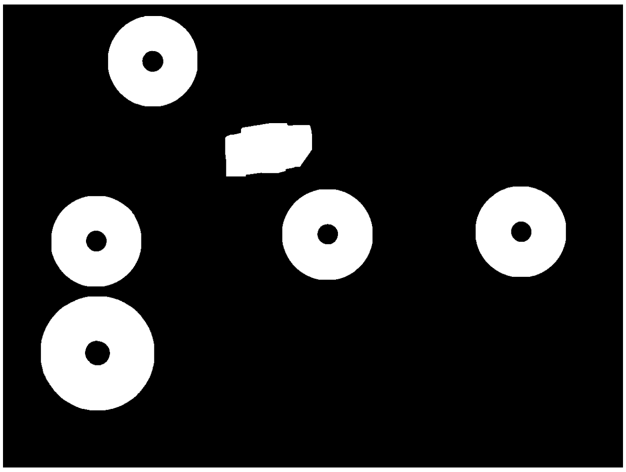
形态学算子需要一个结构元素，以定义其对每个像素的影响区域。为此，可以从枚举ETcVnStructuringElementShape [▶ 197]中选择各种形状并从不同的尺寸中进行选择。

```
hr := F_VN_CreateStructuringElement(
    ipStructuringElement    :=  ipElement,
    eShape                  :=  eSEShape,
    nWidth                  :=  aSESize[0],
    nHeight                 :=  aSESize[1],
    hrPrev                  :=  hr);
```

然后，通过枚举ETcVnMorphologicalOperator [▶ 188]定义一个形态学算子，并将其与创建的结构元素一起应用于输入图像。

```
hr := F_VN_MorphologicalOperator(
    ipSrcImage              :=  ipImageIn,
    ipDestImage             :=  ipImageRes,
    eOperator               :=  eOperator,
    ipStructuringElement    :=  ipElement,
    hrPrev                  :=  hr);
```

例如，通过大小为[31; 31]的矩形结构元素和之前的阈值进行形态学开运算，会产生以下结果。它将较小或较窄的物体从图像中移除。

输入图像	结果图像
	

注意

执行时间长

根据输入图像和结构元素的大小，与其他功能相比，形态学算子的执行可能需要很长的时间。确保设置了相应的周期时间，且必要时使用看门狗！

8.1.8 图像分割

- 通过 RGB 范围检查颜色范围 [▶ 2693]

8.1.8.1 通过 RGB 范围检查颜色范围

在该例程中，函数 F_VN_CheckColorRange [▶ 1285] 在图像中搜索红色、绿色和蓝色对象，并根据它们的颜色对其进行标注。该函数适用于 RGB 图像。也可以将该函数应用于其他颜色空间中的图像，如 HSV、Lab 和 BGR，在这种情况下，必须对参数设置进行相应调整。

变量

```
hr                  :   HRESULT;

// Images
ipImageIn           :   ITcVnImage;
ipImageInDisp       :   ITcVnDisplayableImage;

ipImageWorkCol      :   ITcVnImage;
ipImageWorkColDisp :   ARRAY [0..2] OF ITcVnDisplayableImage;

ipImageRes          :   ITcVnImage;
ipImageResDisp      :   ITcVnDisplayableImage;

// Colors
iColor              :   INT;
aColorTxt           :   ARRAY[0..2] OF STRING := [ 'RED', 'GREEN', 'BLUE' ];
aColor              :   ARRAY[0..2] OF TcVnVector4_LREAL := [ [150, 0, 0], [0, 255, 0], [0, 0,
255] ];
aColorRefLow        :   ARRAY[0..2] OF TcVnVector4_LREAL := [ [150, 50, 20], [35, 90, 60], [20, 40,
130] ];
aColorRefUp         :   ARRAY[0..2] OF TcVnVector4_LREAL := [ [255, 120, 100], [100, 200, 140], [60,
160, 255] ];

// Contours
ipContourList       :   ITcVnContainer;
ipIterator          :   ITcVnForwardIterator;
ipContour           :   ITcVnContainer;
fArea               :   LREAL;
aCenter             :   TcVnPoint2_LREAL;
```

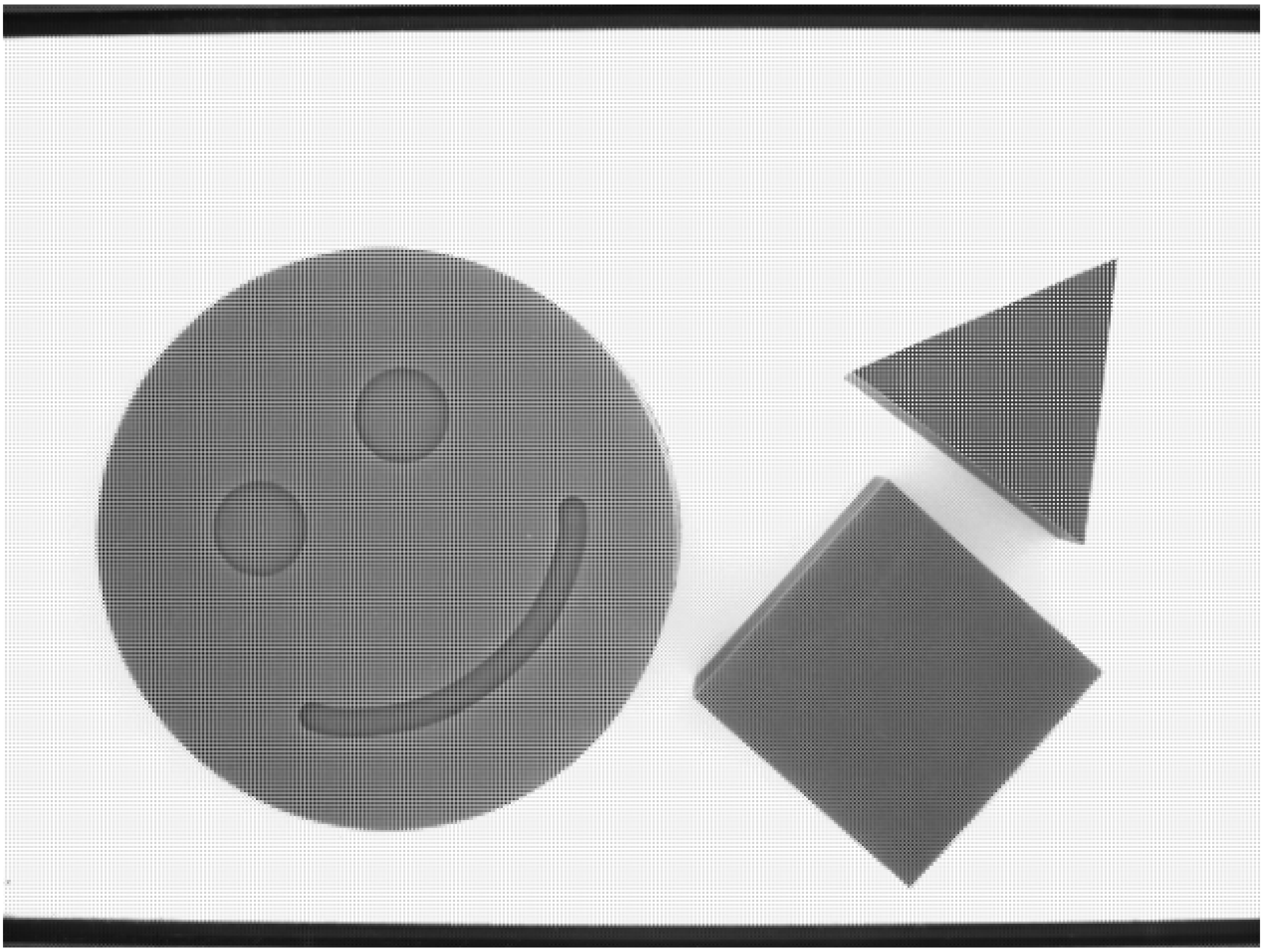
代码

```
// Attention: With other images another color space transformation could be necessary
hr := F_VN_ConvertColorSpace(ipImageIn, ipImageRes, TCVN_CST_Bayer_RG_TO_RGB, hr);

FOR iColor := 0 TO 2 DO

   // Apply a "Color-Threshold" on the image
   hr := F_VN_CheckColorRange( ipSrcImage    := ipImageRes,
                               ipDestImage   := ipImageWorkCol,
                               aLowerBounds  := aColorRefLow[iColor],
                               aUpperBounds  := aColorRefUp[iColor],
                               hrPrev        := hr );

   // Find all objects / contours in the black and white image
   hr := F_VN_FindContours(ipImageWorkCol, ipContourList, hr);
   hr := F_VN_GetForwardIterator(ipContourList, ipIterator, hr);

   // Filter the objects by size and draw the contours
   WHILE SUCCEEDED(hr) AND_THEN ipIterator.CheckIfEnd() <> S_OK DO
      hr := F_VN_GetContainer(ipIterator, ipContour, hr);
      hr := F_VN_IncrementIterator(ipIterator, hr);

      // Filter contours by size
      hr := F_VN_ContourArea(ipContour, fArea, hr);

      IF fArea > 5000 THEN
         // Draw Results into an Image
         hr := F_VN_DrawContours(ipContour, -1, ipImageRes, aColor[iColor], 3, hr);
         hr := F_VN_ContourCenterOfMass(ipContour, aCenter, hr);
         hr := F_VN_PutText(aColorTxt[iColor], ipImageRes, LREAL_TO_UDINT(aCenter[0])-30,
LREAL_TO_UDINT(aCenter[1])+10, TCVN_FT_HERSHEY_PLAIN, 2, aColor[iColor], hr);
      END_IF
```

```
    END_WHILE
END_FOR
```

结果

Bayer RG 格式的输入图像。

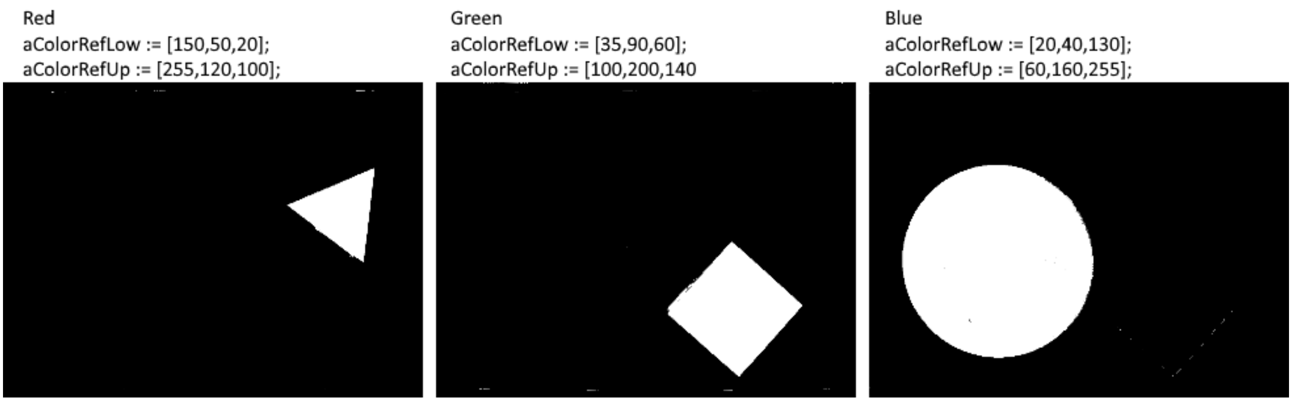


输入图像通过函数 F_VN_ConvertColorSpace [▶ 1197]转换为 RGB 图像。

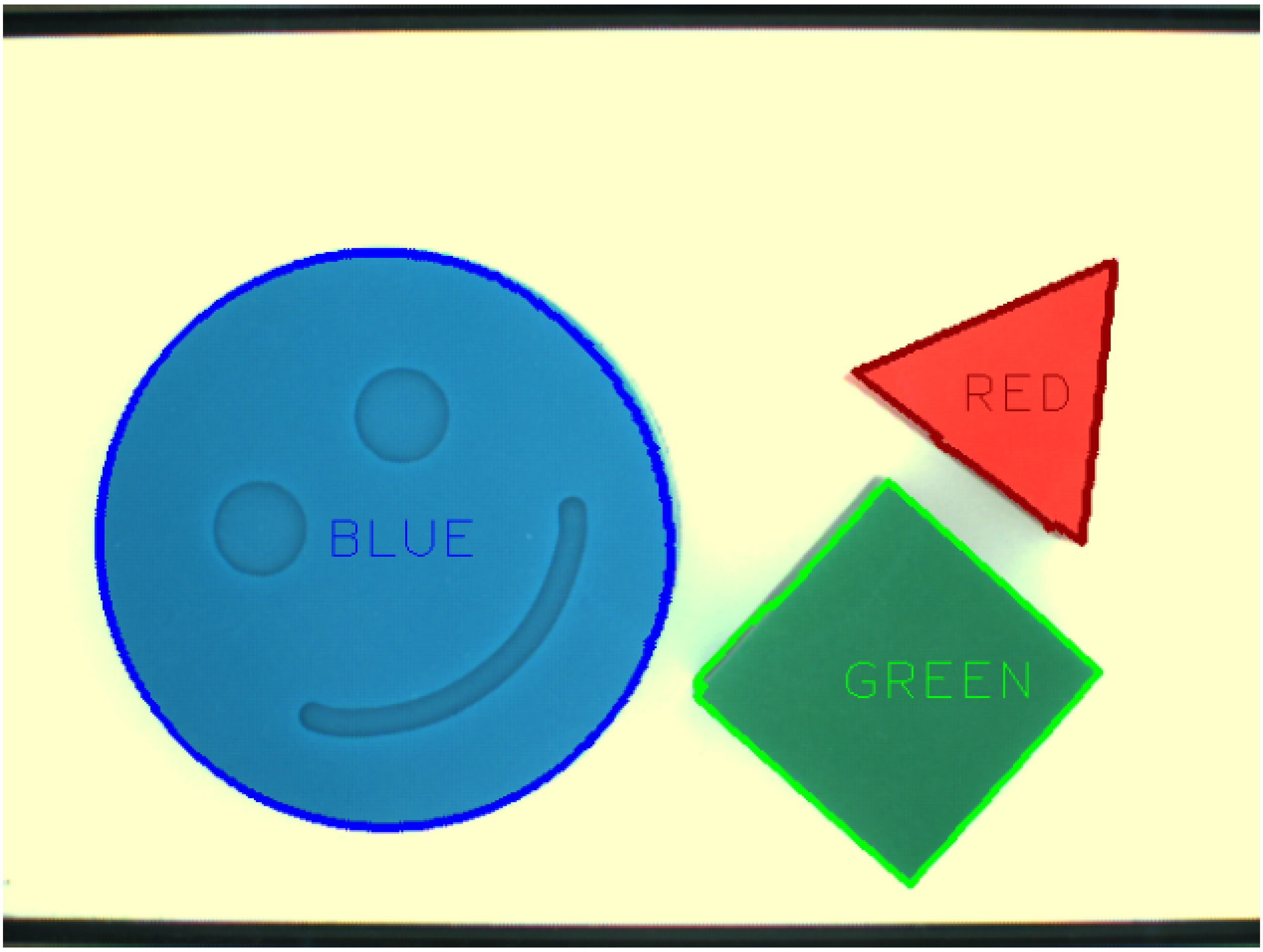
```
hr := F_VN_ConvertColorSpace(ipImageIn, ipImageRes, TCVN_CST_Bayer_RG_TO_RGB, hr);
```

然后就可以使用函数 F_VN_CheckColorRange [▶ 1285] 和相应的上下限阈值。

```
hr := F_VN_CheckColorRange(ipImageRes, ipImageWorkCol, aColorRefLow[iColor], aColorRefUp[iColor],
hr);
```



后续检测后的结果图像



类似1例程

- 与 RGB 参考色的颜色相似度 [▸ 2684]

8.1.9 测量

8.1.9.1 定位圆弧

在这个例子中，函数F_VN_LocateCircularArc [▸ 1435]用于定位和测量圆弧。该函数会返回一个包含中心、半径和边界角的 TcVnCircularArc 结构作为结果。如需测量完整的圆，则使用函数F_VN_LocateEllipse [▸ 1453]。

搜索的起始位置应近似对应于圆弧的预期圆心，在相应的样本图像中是 400，300。

在样本中，相应参数 aCenter 设定为 420，310，以证明此处近似位置就足够。该样本说明，可以补偿图像中待检物体的微小位置变化。

由于样本图像中的实际半径约为 200 像素，而且起点故意不准确，所以搜索半径 fRadius 被设定为 270。在具体的样本中，大约 230 个像素就足够了，但在实践中，有一个额外的安全缓冲区总是好的，例如，如果物体的位置比预期的偏差更大。

对于样本图像，必须选择搜索方向，以便在所有图像中都能正确找到物体。一个好的方向是左下，约 2356 rad（135° ）。为了说明这个方向不一定要准确，在样本中 fDirection 被设置为 2.1 rad（120.3° ）。

此外，从起点开始，从亮到暗的过渡，应被指定为搜索标准。此外，还必须指定最小的对比度，如果超过这个值，函数就会将相邻像素之间的强度差异检测为一个边缘。

如果这些参数不足以达到良好的效果，专家函数F_VN_LocateCircularArcExp [▸ 1438]提供了一些额外的参数，这在本样本中没有考虑。

变量

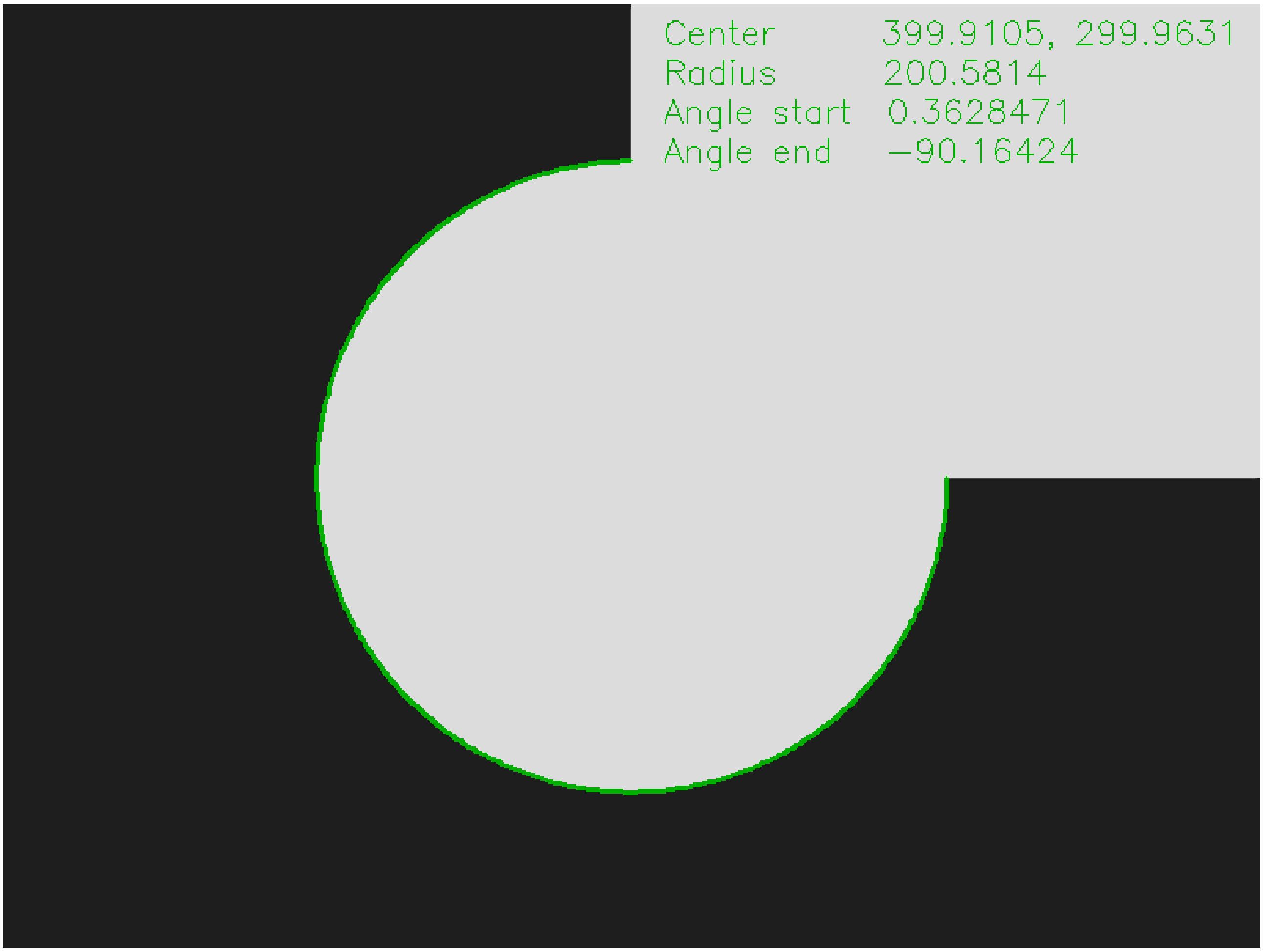
```
hr                : HRESULT;

ipImageIn         : ITcVnImage;
ipImageInDisp     : ITcVnDisplayableImage;
ipImageRes        : ITcVnImage;
ipImageResDisp    : ITcVnDisplayableImage;

// result
stArc             : TcVnCircularArc;

// input parameters (to specify where to start searching for the circular arc)
aCenter           : TcVnPoint2_REAL := [420, 310];
fRadius           : REAL  := 270;
fDirection        : LREAL  := 2.1;

// drawing
aColor            : TcVnVector4_LREAL := [0, 175, 0];
sText             : STRING(255);
```

代码

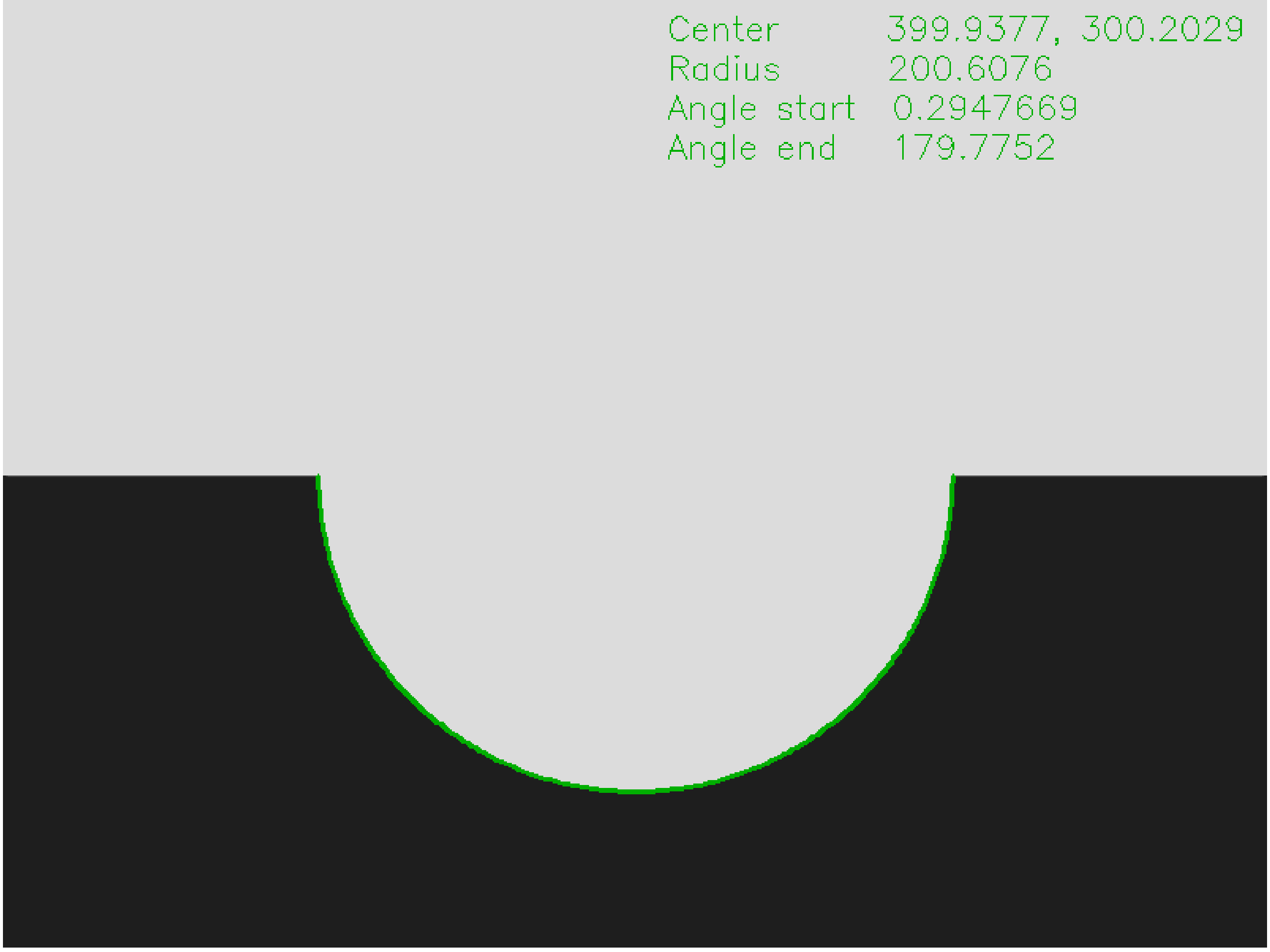
```
hr := F_VN_LocateCircularArc(
                    ipSrcImage          :=   ipImageIn,
                    stCircularArc       :=   stArc,
                    aCenterPoint        :=   aCenter,
                    fSearchRadius       :=   fRadius,
                    fArcDirectionRad    :=   fDirection,
                    eEdgeDirection      :=   TCVN_ED_LIGHT_TO_DARK,
                    fMinStrength        :=   100,
                    hrPrev              :=   hr);

// Draw result for visualization
hr := F_VN_ConvertColorSpace(ipImageIn, ipImageRes, TCVN_CST_GRAY_TO_RGB, hr);
hr := F_VN_DrawCircularArc(stArc, ipImageRes, aColor, 2, hr);
sText := CONCAT(CONCAT(CONCAT('Center ', REAL_TO_STRING(stArc.aCenter[0])), ', '),
REAL_TO_STRING(stArc.aCenter[1]));
hr := F_VN_PutText(sText, ipImageRes, 420, 25, TCVN_FT_HERSHEY_SIMPLEX, 0.7, aColor, hr);
sText := CONCAT('Radius ', REAL_TO_STRING(stArc.fRadius));
hr := F_VN_PutText(sText, ipImageRes, 420, 50, TCVN_FT_HERSHEY_SIMPLEX, 0.7, aColor, hr);
sText := CONCAT('Angle start ', REAL_TO_STRING(stArc.fStartAngle * 180 / LREAL_TO_REAL(PI)));
hr := F_VN_PutText(sText, ipImageRes, 420, 75, TCVN_FT_HERSHEY_SIMPLEX, 0.7, aColor, hr);
sText := CONCAT('Angle end ', REAL_TO_STRING(stArc.fEndAngle * 180 / LREAL_TO_REAL(PI)));
hr := F_VN_PutText(sText, ipImageRes, 420, 100, TCVN_FT_HERSHEY_SIMPLEX, 0.7, aColor, hr);

// Display source and result image
hr := F_VN_TransformIntoDisplayableImage(ipImageIn, ipImageInDisp, S_OK);
hr := F_VN_TransformIntoDisplayableImage(ipImageRes, ipImageResDisp, S_OK);
```

结果

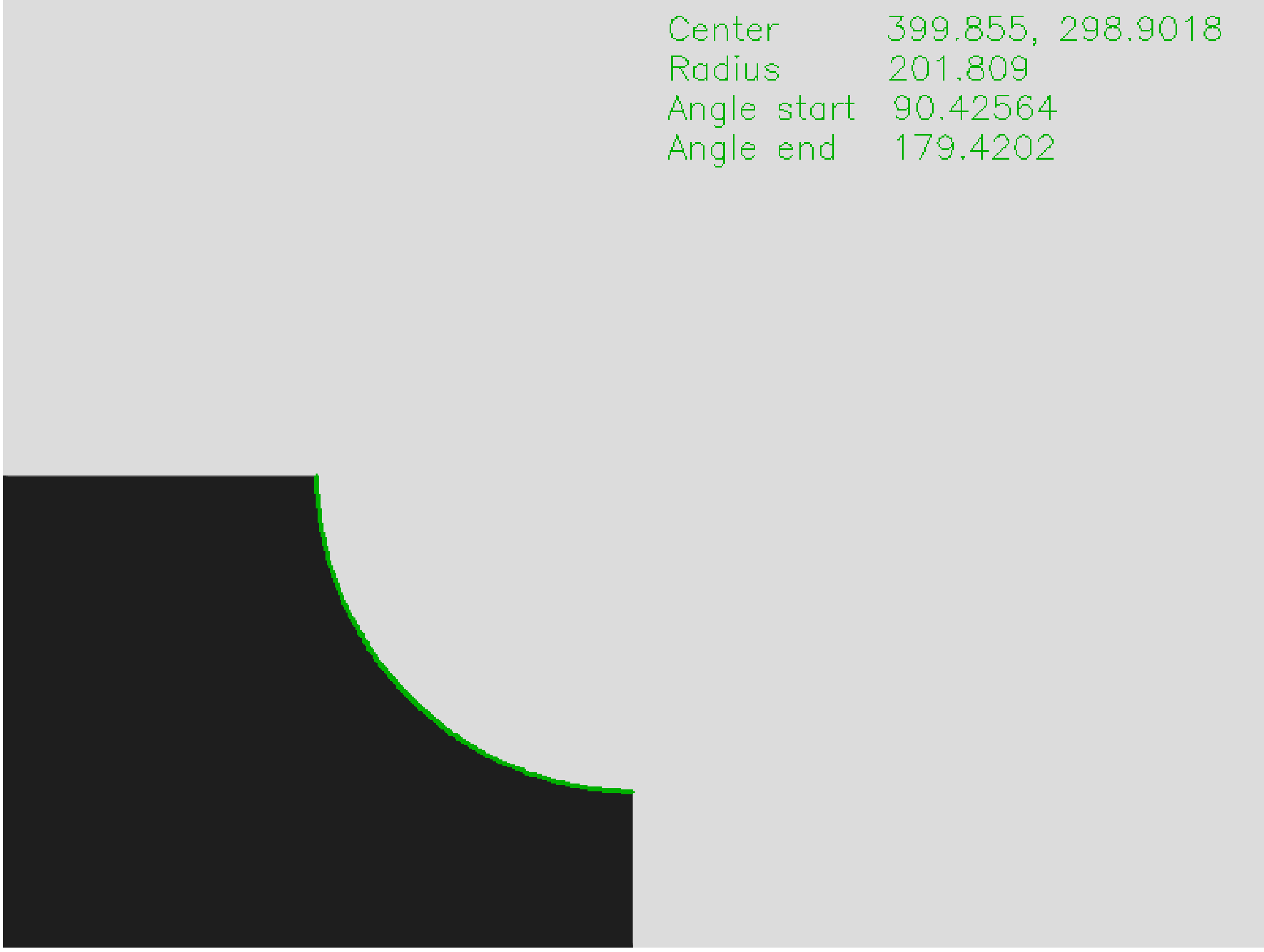
为了实现可视化，使用函数F_VN_DrawCircularArc [▸ 963]将发现的圆弧绘制到结果图像中。基本数值也显示为文本。



附图 28: LocateCircularArc_3QuarterCircle_Result



附图 29: LocateCircularArc_HalfCircle_Result



附图 30: LocateCircularArc_QuarterCircle_Result

类似样本

- 定位边缘 [▶ 2699]
- 定位椭圆 [▶ 2704]
- 测量边缘之间的角度 [▶ 2709]

8.1.9.2 定位边缘

在这个样本中,

- 函数F_VN_LocateEdgeExp [▶ 1444]用于在定义的搜索窗口中定位边缘,
- 且执行时间通过看门狗进行监控,并在必要时加以限制。

说明

通过函数F_VN_LocateEdge [▶ 1441],搜索窗口可用于定位边缘。为了简单起见,一些参数固定不变。另外,此处所用函数F_VN_LocateEdgeExp [▶ 1444]可以完全访问所有参数。这个样本旨在说明各个参数的情况。建议在标准配置之外尝试配置变化,并考虑对于边缘结果和处理时间的影响。在"结果"中可以找到包括描述在内的部分配置变化。

变量

```
hr                : HRESULT;
hrFunc            : HRESULT;

ipImageIn         : ITcVnImage;
ipImageInDisp     : ITcVnDisplayableImage;
ipImageRes        : ITcVnImage;
ipImageResDisp    : ITcVnDisplayableImage;
```

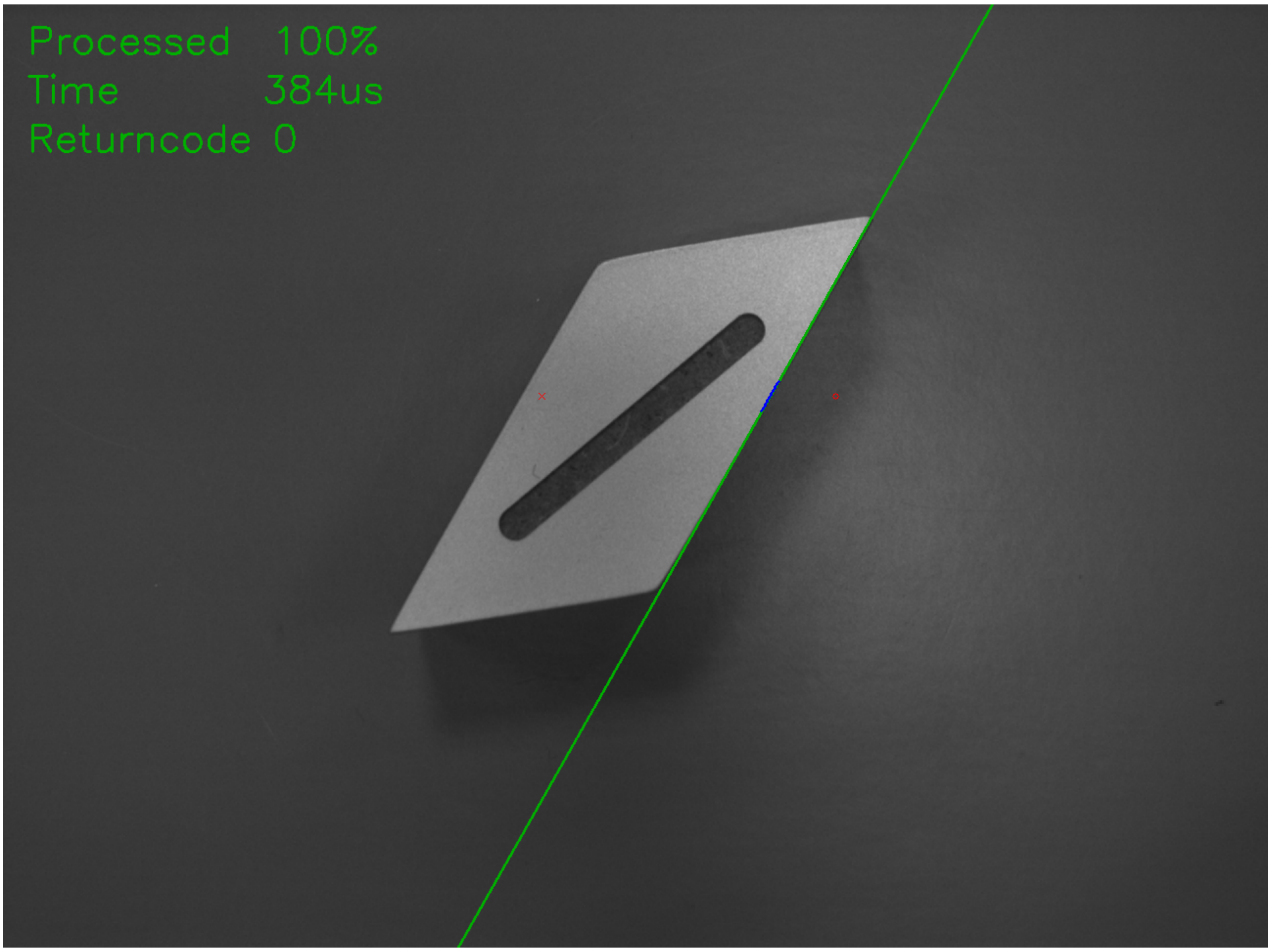
```
// result
ipEdgePoints    : ITcVnContainer;

// parameters
aStartPoint     : TcVnPoint2_REAL := [850, 400];
aEndPoint       : TcVnPoint2_REAL := [550, 400];
eDirection      : ETcVnEdgeDirection := TCVN_ED_DARK_TO_LIGHT;
fMinStrength    : REAL := 50;
nSearchLines    : UDINT := 31;
fSearchLineDist : REAL := 1;
nMaxThickness   : UDINT := 7;
nSubpixIter     : UDINT := 10;
eAlgorithm      : ETcVnEdgeDetectionAlgorithm := TCVN_EDA_INTERPOLATION;
fAvgStrength    : REAL;

// Watchdog
hrWD            : HRESULT;
tStop           : DINT := 15000;
tRest           : DINT;
nFraction       : UDINT;

// drawing
aLine           : TcVnVector4_LREAL;
aColorGreen     : TcVnVector4_LREAL := [0, 175, 0];
aColorBlue      : TcVnVector4_LREAL := [0, 0, 255];
aColorRed       : TcVnVector4_LREAL := [255, 0, 0];
sText           : STRING(255);
```

代码

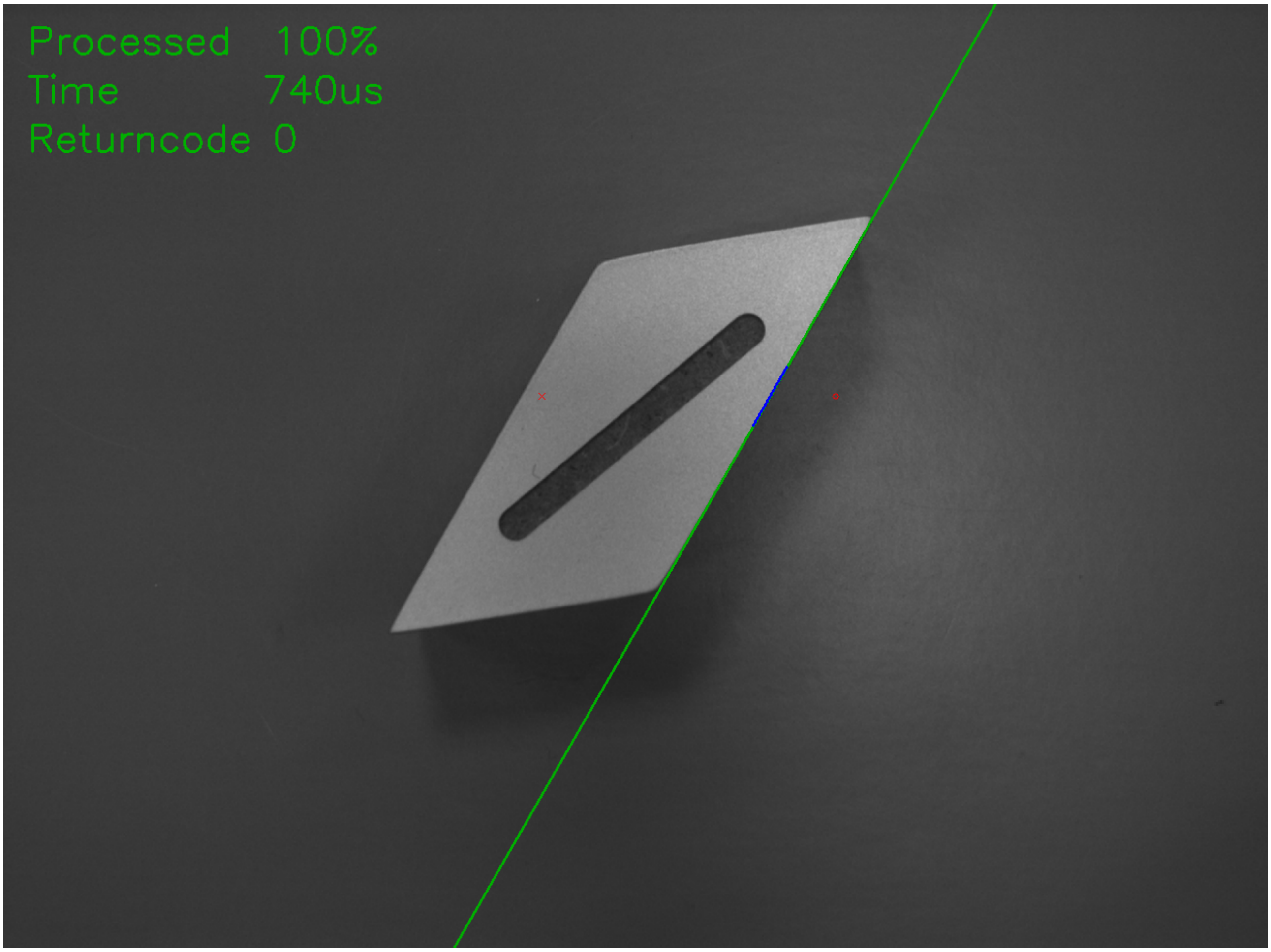
```
hrWD := F_VN_StartRelWatchdog(tStop, hr);
   hrFunc  := F_VN_LocateEdgeExp(
                  ipSrcImage              :=   ipImageIn,
                  ipEdgePoints            :=   ipEdgePoints,
                  aStartPoint             :=   aStartPoint,
                  aEndPoint               :=   aEndPoint,
                  eEdgeDirection          :=   eDirection,
                  fMinStrength            :=   fMinStrength,
                  nSearchLines            :=   nSearchLines,
                  fSearchLineDist         :=   fSearchLineDist
                  nMaxThickness           :=   nMaxThickness,
                  nSubpixelsIterations    :=   nSubpixIter,
                  fApproxPrecision        :=   0.0001,
                  eAlgorithm              :=   eAlgorithm,
                  hrPrev                  :=   hr,
                  fAvgStrength            =>   fAvgStrength);
hrWD := F_VN_StopWatchdog(hrWD, nFractionProcessed=>nFraction, tRest=>tRest);

// Draw result for visualization
hr := F_VN_ConvertColorSpace(ipImageIn, ipImageRes, TCVN_CST_GRAY_TO_RGB, hr);
sText := CONCAT(CONCAT('Processed ', UDINT_TO_STRING(nFraction)), '%');
hr := F_VN_PutTextExp(sText, ipImageRes, 25, 50, TCVN_FT_HERSHEY_SIMPLEX, 1.3, aColorGreen, 2,
TCVN_LT_8_CONNECTED, FALSE, hr);
sText := CONCAT(CONCAT('Time ', DINT_TO_STRING(tStop - tRest)), 'us');
hr := F_VN_PutTextExp(sText, ipImageRes, 25, 100, TCVN_FT_HERSHEY_SIMPLEX, 1.3, aColorGreen, 2,
TCVN_LT_8_CONNECTED, FALSE,hr);
sText := CONCAT('Returncode ', DINT_TO_STRING(hrFunc));
hr := F_VN_PutTextExp(sText, ipImageRes, 25, 150, TCVN_FT_HERSHEY_SIMPLEX, 1.3, aColorGreen, 2,
TCVN_LT_8_CONNECTED, FALSE,hr);
hr := F_VN_DrawPoint(REAL_TO_UDINT(aStartPoint[0]), REAL_TO_UDINT(aStartPoint[1]), ipImageRes,
TCVN_DS_CIRCLE, aColorRed, hr);
hr := F_VN_DrawPoint(REAL_TO_UDINT(aEndPoint[0]), REAL_TO_UDINT(aEndPoint[1]), ipImageRes,
TCVN_DS_X, aColorRed, hr);
hr := F_VN_FitLine(ipEdgePoints, aLine, hr);
hr := F_VN_DrawLine_TcVnVector4_LREAL(aLine, ipImageRes, aColorGreen, 2, hr);
hr := F_VN_DrawPointsExp(ipEdgePoints, ipImageRes, TCVN_DS_PLUS, aColorBlue, 1, 1,
TCVN_LT_8_CONNECTED, hr);

// Display source and result image
hr := F_VN_TransformIntoDisplayableImage(ipImageIn, ipImageInDisp, S_OK);
hr := F_VN_TransformIntoDisplayableImage(ipImageRes, ipImageResDisp, S_OK);
```

结果

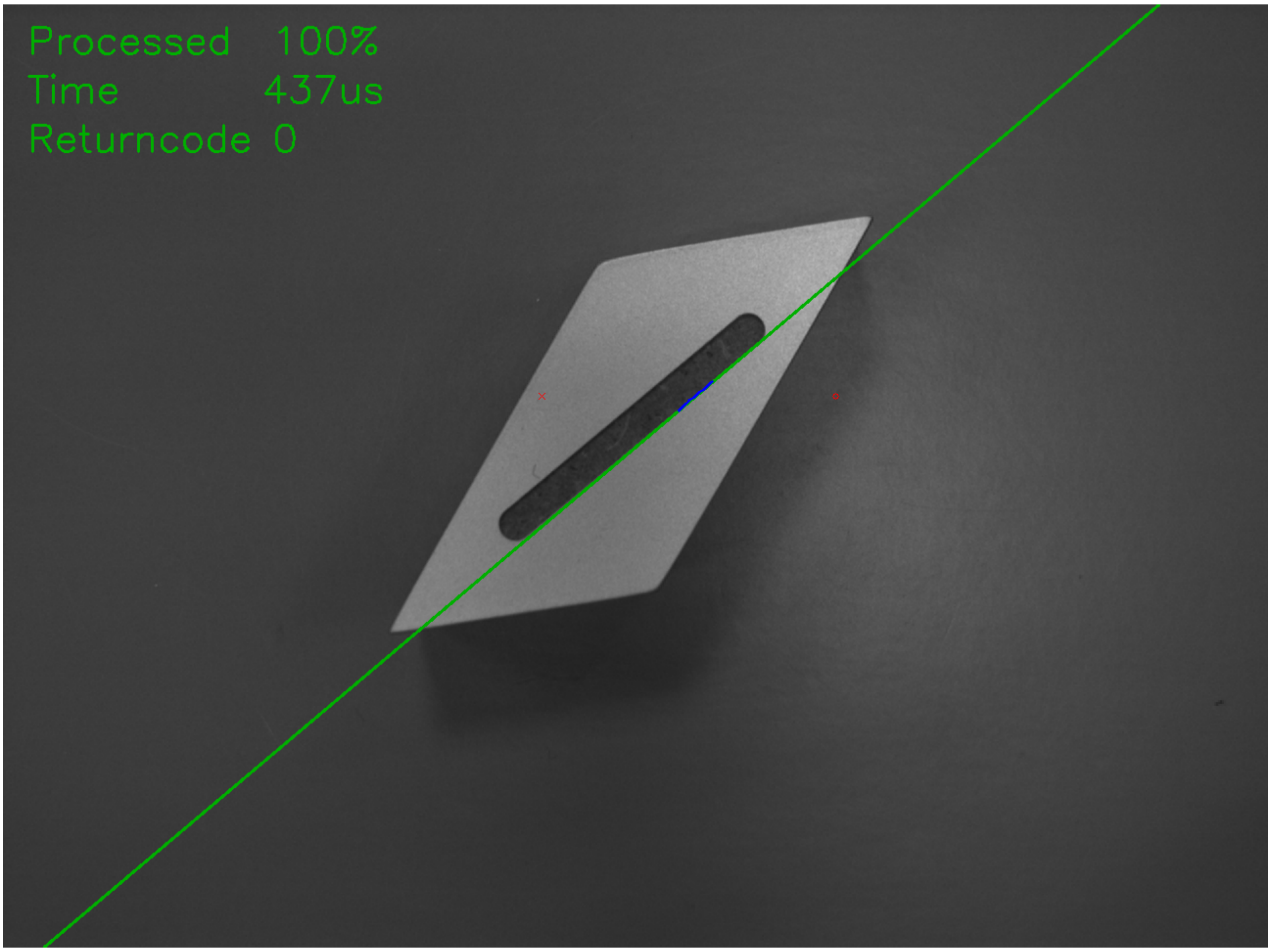
为了实现可视化，发现的边缘点以蓝色显示，并在输出图像中以绿色画出回归线。此外，还画出了起点（红圈）和终点（红 X）。处理百分比、所需执行时间和函数返回值都显示在左上角。
对于这个样本中使用的参数，结果如下：



如果在使用参数nSearchLines的情况下搜索线的数量从 31 增加到 61 条，所需的计算时间也会翻倍。作为回报，回归线变得更加精确：



如果想要找到组件的铣边而不是外边，请将参数eDirection改为TCVN_ED_LIGHT_TO_DARK。这就忽略了从暗到亮的外缘过渡，而是找到了从亮到暗过渡的以下边缘：



缩短执行时间

值得注意的是，尽管参数相同，但执行时间却比定位外缘时要高。这是因为所发现的边缘离起点更远。

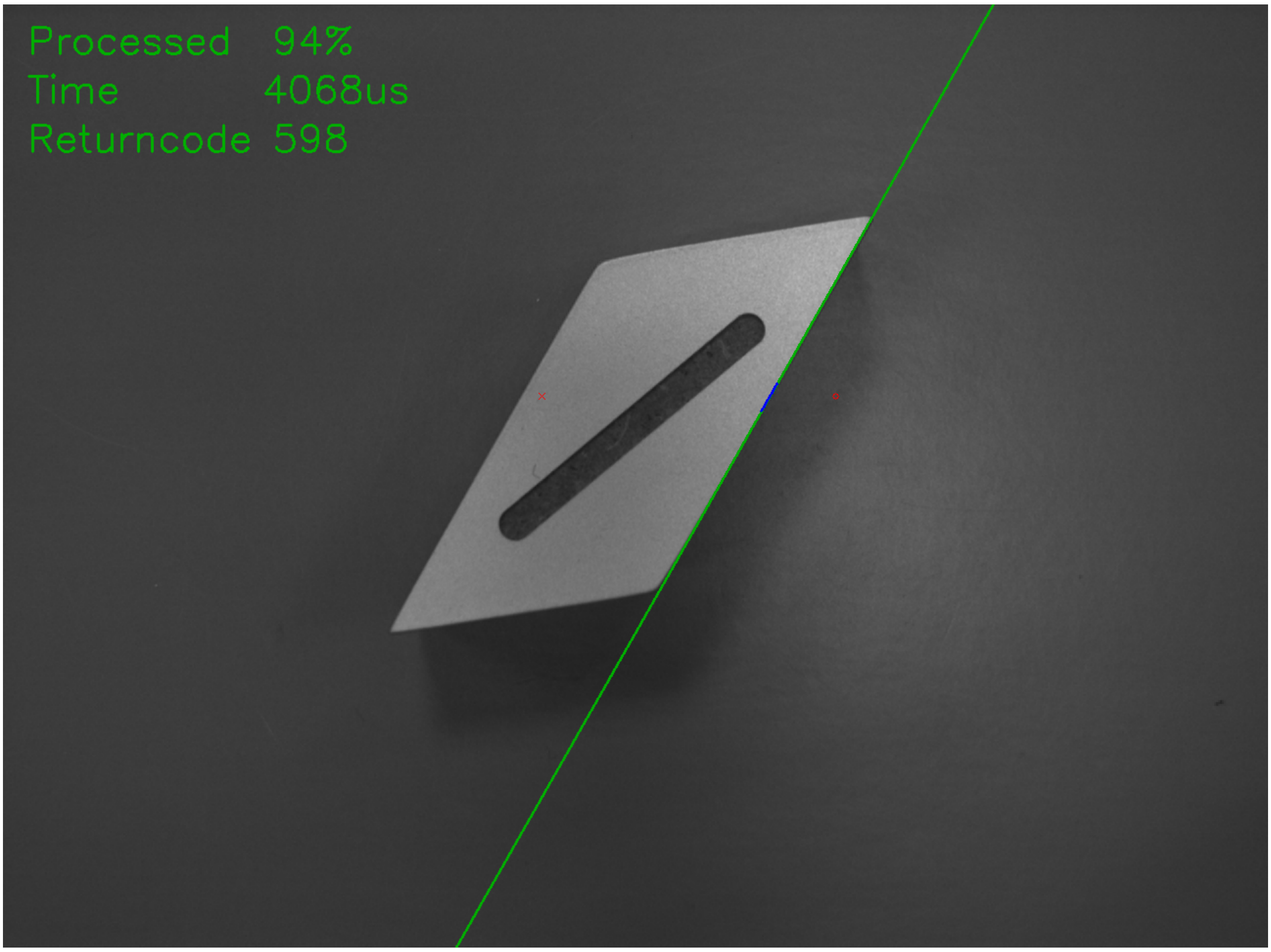
相反，这意味着固定搜索窗口的执行时间根据组件在图像中的位置而略有不同，这在三幅样本图像中也可以看到。

两种近似算法会出现更大的波动，因为它们需要不同的迭代来使用周围的像素强度近似模型参数。

一般来说，如果搜索窗口的搜索线尽可能正交地击中边缘，则需要更少的迭代。如果图像中物体的位置和方向波动很小或没有波动，可以相应地调整搜索窗口以实现更短的执行时间。此外，可以通过减少最大迭代次数来减少所需的最大时间，但这可能会导致结果不太准确。

使用看门狗

此外，可以使用外部看门狗限制时间。如有必要，看门狗会终止函数的执行并返回现有的部分结果。例如，如果算法改为TCVN_EDA_APPROX_ERF且最大迭代次数设置为 60 次，图像 LocateEdge2.bmp（外缘与搜索线正交）的执行几乎正好需要 4 ms（取决于安装的 CPU），而其他两个图像则需要 4.3 和 4.4 毫秒。如果参数 tStop设置为 4000，函数执行会在 4 ms 后尽快停止，并返回到那时为止的部分结果。换句话说，图像 LocateEdge2.bmp 上的函数继续被 100% 处理，而另外两幅图像上的函数（本例中为 LocateEdge3.bmp）则被提前终止：



由于只能在算法中的某些点上中止，并返回部分结果（以前在搜索线上找到的边缘点），因此该函数不是在精确的 4000 µs 后终止，而是需要更长的时间，这在选择中止时间时必须考虑到。所需的最大额外时间一般取决于算法和参数化。

尽管终止，但部分结果足以达到预期的效果，因为只有一或两条搜索线的结果缺失。

类似样本

- 定位圆弧 [▶ 2695]
- 定位椭圆 [▶ 2704]
- 测量边缘之间的角度 [▶ 2709]

8.1.9.3 定位椭圆

在这个样本中，

- 函数F_VN_LocateEllipseExp [▶ 1455]用于在确定的搜索区域内定位一个圆形物体
- 且执行时间通过看门狗进行监控，并在必要时加以限制。

说明

为了对测量功能及其参数有一个基本的了解，请先遵守定位边缘样本 [▶ 2699]的说明。函数F_VN_LocateEllipseExp [▶ 1455]与F_VN_LocateEdgeExp [▶ 1441]不同的是，它使用一个圆形的搜索窗口，而不是矩形窗口。在这个样本中，一个大的、不准确的搜索窗口被参数化了，因为圆形物体的位置略有变化。在所有图像中都可以用相同的搜索窗口来确定圆形。

在这个样本中，参数bInvertSearchDirection（设置为TRUE）用于确定搜索方向相反，即从外部向aCenter。这是必要的，因为样本图像也包含系统寻找的椭圆内部边缘。

搜索方向设置为TCVN_ED_LIGHT_TO_DARK，因为在这个样本中，物体比背景更暗。

参数nMaxThickness（定义必须达到fMinStrength的像素数）设置为“7”。

变量

```
hr              : HRESULT;
hrFunc          : HRESULT;

ipImageIn       : ITcVnImage;
ipImageInDisp   : ITcVnDisplayableImage;
ipImageRes      : ITcVnImage;
ipImageResDisp  : ITcVnDisplayableImage;

// result
stEllipse       : TcVnRotatedRectangle;
ipContourPoints : ITcVnContainer;

// input parameters
aCenter         : TcVnPoint2_REAL := [650, 400];
fRadius         : REAL := 300;
fMinRadius      : REAL := 100;
fMinStrength    : REAL := 30;
nSubPixIter     : UDINT := 10;
nSearchLines    : UDINT := 40;
eAlgorithm      : ETcVnEdgeDetectionAlgorithm := TCVN_EDA_INTERPOLATION;

// Watchdog
hrWD            : HRESULT;
tStop           : DINT := 15000;
tRest           : DINT;
nFraction       : UDINT;

// drawing
aColorRed       : TcVnVector4_LREAL := [200, 0, 0];
aColorGreen     : TcVnVector4_LREAL := [0, 175, 0];
sText           : STRING(255);
```

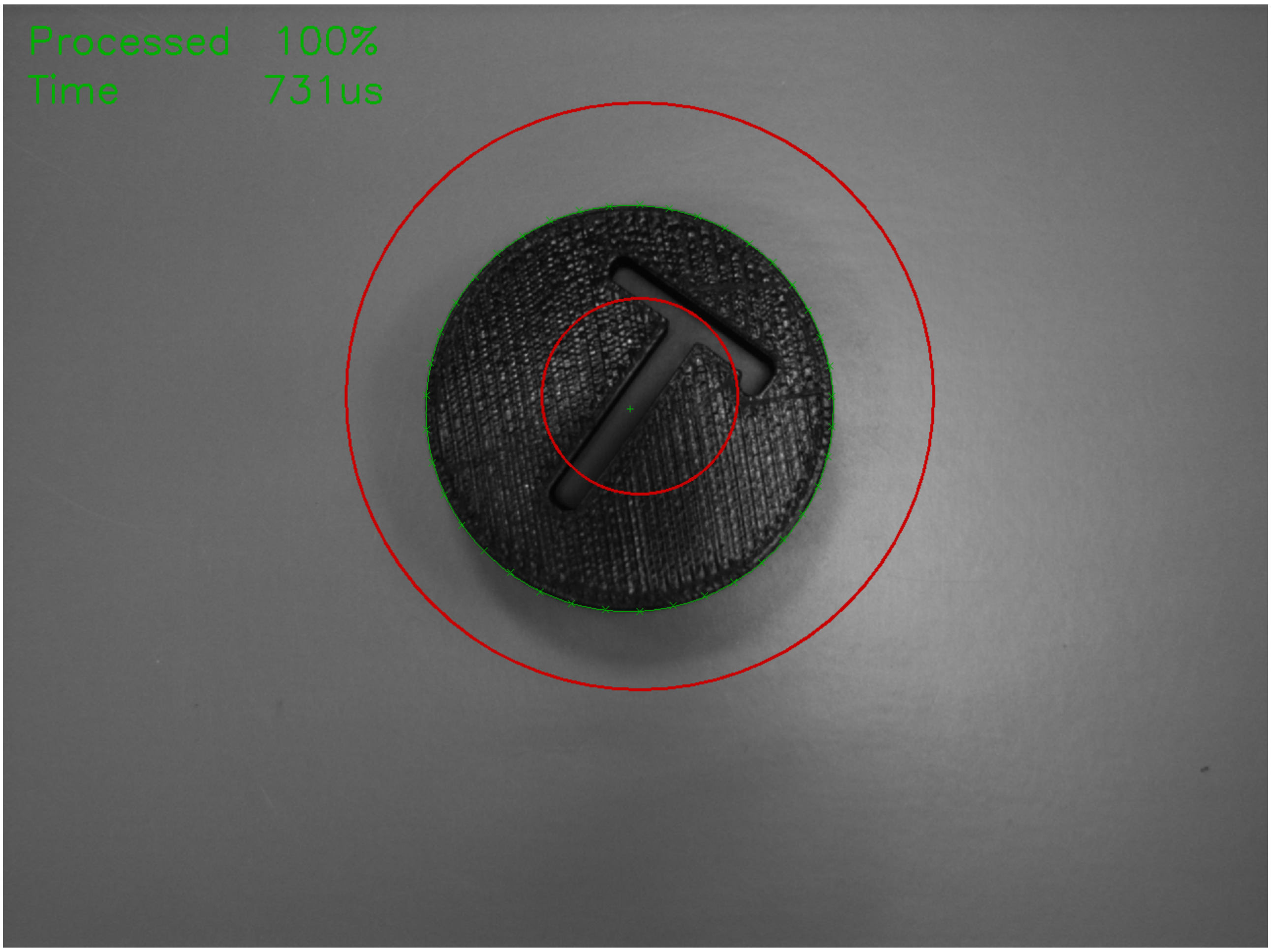
代码

```
hrWD := F_VN_StartRelWatchdog(tStop, hr);
   hrFunc  := F_VN_LocateEllipseExp(
             ipSrcImage                :=   ipImageIn,
             stEllipse                 :=   stEllipse,
             aCenterPoint              :=   aCenter,
             fSearchRadius             :=   fRadius,
             eEdgeDirection            :=   TCVN_ED_LIGHT_TO_DARK,
             fMinStrength              :=   fMinStrength,
             nMaxThickness             :=   7,
             bInvertSearchDirection    :=   TRUE,
             fMinSearchRadius          :=   fMinRadius,
             nSubpixelsIterations      :=   nSubpixIter,
             nSearchLines              :=   nSearchLines,
             fApproxPrecision          :=   0.0001,
             eAlgorithm                :=   eAlgorithm,
             ipContourPoints           :=   ipContourPoints,
             hrPrev                    :=   hr);
hrWD := F_VN_StopWatchdog(hrWD, nFractionProcessed=>nFraction, tRest=>tRest);

// Draw result for visualization
hr  := F_VN_ConvertColorSpace(ipImageIn, ipImageRes, TCVN_CST_GRAY_TO_RGB, hr);
sText  := CONCAT(CONCAT('Processed ', UDINT_TO_STRING(nFraction)), '%');
hr  := F_VN_PutTextExp(sText, ipImageRes, 25, 50, TCVN_FT_HERSHEY_SIMPLEX, 1.3, aColorGreen, 2,
TCVN_LT_8_CONNECTED, FALSE, hr);
sText  := CONCAT(CONCAT('Time ', DINT_TO_STRING(tStop - tRest)), 'us');
hr  := F_VN_PutTextExp(sText, ipImageRes, 25, 100, TCVN_FT_HERSHEY_SIMPLEX, 1.3, aColorGreen, 2,
TCVN_LT_8_CONNECTED, FALSE,hr);
hr  := F_VN_DrawCircle(REAL_TO_UDINT(aCenter[0]), REAL_TO_UDINT(aCenter[1]),
REAL_TO_UDINT(fMinRadius), ipImageRes, aColorRed, 2, hr);
hr  := F_VN_DrawCircle(REAL_TO_UDINT(aCenter[0]), REAL_TO_UDINT(aCenter[1]), REAL_TO_UDINT(fRadius),
ipImageRes, aColorRed, 2, hr);
hr  := F_VN_DrawPoints(ipContourPoints, ipImageRes, TCVN_DS_X, aColorGreen, hr);
hr  := F_VN_DrawPoint(REAL_TO_UDINT(stEllipse.aCenter[0]), REAL_TO_UDINT(stEllipse.aCenter[1]),
ipImageRes, TCVN_DS_PLUS, aColorGreen, hr);
hr  := F_VN_DrawEllipse(stEllipse, ipImageRes, aColorGreen, 1, hr);

// Display source and result image
hr  := F_VN_TransformIntoDisplayableImage(ipImageIn, ipImageInDisp, S_OK);
hr  := F_VN_TransformIntoDisplayableImage(ipImageRes, ipImageResDisp, S_OK);
```

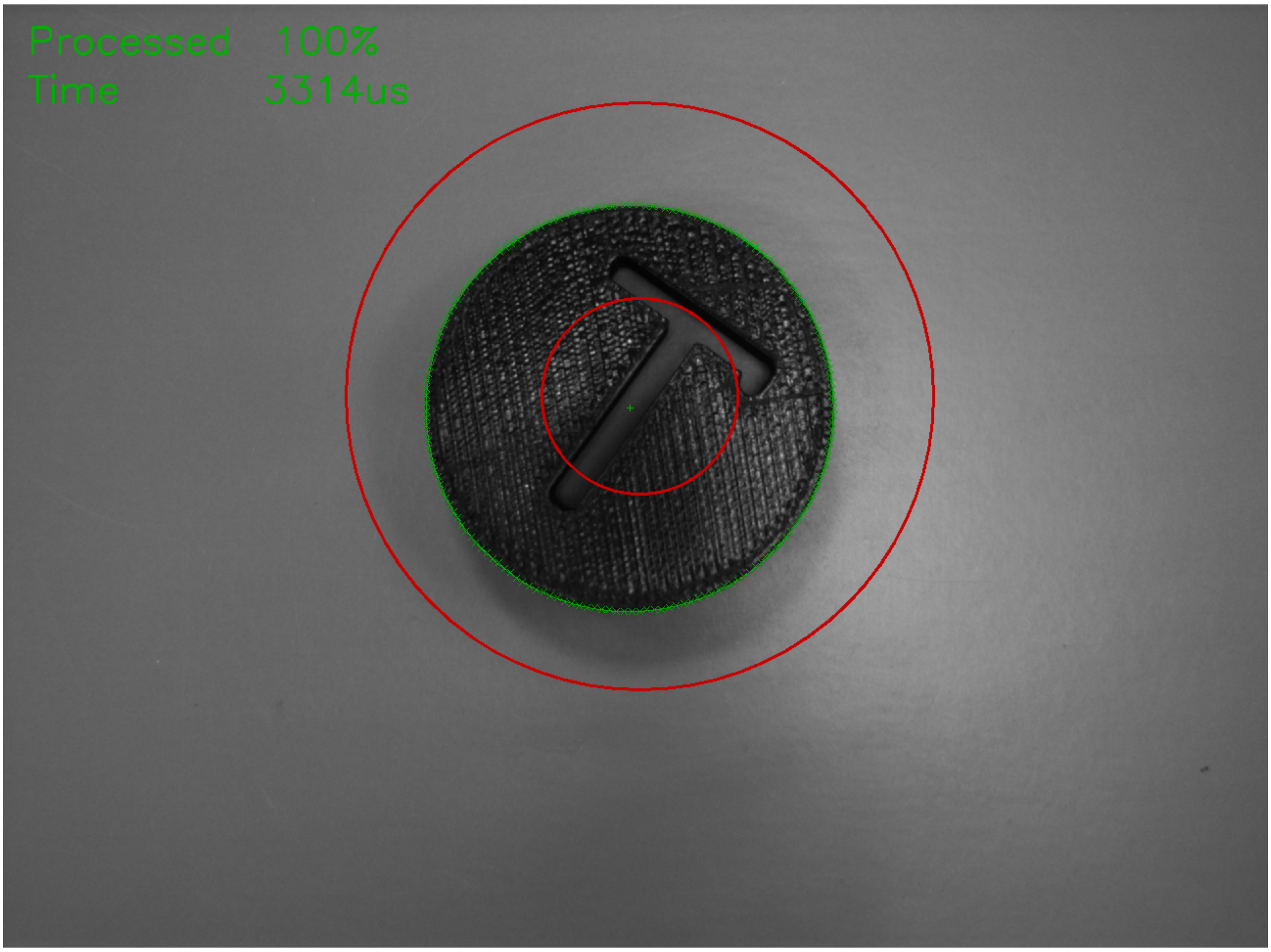
结果

该函数返回TcVnRotatedRectangle结构stEllipse，而该结构定义了由其中心aCenter、其高度和宽度stSize.fHeight/stSize.fWidth和其角度fAngle找到的椭圆。
根据选择，每条搜索线ipContourPoints可以返回一个额外的边缘点，从这个边缘点接近椭圆。

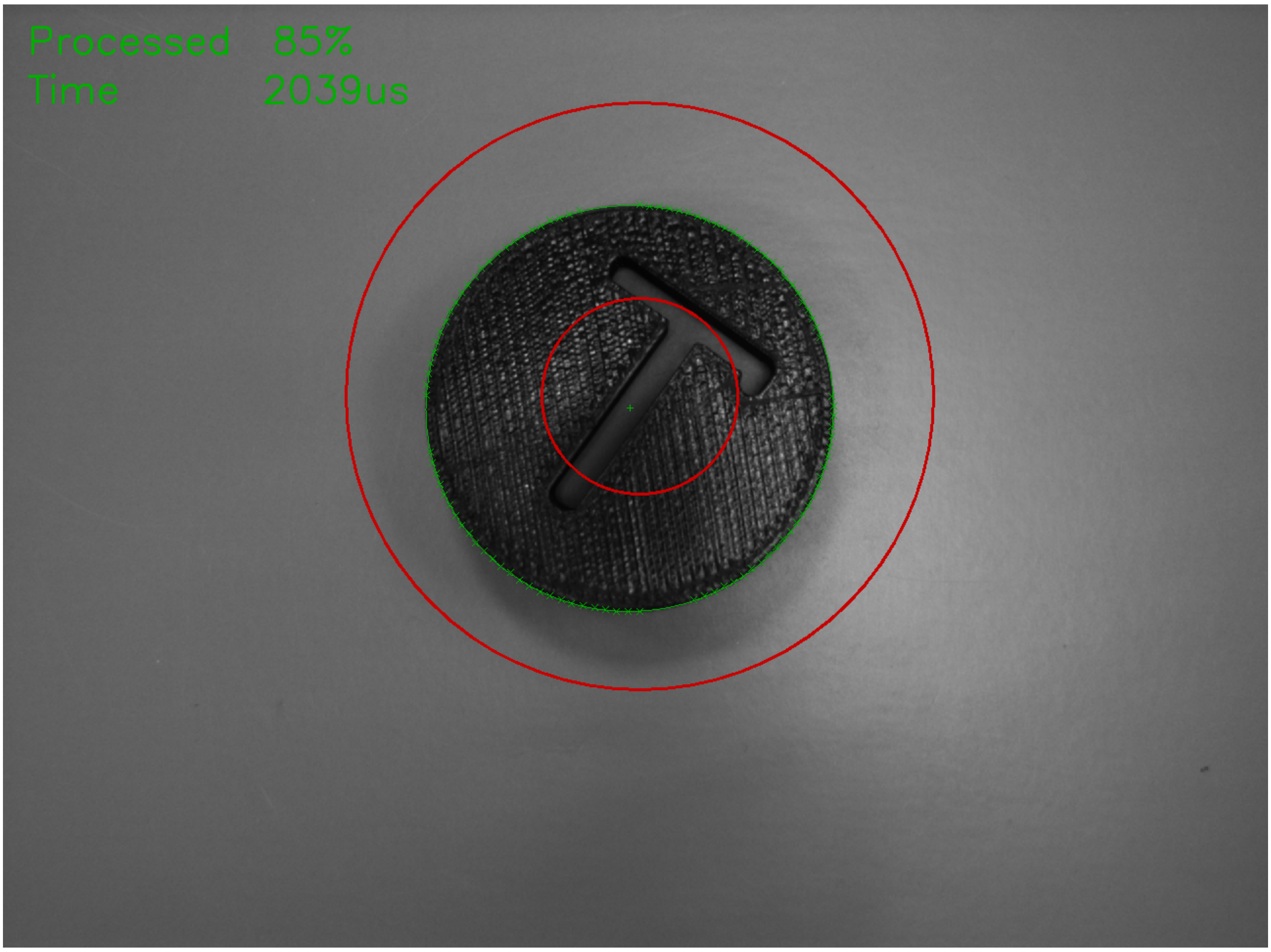
为了实现可视化，搜索区域首先以围绕指定中心点的两个红色圆圈的形式被画进图像。找到的椭圆被画成一条绿色的线，而对应的中心绘制绿色“+”。来自ipContourPoints的各个测量点被画成绿色 x。此外，处理百分比和所需计算时间也显示在左上方。对于本样本中使用的参数，输入图像*LocateEllipse_T.bmp*的结果如下：



如果搜索线的数量从 40 条增加到 180 条，所需计算时间将如预期的那样线性增加：

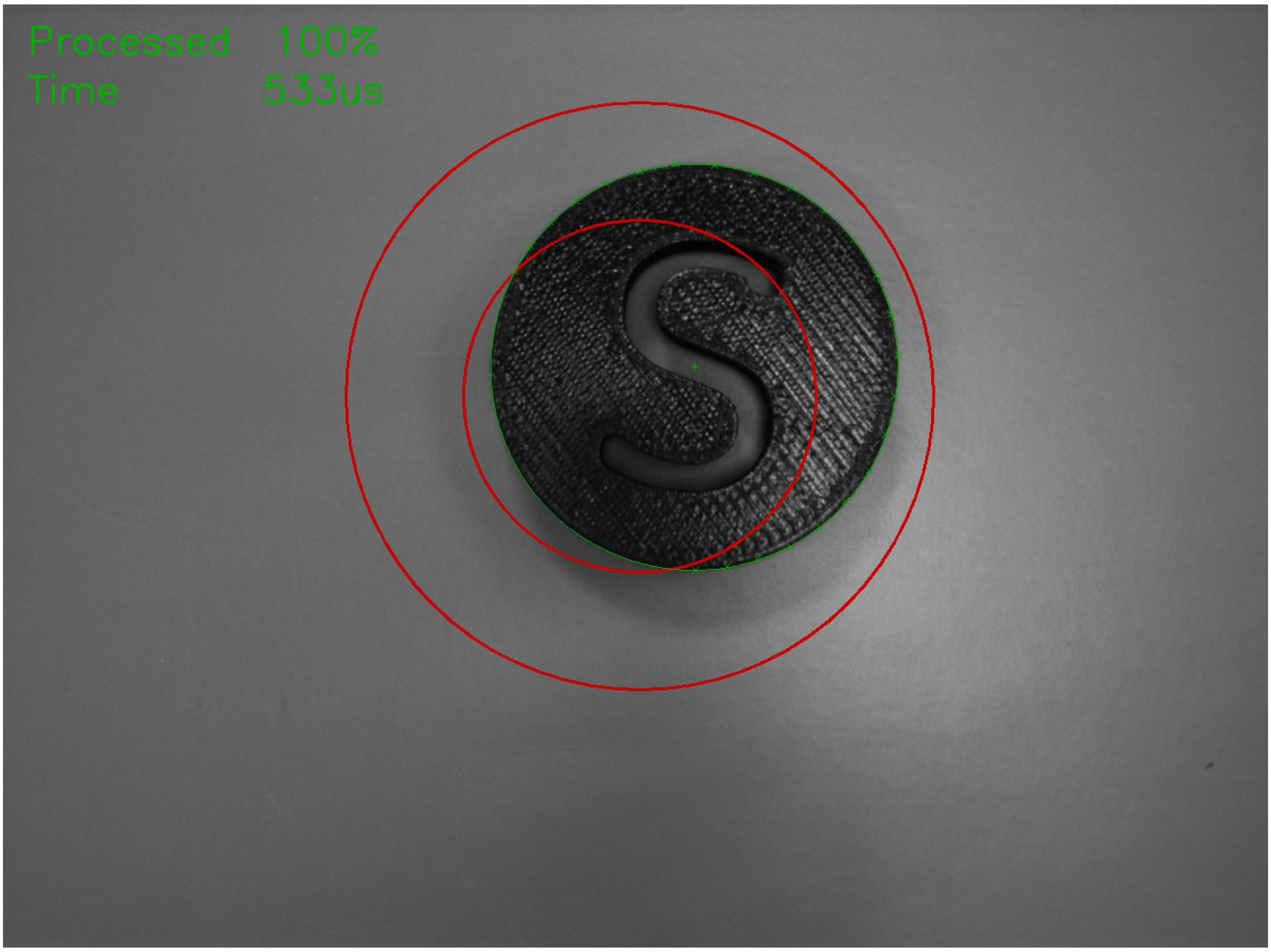


在实践中，根据输入图像，计算时间会有一些变化。因此，在许多情况下，通过看门狗来限制计算时间很有用。例如，如果搜索线数设置为 120 且看门狗超时tStop设置为 2000（取决于所使用的 CPU），那么该函数将在处理时间的 85% 左右终止，并返回此时可用的部分结果：



由于该算法只能在特定的时间点中断并返回部分结果，因此该函数并不是在 2000 µs 后准确终止，而是需要更长的时间，在选择终止时间时必须考虑到这一点。所需的最大额外时间一般取决于算法和参数化。
尽管一些搜索线的结果由于终止而缺失（绿色 x 之间四个相对较大的空隙），但椭圆仍然可以被正确地近似，即部分结果可以像函数被 100% 处理一样使用。

如果一个物体的位置部分在定义的搜索范围之外，例如因为之前的定位步骤不正确，这也可以在有限的范围内实现。例如，如果fMinRadius被设置为 180，一些样本图像的部分外轮廓就会位于指定范围之外。尽管如此，还是有足够的搜索点来近似一个椭圆。然而，在实践中，不应该依赖这一点，而应该将搜索区域设置得足够大：



类似样本

- 定位圆弧 [▶ 2695]
- 定位边缘 [▶ 2699]
- 测量边缘之间的角度 [▶ 2709]

8.1.9.4 测量边缘之间的角度

在这个样本中，

- 函数F_VN_MeasureAngleBetweenEdgesExp [▶ 1460]用于测量一个物体的两个边缘之间的角度
- 且执行时间通过看门狗进行监控，并在必要时加以限制。

说明

在定义了搜索窗口后，搜索方向设置为TCVN_ED_LIGHT_TO_DARK，因为在这个样本中，搜索由内向外进行，且物体比背景颜色更浅。

参数nMaxThickness（定义必须达到fMinStrength的像素数）设置为“7”。

变量

```
hr              : HRESULT;
hrFunc          : HRESULT;

ipImageIn       : ITcVnImage;
ipImageInDisp   : ITcVnDisplayableImage;
ipImageRes      : ITcVnImage;
ipImageResDisp  : ITcVnDisplayableImage;

// result
fAngle          : REAL;
ipEdgePoints1   : ITcVnContainer;
```

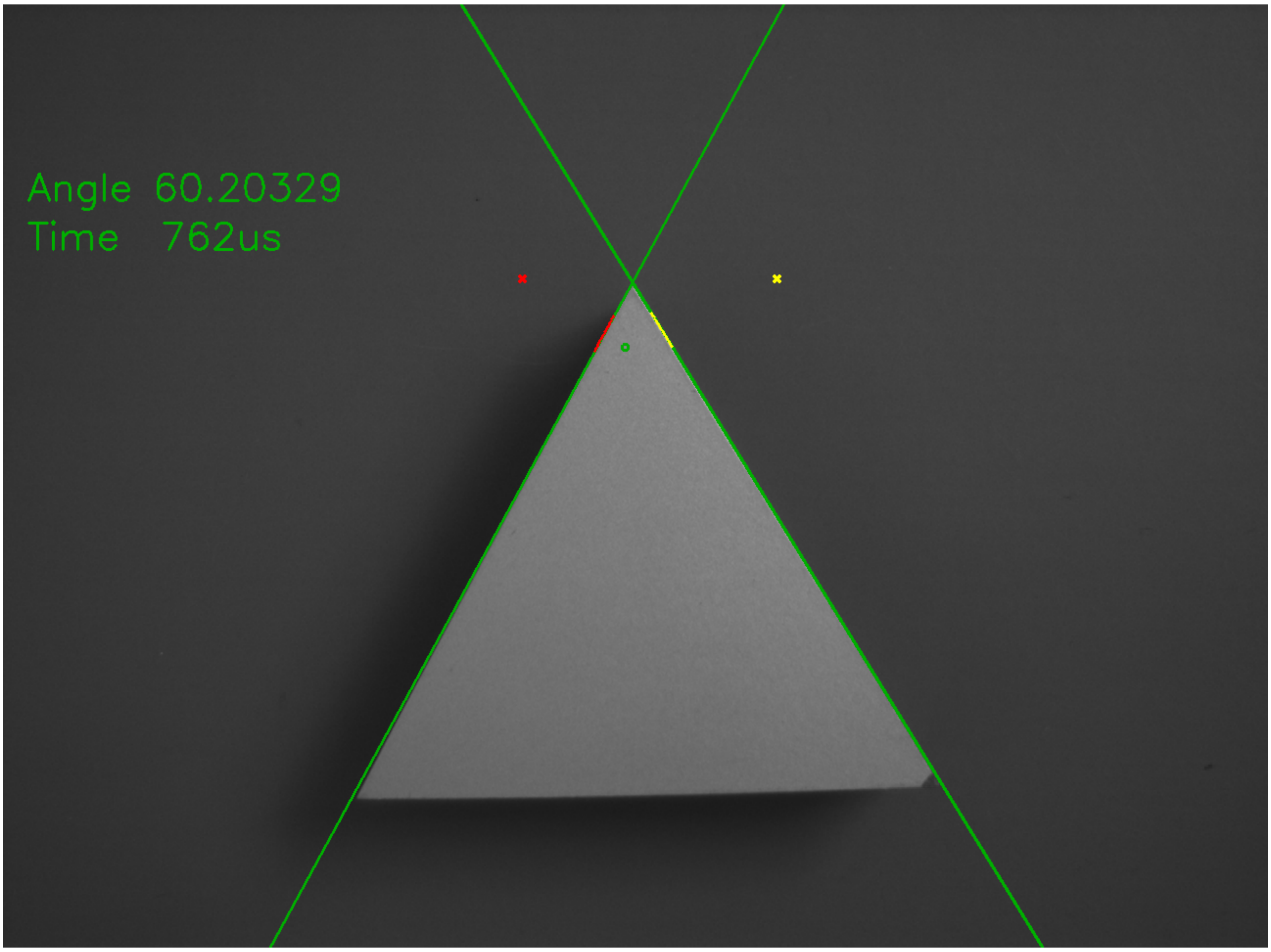
```
ipEdgePoints2    : ITcVnContainer;

// parameters
aInnerPoint      : TcVnPoint2_REAL := [635, 350];
aOuterPoint1     : TcVnPoint2_REAL := [530, 280];
aOuterPoint2     : TcVnPoint2_REAL := [790, 280];
fMinStrength     : REAL := 50;
nSearchLines     : UDINT := 41;
fSearchLineDist  : REAL := 1;
nSubpixIter      : UDINT := 10;
eAlgorithm       : ETcVnEdgeDetectionAlgorithm := TCVN_EDA_INTERPOLATION;

// Watchdog
hrWD             : HRESULT;
tStop            : DINT := 15000;
tRest            : DINT;
nFraction        : UDINT;

// drawing
aLine1           : TcVnVector4_LREAL;
aLine2           : TcVnVector4_LREAL;
aColorRed        : TcVnVector4_LREAL := [255, 0, 0];
aColorGreen      : TcVnVector4_LREAL := [0, 175, 0];
aColorYellow     : TcVnVector4_LREAL := [255, 255, 0];
sText            : STRING(255);
```

代码

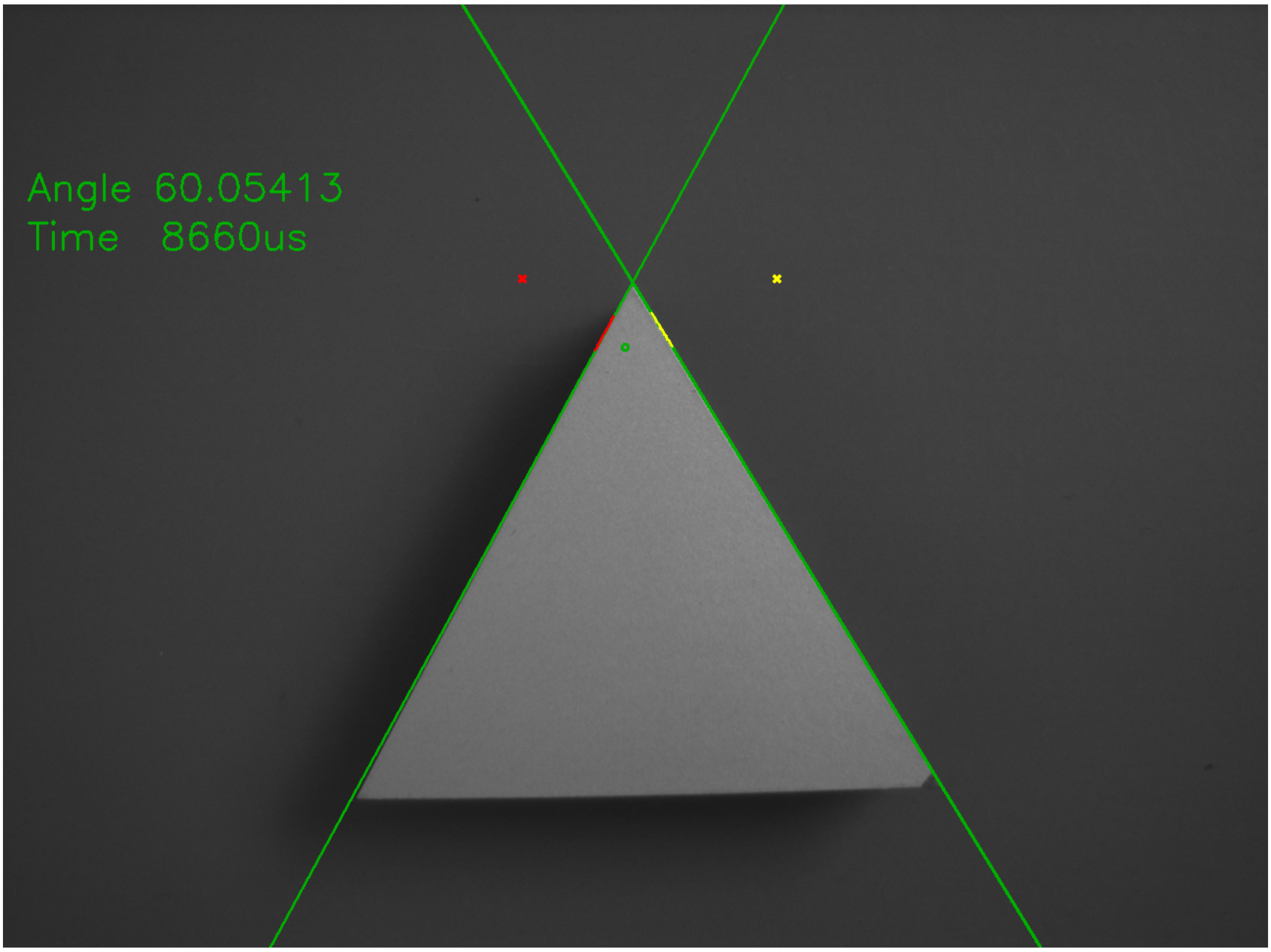
```
hrWD := F_VN_StartRelWatchdog(tStop, hr);
   hrFunc  := F_VN_MeasureAngleBetweenEdgesExp(
                ipSrcImage                := ipImageIn,
                fAngle                    := fAngle
                aInnerPoint               := aInnerPoint,
                aOuterPoint1              := aOuterPoint1
                aOuterPoint2              := aOuterPoint2,
                eEdgeDirection            := TCVN_ED_LIGHT_TO_DARK,
                fMinStrength              := fMinStrength,
                nSearchLines              := nSearchLines,
                fSearchLineDist           := fSearchLineDist,
                nMaxThickness             := 7,
                nSubpixelsIterations      := nSubpixIter,
                bAngleInDegrees           := TRUE,
                fApproxPrecision          := 0.0001,
                eAlgorithm                := eAlgorithm,
                ipEdgePoints1             := ipEdgePoints1,
                ipEdgePoints2             := ipEdgePoints2,
                hrPrev                    := hr);
hrWD := F_VN_StopWatchdog(hrWD, nFractionProcessed=>nFraction, tRest=>tRest);

// Draw result for visualization
hr := F_VN_ConvertColorSpace(ipImageIn, ipImageRes, TCVN_CST_GRAY_TO_RGB, hr);
sText := CONCAT('Angle ', REAL_TO_STRING(fAngle));
hr := F_VN_PutTextExp(sText, ipImageRes, 25, 200, TCVN_FT_HERSHEY_SIMPLEX, 1.3, aColorGreen, 2,
TCVN_LT_8_CONNECTED, FALSE,hr);
sText := CONCAT(CONCAT('Time ', DINT_TO_STRING(tStop - tRest)), 'us');
hr := F_VN_PutTextExp(sText, ipImageRes, 25, 250, TCVN_FT_HERSHEY_SIMPLEX, 1.3, aColorGreen, 2,
TCVN_LT_8_CONNECTED, FALSE,hr);
hr := F_VN_DrawPointExp(REAL_TO_UDINT(aInnerPoint[0]), REAL_TO_UDINT(aInnerPoint[1]), ipImageRes,
TCVN_DS_CIRCLE, aColorGreen, 3, 2, TCVN_LT_8_CONNECTED, hr);
hr := F_VN_DrawPointExp(REAL_TO_UDINT(aOuterPoint1[0]), REAL_TO_UDINT(aOuterPoint1[1]), ipImageRes,
TCVN_DS_X, aColorRed, 3, 2, TCVN_LT_8_CONNECTED, hr);
hr := F_VN_DrawPointExp(REAL_TO_UDINT(aOuterPoint2[0]), REAL_TO_UDINT(aOuterPoint2[1]), ipImageRes,
TCVN_DS_X, aColorYellow, 3, 2, TCVN_LT_8_CONNECTED, hr);
hr := F_VN_FitLine(ipEdgePoints1, aLine1, hr);
hr := F_VN_FitLine(ipEdgePoints2, aLine2, hr);
hr := F_VN_DrawLine_TcVnVector4_LREAL(aLine1, ipImageRes, aColorGreen, 2, hr);
hr := F_VN_DrawLine_TcVnVector4_LREAL(aLine2, ipImageRes, aColorGreen, 2, hr);
hr := F_VN_DrawPointsExp(ipEdgePoints1, ipImageRes, TCVN_DS_PLUS, aColorRed, 1, 1,
TCVN_LT_8_CONNECTED, hr);
hr := F_VN_DrawPointsExp(ipEdgePoints2, ipImageRes, TCVN_DS_PLUS, aColorYellow, 1, 1,
TCVN_LT_8_CONNECTED, hr);

// Display source and result image
hr := F_VN_TransformIntoDisplayableImage(ipImageIn, ipImageInDisp, S_OK);
hr := F_VN_TransformIntoDisplayableImage(ipImageRes, ipImageResDisp, S_OK);
```

结果

为了直观起见，aInnerPoint首先显示为绿色圆圈，aOuterPoint1显示为红色 x，aOuterPoint2显示为黄色x。所找到的相应边缘点ipEdgePoints1和ipEdgePoints2也用红色或黄色画出。边缘点的近似线以绿色绘制。计算出的角度fAngle，单位为度，而所需计算时间以 μs 为单位，显示在图像的左边。对于这个样本中使用的参数，结果如下：



为了得到更准确的结果，可以将eAlgorithm改为TCVN_EDA_APPROX_ERF，将nSubpixIter改为50。然而，这也大大增加了所需的计算时间。



类似样本

- 定位圆弧 [▸ 2695]
- 定位边缘 [▸ 2699]
- 定位椭圆 [▸ 2704]

8.1.10 自行编写的函数

这个样本解释了在为 Vision 应用编写自己的函数时需要注意的事项。它大体上涉及：

- HRESULT [▸ 122]的处理
- 接口指针 [▸ 119]的处理

其他结构

这个函数的实现基于 TwinCAT Vision API 函数的结构。其他结构也可以构想，但请注意这里解释的要点。

方法

在这个样本中，描述了一个函数。然而，它同样适用于方法。

为此，举例来说，我们考虑用一个自行编写的函数来计算和描绘物体：

```
FUNCTION F_CountAndDrawObjects : HRESULT
```

声明部分

在VAR_INPUT部分，函数的属性声明如下：原始图像被直接传输为ITcVnImage。由于在函数或方法中没有存储任何变量值，所以在这种情况下不需要注意其他情况。另一方面，结果图像作为REFERENCE TO ITcVnDisplayableImage传输，因为它在函数中创建，并将在外部返回用于进一步处理。在这种情况下，由于参考原因，没有创建接口指针的副本。

```
VAR_INPUT
    ipSrcImage          :   ITcVnImage;
    ipDestImage         :   REFERENCE TO ITcVnImage;
    aColor              :   TcVnVector4_LREAL;
    nNumberOfObjects    :   REFERENCE TO ULINT;
    hrPrev              :   HRESULT;
END_VAR
```

此外，函数内部需要的一些辅助变量也在VAR中声明。

```
VAR
    hr                  :   HRESULT;
    ipContours          :   ITcVnContainer;
    stParams            :   TcVnParamsBlobDetection;
END_VAR
```

输入检查

在函数开始时，会进行一些输入检查，以确保函数可以正常执行。

最初，FAILED(hrPrev)用于检查前一个处理链中是否发生了错误。如果发生错误，就会显示同样的错误，并直接结束该函数。

```
IF FAILED(hrPrev) THEN
    F_CountAndDrawObjects := hrPrev;
    RETURN;
END_IF
```

还会进行检查以确定所有作为 VAR_INPUT 传送到函数的接口指针是否有效。否则，该函数不能对它们做任何事情，并返回 INVALIDPARM 作为返回代码。

```
IF ipSrcImage = 0 THEN
    F_CountAndDrawObjects := Tc2_System.E_HRESULTAdsErr.INVALIDPARM;
    RETURN;
END_IF
```

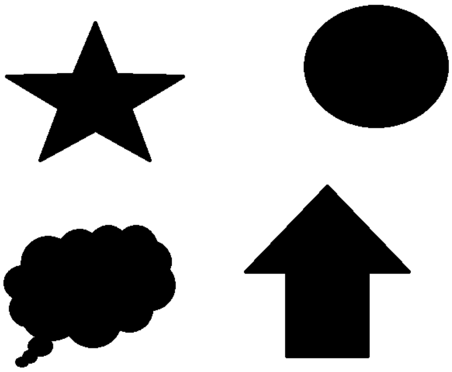
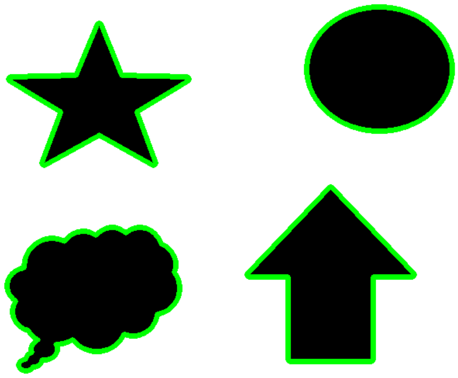
如果所有的输入检查成功，那么作为VAR_INPUT转移的所有接口指针的参考计数器必须增加，因为由于接口指针的转移，创建了一个副本，在功能块执行后被保留下来，必须在参考计数器中被映射。这里可以直接执行方法TcAddRef，而不需要事先进行<> 0检查，因为在输入检查时已经进行了检查。正如在声明部分已经描述的那样，对于函数和方法来说，递增引用计数器是不必要的，因为变量的值（就像这里的指针地址）在执行后不会被保留下来。

主要处理

实际的图像处理序列位于函数的主要部分。这里只是一个例子，不会进一步考虑。F_VN_ConvertColorSpace的调用说明了为了返回结果而作为REFERENCE TO传输的接口指针可以由 TwinCAT Vision API 函数直接写入。

```
hr := F_VN_ConvertColorSpace(ipSrcImage, ipDestImage, TCVN_CST_GRAY_TO_RGB, hr);
stParams.bFilterByArea := TRUE;
stParams.fMinArea := 10_000;
stParams.fMaxArea := 100_000;
hr := F_VN_DetectBlobs(ipSrcImage, ipContours, stParams, hr);
hr := F_VN_GetNumberOfElements(ipContours, nNumberOfObjects, hr);
hr := F_VN_DrawContours(ipContours, -1, ipDestImage, aColor, 5, hr);
```

通过这个图像处理序列，物体被画在结果图像中，且其数量以 ULINT 形式返回：

原始图像	结果图像
	

发现的物体数量：4

整理并返回

在实际的图像处理序列之后，函数内声明的接口指针再次被释放。这是由于在函数中声明的所有变量（即接口指针也是）在结束函数后被删除。如果后台仍有尚未释放的数据，这将导致内存泄漏。

```
FW_SafeRelease(ADR(ipContours));
```

最后，HRESULT作为返回值分配给函数，以便向外部报告任何错误。为了不被证伪，FW_SafeRelease函数的返回值不应该被分配到HRESULT。

```
F_CountAndDrawObjects := hr;
```

8.2 功能块样本

本节包含功能块 [▶ 1502]的样本。

8.2.1 相机寄存器访问

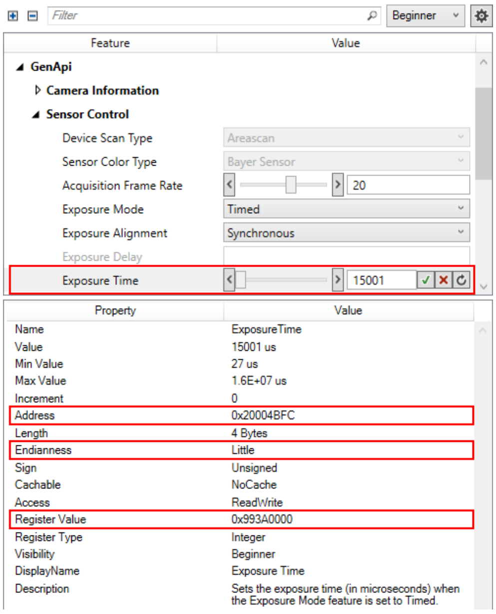
8.2.1.1 读/写寄存器

所有可以通过配置助手在配置模式下读写的相机参数也可以通过功能块FB_VN_ReadRegister_UDINT [▶ 1508]和FB_VN_ReadRegister_REAL [▶ 1506]从 PLC 读取或通过FB_VN_WriteRegister_UDINT [▶ 1515]和FB_VN_WriteRegister_REAL [▶ 1513]写入。所需功能块取决于参数的数据类型，且可以在寄存器类型的属性列表中读取。在这个样本中，曝光时间以整数读取和写入。

```
fbReadValue    :    FB_VN_ReadRegister_UDINT;
fbWriteValue   :    FB_VN_WriteRegister_UDINT;
```

两个功能块都需要参数的地址和字节序列的指示（字节序列）。此外，写入功能块还要求待写入相机的参数值。由于在数值下显示或输入的值不需要直接符合相机内部的处理方式，例如在枚举或其他字节序列的情况下，可以在**数值**寄存器下读取。

这些信息可以从配置助手中读取：



注意

相机特有的参数

此处显示的参数名称、属性和值特定于供应商或相机和固件，因此可能有所不同。

变量

```
bTriggerReadValue  :   BOOL;
nReadValue         :   UDINT;
fbReadValue        :   FB_VN_ReadRegister_UDINT;
nReturnCodeRead    :   UDINT;

bTriggerWriteValue :   BOOL;
nWriteValue        :   UDINT := 5000;
fbWriteValue       :   FB_VN_WriteRegister_UDINT;
nReturnCodeWrite   :   UDINT;
```

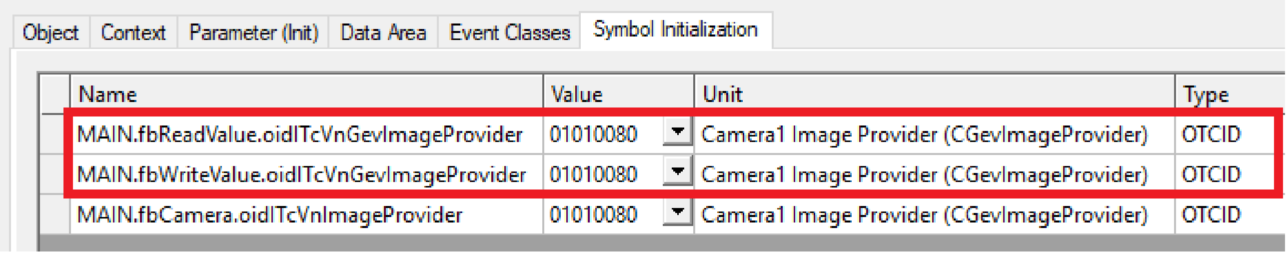
代码

```
fbReadValue(
    nAddress    :=   16#20004BFC, // Check the Address of the Camera Parameter
    nEndian     :=   1,           // 0 = Big, 1 = Little
    bRead       :=   bTriggerReadValue,
    nTimeout    :=   T#5S,
    nValue      =>   nReadValue);

fbWriteValue(
    nAddress    :=   16#20004BFC, // Check the Address of the Camera Parameter
    nValue      :=   nWriteValue, // Check the right Input Format
    nEndian     :=   1,           // 0 = Big, 1 = Little
    bWrite      :=   bTriggerWriteValue,
    nTimeout    :=   T#5S);
```

读取和写入参数的条件

一般来说，仅在打开相机连接的情况下才能读写参数（ETcVnCameraState >= TCVN_CS_OPENED）。关于具体条件，请参考相机的相关供应商文件。

两个功能块必须通过相机的**图像提供者**进行初始化：



Object | Context | Parameter (Init) | Data Area | Event Classes | Symbol Initialization

Name	Value	Unit	Type
MAIN.fbReadValue.oidITcVnGevImageProvider	01010080	Camera1 Image Provider (CGevImageProvider)	OTCID
MAIN.fbWriteValue.oidITcVnGevImageProvider	01010080	Camera1 Image Provider (CGevImageProvider)	OTCID
MAIN.fbCamera.oidITcVnImageProvider	01010080	Camera1 Image Provider (CGevImageProvider)	OTCID

附图 31: 将功能块与相机实例联系起来

8.2.2 文件访问

8.2.2.1 保存 PLC 的图像

在这个样本中，来自 PLC 的图像被保存到本地驱动器。功能块FB_VN_WriteImage [▶ 1536]用于此目的。

保存过程在几个周期内异步进行。

保存过程的时间

保存过程的持续时间取决于各种因素，因此不能笼统地说。

变量

```
hr                      :   HRESULT;

fbCamera                :   FB_VN_SimpleCameraControl;

ipImageIn               :   ITcVnImage;
ipImageInDisp           :   ITcVnDisplayableImage;

// Sample Specific Variables
fbWriteImage            :   FB_VN_WriteImage := (nTimeout := T#500MS);
sFilePath               :   STRING(255)  := '';
bWriteImageTrigger      :   BOOL;
bWriteImageWaitResult:      BOOL;
bWriteImageDone         :   BOOL;
nReturnCode             :   UDINT;
```

在这个样本中，通过将触发变量bWriteImageTrigger设置为TRUE，可以保存一个图像。

代码

在第一次调用fbWriteImage且上升沿处于bWrite时，已经完全接受了转移的ipImage，因此可以在之后立即释放，例如使用F_VN_TransformIntoDisplayableImage。此后，fbWriteImage必须继续循环调用，直到bBusy输出再次为FALSE。这可能是一个错误的情况或者第一次调用的情况，例如，如果没有指定文件扩展名或指定了不正确的扩展名。在下面的 IF 块中，将评估写入是否成功或是否发生错误。如果成功，bWriteImageDone设定且保持不变，直到下次设置bWriteImageTrigger。

```
IF SUCCEEDED(hr) AND ipImageIn <> 0 THEN

    IF NOT fbWriteImage.bBusy AND bWriteImageTrigger THEN
        bWriteImageTrigger := FALSE;
        bWriteImageDone := FALSE;
        bWriteImageWaitResult := TRUE;

        // With setting sFilePath:= '' to an empty string the images are saved under the default
path
        // The default path can be set at the service configuration tab of the Vision Node
        // If sFilePath is set, it must contain the full path, image name and type
        // for example: sFilePath := 'C:\WriteImage\ImageName.bmp'
        fbWriteImage(ipImage := ipImageIn, sFilePath := sFilePath, bWrite := TRUE);
    END_IF
```

```
    hr := F_VN_TransformIntoDisplayableImage(ipImageIn, ipImageInDisp, hr);
END_IF

fbWriteImage(sFilepath:= '', bWrite := FALSE);

IF bWriteImageWaitResult AND NOT fbWriteImage.bBusy THEN
    bWriteImageWaitResult := FALSE;
    // Check if fbWriteImage was finished successfully or implement error handling
    IF fbWriteImage.bError THEN
        nReturnCode := fbWriteImage.nErrorId AND 16#FFF;
    ELSE
        bWriteImageDone := TRUE;
        nReturnCode := 0;
    END_IF
END_IF
```

8.2.2.2 连续加载图像到 PLC

在这个样本中，将以连续方式向 PLC 加载图像。为此，使用：

- FB_VN_ReadImage [▶ 1529]

这个样本同样适用于通过FB_VN_ReadContainer [▶ 1527]加载容器。

说明

一个经常使用的情况是，参考图像被保存，以便与控制器中的实时图像进行比较。这些参考图像应在操作过程中被替换，例如，当引入新的参考时。图像可以被重新加载任何次数和任何时间间隔。

为了不需要为每一个新的参考图像通知 PLC 文件名，定义一个命名惯例。例如，这可以是Image<Index>.png，其中<Index>将被一个连续的整数所取代。然后，为每个新的参考图像提供一个文件名，索引递增 1。因此，PLC 知道下一个参考图像的名称是什么，并可以有目的地尝试加载。

注意

文件损坏的风险

切勿替换图像文件的内容；相反，仅在文件系统中创建新的图像！首先，由于视觉服务的内部缓存机制，被替换的图像未寄存。其次，几个进程同时访问一个图像会导致文件损坏。缓存设置可以交替调整。

应用

6 张彩色图片（名称为image<Index>.png）和空文件夹load附在这个样本上。将load文件夹的绝对路径定义为sBasePath。启动样本 PLC 后，按顺序将图像（从Index=0开始）移动到load文件夹，并在 ADS 图像查看中观察图像ipImageRefDisp的相应变化。

程序

在循环程序中，根据索引nIndex，最初计算待加载文件的路径。另外，调整变量声明中的基本路径sBasePath，使其与你的图像的内存位置相匹配。

```
sPath := CONCAT(CONCAT(sBasePath, TO_STRING(nIndex)), '.png');
```

通过计算出的路径，循环调用类型为FB_VN_ReadImage [▶ 1529]的功能块 fbReadImage。

```
fbReadImage(
    sFilePath    :=  sPath,
    ipDestImage  :=  ipImageIn,
    bRead        :=  TRUE,
    nTimeout     :=  T#1S
);
```

除了调用功能块外，还必须在它准备好时做出反应。一方面，通过调用bRead:=FALSE重置功能块；另一方面，如果执行成功，增加索引并将图像转发到所需的接口指针。必须查询是否发生了错误，因为只有在当前图像已经被成功加载的情况下，才可以增加索引。

```
IF NOT fbReadImage.bBusy THEN
    IF NOT fbReadImage.bError THEN
        nReturnCode := 0;
        nIndex := nIndex + 1;

        IF ipImageIn <> 0 THEN
```

```
            FW_SafeRelease(ADR(ipImageRef));
            ipImageRef := ipImageIn;
            ipImageIn.TcAddRef();
            FW_SafeRelease(ADR(ipImageIn));
        END_IF
    ELSE
        nReturnCode := fbReadImage.nErrorId AND 16#FFF;
    END_IF
    fbReadImage(sFilePath:='', bRead:=FALSE);
END_IF
```

为了节省内存和计算时间，此处的图像转发通过接口有序转移而不是通过拷贝进行。

8.2.3 图像采集

8.2.3.1 通过文件名触发图像

在这个样本中，根据功能块FB_VN_FileSourceControl [▶ 1541]中的文件名，特意触发了图像。

应用

该样本交替显示两个文件名为*Image1.bmp*和*Image2.bmp*的图像（来自ADS Image Watch [▶ 115]的文件源）。样本包括带有相应文件名的图像。你可以用自己的图片替换样本图像，或者另外添加图像。为此，在sFileName中更改文件名或添加到文件名中，如有必要，调整元素的数量。注意，所有待显示的图像必须添加在文件源控制 [▶ 111]中。通过设置bLoop为false，可以停止图像的自动切换，然后可以通过设置nImageIndex选择待触发的图像。

程序

```
hr              :   HRESULT;
sFileName       :   ARRAY [0..1] OF STRING := ['Image1.bmp', 'Image2.bmp'];
fbFileSource    :   FB_VN_FileSourceControl;
eState          :   ETcVnCameraState;
ipImageIn       :   ITcVnImage;
ipImageInDisp   :   ITcVnDisplayableImage;
bTrigger        :   BOOL := TRUE;
bLoop           :   BOOL := TRUE;
nImageIndex     :   UINT := 0;
```

fbFileSource	用于控制类型FB_VN_FileSourceControl [▶ 1541]的文件源 [▶ 110]的功能块。
eState	文件源的状态
ipImageIn	输入图像
ipImageInDisp	可显示的输入图像
bTrigger	触发器标记
sFileName	文件名 – 更改这个变量，以便加载不同的图像。然而，这个图像必须在添加在文件源控制 [▶ 111]中。
hr	类型为HRESULT [▶ 122]的状态变量。

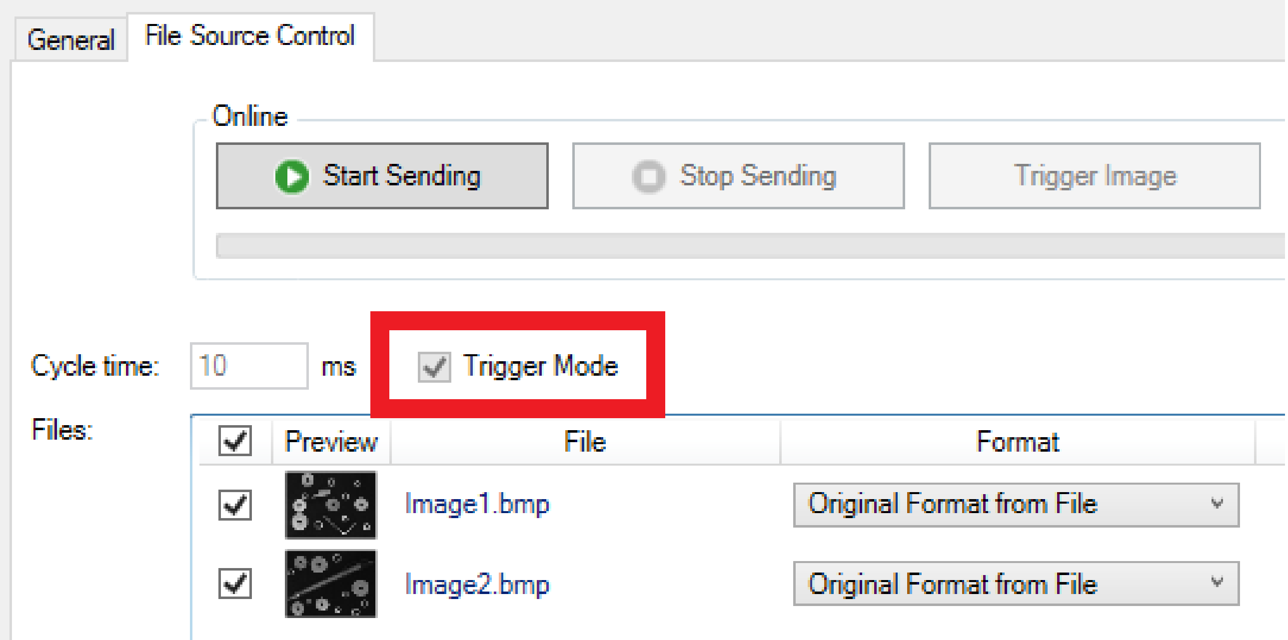
一般来说，通常的状态机被用来操作文件源：

```
eState := fbFileSource.GetState();
CASE eState OF

<...>

END_CASE
```

出错时的复位和图像采集的开始都是在相应的状态下正常进行：

```
TCVN_CS_INITIAL, TCVN_CS_INITIALIZING, TCVN_CS_INITIALIZED, TCVN_CS_OPENING, TCVN_CS_OPENED,
TCVN_CS_STARTACQUISITION:
    fbFileSource.StartAcquisition();

TCVN_CS_ERROR:
    fbFileSource.Reset();
```

然而，为了使文件源在开始采集图像时不立即加载图像，必须在文件源控制中激活**触发模式**：

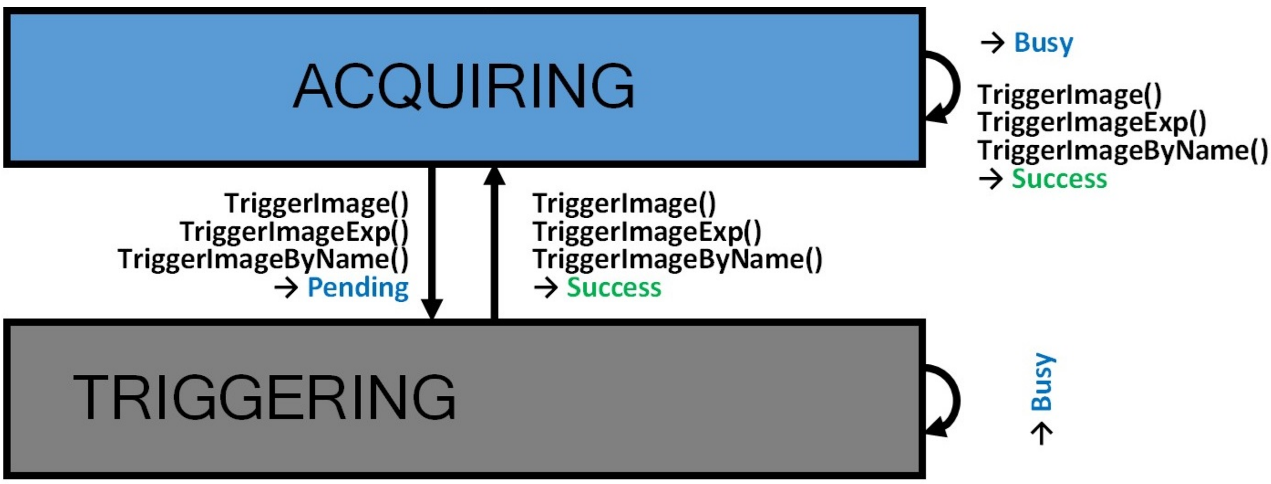


在ACQUIRING状态下，现在将交替地触发和接收具有相应文件名的图像。这里明确使用了TriggerImageByName [▶ 1550]的方法来触发图像。这个方法只存在于功能块FB_VN_FileSourceControl [▶ 1541]中，而不存在于FB_VN_SimpleCameraControl [▶ 1575]中。

```
TCVN_CS_ACQUIRING:
    IF bTrigger THEN
        hr := fbFileSource.TriggerImageByName(sFileName[nImageIndex]);
        IF SUCCEEDED(hr) THEN
            bTrigger := FALSE;

            IF bLoop THEN
                nImageIndex := nImageIndex + 1;
                IF nImageIndex >= 2 THEN
                    nImageIndex := 0;
                END_IF
            END_IF
        END_IF
    ELSE
        hr := fbFileSource.GetCurrentImage(ipImageIn);
        IF SUCCEEDED(hr) AND ipImageIn <> 0 THEN
            bTrigger := TRUE;
            hr := F_VN_TransformIntoDisplayableImage(ipImageIn, ipImageInDisp, hr);
        END_IF
    END_IF
```

由于文件源在触发图像时暂时进入TRIGGERING状态，因此必须在此状态下继续触发。这一直持续到文件源完全加载了图像，从而将其状态改回ACQUIRING。图像的文件名不需要为进一步触发而再次传输；只需调用方法TriggerImage [▶ 1549]即可。

```
TCVN_CS_TRIGGERING:
    fbFileSource.TriggerImage();
```



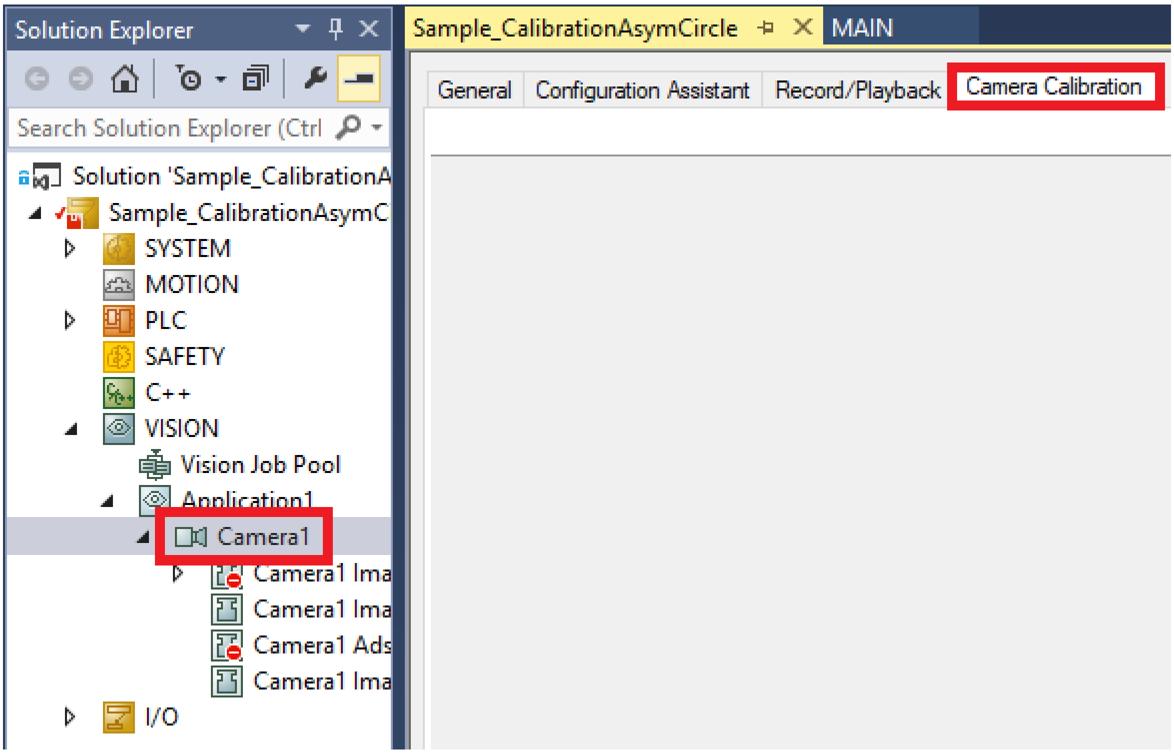
在这个样本中，变量bTrigger被自动设置为TRUE，以在图像成功加载时进行演示。在实际应用中，触发器将被设置在一个在序列上有意义的点上。

8.3 助手样本

8.3.1 校准助手

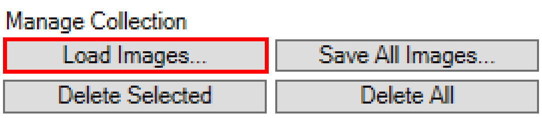
需要几何相机标定以将图像点转换为世界点，在此过程中确定所需的参数。该例程展示使用不对称圆圈模板和标定助手对相机进行几何标定。其他标定模板也可采取类似步骤（请参见 相机校准 [▶ 94]）。只有第 2 步需要做出相应调整。

要运行此例程，首先打开已创建相机的项目，然后打开相机实例的标定助手。



步骤 1 — 图像采集

首先，必须将待标定相机拍摄的标定模板图像加载到助手。这可以通过相机直接采集图像或加载现有图像来实现。对于本例程，您可以在相应的图像文件夹中找到一个不对称圆圈模板的四个合适图像 *ImgCalibAsymCirclesx.bmp*。通过 Load Images.... （**加载图像……**）将这些图像加载到标定助手。

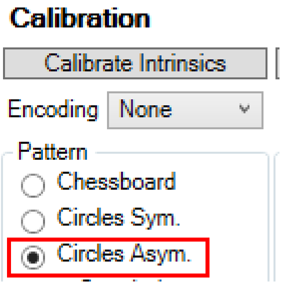


步骤 2 — 确定不对称圆圈模板

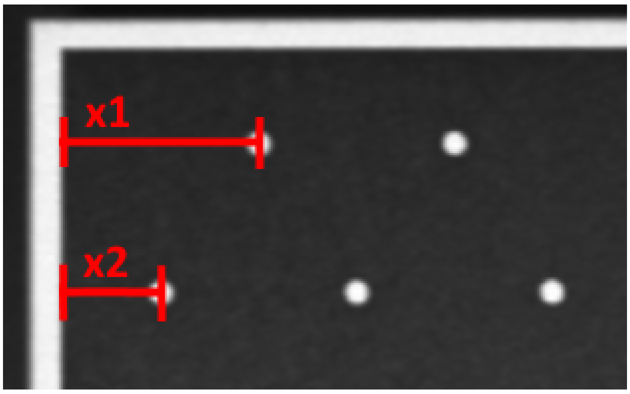
在本例程中，使用了非对称圆圈模板作为标定模板。要使用其他模板类型，请遵守相应定义：

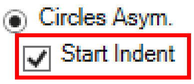
- 棋盘图案 [▶ 102]
- 对称圆形图案 [▶ 103]
- 不对称的圆圈图案 [▶ 104]
- 单个圆圈图案 [▶ 105]

必须在标定助手中为加载的样本图像选择不对称圆圈模板：

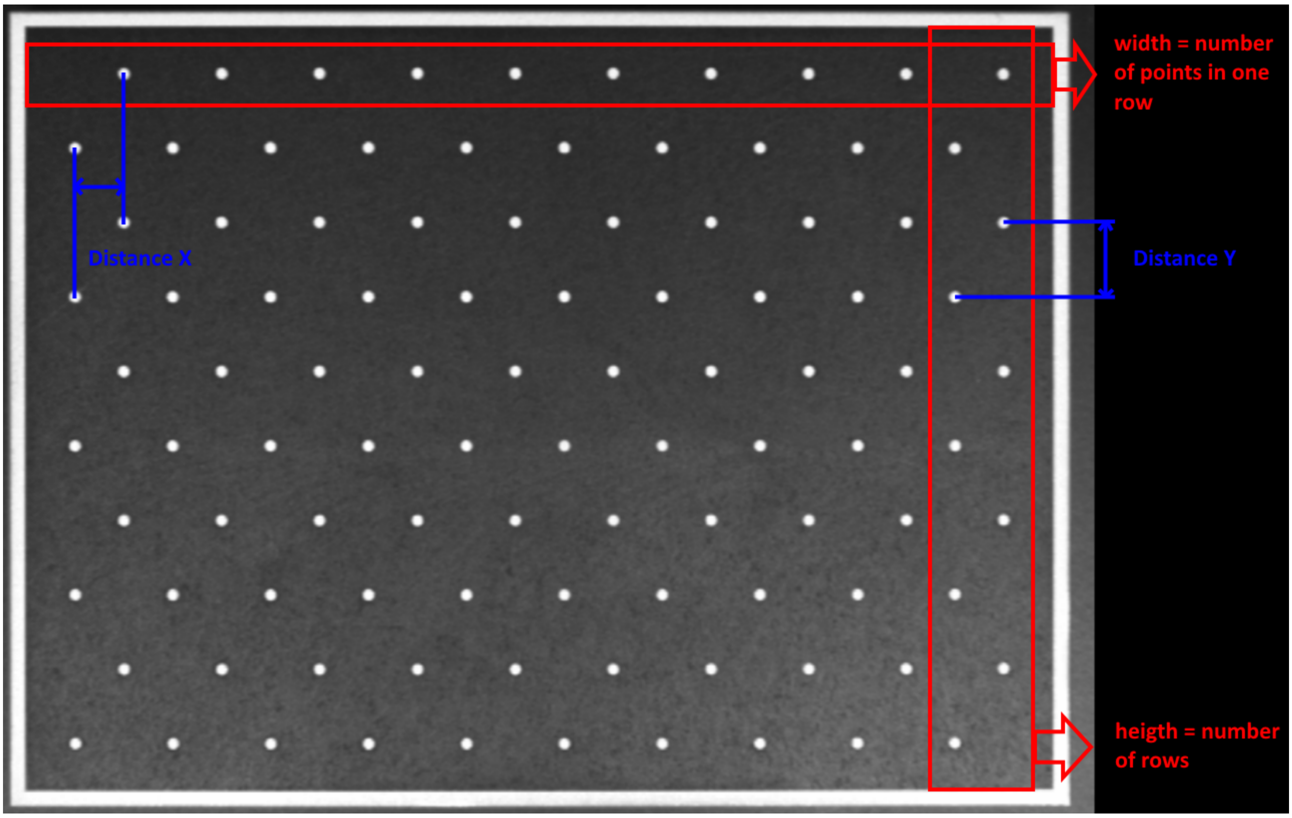


由于不对称，圆圈模板的第一行（x1）或第二行（x2）均缩进。在这种情况下，第一行（x1）缩进，因此应选择 Start Indent （开始缩进）。

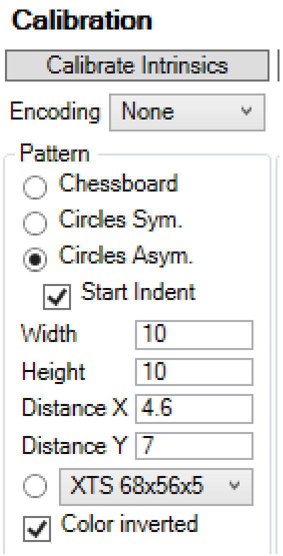




宽度模板的宽度与行中点/圆圈的数量相对应，而**高度**则与行数相对应。**距离** X用于指定两个圆圈在 X 方向上的最短距离。在不对称模板中，这是下一行/上一行到下一个圆的距离。注意图像中的标记。两行之间的距离在**距离** Y 中指定。

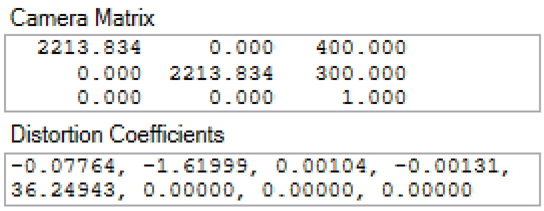


助手希望模板为白底黑圈。由于本示例使用黑底白圈模板，因此必须通过选择**颜色反转**来反转颜色。



步骤 3 — 计算内部参数

标定模板完全确定后，可以使用 Calibrate Intrinsics（标定内部）按钮计算内部参数。**再投影 误差**应介于 0 和 1 之间，并应确定**相机矩阵**和**畸变系数**。

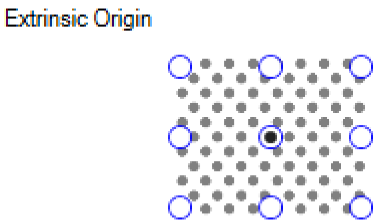


如果不是这种情况，**再投影 误差**工具提示可提供有关可能错误的信息。此外，*C:\ProgramData\Beckhoff\Vision_CalibrationAssistantOutput* 下还会创建一个图像系列，提供有关识别模板的进一步信息。该目录需要写入权限。

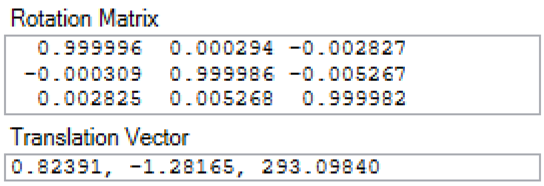
步骤 4 — 计算外部参数

在计算出内部参数后，可以计算外部参数。为此，只需使用一个图像，即模板定位在稍后测量组件等平面上的图像。这是第一个样本图像。因此，点击第一个加载图像，使其出现在图像预览中。

还可以指定**外源**模板的零点。默认情况下，它位于模板的中心。

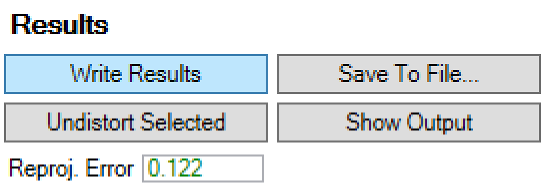


然后，Calibrate Extrinsics（标定外部）按钮会计算**旋转矩阵**和**平移向量**。结果可在 Results（结果）下查看。

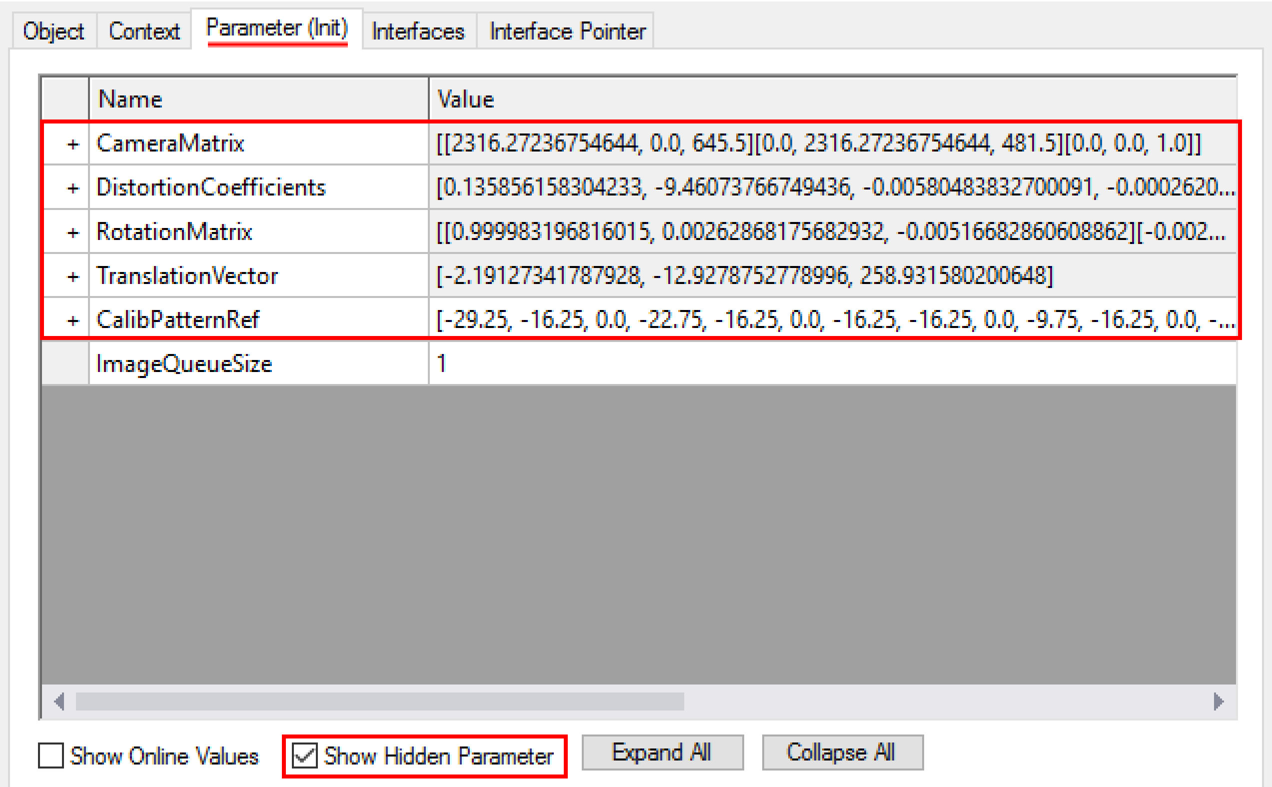


步骤 5 — 将结果写入相机的图像提供程序

如果标定成功，可通过 Write Results（写入结果）将结果写入相机的图像提供程序。这一步骤是必要的，这样 PLC 中的相机功能块 FB_VN_GevCameraControl [▶ 1550] 才能访问标定结果。



在图像提供程序中，如果选择了 Show Hidden Parameters（显示隐藏参数），标定结果可以在 Parameters（参数）（Init）下找到。



完成此步骤后，请务必点击  激活配置。否则 PLC 将无法提供标定结果。

步骤 6 — 将标定结果载入 PLC

功能块 FB_VN_GevCameraControl [▸ 1550] 可用于将标定参数加载到 PLC：

变量

```
hr                          :   HRESULT;
fbCamera                    :   FB_VN_GevCameraControl;
aCameraMatrix               :   TcVnMatrix3x3_LREAL;
aDistortionCoefficients     :   TcVnArray8_LREAL;
aRotationMatrix             :   TcVnMatrix3x3_LREAL;
aTranslationVector          :   TcVnVector3_LREAL;
```

代码

```
hr := fbCamera.GetCameraMatrix(aCameraMatrix);
hr := fbCamera.GetDistortionCoefficients(aDistortionCoefficients);
hr := fbCamera.GetRotationMatrix(aRotationMatrix);
hr := fbCamera.GetTranslationVector(aTranslationVector);
```

步骤 7 — 将像素转换为世界坐标

然后使用函数 F_VN_TransformCoordinatesImageToWorld_Points [▸ 1067] 确定现实世界中图像中的点相对于标定模板零点的坐标，例如：

变量

```
aPointImage                 :   TcVnPoint2_LREAL;
aPointWorld                 :   TcVnPoint3_LREAL
```

代码

```
hr := F_VN_TransformCoordinatesImageToWorld_Point(
        aSrcPoint                    :=  aPointImage,
        aDestPoint                   :=  aPointWorld,
        aCameraMatrix                :=  aCameraMatrix,
        aDistortionCoefficients      :=  aDistortionCoefficients,
        aRotationMatrix              :=  aRotationMatrix,
        aTranslationVector           :=  aTranslationVector,
        fZ                           :=  0,
        hrPrev                       :=  hr);
```

步骤 8 — 将世界点转换为图像坐标。

这种转换可以通过函数 F_VN_TransformCoordinatesWorldToImage_Points [▶ 1072] 进行反转，从而将世界坐标系中的点进行转换，使其能够显示在图像中的正确位置。这对于显示辅助线以便在图像中更好地定向非常有用。下方代码显示了图像中的世界坐标系。

变量

```
i                          :   INT;
aLine                      :   TcVnVector4_DINT;
aCoordinatesWorld          :   ARRAY[0..4] OF TcVnPoint3_LREAL := [ [0,0,0], [50,0,0], [-50, 0, 0],
[0, 50, 0], [0, -50, 0] ];
aCoordinatesImage          :   ARRAY[0..4] OF TcVnPoint2_REAL;
```

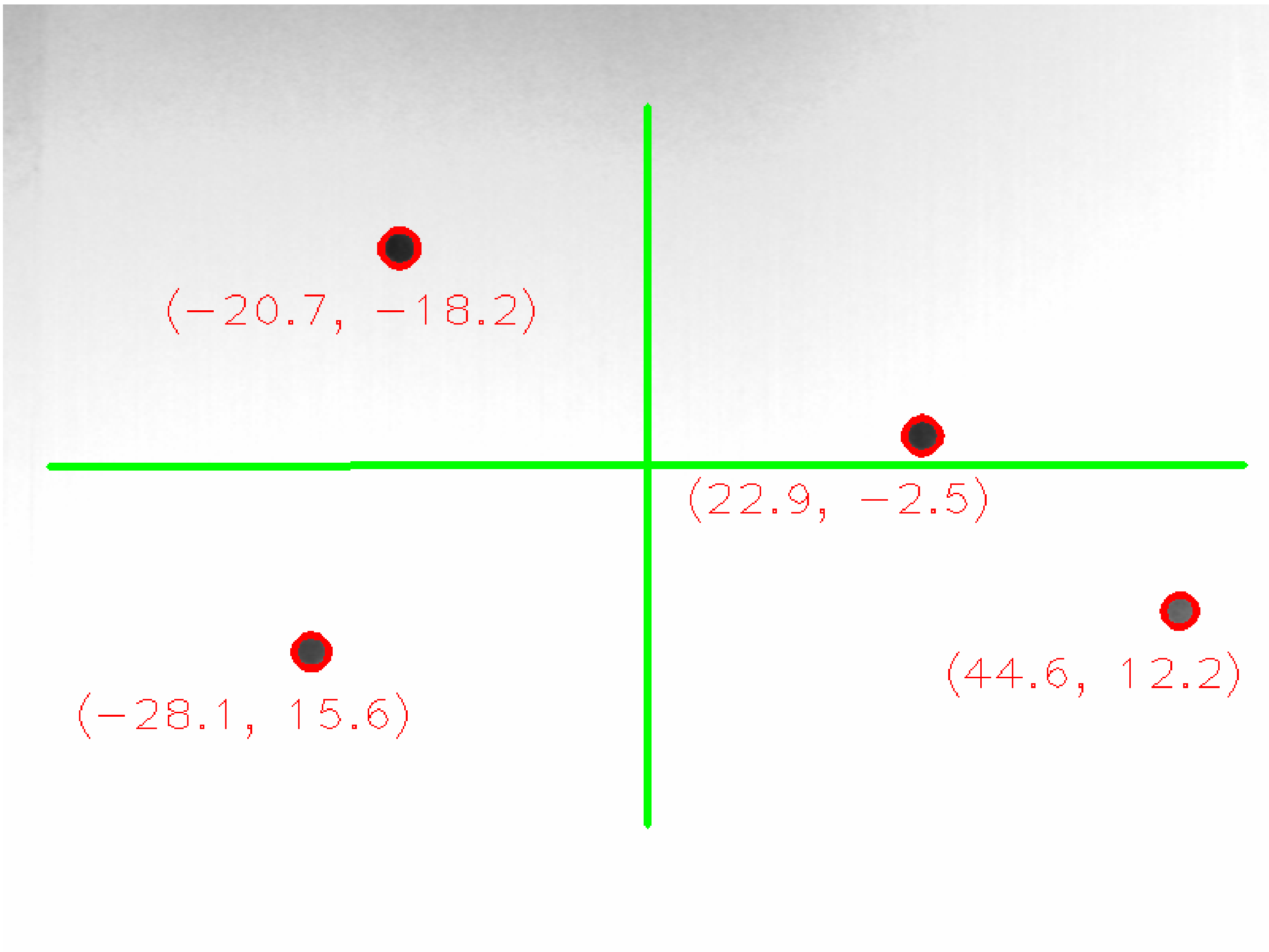
代码

```
FOR i:= 0 TO 4 DO
    hr := F_VN_TransformCoordinatesWorldToImage_Point(
            aSrcPoint                    :=  aCoordinatesWorld[i],
            aDestPoint                   :=  aCoordinatesImage[i],
            aCameraMatrix                :=  aCameraMatrix,
            aDistortionCoefficients      :=  aDistortionCoefficients,
            aRotationMatrix              :=  aRotationMatrix,
            aTranslationVector           :=  aTranslationVector,
            hrPrev                       :=  hr);

    IF i > 0 THEN
        aLine[0] := REAL_TO_DINT(aCoordinatesImage[0][0]);
        aLine[1] := REAL_TO_DINT(aCoordinatesImage[0][1]);
        aLine[2] := REAL_TO_DINT(aCoordinatesImage[i][0]);
        aLine[3] := REAL_TO_DINT(aCoordinatesImage[i][1]);
    hr := F_VN_DrawLine_TcVnVector4_DINT(aLine , ipImageRes, aColorGreen, 3, hr);
    END_IF
END_FOR
```

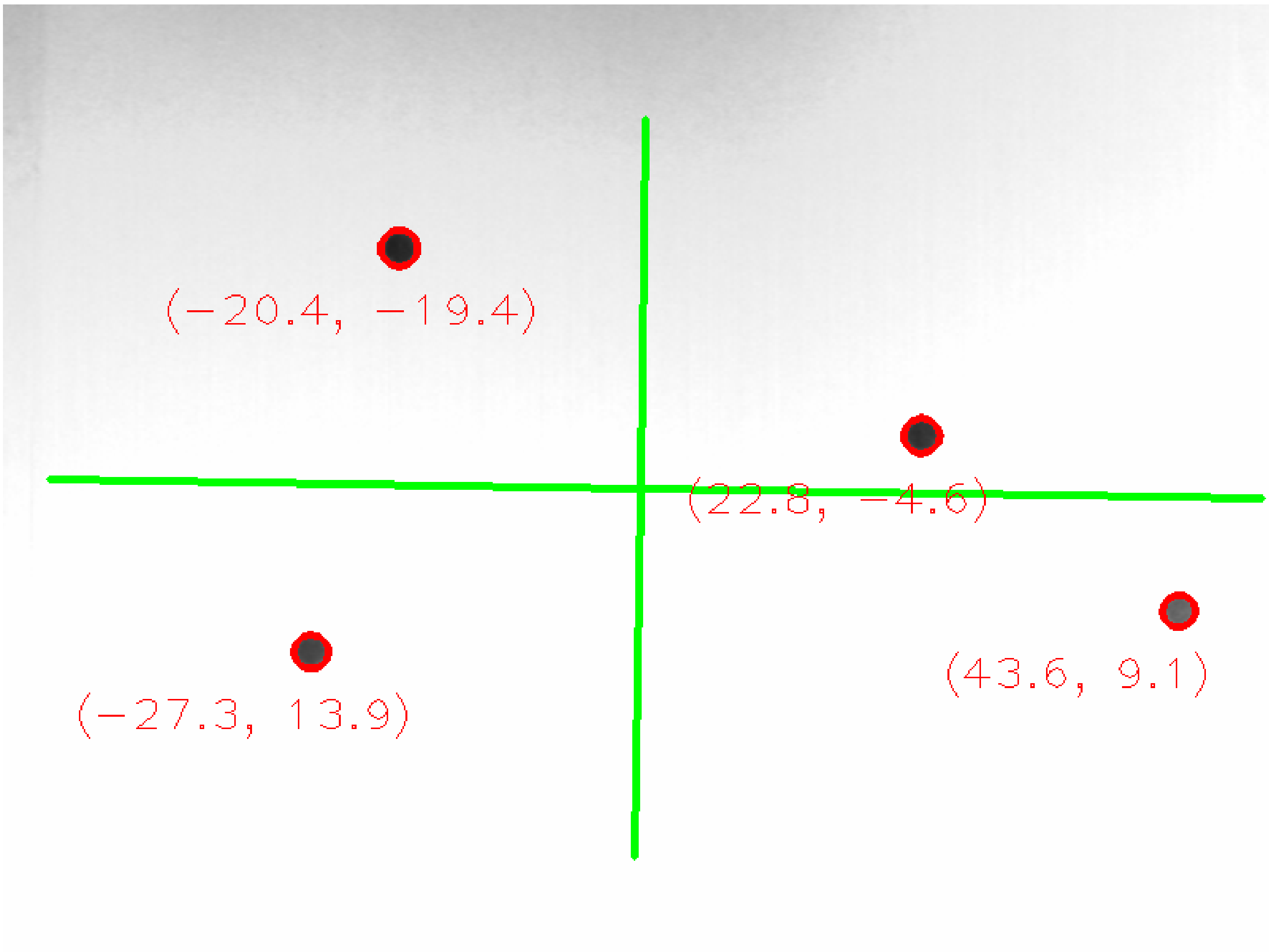
步骤 9 — 查看结果

执行程序时，必须先开始回放相机数据流 [▶ 92]。为此，请打开相机对象的 **Record/Playback（记录/回放）**选项卡，选择相应图像文件夹中的 CameraSimulation.tcs 文件并开始回放。

生成的图像可在 ADS Image Watch [▶ 115] 中进行评估。可以看到图像上绘制的坐标轴和确定的各个点的世界坐标。



如果在步骤 4 中使用不同的图像或坐标原点进行外部标定，结果将有所不同。坐标轴会发生改变，像素的世界坐标也会随之调整。



8.4 相机配置样本

由于每个 GigE Vision 相机的参数可能不同，因此不可能提供完整的描述。下面部分概述了经常使用的相机参数以及如何配置这些参数。关于 GenAPI 参数的配置说明，可以查看配置 [▶ 70]一章。

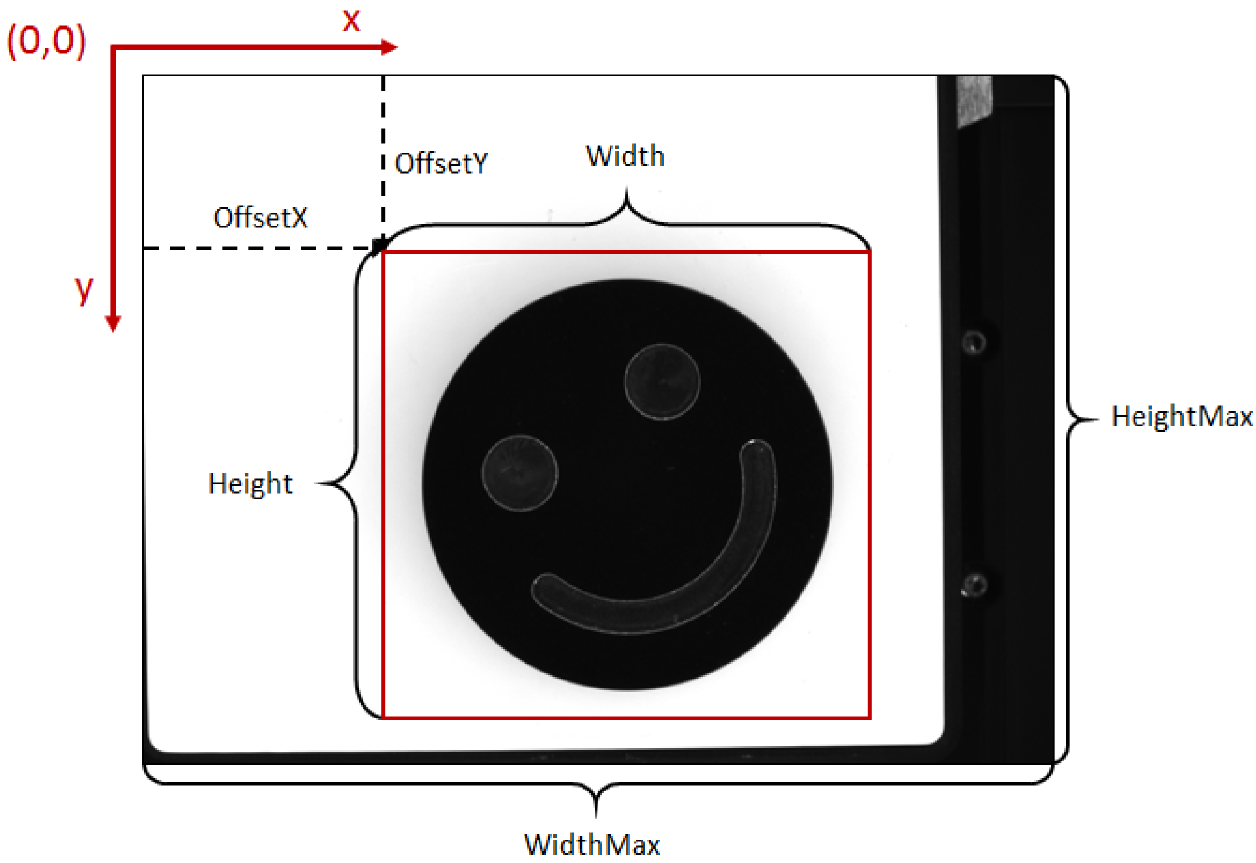
8.4.1 感兴趣区域（ROI）

如果相机传感器记录的视场比需要的大，建议在实际需要的视场上放置感兴趣区域（ROI）：

- 传感器可以更快地读出，具体取决于相机传感器和所选范围。
- 需要传输的数据量减少，也减少了数据传输时间。
- 同时，在整个图像上进行的所有图像处理算法的处理时间也会减少。
- 此外，可以减少图像噪音，使图像处理更加稳健和容易。

检查相机的功能

请参考相机手册，了解相机是否支持相应的功能。这同样适用于实际使用的功能名称。根据 GenICam 标准功能命名公约（2.4 版），使用功能名称的程序如下。



设置 ROI 的功能可以在ImageFormatControl或相机的相应类别下找到：

- WidthMax和HeightMax指定图像的最大宽度和高度，单位：像素。这些值可能因其他功能而不同（如像素合并 [▸ 2727]）。
- Width和Height指定了 ROI 的宽度和高度（即待查看的视场）且必须进行相应设置。
- OffsetX和OffsetY用于定位 ROI 或视场。起点是图像的左上角。

在设置功能时，请注意：

- OffsetX + Width <= WidthMax
- OffsetY + Height <= HeightMax

8.4.2 像素合并

像素合并是将几个像素值相加或平均变成一个。

它可以用于：

- 增加光的敏感性，
- 改善对比度，
- 在相同的视场下降低图像分辨率，使帧率和处理时间更快。

如果在图像分辨率降低时无法再达到所要求的精度，就不应该使用。

在对像素进行相加或平均时，将区分水平和垂直像素合并。

检查相机的功能

请参考相机手册，了解相机是否支持相应的功能。这同样适用于实际使用的功能名称。根据 GenICam 标准功能命名公约（2.4 版），使用功能名称的程序如下。

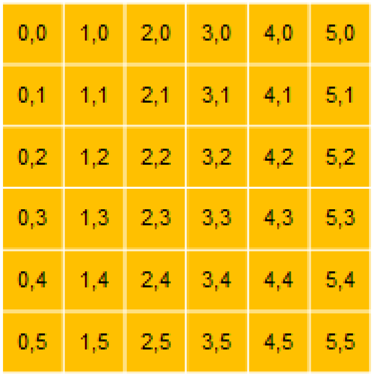
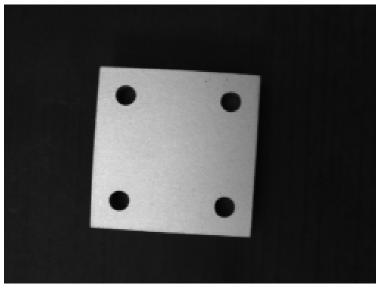
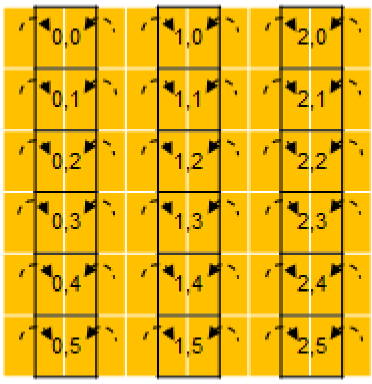
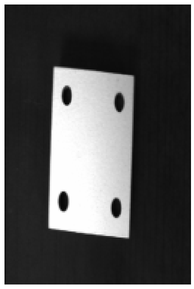
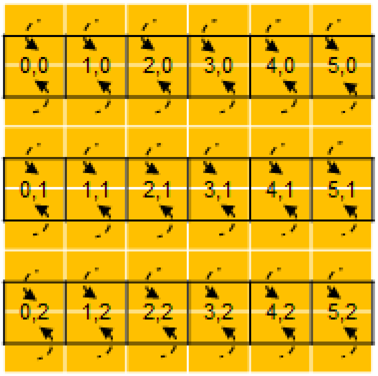
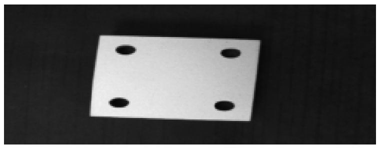
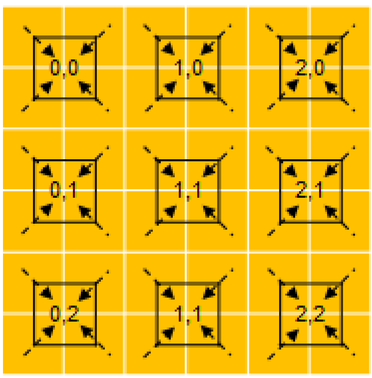
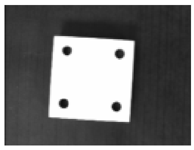
关于设置像素合并的功能，可以在ImageFormatControl下或相机的相应类别下找到：

- BinningSelector表示像素合并是适用于整个sensor，还是适用于ROI [▸ 2726]。
- BinningHorizontalMode和BinningVerticalMode表示是否应将像素相加或平均。

BinningHorizontal和BinningVertical表示在水平和垂直方向上应合并多少个像素。值为 1 表示在相应的方向上没有像素合并。

通过单色相机进行像素合并

举例展示单色相机的四种可能配置（显示了像素的坐标）：

BinningHorizontal = 1 BinningVertical = 1 ⇒ 无“像素合并”		
BinningHorizontal = 2 BinningVertical = 1 BinningHorizontalMode = sum		
BinningHorizontal = 1 BinningVertical = 2 BinningVerticalMode = sum		
BinningHorizontal = 2 BinningVertical = 2 BinningHorizontalMode = sum BinningVerticalMode = sum		

通过彩色相机进行像素合并

相比单色相机，彩色相机的像素合并支持更少。它可以通过不同的方式实施。详情请参考相机用户手册。

8.4.3 图像采集和触发

采集和触发功能相互关联，因此将在下文中一起说明。术语Acquisition基本上描述了可以拍摄图像的相机状态，见ImageAcquisition [▶ 1540]。是否以及何时实际拍摄和发送图像由Trigger参数的配置决定。

关于各种配置选项以及采集和触发设置之间的关系，将在下面的章节中进行更详细的描述（章节按照AcquisitionMode布置）。

目前为止，最常用的图像采集选项是：

- 连续图像采集 [▶ 2730] - 相机流，它在没有任何其他触发的情况下以固定的时间间隔捕捉图像。
- 单一图像触发，带连续图像采集 [▶ 2731] - 相机在触发信号（硬件或软件）后正好拍摄一张图像。相机一直处于ACQUIRING状态，因此在每次录像前不必重复写入AcquisitionStart命令。

检查相机的功能

请参考相机手册，了解相机是否支持相应的功能。这同样适用于实际使用的功能名称。根据 GenICam 标准功能命名公约（2.4 版），使用功能名称的程序如下。

采集模式

AcquisitionMode功能主要指定在采集过程中拍摄多少张图像，并指相机状态机的TCVN_CS_ACQUIRING状态。如需结束记录，必须通过AcquisitionStop命令退出状态，然后才能通过AcquisitionStart命令再次开始下一次记录。因此，Continuous模式是最经常选择的设置，因为这样可以永久地保持在采集状态。

- Continuous
  - 图像采集通过AcquisitionStart命令开始，且仅由AcquisitionStop命令终止。在这之间，根据触发器的设置，连续捕捉图像。
  - 连续的图像采集 [▶ 2730]
- MultiFrame
  - 采集图像，直到达到AcquisitionFrameCount中指定的图像数量。对于每个图像采集序列，都需要一个AcquisitionStart命令。
  - 多图像采集 [▶ 2732]
- SingleFrame
  - 拍摄到单个图像。对于每次图像采集，都需要一个AcquisitionStart命令。
  - 单一图像采集 [▶ 2733]

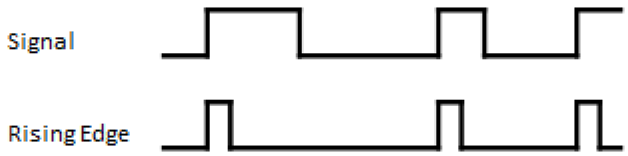
触发源

TriggerSource功能指定哪个内部或物理信号解释为触发器。可能来源清单很长，并且根据相机的不同而不同。下面提供一些样本：

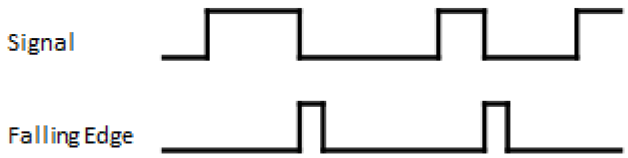
- Software
一个软件命令设置了触发器。PLC 功能块FB_VN_GevCameraControl [▶ 1550]和FB_VN_SimpleCameraControl [▶ 1575]提供了发送命令的方法TriggerImage [▶ 1574]。
- Line0, Line1, ...
物理相机的输入是触发源。注意与之相连的信号源的必要边缘陡度。这可能因相机的不同而不同。
- Encoder0, Encoder1, ...
线扫描相机通常与一个编码器连接，以触发图像采集。如果没有物理编码器，但 TwinCAT 中有相应的轴值，则可以通过 EL2521 或 EL2522 输出编码器信号。

触发器的激活

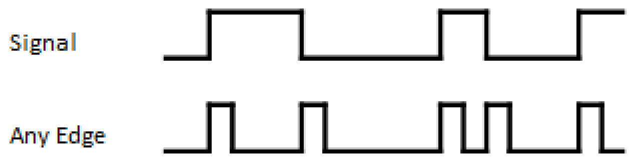
- RisingEdge
  触发器通过TriggerSource的上升沿激活。



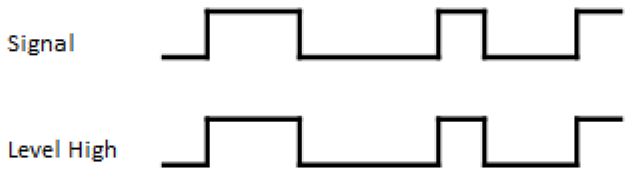
- FallingEdge
  触发器通过TriggerSource的下降沿激活。



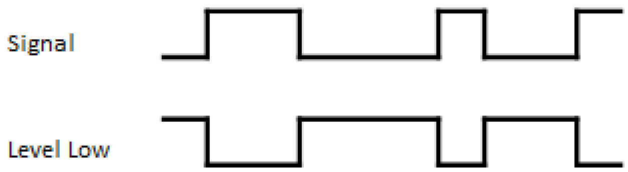
- AnyEdge
  触发器通过TriggerSource的上升或下降沿激活。



- LevelHigh
  只要有信号出现在TriggerSource，触发器就会被激活。



- LevelLow
  只要在TriggerSource没有信号出现，触发器就会被激活。



8.4.3.1 连续的图像采集

默认情况下，图像采集和触发的设置处于互锁状态。下面的解释部分假设AcquisitionMode设置为Continuous。

其他情况描述如下：

- 多图像采集 [▶ 2732]
- 单一图像采集 [▶ 2733]

检查相机的功能

请参考相机手册，了解相机是否支持相应的功能。这同样适用于实际使用的功能名称。根据 GenICam 标准功能命名公约（2.4 版），使用功能名称的程序如下。

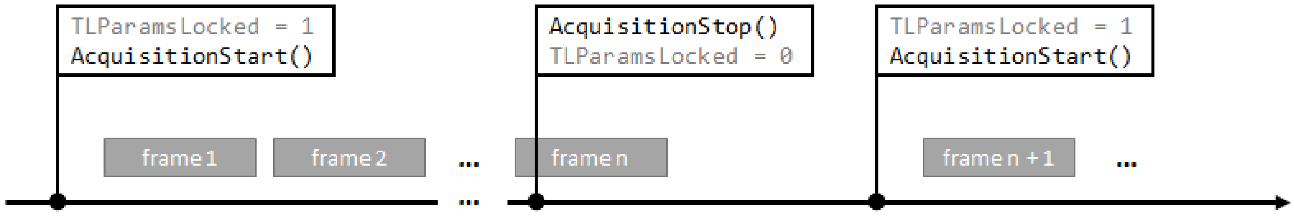
连续的图像采集

如果相机需要进行流式处理，即以均匀的时间间隔拍摄图像而不需要进一步触发，则需要进行以下配置：

- AcquisitionMode = Continuous
- TriggerSelector = AcquisitionStart

- TriggerMode = Off

然后，AcquisitionStart命令启动数据流，且AcquisitionStop命令会结束数据流。当前图像的记录已经完成，且图像已传输。

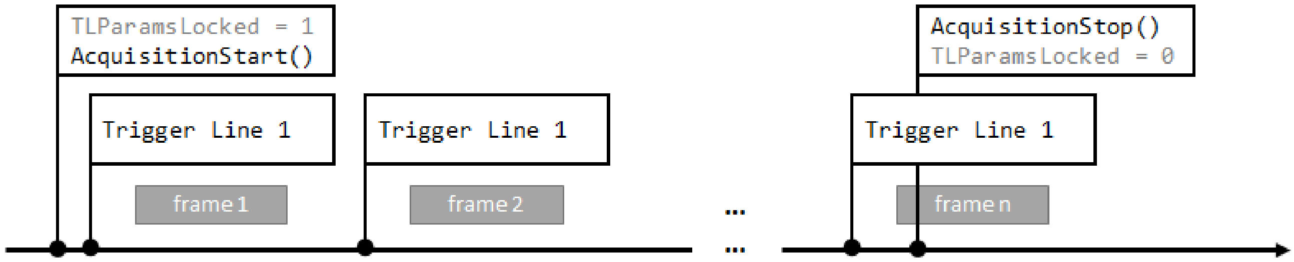


用于连续图像采集的单一图像触发器

如果相机要在触发信号后仅拍摄一幅图像，则需要进行以下配置：

- AcquisitionMode = Continuous
- TriggerSelector = FrameStart
- TriggerMode = On
- TriggerSource = Line1 (exemplarisch)

与其他情况一样，在AcquisitionStart命令之后，相机即可拍摄图像。每章图像都是在选定的信号源发出触发信号后记录。触发信号的详细信息在TriggerActivation中指定。

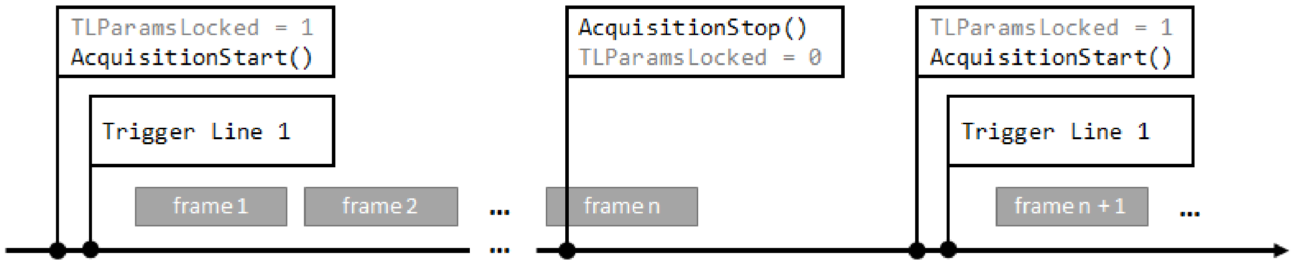


触发启动信号后连续图像采集

如果要求相机在触发信号后以均匀的时间间隔连续拍出图像，则需要进行以下配置：

- AcquisitionMode = Continuous
- TriggerSelector = AcquisitionStart
- TriggerMode = On
- TriggerSource = Line1 (exemplarisch)

在AcquisitionStart命令之后，相机即可捕捉和传输图像。然而，只有在触发信号之后才开始记录，在样本中是在线路 1 的硬件触发信号之后。数据流通过AcquisitionStop命令结束；但是，当前图像的记录仍将完成并传输。



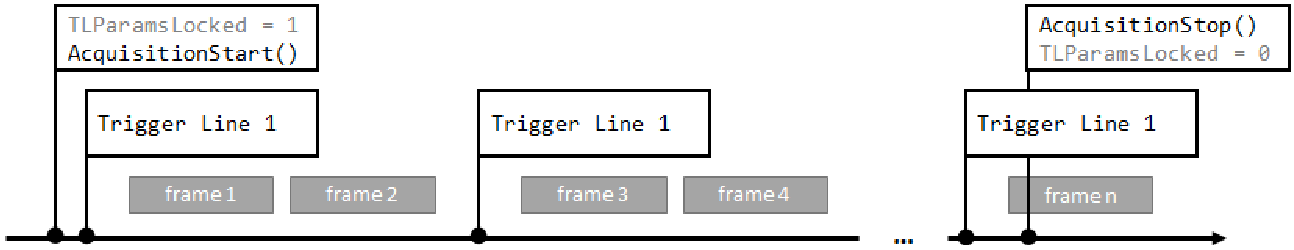
用于连续图像采集的多图像触发器

如果相机要在触发信号后记录一定数量的图像，需要进行以下配置。例如，该数字设置为 2：

- AcquisitionMode = Continuous
- AcquisitionBurstFrameCount = 2 (exemplarisch)
- TriggerSelector = FrameBurstStart

- TriggerMode = On
- TriggerSource = Line1 (exemplarisch)

与其他情况一样，在AcquisitionStart命令之后，相机即可拍摄图像。在选定源的触发信号后，记录定义的图像数量。触发信号的详细信息在TriggerActivation中指定。如果在图像采集/传输过程中发出AcquisitionStop命令，已经开始的图像会完成，但任何缺失的图像都不再创建。



8.4.3.2 多图像采集

默认情况下，图像采集和触发的设置处于互锁状态。下面的解释部分假设AcquisitionMode设置为MultiFrame。

其他情况描述如下：

- 连续的图像采集 [▶ 2730]
- 单一图像采集 [▶ 2733]

检查相机的功能

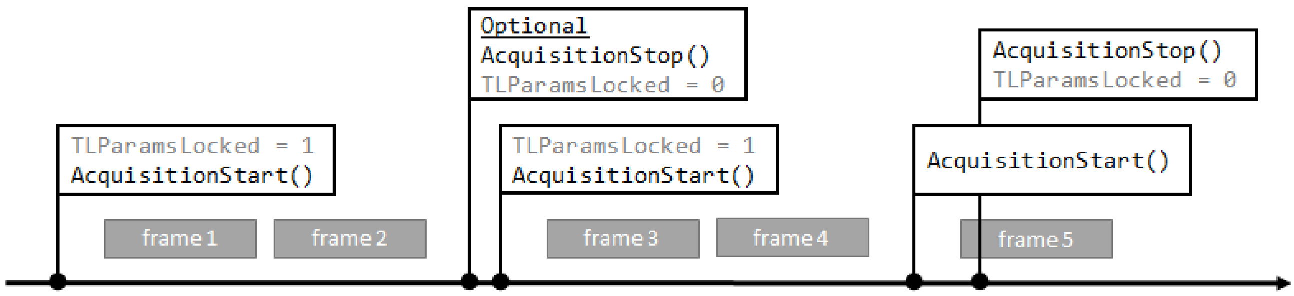
请参考相机手册，了解相机是否支持相应的功能。这同样适用于实际使用的功能名称。根据 GenICam 标准功能命名公约（2.4 版），使用功能名称的程序如下。

多图像采集

如果相机要在不使用触发器的情况下拍出一定数量的图像，需要进行以下配置：

- AcquisitionMode = MultiFrame
- AcquisitionFrameCount = 2 (example)
- TriggerSelector = AcquisitionStart
- TriggerMode = Off

AcquisitionStart命令发出后，数据流开始。在达到指定的帧数（AcquisitionFrameCount）时，将会停止。根据选择，然后发出AcquisitionStop命令，并取消锁定。这使得在下一个AcquisitionStart命令之前可以改变相机的特征。如果在图像采集/传输过程中发出AcquisitionStop命令，已经开始的图像会完成，但任何缺失的图像都不再创建。



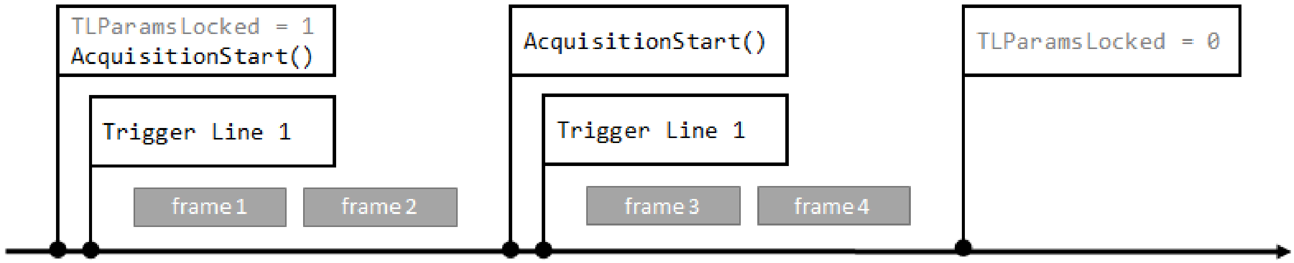
触发启动信号后的多图像采集

如果相机要在触发信号后拍出一定数量的图像，需要进行以下配置：

- AcquisitionMode = MultiFrame
- AcquisitionFrameCount = 2 (example)
- TriggerSelector = AcquisitionStart
- TriggerMode = On

- TriggerSource = Line1 (example)

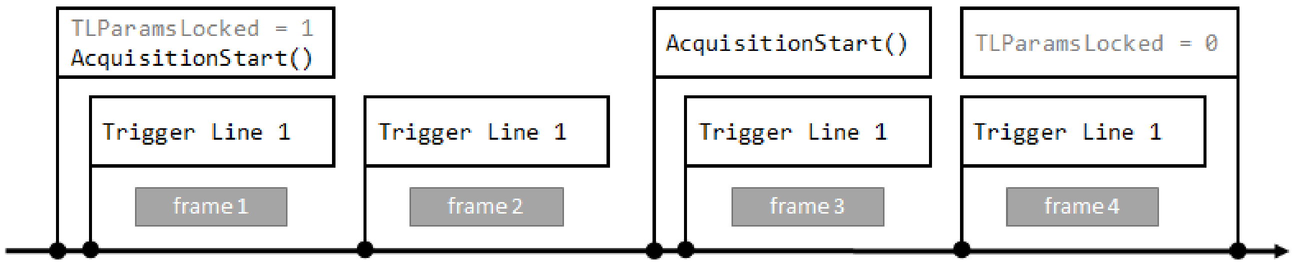
AcquisitionStart命令激活数据流。然而，在触发信号之后，才开始记录指定图像，在样本中为线路 1 上的硬件触发信号之后。与多图像采集（无触发器）一样，AcquisitionStop命令和取消锁定可以选择在达到指定的图像数量后进行。



带单一图像触发的多图像采集

配置：

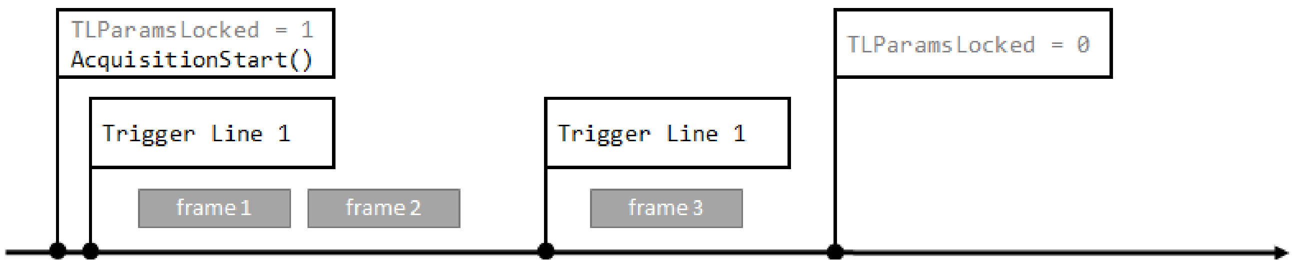
- AcquisitionMode = MultiFrame
- AcquisitionFrameCount = 2 (example)
- TriggerSelector = FrameStart
- TriggerMode = On
- TriggerSource = Line1 (example)



带多图像触发器的多图像采集

配置：

- AcquisitionMode = MultiFrame
- AcquisitionFrameCount = 3 (example)
- AcquisitionFrameBurstCount = 2 (example)
- TriggerSelector = FrameBurstStart
- TriggerMode = On
- TriggerSource = Line1 (example)



8.4.3.3 单一图像采集

默认情况下，图像采集和触发的设置处于互锁状态。下面的解释部分假设AcquisitionMode设置为SingleFrame。

其他情况描述如下：

- 连续的图像采集 [▶ 2730]

- 多图像采集 [▶ 2732]

检查相机的功能

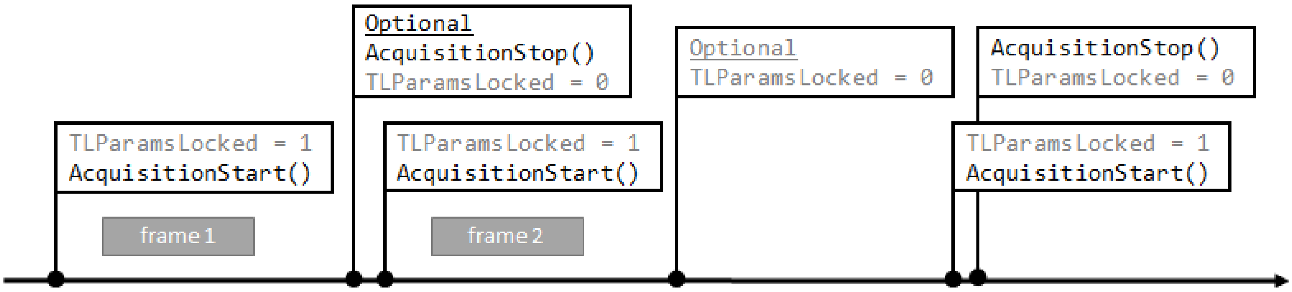
请参考相机手册，了解相机是否支持相应的功能。这同样适用于实际使用的功能名称。根据 GenICam 标准功能命名公约（2.4 版），使用功能名称的程序如下。

单一图像采集

如果在每个AcquisitionStart命令后只记录一个图像，则需要以下配置：

- AcquisitionMode = SingleFrame
- TriggerSelector = AcquisitionStart
- TriggerMode = Off

在AcquisitionStart命令之后，仅拍摄一个图像。在两个AcquisitionStart命令之间，可以发出一个AcquisitionStop命令，并且可以取消锁定。解锁有利于在AcquisitionStart命令之间更改相机功能。如果在AcquisitionStart命令之后过早地发出AcquisitionStop命令，在图像采集尚未开始的情况下，将不捕捉图像。



带有图像采集触发器的单一图像采集
或带有单一图像触发器的单一图像采集

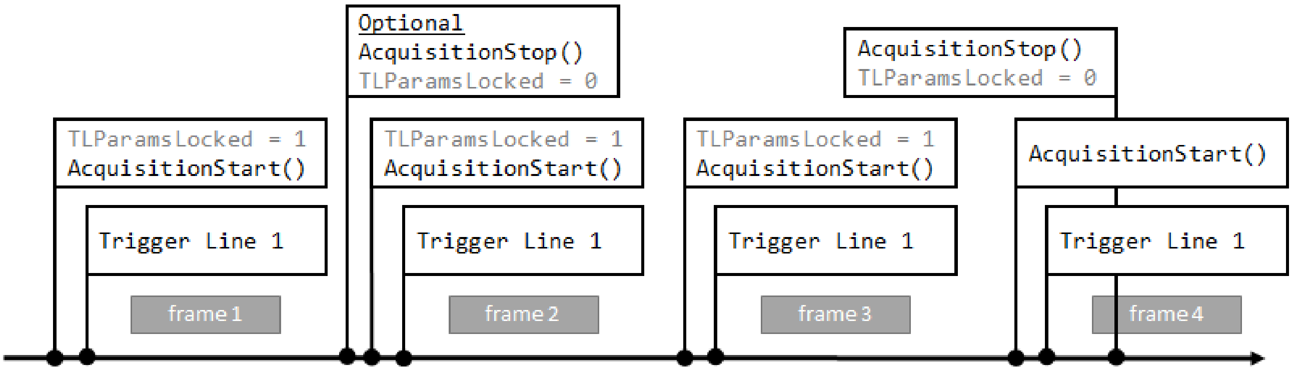
如果在需要触发信号的情况下每次AcquisitionStart命令后只记录一幅图像，可以使用以下配置：

配置 – 带有图像采集触发器的单一图像采集

- AcquisitionMode = SingleFrame
- TriggerSelector = AcquisitionStart
- TriggerMode = On
- TriggerSource = Line1 (example)

配置 – 带单一图像触发器的单一图像采集

- AcquisitionMode = SingleFrame
- TriggerSelector = FrameStart
- TriggerMode = On
- TriggerSource = Line1 (example)



9 附录

在这一章中，您会发现更多的信息，如常见问题解答、TwinCAT Vision 使用的路径概述以及用于纠正已知问题的故障排除章节。

9.1 常见问题解答

TwinCAT Vision 是什么？

TwinCAT Vision 是一款工业图像处理软件。你可以用工业相机捕捉图像，并从中计算出所记录场景的信息。

常见使用案例：

- 产品有无检测
- 对象识别
- 光学测量
- 代码读取

可以免费测试 TwinCAT Vision 吗？

是的，你可以为运行时组件生成 7 天的试用许可证。该工程设计免费。

为了使用 TwinCAT Vision，是否需要 TwinCAT？

是的，TwinCAT Vision 已经完全集成在 TwinCAT中。在安装 TwinCAT Vision 之前，必须先安装一个兼容的TwinCAT版本。

可以在 TwinCAT Vision 中使用哪些相机？

TwinCAT Vision 支持带有 GigE Vision 接口的相机。GigE Vision 相机通过以太网电缆连接到 IPC。

我可以同时使用几台相机吗？

是的，通过 TwinCAT GigE Vision 连接器，可以操作多个相机，具体取决于许可证。

我可以通过一个开关连接几台相机吗？

是的 注意最大的可用带宽（例如，在 1 千兆以太网的情况下为 125 MB/s，实际上接近 110-115 MB/s）。

是否有用于 TwinCAT Vision 的样本？

是的，在样本 [▶ 2639]一章中。

是否有 TwinCAT Vision 功能的基准？

没有，因为视觉算法的执行时间在很大程度上取决于 IPC 性能、图像大小、功能参数和图像内容。因此，对算法如何工作以及哪些因素影响相关计算时间有一个基本的了解是很有帮助的。

其他图像处理软件是否也可以在 TwinCAT 中使用，作为 TwinCAT Vision 的替代？

通过 TC3 GigE Vision 连接器，可以通过 GigE Vision 记录图像。然后可以通过各种方式进行图像处理：

- 通过 PLC 库 Tc3_Vision 使用 TwinCAT Vision API
- 使用自行编写的 TwinCAT PLC 或 C++ 算法
- 在一个 C++ 模块中使用 TwinCAT Vision API
- 使用 Matlab/Simulink
- 通过 ADS 从其他应用程序检索图像数据

TwinCAT Vision 等图像处理软件与智能相机之间有什么区别？

图像处理软件，如 TwinCAT Vision：

- 从相机接收图像数据
- 在电脑上进行处理
- 向控制器发送处理过的信息
- TwinCAT Vision 的优势：控制和图像处理是同一个软件，并在同一个系统中运行。因此，不需要额外的接口和协议。

智能相机：

- 配有内部处理器
- 直接在相机上进行图像处理操作
- 通过附加接口提供处理过的信息，如好/坏信息或工件的位置

TwinCAT Vision 可以在 GPU 上执行吗？

不，TwinCAT Vision 只在 TwinCAT 实时环境中的 CPU 上运行。另外，TwinCAT Vision 通过使用 TwinCAT 的多核功能（与工作任务 [▶ 58]互动）提供了自动并行化执行的选项。

9.2 见重要路径

Vision 数据路径

注意

数据相关路径

确保以下标准路径不受 Write filters 或类似程序的影响。否则可能导致数据丢失。如果手动调整了相应的路径设置，则设置的路径也会相应调整。

应用	路径
文件源图像的存储位置	C:\Users\Public\TcVision\FileSources\<UniqueId> （由文件源自动定义。每个项目的 <UniqueId> 都不同）。
相机数据流的存储位置	C:\Users\Public\TcVision\CameraStreams （可能根据记录/回放 [▶ 89]中的设置而有所不同）。
PLC 功能读取或写入的图像的存储位置。	C:\Users\Public\TcVision\Images （可能根据服务配置 [▶ 59]中的设置而有所不同）。
PLC 功能读取或写入的容器的存储位置。	C:\Users\Public\TcVision\Containers （可能根据服务配置 [▶ 59]中的设置而有所不同）。
PLC 功能读取或写入的 ML 模型的存储位置。	C:\Users\Public\TcVision\MlModels （可能根据服务配置 [▶ 59]中的设置而有所不同）。
PLC 功能读取或写入的标定结果的存储位置。	C:\Users\Public\TcVision\CalibrationResults （可能根据服务配置 [▶ 59]中的设置而有所不同）。
单个标定模板的存储位置	C:\Users\Public\TcVision\CalibrationPattern

日志文件路径

日志文件	路径
TwinCAT Vision 安装日志 (所有其他 TwinCAT 安装日志也位于此文件夹中)	C:*Users**<User>*\ TF7xxx-Vision.exe.log Build 4026 及以上版本： C:\ProgramData\Beckhoff\TcPkg\logs\ 单个产品日志文件位于 MSI 子文件夹中。
TwinCAT Vision Service 日志	C:\ProgramData\Beckhoff\TcVnService\TcVnService.log
一般 TwinCAT Vision 日志	C:\ProgramData\Beckhoff\Vision （可能因 记录 [▶ 61] 中的设置而异）
标定助手 [▶ 94]的结果图像用于分析标定模板上特征点的识别	C: \ProgramData\Beckhoff\Vision_CalibrationAssistantOutput

9.3 故障排除

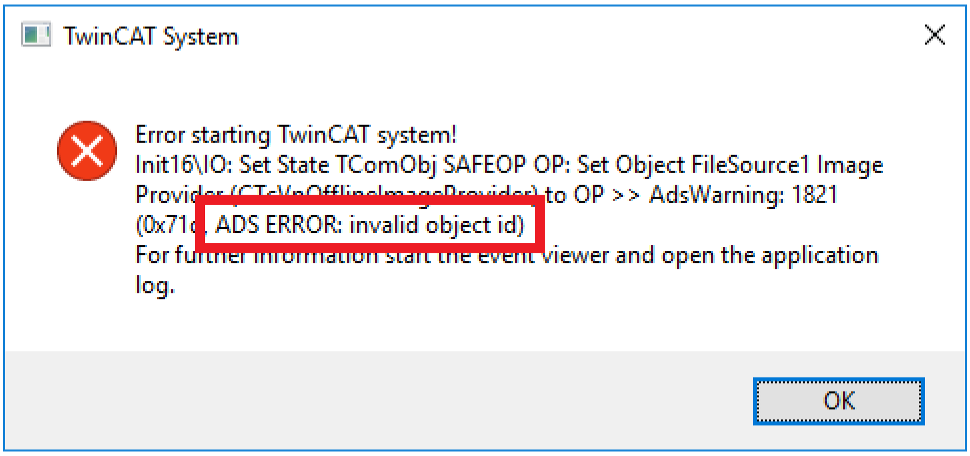
已知的错误情况和相应的解决方案描述如下。如果可以将问题归咎于特定函数或功能块，则首先阅读API 参考 [▶ 119]中的相应条目！

9.3.1 TwinCAT 系统启动

下面解释了 TwinCAT 系统无法启动的情况下的解决方案。

9.3.1.1 TwinCAT 系统不能启动

如果 TwinCAT 系统不能启动，经常出现以下信息。根据带有红色边框的错误代码，可以给出纠正问题的指示。在下文中，提供了有关 TwinCAT Vision 应用程序的常见错误代码的信息。



无效对象 ID

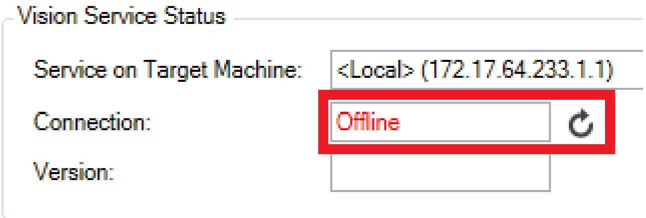
错误信息	ADS ERROR：无效对象 id
解决方案 1	**未链接任务** 1. 检查是否已经为所使用的每个视觉设备创建了相应的任务，并添加缺少的任务。 ◦ 检查是否为每个文件源控制创建了图像采集的任务（默认周期为 10 ms）。 ◦ 检查每个 GigE Vision 相机是否与图像采集任务相关联（默认周期为 1 ms）。如果有 Ads Communicator，也与任务相关联（默认周期为 10 ms）。 2. 这些任务必须分配给视觉设备的相应 TcCom 模块。检查作业，必要时予以纠正。 ◦ 检查文件源的图像提供者任务是否与文件源控制的图像提供者相连。 ◦ 检查相机的图像采集任务是否与相机的图像采集和图像采集模拟相联系。 ◦ 将 Ads Communicator 任务与相机的 Ads Communicator 联系起来。 ◦ 请勿将任务分配给相机的图像提供者。
解决方案 2	**网络适配器无效** 网络适配器未调整，例如在系统改变之后。根据前的目标系统，调整网络适配器。 在相关网络适配器中，打开**适配器**选项卡，并通过**搜索**选择新的兼容适配器。
解决方案 3	**网络适配器已删除** 网络适配器已删除，例如在假设相机的模拟不需要的情况下。然而，每个相机对象始终需要网络适配器；如有必要，可以停用但不能删除。 通过 IpStack 创建新的网络适配器，并在相机采集对象的**接口指针**选项卡上链接 IpStack。

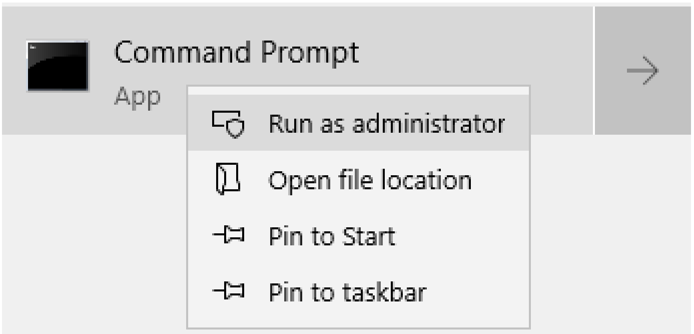
关于连接 Vision TcCOM 对象的更多信息，可查看相机 [▶ 108]和文件源 [▶ 113]的相应章节。

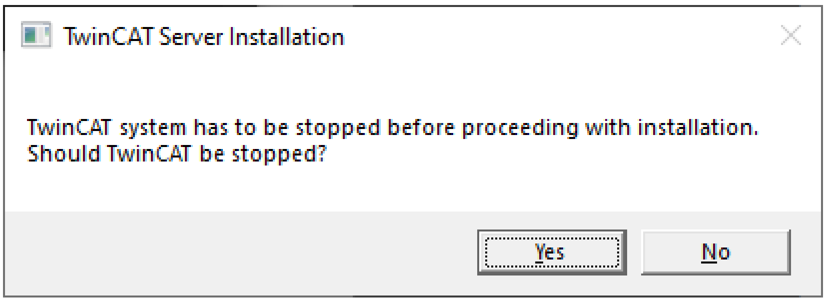
一般性 ADS 错误

错误信息	ADS ERROR：一般 ADS 错误
解决方案 1	**相机比配置的多** 网络适配器的默认设置最多允许 2 台相机同时操作。如果想要在同一适配器上操作更多的相机，请将初始参数ipMaxReceiver和UdpMaxReceiver增加到所连接相机数量的两倍（见网络适配器的配置 [▶ 50]）。

9.3.1.2 视觉服务不启动

<table>
<tr><td>情况</td><td>TwinCAT Vision 服务没有启动。这可以通过视觉节点的服务配置选项卡上的连接状态“离线”查看。

</td></tr>
<tr><td>解决方案</td><td>如果 TwinCAT 版本服务只启动失败一次或失败，可以通过 TwinCAT 重启来重新激活（点击  ）。
然而，如果它通常无法启动（在系统启动后），则必须将该服务重新注册为 TwinCAT 服务。为此，请遵循以下步骤：

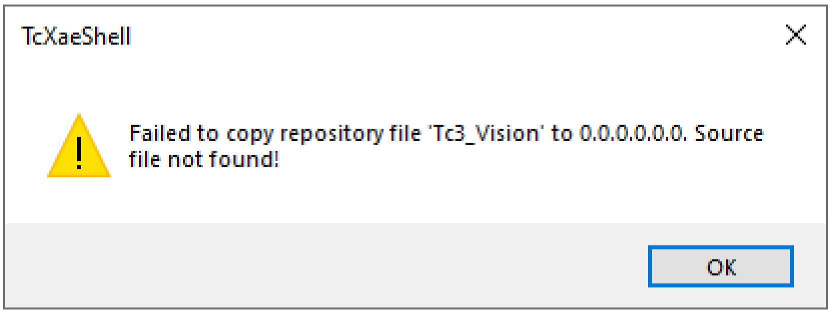
1. 将 TwinCAT 切换到配置模式 
2. 以管理员身份打开提示。


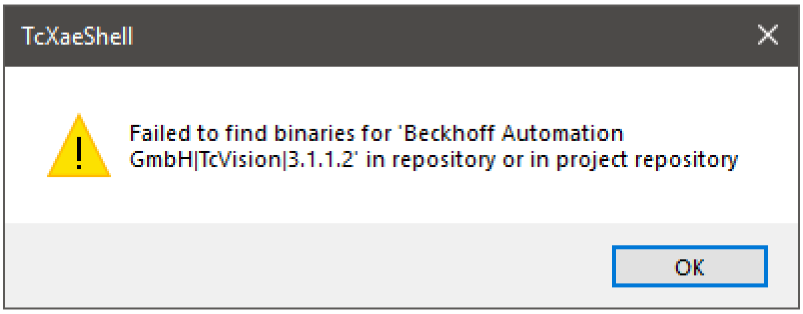
3. 通过“cd C:\TwinCAT\3.1\Components\Vision\TcVision”切换到相应的文件夹。
4. 执行“TcVnService.exe /tc3serverreg”命令，以启动视觉服务。
5. 通过点击是，确认随后的对话框。


⇨ 视觉服务启动。
TwinCAT 必须处于配置模式，才能启动视觉服务。因此，如果需要重新启动视觉服务，必须重新启动机器。
如果需要手动停止视觉服务，可以使用命令“TcVnService.exe /unregtc3server”。</td></tr>
</table>

9.3.1.3 无法找到/加载二进制文件或存储库

<table>
<tr><td>情况</td><td>在配置激活时，会出现两个信息中的一个。根据原因和安装版本，其他文件名或版本号可能出现在信息中。
该信息的原因在于，项目中使用的版本与系统中安装的版本不一致。





</td></tr>
<tr><td>解决方案 1</td><td>检查视觉库
检查项目中包含的 Tc3_Vision 库的版本。
如果版本与工程系统 C:\TwinCAT\3.1\Repository\Beckhoff Automation GmbH\Tc3_Vision 文件夹中的版本不一致，则更新项目中的库，或安装另一个安装版本。</td></tr>
<tr><td>解决方案 2</td><td>检查相机或 FileSorce 的 TMC 版本
在系统 >TcCOM 对象 >项目对象下检查项目所用列出对象的 TMC 版本。如果版本以黄色突出显示，则说明系统中不存在指定的版本或不匹配。
现在，更新相应的 TMC 版本（通过版本的下拉菜单选择版本，然后右击第一列中的一个对象 ID，并选择通过更改版本重新加载 TMI/TMC 描述）到可用版本或安装另一个设置版本。</td></tr>
<tr><td>解决方案 3</td><td>视觉组件不可用
当前系统中缺少相应的存储库文件，因为安装时没有选择视觉组件。基本上，工程和目标系统的版本必须匹配。因此，如果系统不同，请检查这两个装置。在纯运行时系统 XAR 上，仅视觉服务可以安装，因为所有其他文件都由工程系统 XAE 传输。
安装缺失的视觉组件或使用修复选项运行设置。</td></tr>
</table>

9.3.2 PLC 运行时环境

下面解释了在 PLC 运行时发生错误的情况的解决方案。

9.3.2.1 视觉设备处于 ERROR 状态

相机功能块在 ERROR 和 INITIAL 状态之间切换

情况	功能块FB_VN_Gev_CameraControl [▶ 1550]或FB_VN_SimpleCameraControl [▶ 1575]的实例在调用方法GetState时交替返回TCVN_CS_ERROR和TCVN_CS_INITIAL。
	在TCVN_CS_ERROR状态下，功能块的 Reset() 方法已成功执行，且相机实例变为TCVN_CS_INITIAL状态。如果相机一直处于TCVN_CS_ERROR状态，请参见设备长期处于 ERROR 状态 [▶ 2742]。
	`eState := fbCamera.GetState()`
	`IF eState = TCVN_CS_ERROR THEN`
	`    fbCamera.Reset();`
	`END_IF`
	
	在 TCVN_CS_INITIAL 状态下，则调用了 OpenCamera() 或 StartAcquisition() 的功能块方法。调用失败，且实例返回状态 TCVN_CS_ERROR。
	`eState := fbCamera.GetState()`
	`IF eState = TCVN_CS_INITIAL THEN`
	`    fbCamera.OpenCamera();`
	`END_IF`
	或
	`eState := fbCamera.GetState()`
	`IF eState = TCVN_CS_INITIAL THEN`
	`    fbCamera.StartAcquisition();`
	`END_IF`

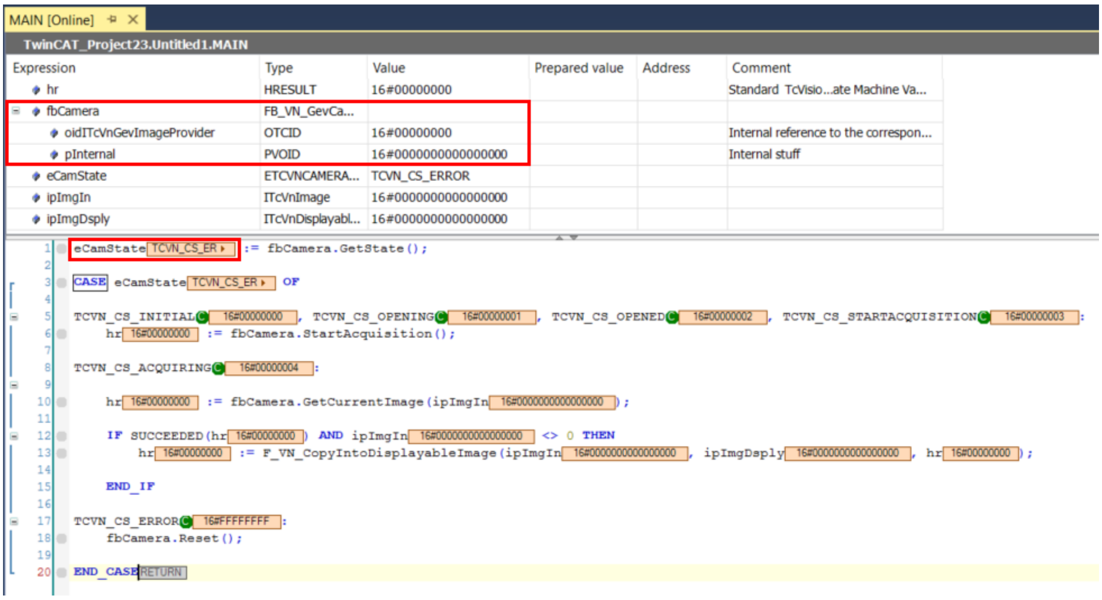
解决方案 1	检查相机的物理连接和相机的电源。
解决方案 2	检查相机连接的设置（相机的 IP 地址、网络端口）。

功能块长期处于 ERROR 状态

<table>
<tr><td>情况</td><td>

功能块的实例FB_VN_GevCameraControl、FB_VN_SimpleCameraControl或FB_VN_FileSourceControl长期返回状态TCVN_CS_ERROR。即使在调用功能块方法 Reset 时，也没有改变状态为TCVN_CS_INITIAL（如果有，见设备在 ERROR 和 INITIAL 状态之间切换 [▶ 2741]）。

```
fbCamera.Reset();
```

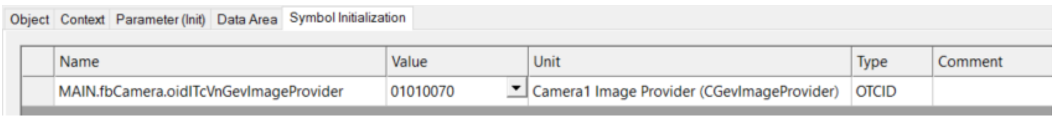


</td></tr>
<tr><td>解决方案 1</td><td>

符号实例必须链接到图像提供者 – FB_VN_GevCameraControl、FB_VN_SimpleCameraControl和FB_VN_FileSourceControl的每个实例必须链接到视觉节点中创建的相机或文件源控制。

请按以下步骤进行：

1. 进行 PLC 项目的成功重建。
2. 然后双击 PLC 项目的实例。
3. 在**符号初始化**选项卡下，将相应相机的图像提供者或相应的文件源控制分配给FB_VN_GevCameraControl的实例。
4. 重新激活配置。



FB_VN_GevCameraControl 只能链接到相机的图像提供者。

FB_VN_FileSourceControl 只能链接到文件源控制的图像提供者。

FB_VN_SimpleCameraControl 可以链接到相机的图像提供者或文件源控制的图像提供者。

</td></tr>
<tr><td>解决方案 2</td><td>

FB_VN_GevCameraControl或FB_VN_SimpleCameraControl或FB_VN_FileSourceControl的实例被指定为相机或文件源控制的图像提供者，但设备却长期处于错误状态。

检查FB_VN_GevCameraControl或FB_VN_SimpleCameraControl或FB_VN_FileSourceControl的实例的方法调用。许多方法仅可在某种状态下被调用。关于视觉设备状态的详细描述，可查看图像采集的功能块 [▶ 1540]部分。

- 在所有状态下
  - GetState
  - GetCalibPatternRef （来自 FB_VN_GevCameraControl）
  - GetCameraMatrix （来自 FB_VN_GevCameraControl）
  - GetDistortionCoefficients （来自 FB_VN_GevCameraControl）
  - GetRotationMatrix （来自 FB_VN_GevCameraControl）
  - GetTranslationVector （来自 FB_VN_GevCameraControl）
  - SetCameraMatrix （来自 FB_VN_GevCameraControl）

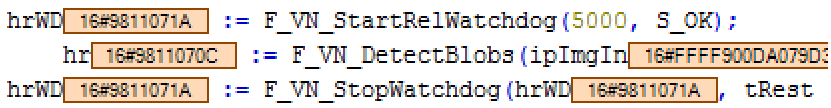
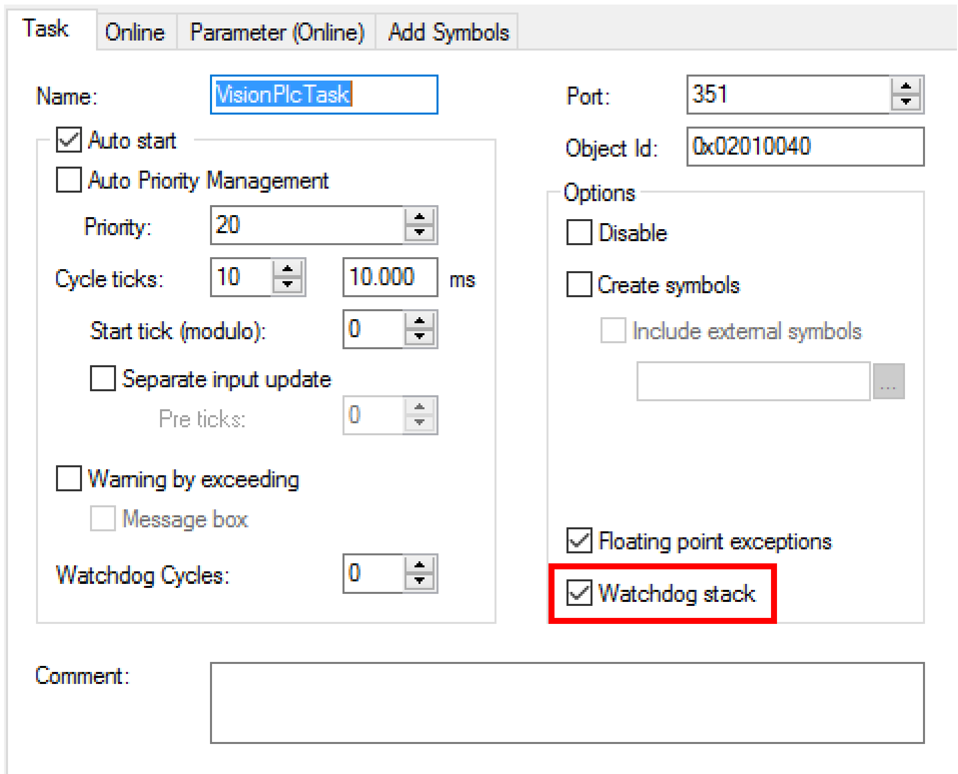
</td></tr>
</table>

  - SetDistortionCoefficients （来自 FB_VN_GevCameraControl）
  - SetRotationMatrix （来自 FB_VN_GevCameraControl）
  - SetTranslationVector （来自 FB_VN_GevCameraControl）
- 在 INITIAL 状态和 INITIALIZING 状态下
  - InitializeCamera（来自 FB_VN_GevCameraControl）
- 在 INITIAL 状态和 OPENING 状态下
  - OpenCamera（来自 FB_VN_GevCameraControl）
  - StartAcquisition
- 在 OPENED 状态和 STARTACQUISITION 状态下
  - StartAcquisition
- 在 OPENED 状态和 CLOSING 状态下
  - CloseCamera （来自 FB_VN_GevCameraControl）
- 在 ACQUIRING 状态下
  - GetCurrentImage
  - GetCurrentImageUndistorted（来自 FB_VN_GevCameraControl）
  - GetCurrentImageWithGvspInfo（来自 FB_VN_GevCameraControl）
  - GetCurrentImageAndFileName（来自 FB_VN_FileSourceControl）
  - ClearImageQueue（来自 FB_VN_GevCameraControl）
  - GetLastImageFromQueue（来自 FB_VN_GevCameraControl）
  - GetOmittedImagesNum（来自 FB_VN_GevCameraControl）
- 在 ACQUIRING 状态和 TRIGGERING 状态下
  - TriggerImage
  - TriggerImageExp（来自 FB_VN_FileSourceControl）
  - TriggerImageByName（来自 FB_VN_FileSourceControl）
- 在 ACQUIRING 状态和 STOPACQUISITION 状态下
  - StopAcquisition
- 在 ERROR 状态下
  - Reset

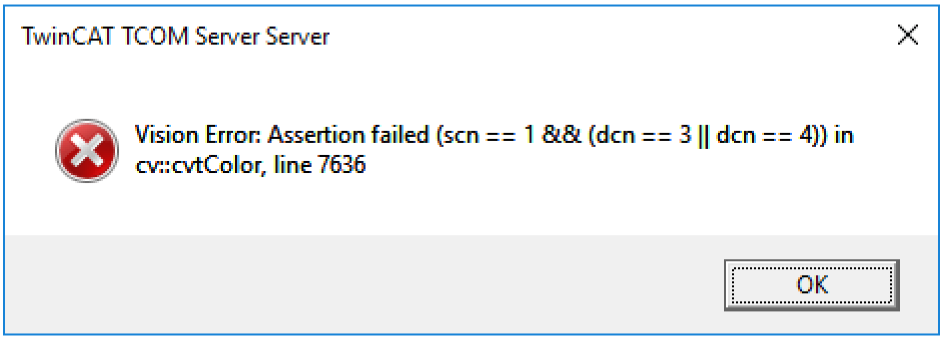
9.3.2.2 重新启动后找不到相机

情况	应用程序完全照常工作。重新启动机器后，无法与相机建立连接。
解决方案 1	IP 地址已经改变。使用相机的配置助手 [▶ 67]中的 Force-IP 选项。

9.3.2.3 启动看门狗时出错

情况	函数F_VN_StartAbsWatchdog [▶ 948]或F_VN_StartRelWatchdog [▶ 949]长期返回错误代码16#71A（NOINTERFACE）。 
解决方案	在相应的 PLC 任务中激活看门狗堆栈，然后重新激活配置 。 

9.3.2.4 视觉错误：断言失败

错误信息	在执行视觉应用程序的过程中，出现了类似以下的错误信息。同时，TwinCAT 进入 EXCEPTION 状态。  这条信息表明，在执行视觉函数的过程中，存在一个无法拦截的冲突。这基本上是指输入数据的正确格式，且应仅在函数没有按照 API 参考使用时才会发生。在“VisionError: Assertion failed”背后的错误规范可能因用例不同而不同。
解决方案与 F_VN_ConvertColorSpace [▶ 1197]	图像可以是单通道，也可以是多通道图像。对于某些函数的调用，预计使用具有定义通道数量的图像。例如，对于函数F_VN_ConvertColorSpace，其中枚举ETcVnColorSpaceTransform用于指定输入图像的格式。如果工程部登录，相应的函数调用会在 PLC 代码中被标记出来。函数F_VN_GetPixelFormat（返回实际通道数量）可以帮助分析为什么实际的通道数量与预期的数量不同。 典型原因： • 文件源控制 在文件源中，设置了与预期不同的图像格式，或者图像格式被设置为“来自文件的原始格式”，而实际图像格式与预期不同。例如，看起来是单色的图像并不总是作为单色图像（单通道图像）存储。 • 相机 对于许多相机，可以在配置中设置像素格式。根据所选择的格式，图像通道的数量也会改变。 • 以前的处理步骤 以前的函数调用可能已经改变了图像，且因此也改变了通道的数量。

9.3.2.5 错误：SSE 无效操作

错误信息	得到错误信息SSE 无效操作。
解决方案	如果在执行的任务上激活了**浮点异常**处理，将以不必要的方式发生这个错误。 按照CPU 内核和任务 [▶ 55]部分的说明，停用选项**浮点异常**。

9.3.2.6 文件来源

错误状态

错误模式	当 PLC 运行时，文件源不断切换到错误状态。
解决方案 1	**内存太少** 路由器内存设置太小。因此，文件源不能将任何图像加载到 PLC 中，并在试图加载时进入错误状态。扩大路由器的内存。

不在 PLC 中的 FS 控制的图像

错误模式	在文件源控制中选择了图像，而 PLC 中的文件源功能块处于ACQUISITION状态。尽管如此，仍无法通过GetCurrentImage检索到任何图像。
解决方案 1	**未设置 BaseDir** 在 TwinCAT 激活后，首次启动了**从目标读取**按钮。在这种情况下，读取文件源图像的路径还没有转移到运行中的 TcCOM 对象。因此，即使在文件源控制中选择了图像，也不能加载任何图像。在按下**从目标读取**后，重新激活 TwinCAT。
解决方案 2	**触发模式** 在文件源控制中选择了触发模式，但在 PLC 中只调用了方法GetCurrentImage，没有调用TriggerImage。通过调用 PLC 中的方法TriggerImage，触发一个图像，或者在文件源控制中停用触发模式。

9.3.3 助手

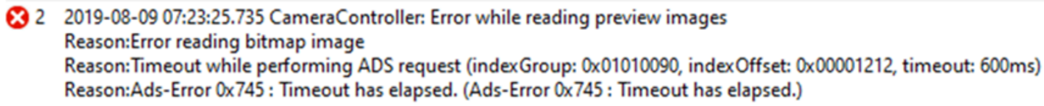
下面解释了助手操作出现问题的情况下的解决方案。

9.3.3.1 超时

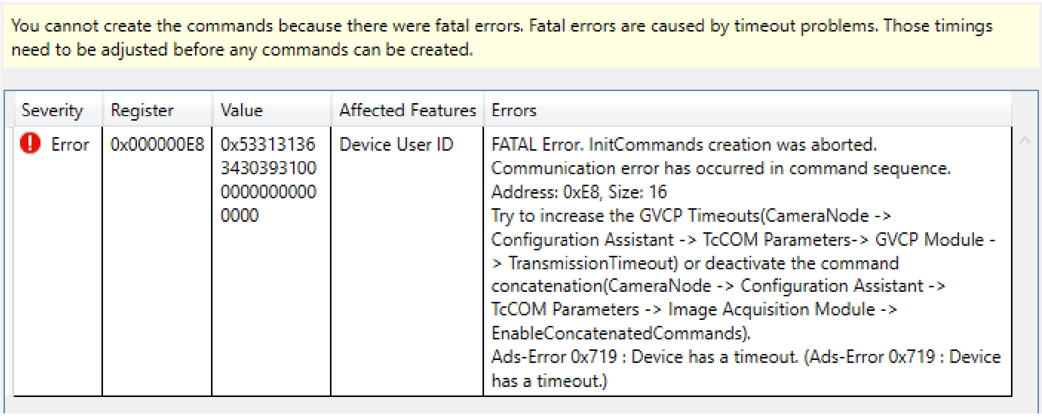
<table>
<tr><td>情况</td><td>

在与相机通信过程中发生超时错误。正如以下案例所述，这可能发生在不同的地方。

在开发环境中的图像显示期间：



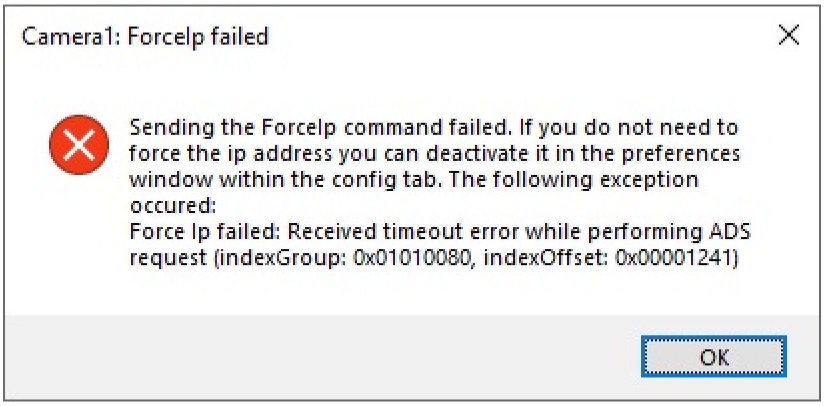
在创建初始化命令时：



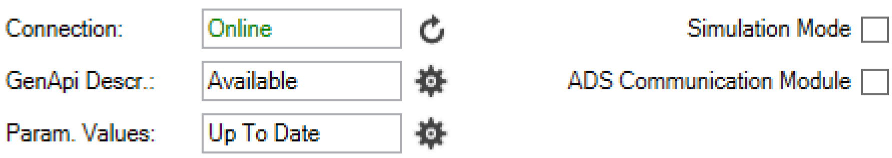
You cannot create the commands because there were fatal errors. Fatal errors are caused by timeout problems. Those timings need to be adjusted before any commands can be created.

Severity	Register	Value	Affected Features	Errors
Error	0x000000E8	0x53313136 3430393100 0000000000 0000	Device User ID	FATAL Error. InitCommands creation was aborted. Communication error has occurred in command sequence. Address: 0xE8, Size: 16 Try to increase the GVCP Timeouts(CameraNode -> Configuration Assistant -> TcCOM Parameters-> GVCP Module -> TransmissionTimeout) or deactivate the command concatenation(CameraNode -> Configuration Assistant -> TcCOM Parameters -> Image Acquisition Module -> EnableConcatenatedCommands). Ads-Error 0x719 : Device has a timeout. (Ads-Error 0x719 : Device has a timeout.)

当使用 ForceIP 命令时：



在相机对象中建立连接时：



</td></tr>
<tr><td>解决方案</td><td>

在超时错误的情况下，两种不同类型的超时基本上相关：一种是传输超时，它限制了图像采集模块和相机之间的通信。另一种是 ADS 通信超时，它限制了用户界面和图像采集模块之间的通信。

在配置树 [▶ 70]中调整 GVCP 模块的 TcCOM 参数中的传输超时。同时调整 MaxTimeouts 的数量。

如有必要，确保在设置 [▶ 74]中调整 ADS 通信超时，因为这应该大于 MaxTimeouts 和传输超时的乘积。

逐步增加次数，必要时增加尝试次数，直到不再出现错误信息。然而，确保不要把时间和尝试次数设置得过高，否则会出现非常长的等待时间。

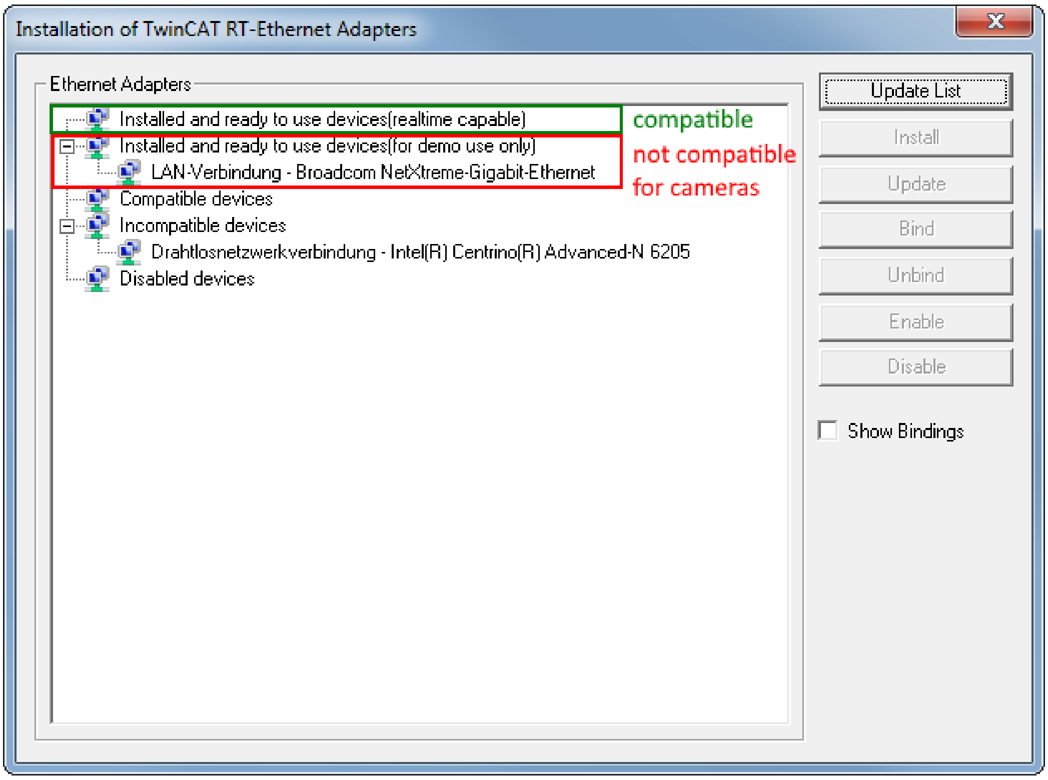
</td></tr>
</table>

9.3.3.2 图像显示停滞

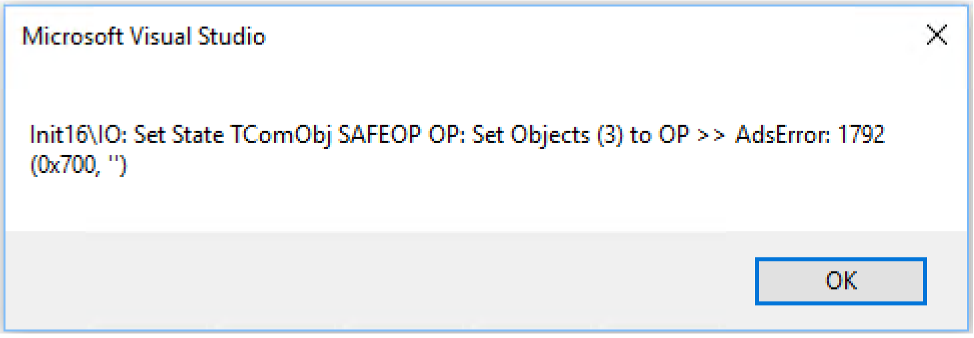
错误模式	ADS 图像查看和相机助手中的图像显示偶尔会停顿。
解决方案 1	**远程桌面协议** 在个别情况下，远程桌面协议和网卡可能与 TwinCAT Vision 开发环境或图像显示工具的传输不匹配。 请在本地操作并连接 TwinCAT 中的目标设备，或者直接在目标计算机上操作。

错误模式	部分图像无法显示在 ADS 图像查看和相机助手中。
解决方案 1	**减少数据量** 由于图像是由 ADS 从控制器中检索以便显示，因此可能无法检索到所有的图像数据，特别是在大图像、高帧率或几个 ADS 图像查看窗口的情况下。 因此，使用 ROI 减少同时显示的图像数量，并降低每张图像的帧率或数据量。

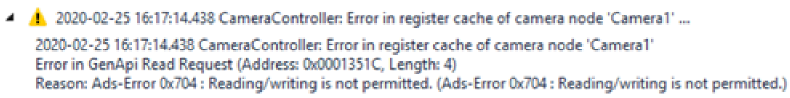
9.3.3.3 未通过设备发现找到相机

情况	你正在创建一个 GigE Vision 相机对象。您已经成功设置了一个带有 IP 堆栈的 TwinCAT RT-Ethernet 适配器。 然而，“设备发现”对话框仍然为空，且没有错误信息。
解决方案 1	**选择了错误的网络适配器**
解决方案 2	**没有 GigE Vision 相机** 具有千兆以太网接口的相机不一定要“讲”GigE Vision。请密切注意制造商对传输协议的说明。
解决方案 3	**不兼容的网络适配器** 使用的网络适配器... 

9.3.3.4 加载相机对象时出错

错误信息	 在配置模式下，当重新加载相机模块或需要通过选择目标相机对话框中的发现设备按钮选择相机时，会出现以下 Visual Studio 信息。 
解决方案	连接的相机比配置的要多。 网络适配器的默认设置最多允许 2 台相机同时操作。如果想要操作更多相机，请将网络适配器的初始参数ipMaxReceiver和UdpMaxReceiver增加到所连接相机数量的两倍（见网络适配器的配置 [▶ 53]）。

9.3.3.5 警告：不允许读取/写入

警告	 在连接相机时可能会出现这个警告。
解决方案	该警告表明，TwinCAT Vision 向相机发送的写或读命令不被允许。如果 GenAPI 描述中对寄存器的访问权限的定义偏离了相机的实际行为，就会发生这种情况。通常情况下，这个相机寄存器对于用户来说并不重要。如需确定这与哪个寄存器有关，可以在配置助手 [▶ 67]中搜索显示的寄存器地址。

9.3.4 TwinCAT HMI 软件包

对于 TwinCAT 人机界面软件包不能按预期工作的情况，下面将说明解决方案。

9.3.4.1 服务器扩展

不显示图像

情况	尽管安装了服务器扩展 [▶ 2599]，并且适当地链接了图像符号，但来自 PLC 的图像没有显示在 HMI 中。
解决方案 1	缺少管理员权限 您选择了“数据流”作为图像格式，但您还没有以管理员身份启动 HMI 服务器。选择不同的图像格式（BMP、PNG 或 JPG），或以管理员身份启动 HMI 服务器。

已发布的页面中缺少图片

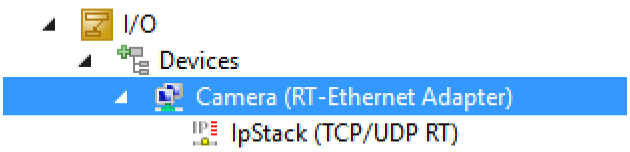
情况	TcHmiVision [▶ 2599]服务器扩展已经成功集成到项目中。图像显示在实时视图或工程服务器提供的 HMI 上。该项目也可以在独立的 TcHmi 服务器上无错误地发布。然而，在公布的 HMI 上没有显示图像。
解决方案 1	**缺少目标许可证** TC3 HMI 服务器许可证中包括一个目标许可证。如果 HMI 服务器需要与多个目标（ADS 连接）进行通信，则需要额外的许可。
解决方案 2	**运行时环境的配置错误** 工程中使用的默认配置可能与公布的配置不同（不同的 AmsNetIds）。因此，请检查运行时环境的配置，并在必要时进行调整。
解决方案 3	**用户权利受限** 使用 TwinCAT HMI 用户管理，而当前登录的用户没有权限查看图像或接收扩展数据。检查你的用户组的授权。
解决方案 4	**限制数据量或刷新时间间隔** 由于数据量可能相当大，特别是在几个图像必须同时由 ADS 从控制器检索并由 HMI 服务器转换的情况下，这可能导致瓶颈。有几个选项可以补救：降低控制器中的图像大小，增加单个图像的刷新率，使用不同的刷新率或使用压缩的图像格式。

9.3.5 相机通信

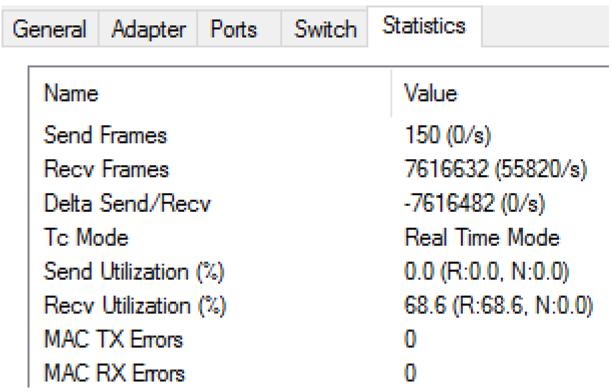
如果连接有问题，特别是发送或接收的帧有问题，以及参数化有问题或丢失图像，你会在网络适配器和相机对象的图像采集模块下找到以下诊断信息。

RT 以太网适配器中的诊断选项

1. 首先选择用于各自相机通信的 RT 以太网适配器。



2. 通过双击打开，并进入**统计**选项卡。



⇨ 在这里可以看到帧是否被普遍发送和接收，或者是否发生错误。此外，还可以通过Recv Utilization，了解连接的利用率（百分比）。超过 98% 的非常高的利用率已经表明过载了，特别是如果在一个适配器上操作几台相机。

进一步的诊断信息可以在 IpStack 中找到。

1. 如需查看，首先通过双击打开底层的 IpStack。
2. 转到**参数（在线）**选项卡。

⇨ 在这里可以看到各个类别的计数器值。

Object | Context | Parameter (Init) | Parameter (Online) | Interfaces | Interface Pointer

	Name	Online	CS	Type	PTCID
-	IpStackDiagnosis	...	☐		0x03011108
	.ip.nSendCnt	1372		UDINT	
	.ip.nSendFailCnt	0		UDINT	
	.ip.nRecvCnt	392048		UDINT	
	.ip.nRecvFailCnt	0		UDINT	
	.arpRequest.nSendCnt	1		UDINT	
	.arpRequest.nSendFailCnt	0		UDINT	
	.arpRequest.nRecvCnt	0		UDINT	
	.arpRequest.nRecvFailCnt	0		UDINT	
	.arpReply.nSendCnt	0		UDINT	
	.arpReply.nSendFailCnt	0		UDINT	
	.arpReply.nRecvCnt	1		UDINT	
	.arpReply.nRecvFailCnt	0		UDINT	
	.pingRequest.nSendCnt	0		UDINT	
	.pingRequest.nSendFailCnt	0		UDINT	
	.pingRequest.nRecvCnt	0		UDINT	
	.pingRequest.nRecvFailCnt	0		UDINT	
	.pingReply.nSendCnt	0		UDINT	
	.pingReply.nSendFailCnt	0		UDINT	
	.pingReply.nRecvCnt	0		UDINT	
	.pingReply.nRecvFailCnt	0		UDINT	
	.nLinkStatusChangedCnt	1		UDINT	
	.nAllocFailCnt	0		UDINT	
	.nArpTimeoutFrames	0		UDINT	
	.nDroppedFrames	0		UDINT	

图像采集模块中的诊断选项

1. 首先选择相应的相机对象。

- Camera1
  - Camera1 Image Acquisition (CGevImageAcquisition)
    - Gvcp Module (CGvcpProtocol)
    - Gvsp Module (CGvspProtocol)
  - Camera1 Image Acquisition Simulation (CGevImageAcquisitionSimulation)
  - Camera1 Ads Communicator (CGevAdsCommunicator)
  - Camera1 Image Provider (CGevImageProvider)

2. 打开 Gvcp 模块下的**参数（在线）**选项卡。

⇨ 在这里可以得到有关控制协议通信的信息。这包括例如心跳、寄存器的读写和命令（如开始采集）。如果一切正常，仅两个 nSuccess 值应该计数。如果发生超时，请参考本章超时 [▶ 2748]了解补救措施。对于其他问题，需要使用 Wireshark 进行更深入的诊断。

Object | Context | Parameter (Init) | Parameter (Online) | Interfaces

	Name	Online	PV	CS	Unit	Type	PTCID	Comment
-	GvcpProtocolDiagnosis	...		☐			0x00010001	Contains statistics for GVCP protocol diagnosis.
	.stSend.nSuccess	1242				ULINT		Number of successfully sent packets
	.stSend.nError	0				ULINT		Number of packets causing errors during transmission
	.stSend.nTimeout	0				ULINT		Number of timeouts that have occurred
	.stRecv.nSuccess	1189				ULINT		Number of successfully received packets
	.stRecv.nError	0				ULINT		Number of received packets signalling or causing errors
	.stRecv.nIgnored	0				ULINT		Number of ignored packets

关于数据流协议的诊断信息，可查看 Gvsp 模块下的**参数（在线）**选项卡。这些计数器纯粹是指传输的图像数据，且分为单个电报/数据包和完整图像/块。

如果一切正常，仅应计数 nSuccess 和 nCompletedBlocks 的值。

nResendRequests 已经表明了问题，但在这种情况下，它们仍然可以被回答，因为所有其他的计数器仍然为零。在超时的情况下，应在配置助手中GVSP 模块设置 [▶ 89]下增加 BlockTimeout。计算出 nBlockIdOverflows 是正常的，并不代表一个错误。计数器最终会因数值范围而数据过载，或在相机重新启动时被重置。

Object | Context | Parameter (Init) | Parameter (Online) | Interfaces | Interface Pointer

	Name	Online	CS	Type	PTCID	Comment
-	GvspProtocolDiagnosis	...	☐		0x00010001	Contains statistics for GVSP protocol diagnosis.
	.stRecv.nSuccess	390729		ULINT		Number of successfully received packets
	.stRecv.nError	0		ULINT		Number of received packets signalling or causing errors
	.stRecv.nIgnored	0		ULINT		Number of ignored packets
	.nResendRequests	53		ULINT		Number of packet resend requests
	.nCompletedBlocks	139		ULINT		Number of successfully received images
	.nAbortedBlocks	0		ULINT		Number of aborted images
	.nTimedOutBlocks	0		ULINT		Number of timed out images
	.nInvalidBlocks	0		ULINT		Number of invalid images
	.nFurtherErroneousBlocks	0		ULINT		Number of other erroneous images
	.nDuplicateBlocks	0		ULINT		Number of duplicate images
	.nMissingBlocks	0		ULINT		Number of missing images
	.nUnexpectedBlocks	0		ULINT		Number of unexpected images
	.nBlockIdOverflows	0		ULINT		Number of block ID overflows (not an error)

9.4 ADS 返回代码

错误代码分组：
全局错误代码：ADS 返回代码 [▶ 2753]... （0x9811_0000 ...）
路由器错误代码：ADS 返回代码 [▶ 2754]... （0x9811_0500 ...）
一般 ADS 错误：ADS 返回代码 [▶ 2754]... （0x9811_0700 ...）
RTime 错误代码：ADS 返回代码 [▶ 2756]... （0x9811_1000 ...）

全局错误代码

Hex	Dec	HRESULT	名称	描述
0x0	0	0x98110000	ERR_NOERROR	无错误。
0x1	1	0x98110001	ERR_INTERNAL	内部错误。
0x2	2	0x98110002	ERR_NORTIME	不具有实时性。
0x3	3	0x98110003	ERR_ALLOCLOCKEDMEM	分配已锁定 – 内存错误。
0x4	4	0x98110004	ERR_INSERTMAILBOX	邮箱已满 – 无法发送 ADS 消息。减少每个周期的 ADS 消息数量将有所帮助。
0x5	5	0x98110005	ERR_WRONGRECEIVEHMSG	HMSG 错误。
0x6	6	0x98110006	ERR_TARGETPORTNOTFOUND	未找到目标端口 – ADS 服务器未启动或无法访问。
0x7	7	0x98110007	ERR_TARGETMACHINENOTFOUND	未找到目标计算机 – 未找到 AMS 路由。
0x8	8	0x98110008	ERR_UNKNOWNCMDID	未知命令 ID。
0x9	9	0x98110009	ERR_BADTASKID	任务 ID 无效。
0xA	10	0x9811000A	ERR_NOIO	No IO。
0xB	11	0x9811000B	ERR_UNKNOWNAMSCMD	未知 AMS 命令。
0xC	12	0x9811000C	ERR_WIN32ERROR	Win32 错误。
0xD	13	0x9811000D	ERR_PORTNOTCONNECTED	端口未连接。
0xE	14	0x9811000E	ERR_INVALIDAMSLENGTH	AMS 长度无效。
0xF	15	0x9811000F	ERR_INVALIDAMSNETID	AMS Net ID 无效。
0x10	16	0x98110010	ERR_LOWINSTLEVEL	安装级别 – TwinCAT 2 授权错误。
0x11	17	0x98110011	ERR_NODEBUGINTAVAILABLE	无调试可用。
0x12	18	0x98110012	ERR_PORTDISABLED	端口已禁用 – TwinCAT 系统服务未启动。
0x13	19	0x98110013	ERR_PORTALREADYCONNECTED	端口已连接。
0x14	20	0x98110014	ERR_AMSSYNC_W32ERROR	AMS Sync Win32 错误。
0x15	21	0x98110015	ERR_AMSSYNC_TIMEOUT	AMS Sync 超时。
0x16	22	0x98110016	ERR_AMSSYNC_AMSERROR	AMS Sync 错误。
0x17	23	0x98110017	ERR_AMSSYNC_NOINDEXINMAP	不存在适用于 AMS Sync 的索引映射。
0x18	24	0x98110018	ERR_INVALIDAMSPORT	AMS 端口无效。
0x19	25	0x98110019	ERR_NOMEMORY	无内存。
0x1A	26	0x9811001A	ERR_TCPSEND	TCP 发送错误。
0x1B	27	0x9811001B	ERR_HOSTUNREACHABLE	主机无法访问。
0x1C	28	0x9811001C	ERR_INVALIDAMSFRAGMENT	AMS 片段无效。
0x1D	29	0x9811001D	ERR_TLSSEND	TLS 发送错误 – ADS 安全连接失败。
0x1E	30	0x9811001E	ERR_ACCESSDENIED	拒绝访问 – 拒绝 ADS 安全访问。

路由器错误代码

Hex	Dec	HRESULT	名称	描述
0x500	1280	0x98110500	ROUTERERR_NOLOCKEDMEMORY	无法分配锁定的内存。
0x501	1281	0x98110501	ROUTERERR_RESIZEMEMORY	路由器内存大小无法更改。
0x502	1282	0x98110502	ROUTERERR_MAILBOXFULL	邮箱已达到最大消息数。
0x503	1283	0x98110503	ROUTERERR_DEBUGBOXFULL	调试邮箱已达到最大消息数。
0x504	1284	0x98110504	ROUTERERR_UNKNOWNPORTTYPE	端口类型未知。
0x505	1285	0x98110505	ROUTERERR_NOTINITIALIZED	路由器未初始化。
0x506	1286	0x98110506	ROUTERERR_PORTALREADYINUSE	端口号已分配。
0x507	1287	0x98110507	ROUTERERR_NOTREGISTERED	端口未注册。
0x508	1288	0x98110508	ROUTERERR_NOMOREQUEUES	已达到最大端口数。
0x509	1289	0x98110509	ROUTERERR_INVALIDPORT	端口无效。
0x50A	1290	0x9811050A	ROUTERERR_NOTACTIVATED	路由未激活。
0x50B	1291	0x9811050B	ROUTERERR_FRAGMENTBOXFULL	邮箱已达到最大片段消息数。
0x50C	1292	0x9811050C	ROUTERERR_FRAGMENTTIMEOUT	发生片段超时。
0x50D	1293	0x9811050D	ROUTERERR_TOBEREMOVED	端口已移除。

一般性 ADS 错误代码

Hex	Dec	HRESULT	名称	描述
0x700	1792	0x98110700	ADSERR_DEVICE_ERROR	一般性设备错误。
0x701	1793	0x98110701	ADSERR_DEVICE_SRVNOTSUPP	服务器不支持该服务。
0x702	1794	0x98110702	ADSERR_DEVICE_INVALIDGRP	索引组无效。
0x703	1795	0x98110703	ADSERR_DEVICE_INVALIDOFFSET	索引偏移量无效。
0x704	1796	0x98110704	ADSERR_DEVICE_INVALIDACCESS	不允许读取或写入。
0x705	1797	0x98110705	ADSERR_DEVICE_INVALIDSIZE	参数大小不正确。
0x706	1798	0x98110706	ADSERR_DEVICE_INVALIDDATA	无效数据值。
0x707	1799	0x98110707	ADSERR_DEVICE_NOTREADY	设备尚未准备好运行。
0x708	1800	0x98110708	ADSERR_DEVICE_BUSY	设备正忙。
0x709	1801	0x98110709	ADSERR_DEVICE_INVALIDCONTEXT	操作系统上下文无效。这可能是由于在不同的任务中使用 ADS 函数块造成的。可以通过在 PLC 中进行多任务同步解决这个问题。
0x70A	1802	0x9811070A	ADSERR_DEVICE_NOMEMORY	内存不足。
0x70B	1803	0x9811070B	ADSERR_DEVICE_INVALIDPARM	参数值无效。
0x70C	1804	0x9811070C	ADSERR_DEVICE_NOTFOUND	未找到（文件…）。
0x70D	1805	0x9811070D	ADSERR_DEVICE_SYNTAX	文件或命令中存在语法错误。
0x70E	1806	0x9811070E	ADSERR_DEVICE_INCOMPATIBLE	对象不匹配。
0x70F	1807	0x9811070F	ADSERR_DEVICE_EXISTS	对象已存在。
0x710	1808	0x98110710	ADSERR_DEVICE_SYMBOLNOTFOUND	符号未找到。
0x711	1809	0x98110711	ADSERR_DEVICE_SYMBOLVERSIONINVALID	符号版本无效。这可能是由于联机更改造成的。创建新句柄。
0x712	1810	0x98110712	ADSERR_DEVICE_INVALIDSTATE	设备（服务器）处于无效状态。
0x713	1811	0x98110713	ADSERR_DEVICE_TRANSMODENOTSUPP	不支持 AdsTransMode。
0x714	1812	0x98110714	ADSERR_DEVICE_NOTIFYHNDINVALID	通知句柄无效。
0x715	1813	0x98110715	ADSERR_DEVICE_CLIENTUNKNOWN	通知客户端未注册。
0x716	1814	0x98110716	ADSERR_DEVICE_NOMOREHDLS	没有更多的句柄可用。
0x717	1815	0x98110717	ADSERR_DEVICE_INVALIDWATCHSIZE	通知大小过大。
0x718	1816	0x98110718	ADSERR_DEVICE_NOTINIT	设备未初始化。
0x719	1817	0x98110719	ADSERR_DEVICE_TIMEOUT	设备超时。
0x71A	1818	0x9811071A	ADSERR_DEVICE_NOINTERFACE	接口查询失败。
0x71B	1819	0x9811071B	ADSERR_DEVICE_INVALIDINTERFACE	请求的接口错误。
0x71C	1820	0x9811071C	ADSERR_DEVICE_INVALIDCLSID	Class ID 无效。
0x71D	1821	0x9811071D	ADSERR_DEVICE_INVALIDOBJID	Object ID 无效。
0x71E	1822	0x9811071E	ADSERR_DEVICE_PENDING	请求待定。
0x71F	1823	0x9811071F	ADSERR_DEVICE_ABORTED	请求已中止。
0x720	1824	0x98110720	ADSERR_DEVICE_WARNING	警告信号。
0x721	1825	0x98110721	ADSERR_DEVICE_INVALIDARRAYIDX	数组索引无效。
0x722	1826	0x98110722	ADSERR_DEVICE_SYMBOLNOTACTIVE	符号未激活。
0x723	1827	0x98110723	ADSERR_DEVICE_ACCESSDENIED	拒绝访问。
0x724	1828	0x98110724	ADSERR_DEVICE_LICENSENOTFOUND	缺少授权。
0x725	1829	0x98110725	ADSERR_DEVICE_LICENSEEXPIRED	授权已过期。
0x726	1830	0x98110726	ADSERR_DEVICE_LICENSEEXCEEDED	超出授权。
0x727	1831	0x98110727	ADSERR_DEVICE_LICENSEINVALID	授权无效。
0x728	1832	0x98110728	ADSERR_DEVICE_LICENSESYSTEMID	授权问题：系统 ID 无效。
0x729	1833	0x98110729	ADSERR_DEVICE_LICENSENOTIMELIMIT	授权不受时间限制。
0x72A	1834	0x9811072A	ADSERR_DEVICE_LICENSEFUTUREISSUE	授权问题：未来的时间。
0x72B	1835	0x9811072B	ADSERR_DEVICE_LICENSETIMETOLONG	授权期限太长。
0x72C	1836	0x9811072C	ADSERR_DEVICE_EXCEPTION	系统启动异常。
0x72D	1837	0x9811072D	ADSERR_DEVICE_LICENSEDUPLICATED	授权文件读取了两次。
0x72E	1838	0x9811072E	ADSERR_DEVICE_SIGNATUREINVALID	签名无效。
0x72F	1839	0x9811072F	ADSERR_DEVICE_CERTIFICATEINVALID	证书无效。
0x730	1840	0x98110730	ADSERR_DEVICE_LICENSEOEMNOTFOUND	OEM未知公钥。
0x731	1841	0x98110731	ADSERR_DEVICE_LICENSERESTRICTED	此系统 ID 授权无效。
0x732	1842	0x98110732	ADSERR_DEVICE_LICENSEDEMODENIED	演示授权已禁止。
0x733	1843	0x98110733	ADSERR_DEVICE_INVALIDFNCID	无效函数 ID。
0x734	1844	0x98110734	ADSERR_DEVICE_OUTOFRANGE	超出有效范围。
0x735	1845	0x98110735	ADSERR_DEVICE_INVALIDALIGNMENT	无效对齐。
0x736	1846	0x98110736	ADSERR_DEVICE_LICENSEPLATFORM	无效平台级别。
0x737	1847	0x98110737	ADSERR_DEVICE_FORWARD_PL	上下文 – 转到被动级别。

Hex	Dec	HRESULT	名称	描述
0x738	1848	0x98110738	ADSERR_DEVICE_FORWARD_DL	上下文 – 转到调度级别。
0x739	1849	0x98110739	ADSERR_DEVICE_FORWARD_RT	上下文 – 转到实时。
0x740	1856	0x98110740	ADSERR_CLIENT_ERROR	客户端错误。
0x741	1857	0x98110741	ADSERR_CLIENT_INVALIDPARM	服务包含无效参数。
0x742	1858	0x98110742	ADSERR_CLIENT_LISTEMPTY	轮询列表为空。
0x743	1859	0x98110743	ADSERR_CLIENT_VARUSED	Var 连接已在使用中。
0x744	1860	0x98110744	ADSERR_CLIENT_DUPLINVOKEID	调用的 ID 已在使用中。
0x745	1861	0x98110745	ADSERR_CLIENT_SYNCTIMEOUT	已发生超时 – 远程终端在指定的 ADS 超时中未响应。远程终端的路由设置可能配置不正确。
0x746	1862	0x98110746	ADSERR_CLIENT_W32ERROR	Win32 子系统中发生错误。
0x747	1863	0x98110747	ADSERR_CLIENT_TIMEOUTINVALID	客户端超时值无效。
0x748	1864	0x98110748	ADSERR_CLIENT_PORTNOTOPEN	端口未打开。
0x749	1865	0x98110749	ADSERR_CLIENT_NOAMSADDR	无 AMS 地址。
0x750	1872	0x98110750	ADSERR_CLIENT_SYNCINTERNAL	Ads sync 中发生内部错误。
0x751	1873	0x98110751	ADSERR_CLIENT_ADDHASH	哈希表溢出。
0x752	1874	0x98110752	ADSERR_CLIENT_REMOVEHASH	未在表格中找到密钥。
0x753	1875	0x98110753	ADSERR_CLIENT_NOMORESYM	缓存中没有符号。
0x754	1876	0x98110754	ADSERR_CLIENT_SYNCRESINVALID	收到无效响应。
0x755	1877	0x98110755	ADSERR_CLIENT_SYNCPORTLOCKED	同步端口已锁定。
0x756	1878	0x98110756	ADSERR_CLIENT_REQUESTCANCELLED	请求已取消。

RTime 错误代码

Hex	Dec	HRESULT	名称	描述
0x1000	4096	0x98111000	RTERR_INTERNAL	实时系统中发生内部错误。
0x1001	4097	0x98111001	RTERR_BADTIMERPERIODS	计时器值无效。
0x1002	4098	0x98111002	RTERR_INVALIDTASKPTR	任务指针提供了无效值 0（零）。
0x1003	4099	0x98111003	RTERR_INVALIDSTACKPTR	堆栈指针提供了无效值 0（零）。
0x1004	4100	0x98111004	RTERR_PRIOEXISTS	请求任务优先级已分配。
0x1005	4101	0x98111005	RTERR_NOMORETCB	没有可用的空闲 TCB（任务控制块）。TCB 的最大数量为 64。
0x1006	4102	0x98111006	RTERR_NOMORESEMAS	没有可用的空闲信号。信号的最大数量为 64。
0x1007	4103	0x98111007	RTERR_NOMOREQUEUES	队列中没有可用的空闲空间。队列中的最大位置数为 64。
0x100D	4109	0x9811100D	RTERR_EXTIRQALREADYDEF	已应用外部同步中断。
0x100E	4110	0x9811100E	RTERR_EXTIRQNOTDEF	未应用外部同步中断。
0x100F	4111	0x9811100F	RTERR_EXTIRQINSTALLFAILED	外部同步中断应用失败。
0x1010	4112	0x98111010	RTERR_IRQLNOTLESSOREQUAL	在错误的上下文中调用服务函数
0x1017	4119	0x98111017	RTERR_VMXNOTSUPPORTED	不支持 Intel VT-x 扩展。
0x1018	4120	0x98111018	RTERR_VMXDISABLED	BIOS 中未启用 Intel VT-x 扩展。
0x1019	4121	0x98111019	RTERR_VMXCONTROLSMISSING	Intel VT-x 扩展中缺少函数。
0x101A	4122	0x9811101A	RTERR_VMXENABLEFAILS	Intel VT-x 激活失败。

具体的正向 HRESULT 返回代码：

HRESULT	名称	描述
0x0000_0000	S_OK	无错误。
0x0000_0001	S_FALSE	无错误。 示例：处理成功，但有一个负面或不完整的结果。
0x0000_0203	S_PENDING	无错误。 示例：处理成功，但还没有结果。
0x0000_0256	S_WATCHDOG_TIMEOUT	无错误。 示例：处理成功，但发生了超时。

TCP Winsock 错误代码

Hex	Dec	名称	描述
0x274C	10060	WSAETIMEDOUT	发生连接超时 - 在创建连接时发生错误，因为远程终端在特定时间后未正确响应，或者所建立连接因所连接主机未响应而无法维持。
0x274D	10061	WSAECONNREFUSED	连接遭到拒绝 - 无法建立连接，因为目标计算机明确拒绝了该连接。此错误通常由尝试连接到外部主机上处于非活动状态的服务（即没有运行服务器应用程序的服务）导致。
0x2751	10065	WSAEHOSTUNREACH	不存在至主机的路由 - 套接字操作引用了不可用的主机。
更多 Winsock 错误代码：Win32 错误代码			

9.5 技术支持和服务

倍福公司及其合作伙伴在世界各地提供全面的技术支持和服务，对与倍福产品和系统解决方案相关的所有问题提供快速有效的帮助。

下载搜索器

我们的下载搜索器包含我们供您下载的所有文件。您可以通过它搜索我们的应用案例、技术文档、技术图纸、配置文件等等。

可供下载的文件格式多种多样。

倍福分公司和代表处

若需要倍福产品的本地支持和服务，请联系倍福分公司或代表处！

倍福遍布世界各地的分公司和代表处地址可在倍福官网上找到：http://www.beckhoff.com.cn

该网页还提供更多倍福产品组件的文档。

倍福技术支持

技术支持部门为您提供全面的技术援助，不仅帮助您应用各种倍福产品，还提供其他广泛的服务：

- 技术支持
- 复杂自动化系统的设计、编程和调试
- 以及倍福系统组件的各种培训课程

热线电话： +49 5246 963-157
电子邮箱： support@beckhoff.com

倍福售后服务

倍福服务中心提供所有售后服务：

- 现场服务
- 维修服务
- 备件服务
- 热线服务

热线电话： +49 5246 963-460
电子邮箱： service@beckhoff.com

倍福公司总部

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

电话： +49 5246 963-0
电子邮箱： info@beckhoff.com
网址： www.beckhoff.com

9.6 第三方组件

本软件包含第三方组件。有关详细信息，请参见以下文件夹中提供的许可文件：

TwinCAT 3.1 4024 和 TwinCAT 3.1 4026 中的不同内存路径

Build 4024 及以下版本：

<TwinCatInstallPath>\3.1\System\Legal\TF7xxx Additional license information.txt

Build 4026 及以上版本：

- *C:/Program Files (x86)/Beckhoff\Legal\TwinCAT-XAE-GigEVision*
- *C:/Program Files (x86)/Beckhoff\Legal\TwinCAT-XAE-TMX-Vision*
- *C:/Program Files (x86)/Beckhoff\Legal\TwinCAT-XAE-Vision*
- *C:/Program Files (x86)/Beckhoff\Legal\TwinCAT-XAR-VisionService*

更多信息：
www.beckhoff.com/twincat-vision/

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
电话号码：+49 5246 9630
info@beckhoff.com
www.beckhoff.com