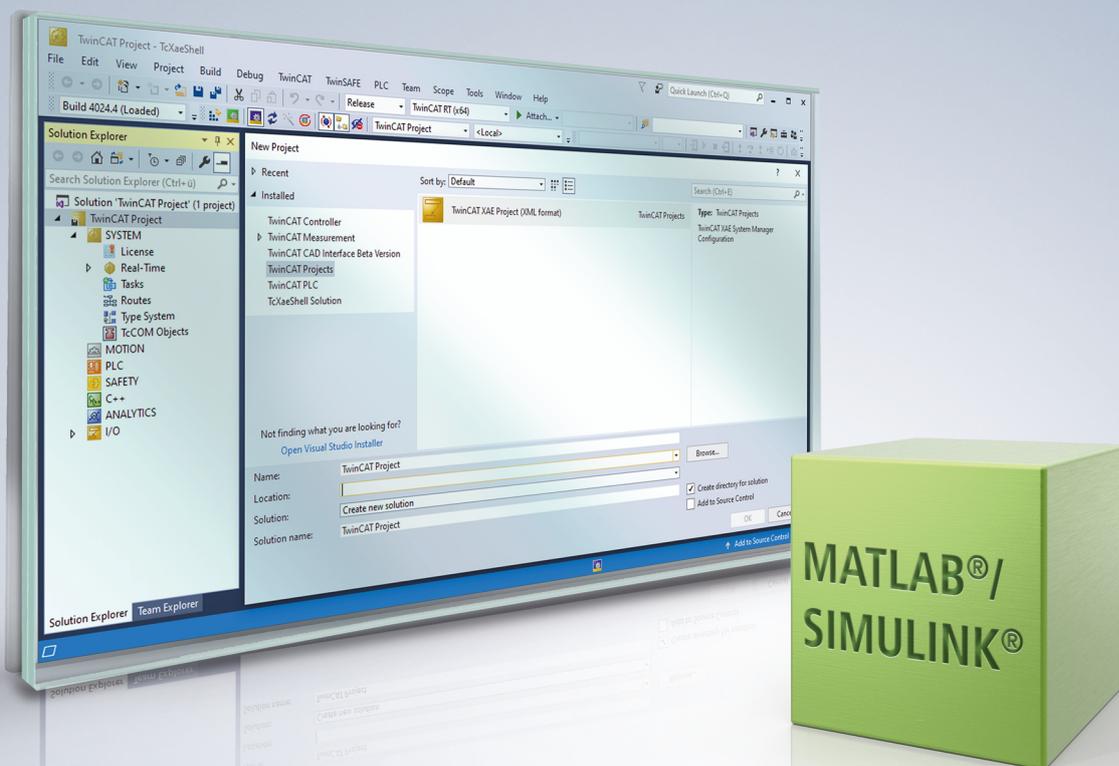


BECKHOFF New Automation Technology

手册 | ZH

TwinCAT 3

MATLAB®/Simulink®



目录

1 前言	5
1.1 文档说明	5
1.2 安全信息	6
1.3 信息安全说明	6
1.4 文档发行状态	8
2 概述	9
3 框图	11
3.1 Simulink®-TcCOM	12
3.1.1 使用框图	12
3.1.2 显示信号曲线	12
3.1.3 框图中的模块参数设置	14
3.1.4 调试	15
3.2 MATLAB®-TcCOM	20
3.2.1 框图窗口的操作	21
3.2.2 信号曲线的显示	22
3.3 集成框图控件	24
4 TwinCAT 自动化接口：在 MATLAB® 中使用	26
4.1 示例：Tc3AutomationInterface	26

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®

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

© 德国倍福自动化有限公司。
未经明确授权，不得复制、分发、使用和传播本档内容。
违者将被追究赔偿责任。德国倍福自动化有限公司保留所有发明、实用新型和外观设计专利权。

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。

1.4 文档发行状态

版本	变更
1.4.x	新功能： <ul style="list-style-type: none">• 查看框图的版本号。• 关于安装 TwinCAT 3.1 Build 4026 [▶_11] 的简要信息

2 概述

MATLAB® 和 Simulink®

MATLAB® 和 Simulink® 已成为全球多种应用领域的主流开发环境，适用于各种应用，甚至在初露头角的工程师群体中也得到广泛采用。此发展趋势的形成有多重原因。MATLAB® 和 Simulink® 提供的开发环境能让您专注于工程技术任务本身。这不仅十分契合教学环境中的授业理念，也能高效满足工业应用需求。MATLAB® 以其大量工具箱，为开发算法和分析数据提供了一个理想的软件工程环境。MATLAB® 提供多种功能，以便轻松接入不同数据格式，并能与 TwinCAT 的各种数据记录机制无缝协同。Simulink® 可为基于模型的设计（MBD）提供全面支持。这包括基于系统模型的开发、测试和验证。其后，可为如 TwinCAT 这样的平台自动生成代码，这种方式为在生产中应用经过测试的代码提供了最优解决方案。Simulink® 提供多物理场仿真建模和生成用于控制、调节及人工智能领域的算法所需的所有资源。因此，只有在模型上经过测试的高质量代码才会被用在控制器上。

TwinCAT 3 的技术和产品

TE1400 TwinCAT 3 Target for Simulink®

使用 TwinCAT 3 Target for Simulink® 可以将 Simulink® 中开发的模型应用于 TwinCAT 3 中。在 Simulink® 中可集成 SimScape™、Stateflow™ 或 DSP System Toolbox™ 等工具箱。它还支持嵌入 MATLAB® 功能块。这些模型通过 Simulink Coder™ 自动编译为 C/C++ 代码，并可通过 TwinCAT 3 Target for Simulink® 转换为可实时运行的 TwinCAT 对象。从 Simulink® 中创建的 TwinCAT 对象与所有其他 TwinCAT 对象一样具有相同的接口和属性。它们完全可以在 TwinCAT 3 开发环境中使用，例如，在一个包含 PLC 源代码的工程项目上扩展、调试以及与现场总线设备关联等，具体请参见文档 [TE1400 TwinCAT 3 Target for Simulink®](#)。

TE1401 TwinCAT 3 Target for MATLAB®

TwinCAT 3 Target for MATLAB® 让您可以在 TwinCAT 3 中使用 MATLAB® 函数。这些函数会自动转换成 TwinCAT 对象，并在 TwinCAT 3 开发环境中无缝应用。自动生成的模块可以作为 TcCOM 对象集成到 TwinCAT 解决方案中也可以专门创建成 PLC 库以 PLC 功能块的方式使用。嵌入的模块与整个 TwinCAT 项目一起被下载到 TwinCAT 3 Runtime 中，并像所有其他对象一样在实时环境中执行，具体请参见文档 [TE1401 TwinCAT 3 Target for MATLAB®](#)。

TE1410 TwinCAT 3 Interface for MATLAB®/Simulink®

TwinCAT 3 Interface for MATLAB®/Simulink® 可在 TwinCAT Runtime 和 MATLAB® 或 Simulink® 之间建立高性能双向通信。在系统开发阶段，您甚至可以借助 Simulink® 实现软件在环仿真。例如，在设备运行期间，您可以通过本产品实现以下功能：

- 基于 MATLAB® 应用程序的人机界面（HMI）
- 基于 MATLAB® 编译器运行时的计算服务器，用于预测性维护或参数优化等功能

[TwinCAT 3 Interface for MATLAB®/Simulink® 的文档](#)

用于 MATLAB® 和 Simulink® 的 TwinCAT 基础技术

下文所述工具无授权费用，均包含在 TwinCAT 3 XAE 或完整设置中。

TwinCAT 3 框图

框图查看器可通过在 TwinCAT 开发环境系统中显示的框图对 TcCOM 模块进行显示、调试和参数设置。此选项卡仅适用于使用 Target for MATLAB® 或 Target for Simulink® 生成的 TcCOM 模块。框图导出功能对这两个目标封装工具而言均为可选项。

[框图 \[▶ 11\]](#)

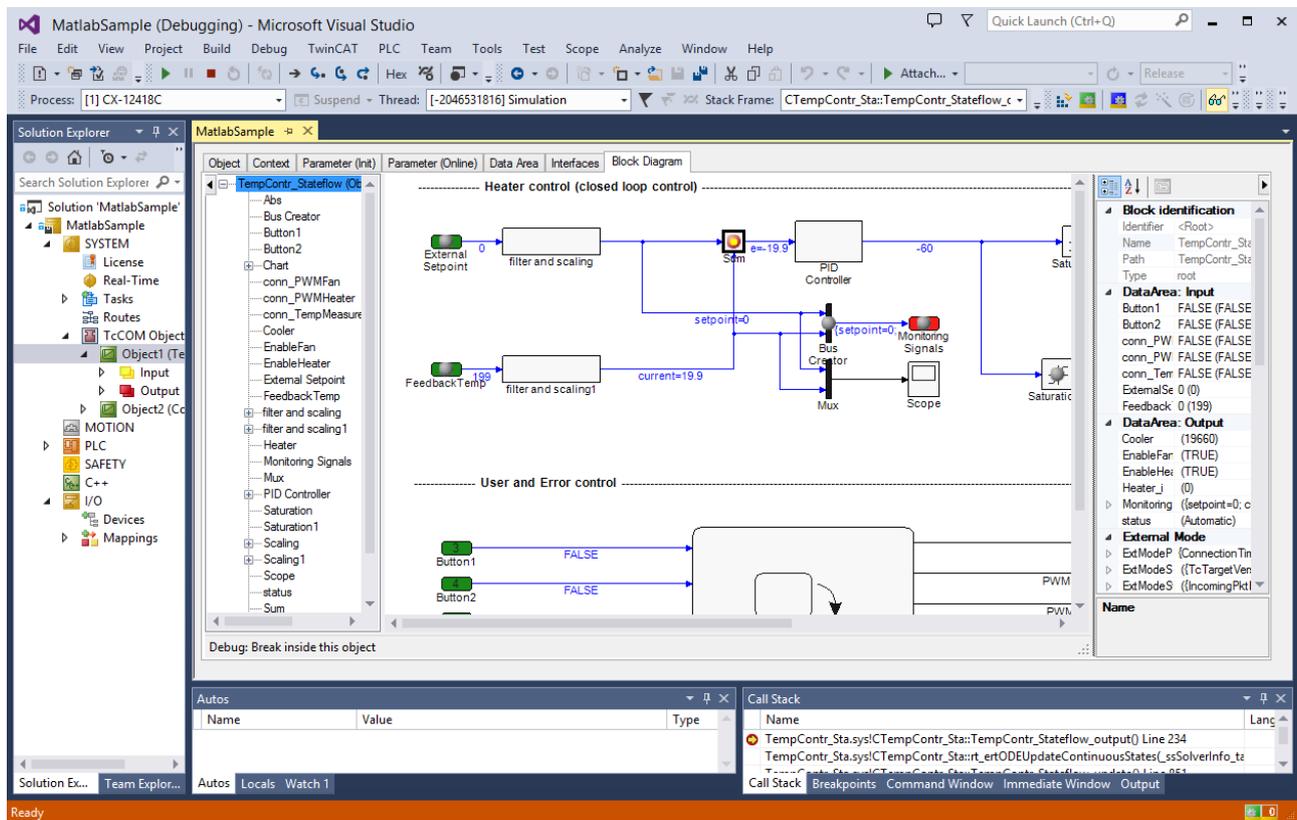
TwinCAT 3 自动化接口

TwinCAT 3 自动化接口是 TwinCAT 项目的编程接口，即您可以在系统管理器中创建配置，创建 PLC 并插入代码，以及实例化 TcCOM 对象，而无需通过 Visual Studio 或 TwinCAT XAE 打开 TwinCAT。此外，该 API 也适用于 MATLAB®。

TwinCAT 自动化接口：在 MATLAB® 中使用 [▶ 26]

3 框图

从 MATLAB® 或 Simulink® 生成 TwinCAT 对象时，也可以选择导出框图（Target for Simulink®）或 MATLAB® 代码（Target for MATLAB®）。在这种情况下，框图或 MATLAB® 代码可以在 TwinCAT 开发环境中的 TcCOM 实例的 **Block Diagram**（框图）选项卡下显示：



使用此控件，您可以浏览框图的整个结构或查看底层 MATLAB® 代码，还可以查看和更改参数值、显示信号值和曲线，以及在调试模式下使用断点通过模块进行调试。该控件被设计成可在独立的视图中使用。

在该控件中点击右键并选择 **About TC3 BlockDiagram**，即可查看框图的版本号。或者，您也可以 Windows **Control Panel**（控制面板）的 **Programs and Features**（程序和功能）下查看版本号（条目：Beckhoff TwinCAT 3 BlockDiagram）。

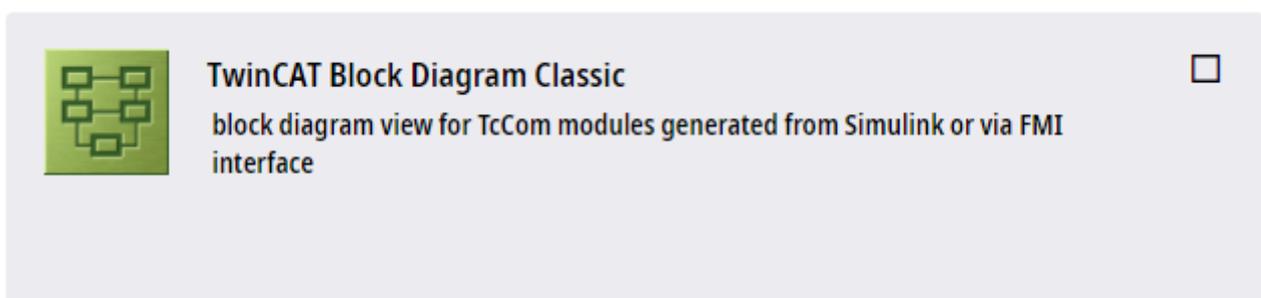
安装

TwinCAT 3.1 Build 4024

TwinCAT Block Diagram 安装程序在 TwinCAT 3 XAE 或 FULL 安装包安装过程中被一并安装，无需单独安装。通过 TE14xx TwinCAT Tools for MATLAB® 安装包也可以找到 TwinCAT Block Diagram 安装程序。

TwinCAT 3.1 Build 4026

通过 TwinCAT Package Manager 安装以下工作负载：



命令行: `tcpkg install TwinCAT.XAE.BlockDiagramClassic`

3.1 Simulink®-TcCOM

如果使用 TwinCAT Target for Simulink® 创建了 TwinCAT object，并在过程中导出了框图，则 Simulink® 模型的框图可以作为控件显示在 TwinCAT XAE 中。

3.1.1 使用框图

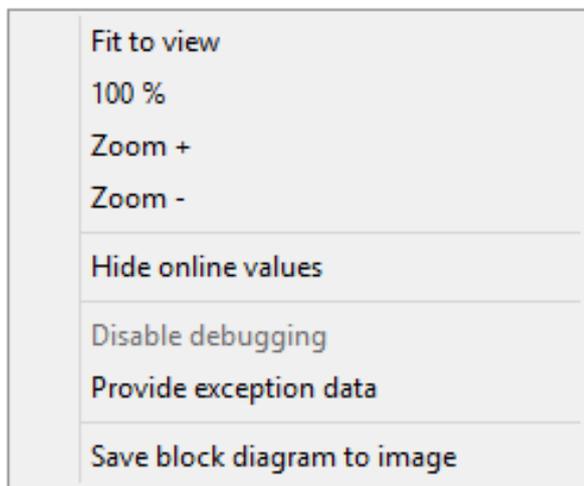
在从 MATLAB® 或 Simulink® 生成 TcCOM 模块时，可对框图导出进行配置。如果已启用导出功能，则可在 TwinCAT 开发环境中模块实例的“框图”（Block Diagram）选项卡下找到框图。

通过快捷键、拖放功能和上下文菜单，可以浏览 TcCOM 模块的层次结构，查看参数值，显示信号值并获取可选的附加调试信息。

快捷键功能：

快捷键	功能
Space	缩放至框图选项卡的当前大小
Backspace	切换到下一个更高的层级
ESC	切换到下一个更高的层级
CTRL + “+”	放大
CTRL + “-”	缩小
F5	连接调试器 (必须激活系统 -> 实时 -> C++ 调试器 -> 启用 C++ 调试器 (System-> Real-Time -> C++ Debugger -> Enable C++ Debugger))

上下文菜单功能：

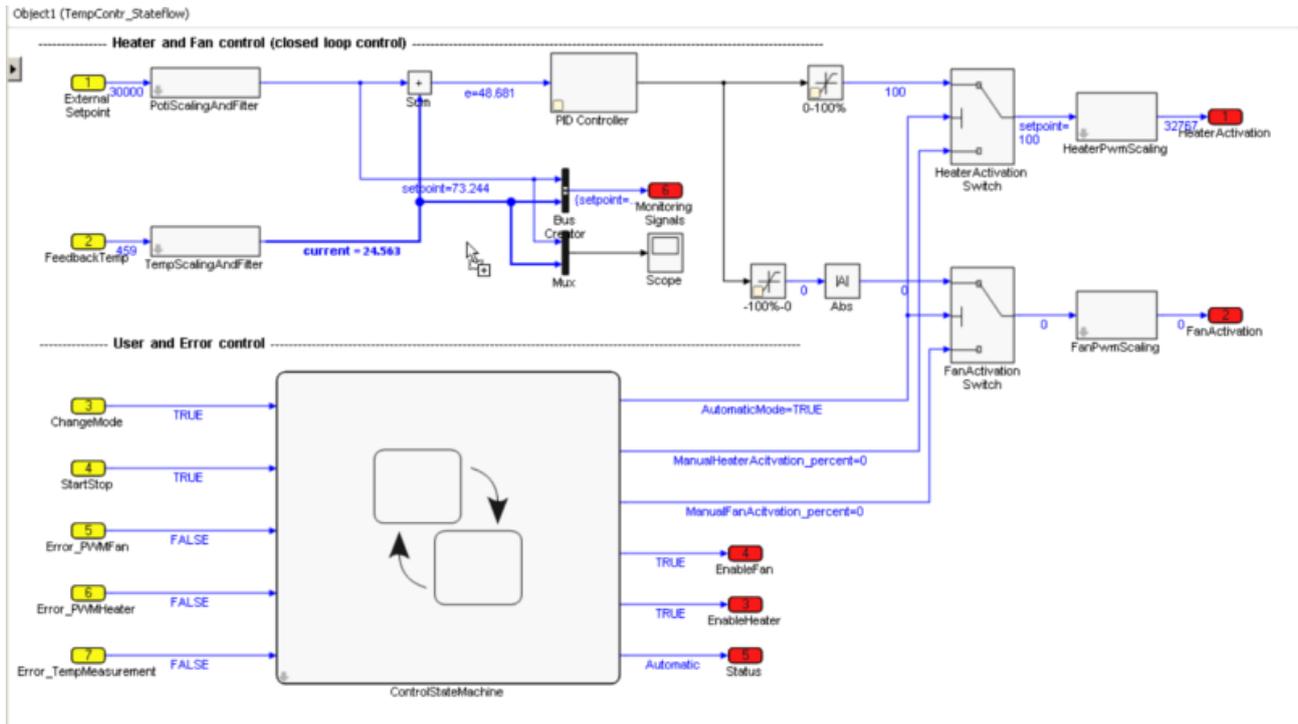


3.1.2 显示信号曲线

显示信号曲线通常有助于验证和故障排除。框图提供以下选项：

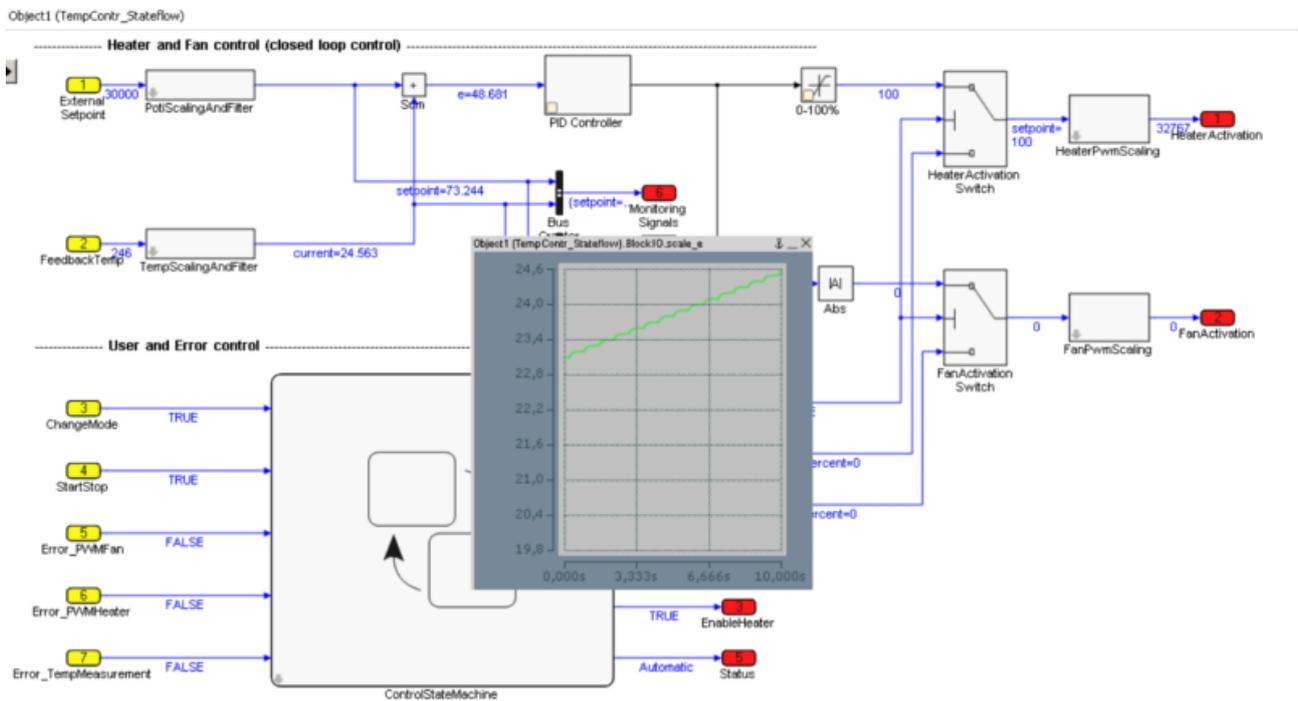
在框图中显示信号曲线

框图中提供在窗口中显示信号曲线的选项。为此，可将信号或图块拖放到框图的空闲区域。



在框图中创建作用域

下拉后，框图中将打开一个作用域窗口。



在框图中显示作用域

作用域窗口的标题栏提供以下选项：

✕	关闭窗口
⚓	将窗口保持在所有框图层次结构的最前面
—	窗口最小化到标题栏



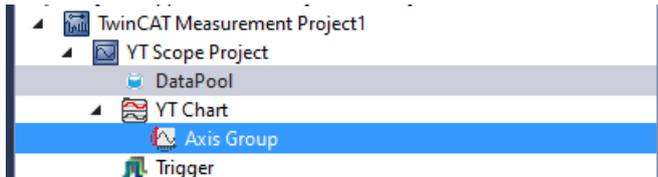
在框图控件 [▶ 24] 中显示作用域需要 Scope View Professional (TE1300) 许可证。TwinCAT XAE 中不需要 Scope View Professional 许可证。

在 Simulink® 总线框图中创建作用域窗口时，总线的所有信号都会直接显示在作用域窗口中。

框图中的作用域窗口可用于快速概览。如要进行更详细的分析，建议在 TwinCAT Measurement 项目中分析信号。

在 TwinCAT 3 Scope 中显示信号曲线

如果不是在框图控件中，而是在 TwinCAT Measurement 项目的轴组中下拉，则信号会添加到该轴组中。



在 TwinCAT 3 Scope 中添加信号

3.1.3 框图中的模块参数设置

如要对 TcCOM 实例进行参数设置，可直接在框图中使用**参数窗口**。此外，还可以使用“属性表”，该表可在框图右侧边展开或折叠。不同的参数值之间存在基本区别：

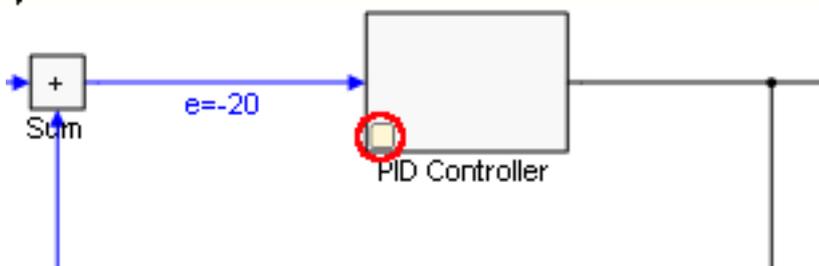
“默认”、“启动”、“在线”和“预备”

在框图属性表的下拉菜单中可以找到以下类型的值：

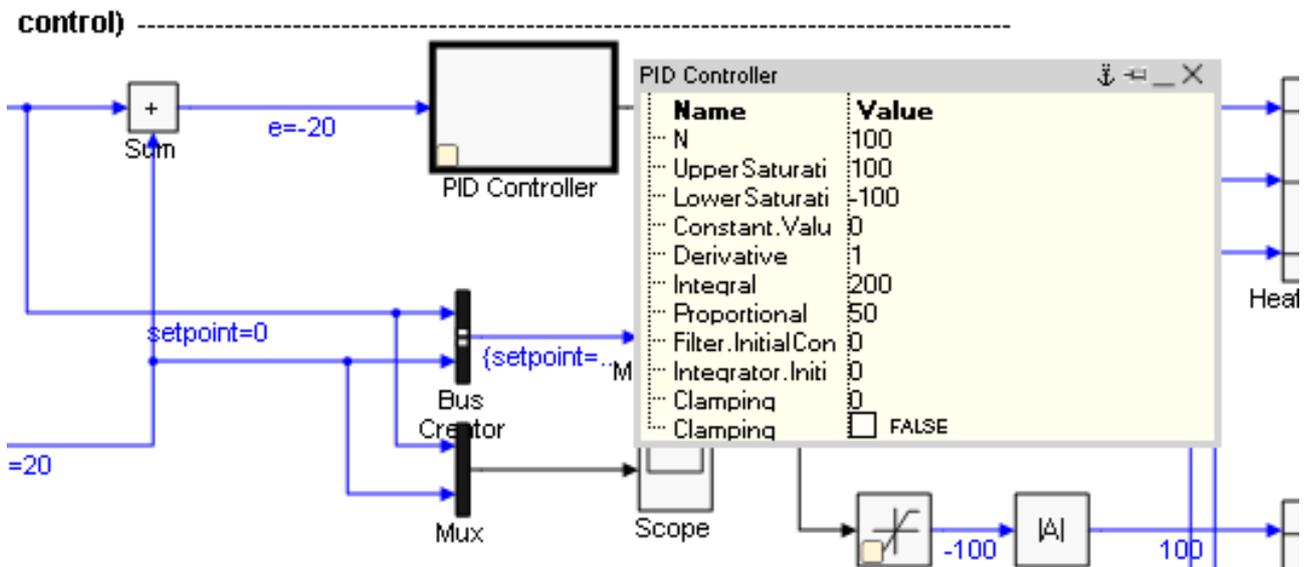
- **默认值**是代码生成过程中的参数值。它们总是存储在模块描述文件中，能够在参数更改后恢复生产设置。
- TwinCAT 启动模块实例后，**启动值**储存在 TwinCAT 项目文件中并下载到模块实例中。输入过程映像的启动值也可以在 Simulink® 模块中指定。这样，模块就可以在输入值不为零的情况下启动，无需将输入值链接至其他过程映像。内部信号和输出信号没有起始值，因为它们始终会在第一个周期内被覆盖。
- 只有当模块在目标系统上启动时，才能使用**在线值**。它们显示运行模块中的当前参数值。还可以在运行时期间更改该值。不过在这种情况下，必须通过上下文菜单启用相应的输入字段，以防止意外输入。
- 只要可以使用在线值，就可以指定**预备值**。它们可用于保存各种值，从而将其统一写入模块。如果已指定预备值，这些值将显示在框图下方的表格中。使用列表右侧的按钮可将预备值下载为在线值并/或保存为起始值，或将其删除。

框图中的参数设置

可参数化的图块在框图中用黄色方框标出。



双击图块或单击黄色方框，会弹出一个窗口，其中显示可更改的参数。



如果更改了某个值，可以通过以下键盘命令来应用：

CTRL + Enter	直接设置在线值
SHIFT + Enter	设置启动值
Enter	设置预备值

标题栏中的图标具有以下功能：

	关闭窗口
	将窗口保持在所有框图层级的最前面
	在当前框图层级保持窗口打开状态
	窗口最小化到标题栏

3.1.4 调试

有多种方法可用于查找使用 MATLAB®/Simulink® 创建的 TcCOM 模块中的错误，或分析 TwinCAT 项目整体架构中的模块行为。

框图中的调试

如果在生成 TcCOM 模块时导出了框图，则框图可以显示在 TwinCAT 开发环境中，并用于在相应模块实例中进行调试。为此，框图使用 Microsoft Visual Studio 调试器，该调试器可通过 TwinCAT 调试器端口与 TwinCAT Runtime 连接。按照“调试” () 下的 C++ 部分所述连接调试器。

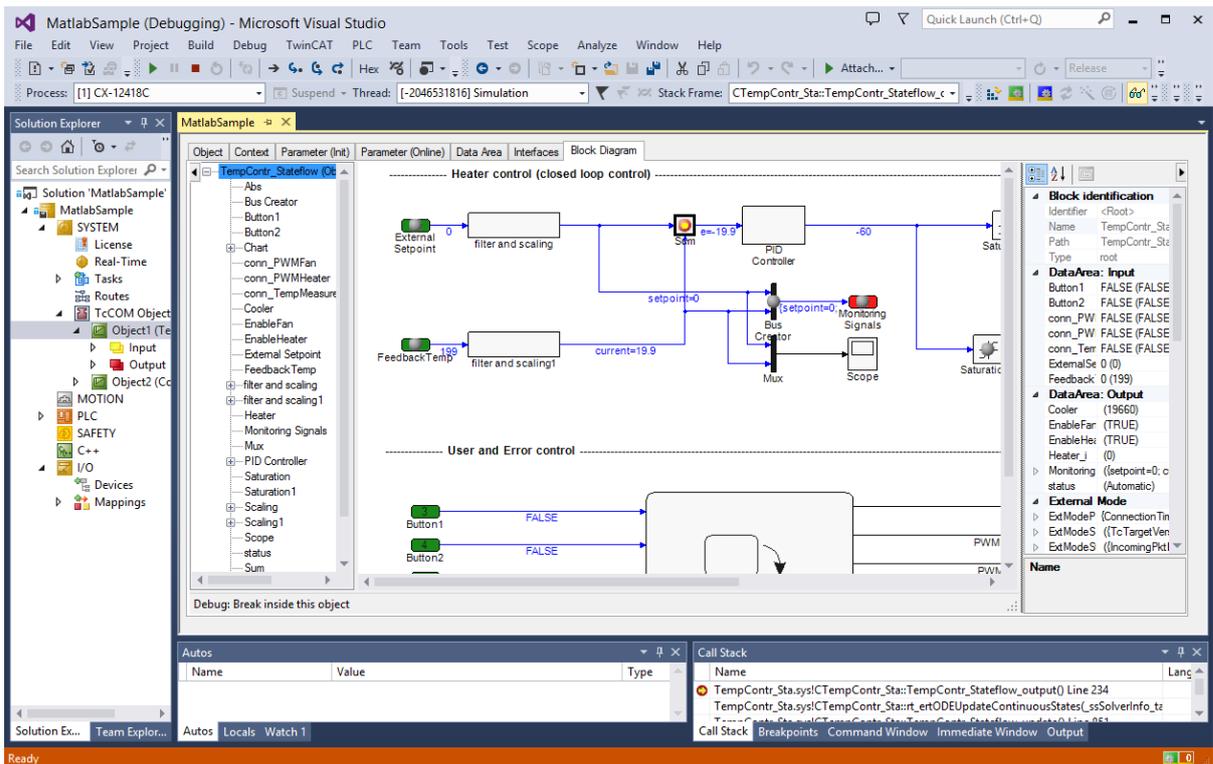
在框图内进行调试的前提条件：

- 工程系统中必须有 TcCOM 模块的 C/C++ 源代码，且 Visual Studio 调试器必须能够找到它。理想情况下，应在生成代码的系统上进行调试。如果模块是在其它系统上创建的，通常可以通过将 Visual Studio 项目集成到 TwinCAT C++ 部分来获取相关的 C/C++ 源代码。文件 <ModelName>.vcxproj 位于构建目录中，请参见
- 该模块必须是使用**调试** (Debug) 配置创建的。在代码生成后直接发布时，请在**发布配置** (publish configuration) 下的区域选择**调试** (Debug) 设置。从 TwinCAT 的 C++ 部分发布模块时，必须启用解决方案 C++ 节点中的调试器；参见 C/C++ 文档中的“调试”章节。

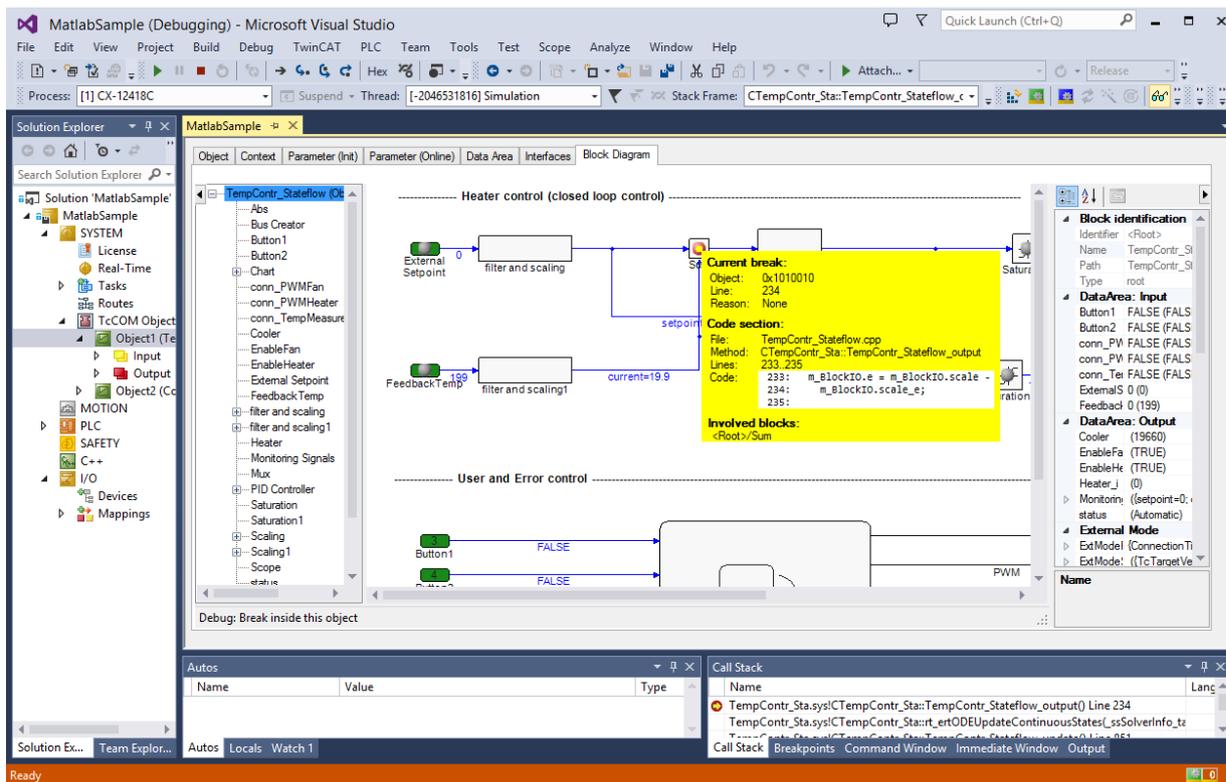
- 在代码生成过程中，必须在 **Tc 高级**（Tc Advanced）下的编码器设置中启用**导出框图**（Export block diagram）和**导出框图调试信息**（Export block diagram debug information）选项。
- 如中所述，在 TwinCAT 项目中，必须启用调试器端口。

在框图中设置断点

1. 将调试器连接到 TwinCAT Runtime 后，可能出现的断点将分配给框图中的图块，并以点的形式表示。单击所需的断点可将其激活，以便在下次执行相关功能块时使模块实例停止执行。点的颜色提供有关断点当前状态的信息：
 - 灰色：断点未启用
 - 红色：断点已启用。下一次执行该功能块时，程序将停止运行
 - 中间黄点：断点触发。程序在此时停止执行
 - 中间蓝点：断点触发（与黄色相同），但是在模块的不同实例中。



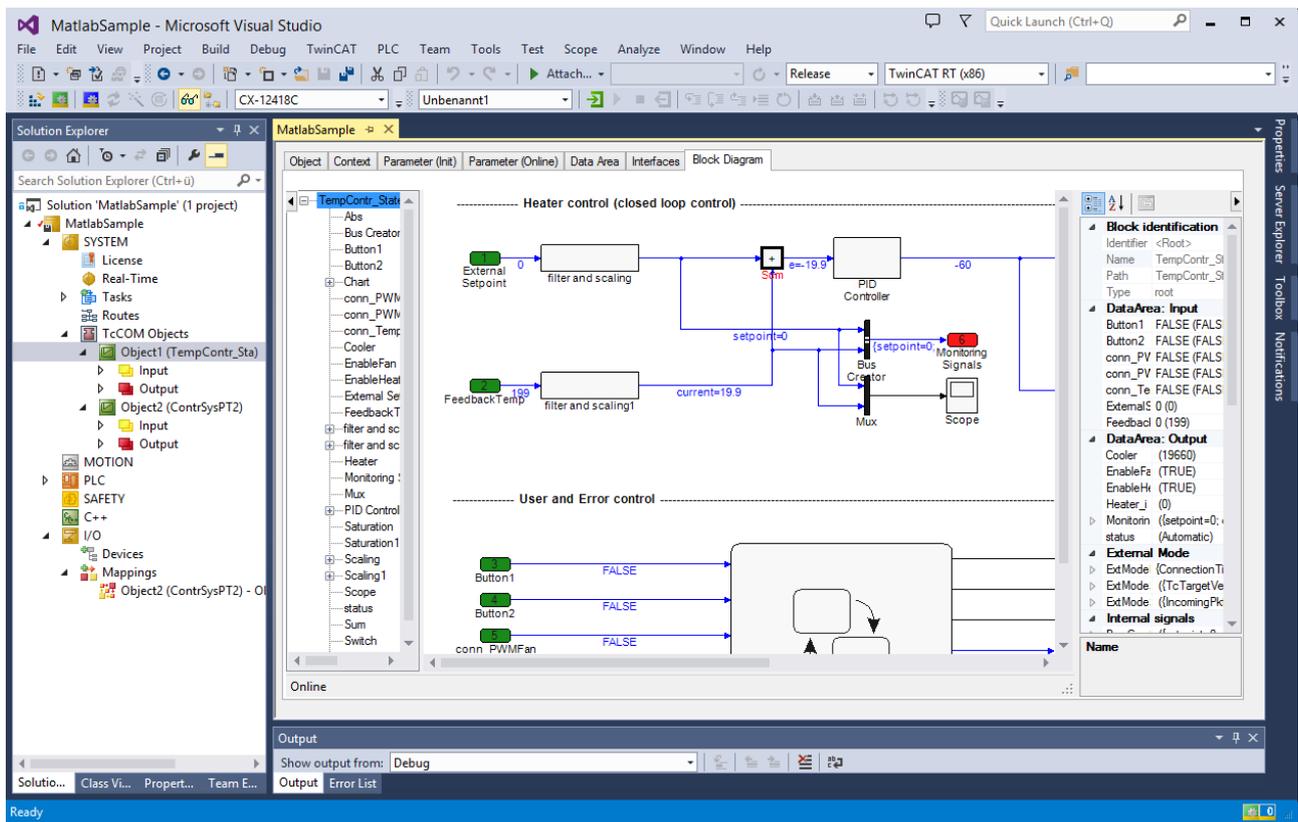
2. 其它信息（例如，相应的 C++ 代码段）可在断点的工具提示中找到：



断点并非总是分配给单个功能块。在许多情况下，多个功能块的功能会被合并在一个代码段，甚至是底层 C++ 代码的一行中。这意味着多个功能块可以共享同一个断点。因此，激活框图中的断点也可能导致其它框图中的点显示发生变化。

评估例外情况

如果在 TcCOM 模块的处理过程中出现异常，例如除以零，则可在框图中显示发生异常的点。为此，TcCOM 模块必须满足上述要求，并且必须在 TwinCAT 项目中启用 C++ 调试器 ()。连接调试器后（可能在异常发生前，也可能在异常发生后），只要能将导致异常的代码行分配给一个功能块，框图中就会突出显示导致异常的功能块。功能块名称以红色显示，功能块本身以粗体标出。



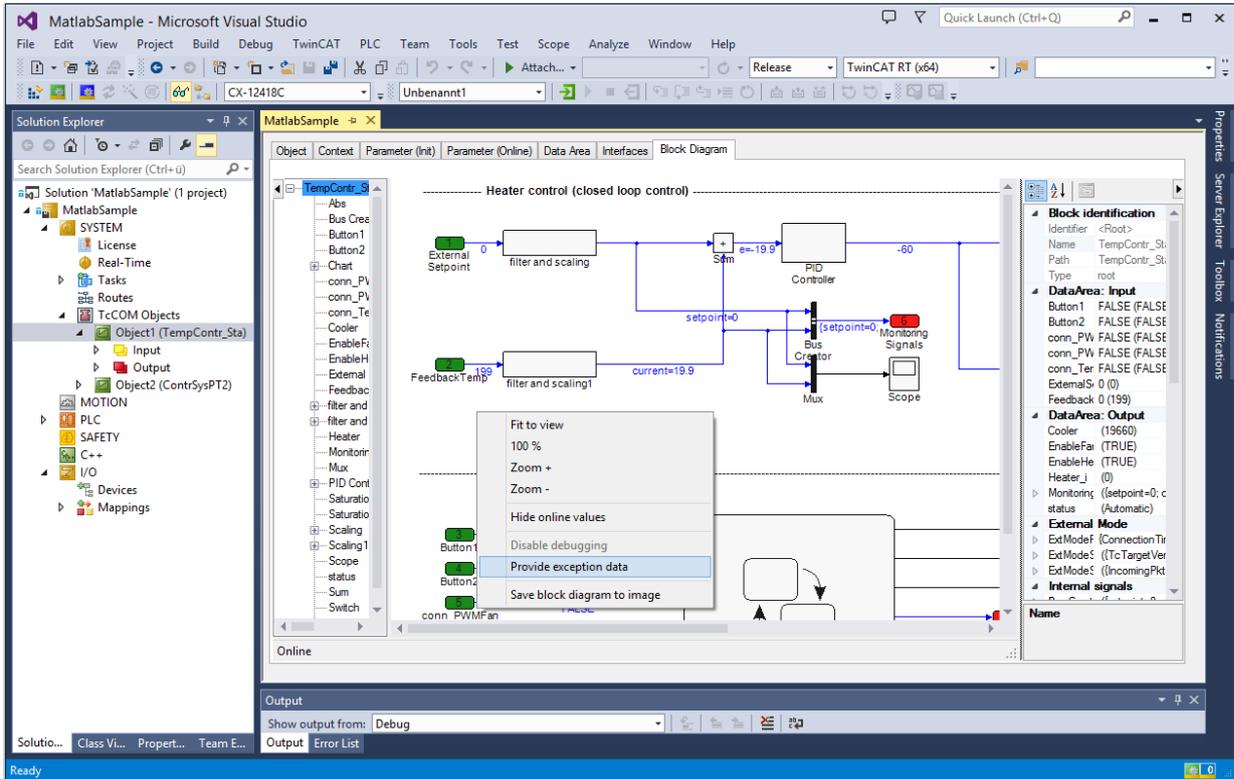
无需源代码即可手动评估异常

即使工程系统中没有模块源代码或 C++ 调试器未激活，一旦出现异常，还是可以在框图中突出显示出错位置。

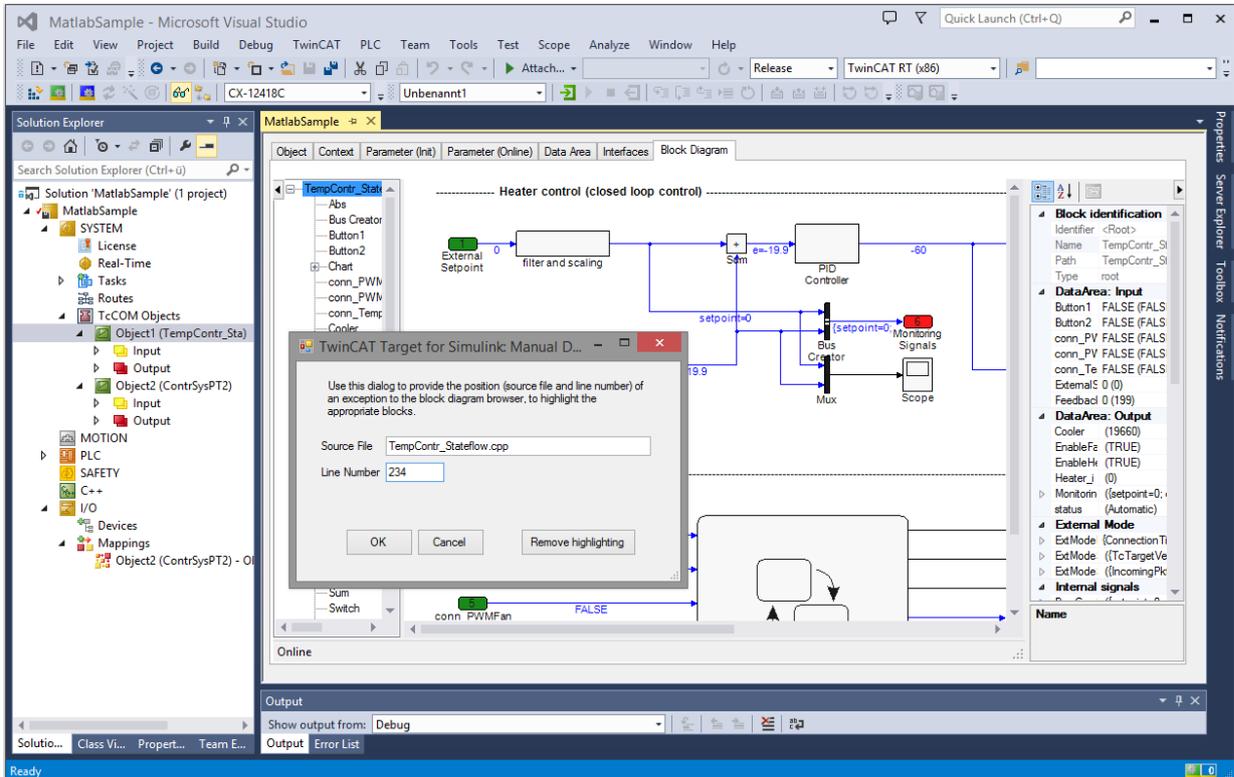
通常，发生错误时总会生成一条错误信息，指示源文件和源代码中的哪一行。在许多情况下，该信息可用于将异常分配给框图中的某个功能块。具体操作如下：

- ✓ 高亮显示框图中出错位置的前提条件是已生成调试信息（**Tc 高级**（Tc Advanced）下编码器设置中的选项 **导出框图调试信息**（Export block diagram debug information））。

3. 从框图的上下文菜单中选择条目提供异常数据 (Provide exception data) :



4. 在打开的对话框中, 输入错误信息中提供的源代码文件和行编号:




```

TwinCAT Project50
Object Context Parameter (Init) Parameter (Online) Data Area Interfaces Block Diagram
Object1 (BaseStatisticIterative) > BaseStatisticIterative
% iterative implementation of mean and standard deviation
% persistent variable -> FB with internal state in PLC

function [ mean_x [ 0 ], std_x [ 0 ] ] = BaseStatisticIterative( x [ 0 ] )

persistent n [ 2385 ] Mn [ 0 ] Sn

% init of persistent vars
if isempty( n [ 2384 ] )
    n [ 2383 ] = 0;
    Mn [ 0 ] = 0;
    Sn [ 0 ] = 0;
end

% update standard deviation
if n [ 2382 ] > 0
    Sn [ 0 ] = Sn [ 0 ] + ( n [ 2380 ] * x [ 0 ] - Mn [ 0 ] )^2 / ( n [ 2379 ]^2 + n [ 2378 ] );
end

% update mean
Mn = Mn [ 0 ] + x [ 0 ] ;

% update population count
n = n [ 2377 ] + 1;

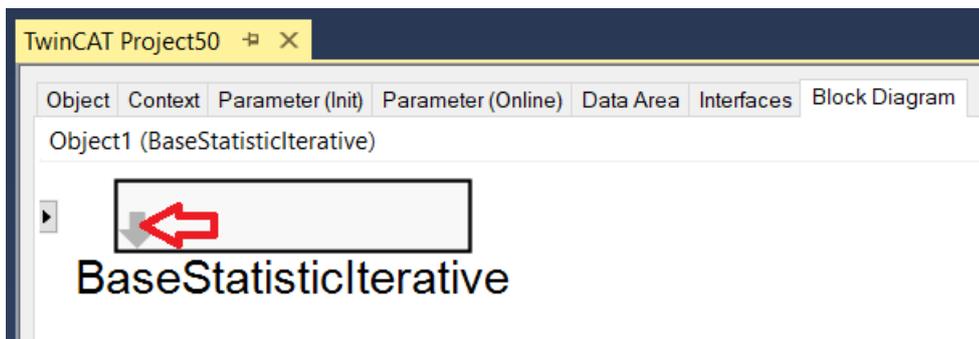
% output scaled values
if n [ 2376 ] > 1
    std_x [ 0 ] = sqrt( Sn [ 0 ] / ( n [ 2375 ] - 1 ) );
else
    std_x [ 0 ] = 0;
end
mean_x = Mn [ 0 ] / n [ 2381 ] ;

Online
    
```

3.2.1 框图窗口的操作

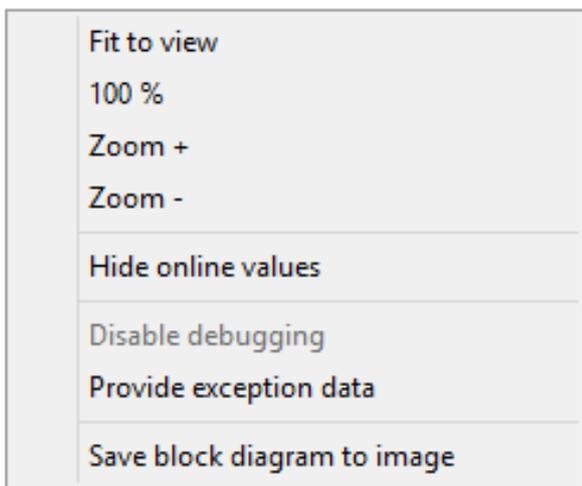
在从 MATLAB® 生成 TcCOM 模块时，可对 MATLAB® 代码导出进行配置。如果已启用导出功能，则可在 TwinCAT 开发环境中的模块实例的 **Block Diagram**（框图）选项卡下找到代码。

在顶层，您将看到以块表示的已创建模块。选择块中的灰色箭头，以显示内容。



快捷键功能:

快捷键	功能
Space	缩放至框图选项卡的当前大小
Backspace	切换到下一个更高的层级
ESC	切换到下一个更高的层级
CTRL + “+”	放大
CTRL + “-”	缩小
F5	连接调试器 (必须激活系统 -> 实时 -> C++ 调试器 -> 启用 C++ 调试器 (System -> Real-Time -> C++ Debugger -> Enable C++ Debugger))

上下文菜单功能:

3.2.2 信号曲线的显示

通过 ADS 可以在 TwinCAT XAE 中检索所选的变量。因此，既可在框图内的微型示波器中显示它们，也可通过测量项目中的 TwinCAT Scope 显示它们。

可在示波器窗口中显示的变量在代码显示时会有一个拖尾的黑框。操作期间在该框中的值显示为蓝色。

将一个“蓝色变量”拖放到框图窗口中，即可打开 Mini.Scope。

Object Context Parameter (Init) Parameter (Online) Data Area Interfaces Block Diagram

Object1 (BaseStatisticIterative) > BaseStatisticIterative

```

% iterative implementation of mean and standard deviation
% persistent variable -> FB with internal state in PLC

function [ mean_x [ ... ], std_x [ 1.977437994 ] ] = BaseStatisticIterative( x [ 4 ] )

persistent n [ 8765 ] Mn [ 14824 ] Sn

% init of persistent vars
if isempty( n [ 8764 ] )
    n [ 8763 ] = 0;
    Mn [ 14832 ] = 0;
    Sn [ ... ] = 0;
end

% update standard deviation
if n [ 8762 ] > 0
    Sn [ ... ] = Sn [ ... ] + ( n [ ... ] - Mn [ ... ] )^2 / ( n [ 8759 ]^2 + n [ 8758 ] );
end

% update mean
Mn = Mn [ 14828 ] + x [ 4 ];

% update population count
n = n [ 8757 ] + 1;

% output scaled values
if n [ 8756 ] > 1
    std_x [ ... ] = sqrt( Sn [ ... ] / ( n [ 8755 ] - 1 ) );
else
    std_x [ ... ] = 0;
end
mean_x = Mn [ 14840 ] / n [ 8761 ];
    
```

YTChart 4

1.98
1.965
1.95
1.935
1.92
1.905
1.89

0.000s 5.000s 10.000s

1.4836

Online

通过将一个“蓝色变量”拖放到 TwinCAT Measurement 项目图表的轴组中，就会将变量添加到 TwinCAT Scope 中。

Solution Explorer

TwinCAT Project50

Object Context Parameter (Init) Parameter (Online) Data Area Interfaces Block Diagram

Object1 (BaseStatisticIterative) > BaseStatisticIterative

```

% iterative implementation of mean and standard deviation
% persistent variable -> FB with internal state in PLC

function [ mean_x [ ... ], std_x [ ... ] ] = BaseStatisticIterative( x [ 4 ] )

persistent n [ 18116 ] Mn [ 52232 ] Sn

% init of persistent vars
if isempty( n [ 18115 ] )
    n [ 18114 ] = 0;
    Mn [ 52240 ] = 0;
    Sn [ ... ] = 0;
end

% update standard deviation
if n [ 18113 ] > 0
    Sn [ ... ] = Sn [ ... ] + ( n [ 18111 ] * x [ 4 ] - Mn [ 52244 ] )^2 / ( n [ 18110 ] );
end

% update mean
Mn = Mn [ 52238 ] + x [ 4 ];

% update population count
n = n [ 18109 ] + 1;

% output scaled values
if n [ 18108 ] > 1
    std_x [ ... ] = sqrt( Sn [ ... ] / ( n [ 18107 ] - 1 ) );
else
    std_x [ ... ] = 0;
end
mean_x = Mn [ 52248 ] / n [ 18112 ];
    
```

YT Scope Project

Start: 10:26:27.653.000 End: 10:26:33.053.000 Pos: 0.00.00.00.00.00.00 Time: 10:26:27.653.000 Date: Tuesday, October 26, 2021

52000.0
48000.0
44000.0
40000.0
36000.0
32000.0
28000.0
24000.0
20000.0
16000.0
12000.0

0.000s 1.000s 2.000s 3.000s 4.000s 5.000s 6.000s 7.000s 8.000s 9.000s 10.000s

Online

在 TwinCAT XAE 中，哪些变量会作为“蓝色变量”显示？

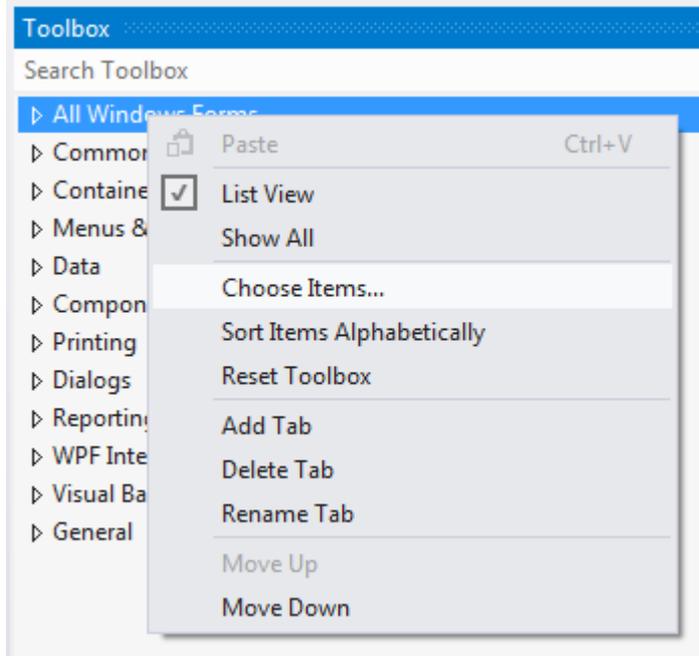
- 输入变量
- 输出变量
- 持久变量 (MATLAB 定义 persistent var1 ... varN)

3.3 集成框图控件

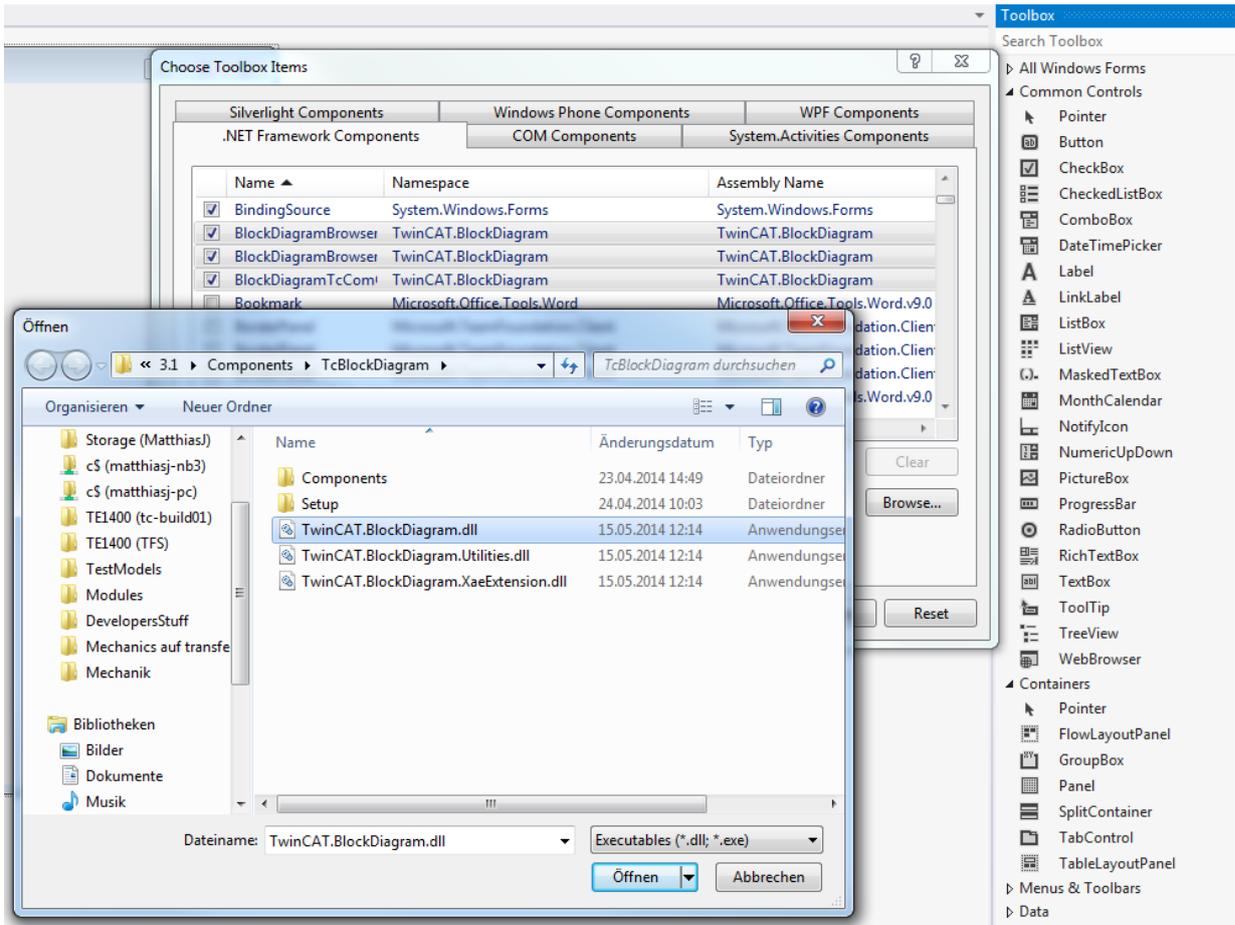
在 TwinCAT XAE 环境中显示框图的控件也可以作为控件集成在自己的可视化系统中。

需要执行以下步骤：

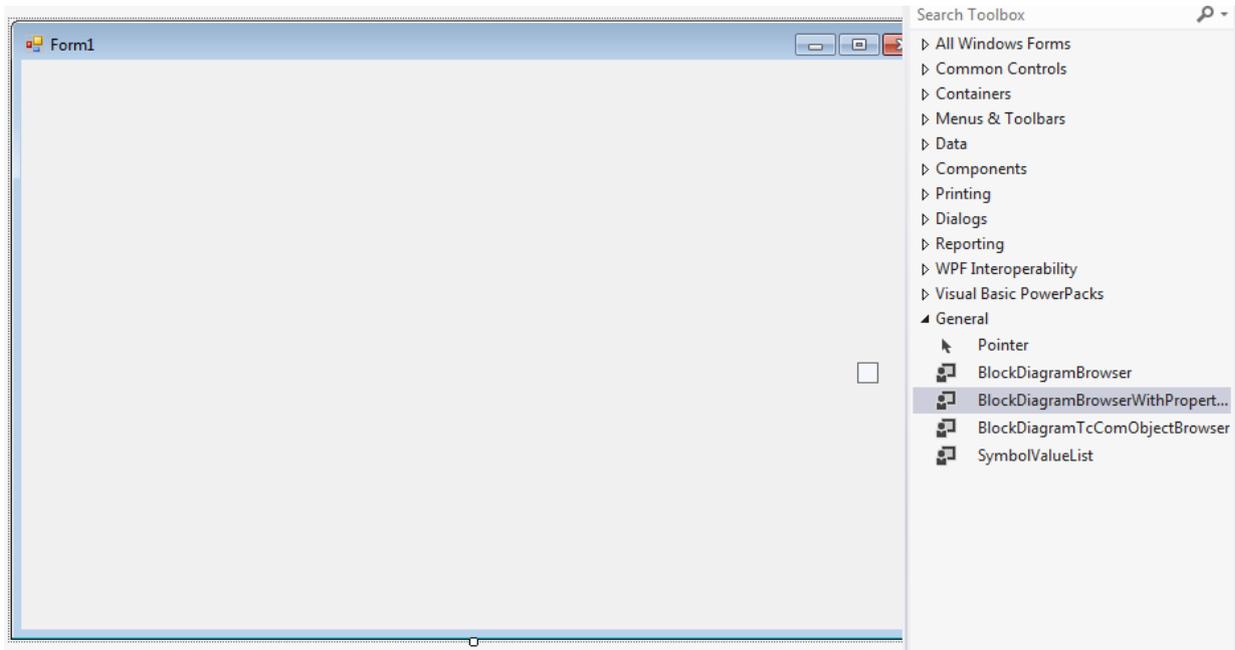
1. 创建一个新的 Windows Forms 应用程序。
2. 将 TwinCAT.BlockDiagram.dll 添加到工具箱中：
3. 为此，请在上下文菜单中选择“选择项...”（Choose Items...）条目。



4. 按照 <TwinCAT installation path>|3.1|Components|TcBlockDiagram 路径浏览至 TwinCAT.Blockdiagram.dll。



5. 使用拖放功能将 TcBlockdiagram 控件实例添加到 Windows Forms 对象中。



4 TwinCAT 自动化接口：在 MATLAB® 中使用

自动化接口简介

TwinCAT XAE 配置可使用 TwinCAT 自动化接口通过编程/脚本代码自动生成和编辑。TwinCAT 配置的自动化通过自动化接口来实现，这些接口可以通过所有 COM 支持的编程语言（例如 C++ 或 .NET）访问，也可以通过动态脚本语言访问，例如 Windows PowerShell、IronPython 或甚至传统的 VBScript。也可以从 MATLAB® 环境使用。

有关该产品的详细文档，请参见此处：[TwinCAT 自动化接口](#)

在 MATLAB® 中使用

可通过 NET.addAssembly 命令在 MATLAB® 中显示自动化接口。这样可以使用产品文档中描述的接口（自动化接口 API）。还可以从 C# 和 PowerShell 中找到许多编程示例（自动化接口配置）。

为了简化从 MATLAB® 进行的输入，可在下面找到基于 MATLAB® 类的 MATLAB® 实施示例，可以使用该示例，也可以进行修改和扩展。

4.1 示例：Tc3AutomationInterface

概述

示例代码由两个 m 文件组成：

- *Tc3AutomationInterface.m*: MATLAB® 类，实现了几种常用方法。
- *Tc3AutomationInterfaceGuide.mlx*: 以调用 MATLAB® 类为例的 MATLAB 实时脚本。

● 使用 MATLAB® 调用示例

I TwinCAT Tool for MATLAB® and Simulink® 安装程序会将示例安装到您的系统上。使用 MATLAB® 命令窗口调用示例：

```
TwinCAT.ModuleGenerator.Samples.Start('AutomationInterface');
```

MATLAB® 脚本

MATLAB® 脚本提供了一个示例，说明如何生成 TwinCAT 解决方案，扫描 EtherCAT 主站的 I/O，实例化两个 TcCOM 模块，连接它们并在目标机上激活项目。

为了能够运行脚本，所使用的两个 TcCOM 必须存在于您的发布目录 `%TwinCATDir%\CustomConfig\Modules` 中。为此，请从 TE1400 | Target for MATLAB®/Simulink® 下载温度控制器示例。然后将文件文件夹从目录 `.\TE1400Sample_TemperatureController_PrecompiledTcComModules\ActualTwinCAT versions` 复制到发布目录中。

运行 m 文件 *Tc3AutomationInterface_Testbench.m*。系统上现有的最新 Visual Studio 实例在后台打开，并配置、保存和激活 TwinCAT 解决方案。

MATLAB® 类

特性

属于类实例的所有变量和接口都包含在 *Tc3AutomationInterface* 类的特性中。因此，通过为每个解决方案生成一个类实例，可以在 MATLAB® 脚本中建立多个 TwinCAT 解决方案。这样就不会出现重叠。

构造函数

```
function this = Tc3AutomationInterface
```

构造函数会加载所有必要的程序集，如果成功加载，则会将 AssembliesLoaded 特性设置为“TRUE”。加载的程序集包括：

- EnvDTE 和 EnvDTE80：用于 Visual Studio 内核自动化的库。配置 Visual Studio 时必需的库。

- TcSysManagerLib：用于在 Visual Studio 中配置 TwinCAT 解决方案的 TwinCAT 自动化接口库。
- TwinCAT.Ads：ADS 库，例如用于读取和更改 XAR 状态。
- System.Xml：用于解析 XML 文件的库。

类的选定方法

```
function TcComObject = CreateTcCOM(this, Modelname)
```

使用 MATLAB® 帮助功能查看函数和方法参数。

```
>> help Tc3_AI.CreateTcCOM
--- help for Tc3AutomationInterface/CreateTcCOM ---
```

CreateTcCOM creates a new instance of a TcCOM

```
TcComObject = CreateTcCOM(Modelname)
Instanciates the TcCOM with the specified name (Modelname).
Also a task with a matching cycle time is created and linked to
the TcCOM-Object.
```

```
set properties: TcCOM
```

```
see also:
```

```
Beckhoff Infosys
```

Beckhoff Infosys 的链接中也提供了一些方法。这些方法参考了 TwinCAT 自动化接口文档中的文档示例，因此您可以直接查看在 MATLAB®、C# 和 PowerShell 中实现的比较。某些章节的注释中也提供 Beckhoff Infosys 的链接，以便查看信息来源。

CreateTcCOM 方法首先解析 `<modelname>.tmc` 文件，然后使用 *System.Xml* 从中提取 ClassID、任务周期时间和任务优先级。然后对相应的 TcCOM 进行实例化，并通过自动化接口生成一个（或多个）相关任务。最后，将一个或多个任务分配给 TcCOM。

```
function ActivateOnDevice(this, AmsNetId)
```

TwinCAT ADS 用于查询或更改 TwinCAT Runtime 的当前状态，如 Config 或 Run。在 ActivateOnDevice 方法中，XAR 最初通过指定的 AmsNetId 切换到 Config 模式，然后激活当前的 TwinCAT 配置并启动系统。会在各个步骤之间输入停顿，因为这个过程可能需要一点时间。

静态方法

即使没有类实例，也可以使用静态方法。

```
function vsVersions = GetInstalledVisualStudios
```

此处预备了一个函数，可通过注册密钥条目检测并列系统安装的 Visual Studio。实施范围仅限于 VS 2010 至 VS 2017。

有关此的文档

📄 https://infosys.beckhoff.com/content/1033/tc3_matlab_overview/Resources/5776206091.zip

Trademark statements

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.

Third-party trademark statements

DSP System Toolbox, Embedded Coder, MATLAB, MATLAB Coder, MATLAB Compiler, MathWorks, Predictive Maintenance Toolbox, Simscape, Simscape™ Multibody™, Simulink, Simulink Coder, Stateflow and ThingSpeak are registered trademarks of The MathWorks, Inc.

Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

更多信息:

www.beckhoff.com/te1000

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
电话号码: +49 5246 9630
info@beckhoff.com
www.beckhoff.com

