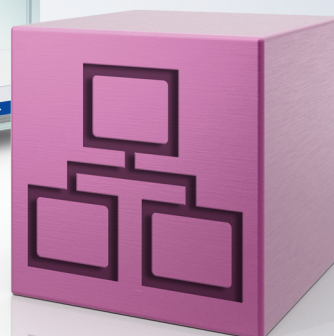
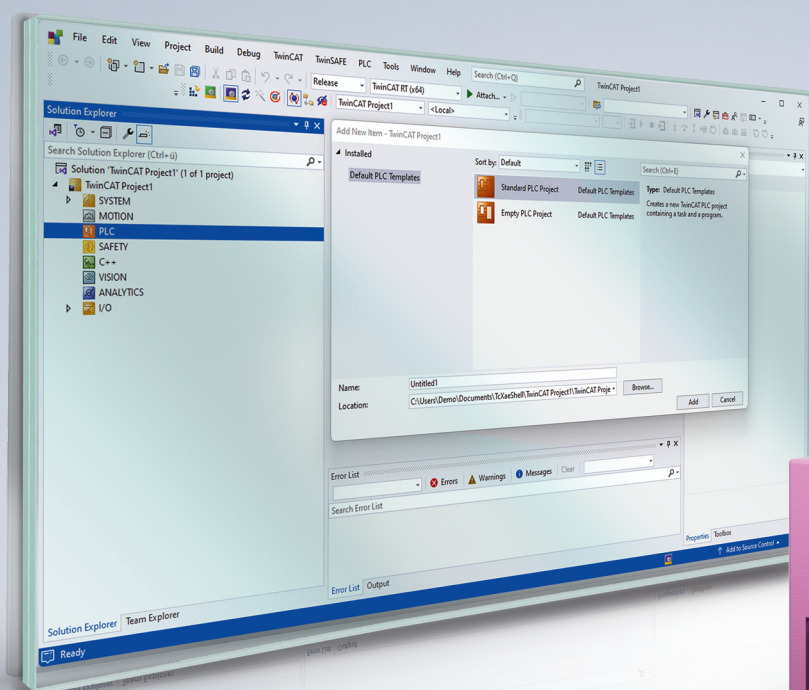


手册 | ZH

TF6255

TwinCAT 3 | Modbus RTU



目录

1 前言 5

1.1 文档说明 5

1.2 安全信息 5

1.3 信息安全说明 6

2 概述 7

3 安装 8

3.1 系统要求 8

3.2 安装 8

3.3 授权 9

4 配置 12

4.1 端子模块配置 12

4.2 Modbus 地址数组 12

5 PLC API 14

5.1 功能块 14

5.1.1 [已过时] 14

5.1.2 ModbusRtuMasterV2_PcCOM 20

5.1.3 ModbusRtuMasterV2_KL6x22B 23

5.1.4 ModbusRtuMasterV2_KL6x5B 26

5.1.5 ModbusRtuMasterV2_Generic 29

5.1.6 ModbusRtuSlave_PcCOM 32

5.1.7 ModbusRtuSlave_KL6x22B 33

5.1.8 ModbusRtuSlave_KL6x5B 34

5.1.9 ModbusRtuSlave_Generic 36

5.2 Datatypes 37

5.2.1 MODBUS_UNITID 37

5.2.2 MODBUS_ERRORS 37

5.3 全局常量 38

5.3.1 Global_Version 38

6 技术支持和服务 39

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

本文档经过精心准备。然而，所述产品正在不断开发中。

我们保留随时修改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®、ATRO®、EtherCAT®、EtherCAT G®、EtherCAT G10®、EtherCAT P®、MX-System®、Safety over EtherCAT®、TC/BSD®、TwinCAT®、TwinCAT/BSD®、TwinSAFE®、XFC®、XPlanar® 和 XTS® 是 Beckhoff Automation GmbH

的注册商标并由其授权使用。本出版物中所使用的其它名称可能是商标名称，任何第三方出于其自身目的使用它们可能会侵犯商标所有者的权利。



EtherCAT® 是注册商标和专利技术，由 Beckhoff Automation GmbH 授权使用。

版权所有

© Beckhoff Automation GmbH。

未经明确授权，不得复制、分发、使用和传播本文档内容。

违者将被追究赔偿责任。Beckhoff Automation GmbH 保留所有发明、实用新型和外观设计专利权。

第三方商标

本文档可能使用了第三方商标。有关商标信息，可以访问：<https://www.beckhoff.com/trademarks>。

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。

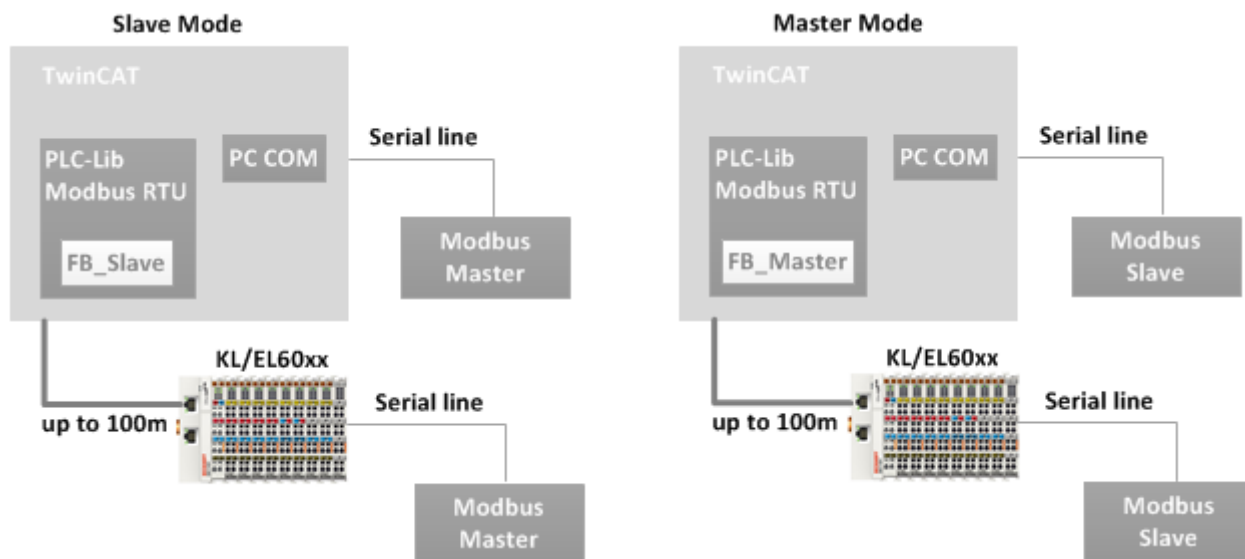
此外，还应遵守 Beckhoff 关于采取适当保护措施的建议。关于信息安全和工业安全的更多信息，请访问本公司网站 <https://www.beckhoff.com/secguide>。

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 3 Modbus RTU 提供了用于与 Modbus 端子模块设备进行串行通讯的功能块。



Modbus RTU 设备通过串行接口与倍福控制器连接。TwinCAT PLC 使用 Modbus RTU 库的从站功能块与 Modbus 主站进行通信（从站模式）。此外，主站功能块还可用于寻址多个 Modbus 从站（主站模式）

支持的接口

- PC 或 CX 的串行 COM 端口
- 串行总线端子模块 KL60xx
- 串行 EtherCAT 端子模块 EL60xx
- PC 或 CX 的虚拟串行 COM 端口（USB 端口）
 - 额外使用（和授权）[TF6340 TwinCAT 3 串行通信](#)

更多文档

有关 Modbus 的技术细节和规格，请访问：<http://www.modbus.org>

边界条件

Modbus 协议定义了精确的时间要求，以确保（例如）电文的所有字符能够完整传输。由于 Modbus RTU 通信是在 PLC 控制器上实现的，受限于 PLC 程序的周期性执行方式，无法保证精确的时序。大多数终端设备都具有很高的容错性，即使字符之间存在短时间间隔，也能正常工作。在个别情况下，仍需检查终端设备的具体行为。

EL60x2 的第二个通道不适合进行 Modbus RTU 通信。因为该通道的处理优先级较低，会导致发送的帧数据之间存在间隙，而这些间隙可能会被远程终端误判为帧错误。



对于某些串行接口终端模块，可以在发送之前先填充内部缓冲区（连续发送选项）。如果在相应的串行终端模块中进行了设置，ModbusRTU 库就可以利用这一特性。例如，在 KL6031 模块上，可以通过 KL6configuration 配置功能块激活连续模式（寄存器 34 的第 6 位）。随后，最多 128 字节的数据将被放入总线端子模块的内部缓冲区，并以连续方式发送。

3 安装

3.1 系统要求

技术数据		TF6255 TC3 Modbus RTU
目标系统	Windows	WEC7、Win10、Win11 IPC 或 CX (x86, x64, Arm®)
	TwinCAT/BSD	TwinCAT/BSD >= 13.0 IPC 或 CX (x64)
	Beckhoff RT Linux®	Beckhoff RT Linux® >= 13.3 IPC 或 CX (x64, Arm®)
TwinCAT	最低 TwinCAT 版本	3.1.4012
	最低 TwinCAT 等级	TC1200 TwinCAT 3 PLC
	所需 TwinCAT 授权	TF6255 TwinCAT 3 Modbus RTU
所需软件包 Engineering (XAE)	Windows (TcPkg)	TF6255-ModbusRTU-XAE

3.2 安装

根据所使用的 TwinCAT 版本和操作系统，该 TwinCAT 3 功能组件可以采用不同的安装方式，下面将详细介绍。



更新安装

更新安装总是会卸载之前安装版本。

- 在更新之前，请先备份您的配置文件。

TwinCAT 3 Package Manager : 安装 (TwinCAT 3.1 Build 4026)

如果在 Microsoft Windows 操作系统上使用 TwinCAT 3.1 Build 4026（及更高版本），则可以通过 TwinCAT Package Manager (TcPkg) 安装此功能。有关安装产品的详细说明，请参阅 [TwinCAT 3.1 Build 4026 安装说明](#) 中的管理 TwinCAT 软件。

注意

未做好准备的 TwinCAT 重启会导致数据丢失

安装该功能可能会导致 TwinCAT 重启。

- 确保系统上没有运行重要的 TwinCAT 应用程序，或者先有序关闭这些应用程序。

TwinCAT Package Manager : 安装 (TwinCAT 3.1 Build 4026)

有关安装产品的详细说明，请参阅 [TwinCAT 3.1 Build 4026 安装说明](#) 中的“安装工作负载”章节。

安装以下工作负载才能使用产品：

- TF6255 | TwinCAT 3 Modbus RTU

TwinCAT Package Manager CLI

您可以使用 [TcPkg Command Line Interface](#)（命令行界面，CLI）来显示系统上的可用工作负载：

```
tcpkg list -t workload
```

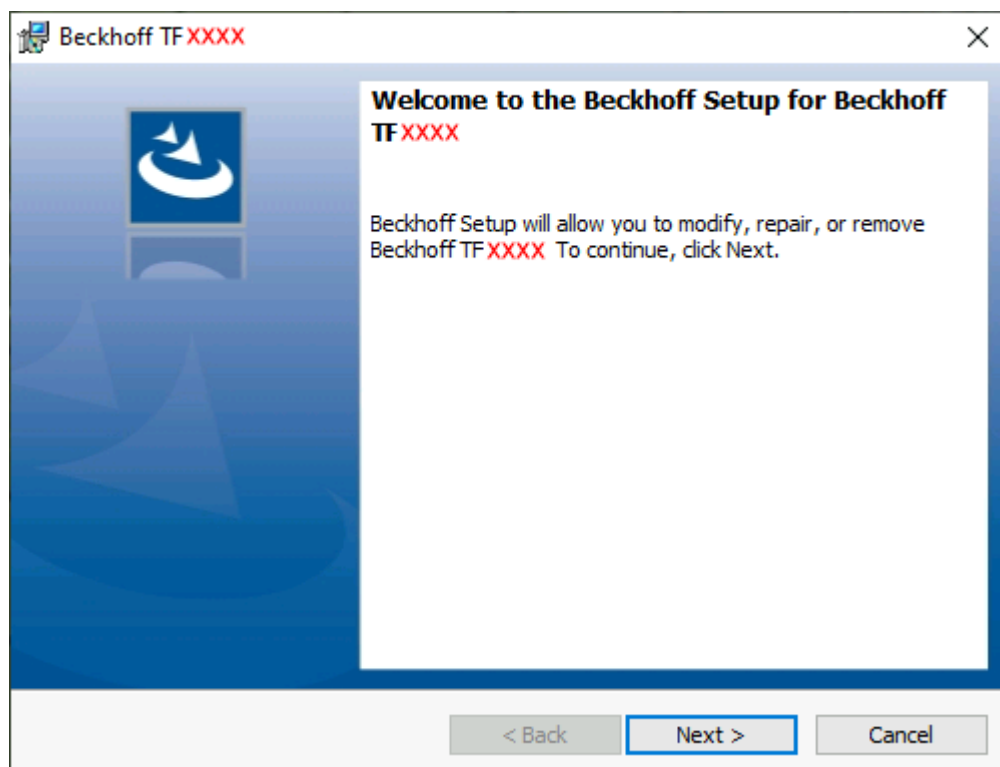
请使用以下命令安装工作负载：

```
tcpkg install TF6255.ModbusRTU.XAE
```

TwinCAT 3 设置：安装（TwinCAT 3.1 Build 4024 及更早版本）

如果您在 Microsoft Windows 操作系统上使用的是 TwinCAT 3.1 Build 4024，可以通过一个安装包来安装该功能，该安装包可以从倍福网站 <https://www.beckhoff.com/download> 上下载。

根据需要使用该功能的系统，可以在工程设计或运行时端进行安装。下面的截图是设置界面示例。



要完成安装过程，请按照“设置”对话框中的说明进行操作。

注意

未做好准备的 TwinCAT 重启会导致数据丢失

安装该功能可能会导致 TwinCAT 重启。

- 确保系统上没有运行重要的 TwinCAT 应用程序，或者先有序关闭这些应用程序。

3.3 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

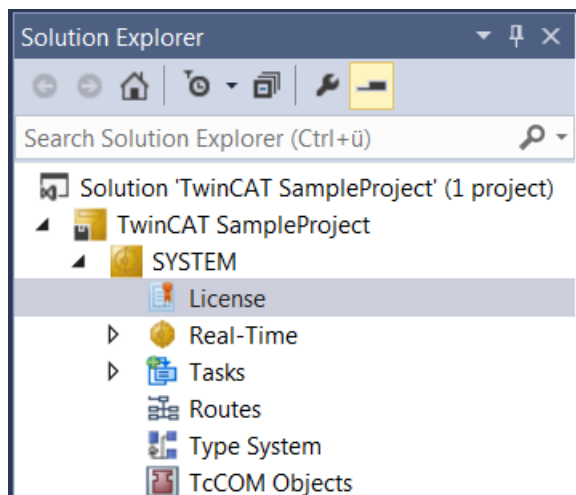


对于 TwinCAT 3 许可证加密狗无法启用 7 天测试版本。

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。

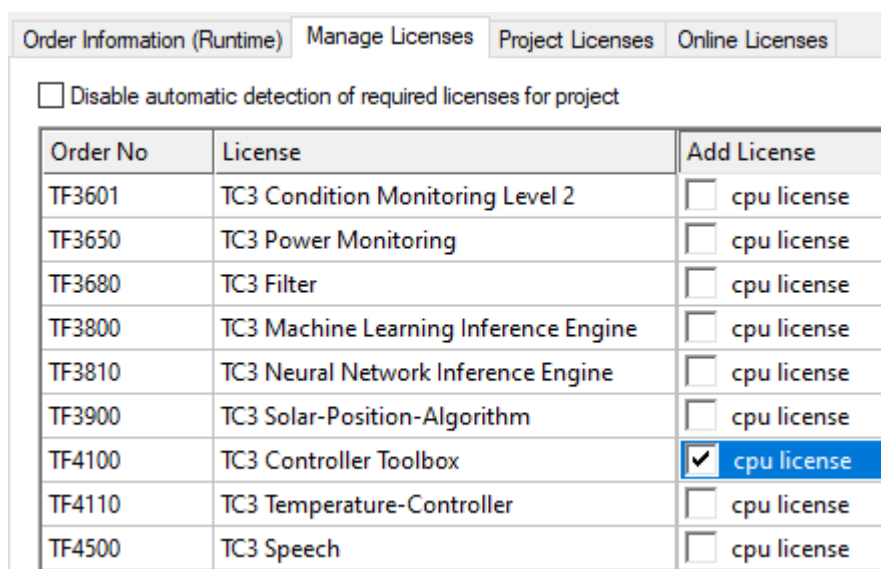
⇒ 授权设置总是指选定的目标系统。在目标系统上激活项目时，自动将相应的 TwinCAT 3 许可证复制到该系统中。

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。

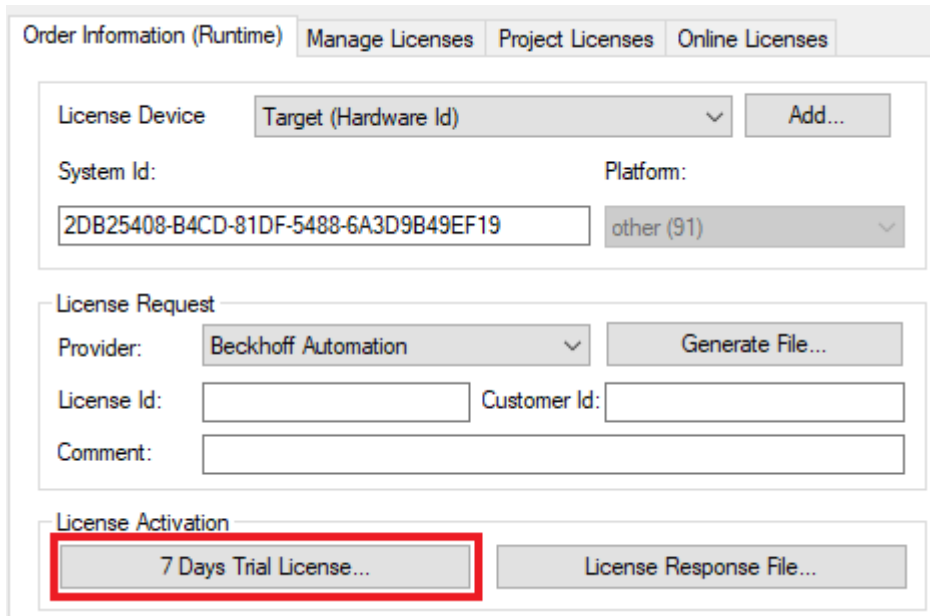
5. 打开**管理许可证**选项卡。在**添加许可证**栏中，选中希望添加到项目的许可证的复选框（例如“TF4100 TC3 控制器工具箱”）。



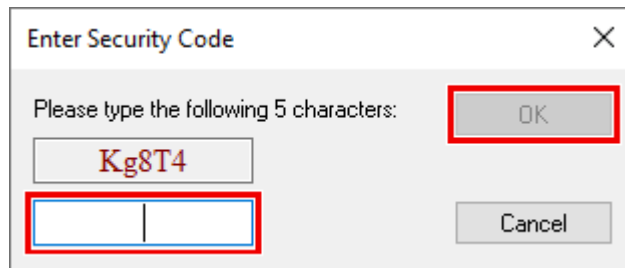
6. 打开**订单信息（运行时间）**选项卡。

⇒ 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 配置

4.1 端子模块配置

KL6001、KL6011、KL6021、KL6031 和 KL6041 总线端子模块可通过 KS2000 组态软件进行参数设置。

此外，还可以通过 `Tc2_SerialCom` 串行通信库中的 PLC 功能块对系统进行配置。这意味着 KL6configuration 功能块无需授权即可用于配置总线端子模块。



请注意，其他用于实现通信（而非配置）的 `Tc2_SerialCom` 库功能块，需要 TF6340 TwinCAT 3 Serial Communication 串行通信授权。

4.2 Modbus 地址数组

Modbus 针对不同的数据区域定义了相应的访问功能。这些数据区域在 TwinCAT PLC 程序中被声明为变量（例如 Word 数组），并作为输入参数传递给 Modbus 从站功能块。每个区域都有不同的 Modbus 起始地址，以便能够明确区分各个区域。在寻址时，必须考虑到这一偏移。

输入

输入数据区通常描述具有只读访问权限的物理输入数据。它们可以是数字量输入（位）或模拟量输入（字）。PLC 程序员可以决定是否允许通信伙伴直接访问物理输入点。也可以为 Modbus 通信定义一个与物理输入并不完全一致的输入区域：

将 Modbus 输入数据定义为物理输入的直接镜像。数据区的大小可以自由指定。它受到所用控制器输入过程映像实际大小的限制。

```
VAR
Inputs_AT%I* : ARRAY[0..255] OF WORD;
END_VAR
```

将 Modbus 输入数据定义为独立于物理输入的特定 Modbus 数据区。

```
VAR
Inputs : ARRAY[0..255] OF WORD;
END_VAR
```

输入数据区的最大大小为 2048 个字（`ARRAY[0...2047] OF WORD`）。

通过以下 Modbus 功能，可以通过 Modbus 主站访问输入区域：

```
2 : Read Input Status
4 : Read Input Registers
```

寻址

输入区域的地址偏移为 0，即报文中传输的地址 0 是输入数据区的第一个元素。

示例：

PLC 变量	访问类型	Modbus 报文中的地址	终端设备中的地址 (取决于设备)
输入 [0]	字	16#0	30001
输入 [1]	字	16#1	30002
输入区 [0]，第 0 位	位	16#0	10001
输入区 [1]，第 14 位	位	16#1E	1001F

输出

输出数据区域通常描述可读写的物理输出数据。输出可以是数字量输出（线圈）或模拟量输出（输出寄存器）。与输入一样，该区域既可以声明为物理输出变量，也可以声明为普通变量。

将 Modbus 输出数据定义为物理输出的直接镜像。数据区的大小可以自由指定。它受限于所使用的控制器输出进程映像的实际大小。

```
VAR
Outputs AT%Q* : ARRAY[0..255] OF WORD;
END_VAR
```

将 Modbus 输出数据定义为独立于物理输入的特定 Modbus 数据区。

```
VAR
Outputs : ARRAY[0..255] OF WORD;
END_VAR
```

输出数据区域的最大大小为 14336 字 (ARRAY[0...14335] OF WORD) 。

通过以下 Modbus 功能，可以通过 Modbus 主站访问输出区域：

```
1 : Read Coil Status
3 : Read Holding Registers
5 : Force Single Coil
6 : Preset Single Register
15 : Force Multiple Coils
16 : Preset Multiple Registers
```

寻址

输出区域以 16#800 偏移寻址，即报文中传输的地址 16#800 寻址输出数据区域的第一个元素。

示例：

PLC 变量	访问类型	Modbus 报文 中的地址	终端设备中的地址 (取决于设备)
输出 [0]	字	16#800	40801
输出 [1]	字	16#801	40802
输出区 [0]，第 0 位	位	16#800	00801
输出区 [1]，第 14 位	位	16#81E	0081F

内存

内存数据区域描述的是一个没有物理 I/O 映射的 PLC 变量区域。

将 Modbus 内存数据定义为变量。

```
VAR
Memory : ARRAY[0..255] OF WORD;
END_VAR
```

内存数据区域的最大大小为 16384 个字 (ARRAY[0...16383] OF WORD) 。

通过以下 Modbus 功能，可以通过 Modbus 主站访问内存区：

```
3 : Read Holding Registers
6 : Preset Single Register
16 : Preset Multiple Registers
```

寻址

内存区域的地址偏移为 16#4000，即报文中传送的地址 16#4000 是内存数据区的第一个字。

示例：

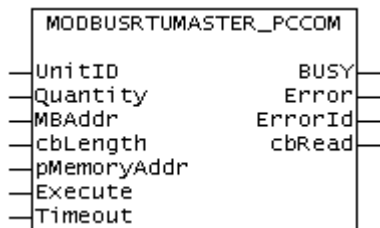
PLC 变量	访问类型	Modbus 报文 中的地址	终端设备中的地址 (取决于设备)
内存 [0]	字	16#4000	44001
内存 [1]	字	16#4001	44002

5 PLC API

5.1 功能块

5.1.1 [已过时]

5.1.1.1 ModbusRtuMaster_PcCOM



功能块 `ModbusRtuMaster_PcCOM` 实现了一个 Modbus 主站，通过 PC 串行接口（COM 端口）进行通信。功能块 `ModbusRtuMaster_KL6x5B` [► 16] 可通过串行总线端子模块 KL6001、KL6011 或 KL6021 进行通信。

● 硬件连接

i 与通信端口连接所需的数据结构已集成在该功能块中。一旦 PLC 程序集成后，它们就会显示在 TwinCAT System Manager 中，并可以与 COM 端口进行连接。操作步骤类似于 TF6340 TwinCAT 3 串行通信文档中串行 PC 接口一章的描述。

该功能块不以其基本形式进行调用，而是在 PLC 程序中使用该功能块内部的单个动作。每个 Modbus 功能都是以动作的形式实现的。

支持的 Modbus 功能（动作）

- **ModbusMaster.ReadCoils**

Modbus 功能 1 = 读取线圈

从连接的从站中读取二进制输出（线圈）。数据以压缩格式（每个字节存储 8 位）从指定的地址 `pMemoryAddr` 开始存储。

- **ModbusMaster.ReadInputStatus**

Modbus 功能 2 = 读取输入状态

从连接的从站中读取二进制输入。数据以压缩格式（每个字节存储 8 位）从指定的地址 `pMemoryAddr` 开始存储。

- **ModbusMaster.ReadRegs**

Modbus 功能 3 = 读取保持寄存器

从连接的从站读取数据。

- **ModbusMaster.ReadInputRegs**

Modbus 功能 4 = 读取输入寄存器

从连接的从站中读取输入寄存器。

- **ModbusMaster.WriteSingleCoil**

Modbus 功能 5 = 写入单线圈

向连接的从站发送一个二进制输出（线圈）。数据必须以压缩形式（每字节 8 位）从指定地址 `pMemoryAddr` 发送。

- **ModbusMaster.WriteSingleRegister**

Modbus 功能 6 = 写入单寄存器

向连接的从站发送一个单个数据字

- **ModbusMaster.WriteMultipleCoils**

Modbus 功能 15 = 写入多个线圈

向连接的从站发送二进制输出（线圈）。数据必须以压缩形式（每字节 8 位）从指定地址 pMemoryAddr 发送。

- **ModbusMaster.WriteRegs**

Modbus 功能 16 = 预置多寄存器

向连接的从站发送数据

- **ModbusMaster.Diagnostics**

Modbus 功能 8 = 诊断

向连接的从站发送诊断请求，并附带用户指定的子功能码。由于该功能（诊断）并不访问内存地址，因此其功能代码是在数据字 MBAddr 中传输。该功能所需的任何数据都包含在 pMemoryAddr 中。

输入

```
VAR_INPUT
    UnitID      : UINT;
    Quantity    : WORD;
    MBAddr      : WORD;
    cbLength     : UINT;
    pMemoryAddr : POINTER TO BYTE;
    Execute     : BOOL;
    Timeout     : TIME;
END_VAR
```

名称	类型	描述
单位标识	UINT	Modbus 站地址 [► 37] (1..247)。Modbus 从站只有在接收到包含其自身站地址的报文时才会做出响应。或者，也可选择使用广播地址来回复任何请求。地址 0 保留用于广播报文，因此不是有效的站地址（见 UnitID [► 37]）
数量	WORD	对于寻址的 Modbus 功能，该参数指定了需要读取或写入的数据字数。对于位寻址的 Modbus 功能，数量参数指定了需要读取或写入的位数（输入或线圈）。
MBAddr	WORD	Modbus 数据地址：指从终端设备（从站）读取数据时所指向的起始地址。该地址会原封不动地传输至从站，并在从站端被解释为数据地址。 但在执行诊断功能 (8) 时，此位置传输的是功能代码（即子功能码）。
cbLength	UINT	用于发送或读取动作的数据变量的大小（以字节为单位）。cbLength 必须大于或等于由数量决定的实际传输数据量。对于字访问，例如 [cbLength >= Quantity * 2]。cbLength 可以通过 SIZEOF(Modbus 数据) 计算得出。
pMemoryAddr	BYTE	PLC 中的内存地址，通过 ADR (Modbus 数据) 计算得出。对于读取操作，读取到的数据将存储在所指定的变量中。对于发送操作，数据将从指定的变量传输至终端设备。
执行	BOOL	开始信号。该操作由执行输入的上升沿触发。
超时	TIME	等待所寻址从站响应的超时时间。

输出

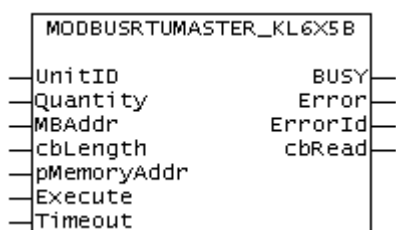
```
VAR_OUTPUT
    BUSY      : BOOL;
    Error     : BOOL;
    ErrorId   : MODBUS_ERRORS;
    cbRead    : UINT;
END_VAR
```

名称	类型	描述
繁忙	BOOL	表示功能块处于激活状态。当执行出现上升沿时，Busy 变为 TRUE；一旦启动的操作执行完毕，Busy 将重新变为 FALSE。在任何时候，只能有一项操作处于活动状态。
错误	BOOL	表示在执行操作过程中发生错误。
ErrorId	MODBUS_ERRORS	指示通信受到干扰或发生故障时的错误编号 [► 37]。
cbRead	UINT	提供读取操作的读取数据字节数。

要求

开发环境	目标平台	要包括的 PLC 库
TwinCAT v3.1.0	PC 或 CX (x86、x64、Arm®)	Tc2_ModbusRTU

5.1.1.2 ModbusRtuMaster_KL6x5B



功能块 `ModbusRtuMaster_KL6x5B` 可实现 Modbus 主站功能，通过串行总线端子模块 KL6001、KL6011 或 KL6021 进行通信。



硬件连接

与通信端口连接所需的数据结构已集成在该功能块中。在 PC 上，TwinCAT 系统管理器中的分配类似于 TF6340 TwinCAT 3 串行通信文档中串行总线端子模块章节中的描述。

该功能块不以其基本形式进行调用，而是在 PLC 程序中使用该功能块内部的单个动作。每个 Modbus 功能都是以动作的形式实现的。

支持的 Modbus 功能（动作）

- **ModbusMaster.ReadCoils**

Modbus 功能 1 = 读取线圈

从连接的从站中读取二进制输出（线圈）。数据以压缩格式（每个字节存储 8 位）从指定的地址 `pMemoryAddr` 开始存储。

- **ModbusMaster.ReadInputStatus**

Modbus 功能 2 = 读取输入状态

从连接的从站中读取二进制输入。数据以压缩格式（每个字节存储 8 位）从指定的地址 `pMemoryAddr` 开始存储。

- **ModbusMaster.ReadRegs**

Modbus 功能 3 = 读取保持寄存器

从连接的从站读取数据。

- **ModbusMaster.ReadInputRegs**

Modbus 功能 4 = 读取输入寄存器

从连接的从站中读取输入寄存器。

- **ModbusMaster.WriteSingleCoil**

Modbus 功能 5 = 写入单线圈

向连接的从站发送一个二进制输出（线圈）。数据必须以压缩形式（每字节 8 位）从指定地址 pMemoryAddr 发送。

- **ModbusMaster.WriteSingleRegister**

Modbus 功能 6 = 写入单寄存器

向连接的从站发送一个单个数据字

- **ModbusMaster.WriteMultipleCoils**

Modbus 功能 15 = 写入多个线圈

向连接的从站发送二进制输出（线圈）。数据必须以压缩形式（每字节 8 位）从指定地址 pMemoryAddr 发送。

- **ModbusMaster.WriteRegs**

Modbus 功能 16 = 预置多寄存器

向连接的从站发送数据

- **ModbusMaster.Diagnostics**

Modbus 功能 8 = 诊断

向连接的从站发送诊断请求，并附带用户指定的子功能码。由于该功能（诊断）并不访问内存地址，因此其功能代码是在数据字 MAddr 中传输。该功能所需的任何数据都包含在 pMemoryAddr 中。

输入

```
VAR_INPUT
    UnitID      : UINT;
    Quantity    : WORD;
    MAddr       : WORD;
    cbLength     : UINT;
    pMemoryAddr : POINTER TO BYTE;
    Execute     : BOOL;
    Timeout     : TIME;
END_VAR
```

名称	类型	描述
单位标识	UINT	Modbus 站地址 [► 37] (1..247)。Modbus 从站只有在接收到包含其自身站地址的报文时才会做出响应。或者，也可选择使用广播地址来回复任何请求。地址 0 保留用于广播报文，因此不是有效的站地址（见 UnitID [► 37]）
数量	WORD	对于寻址的 Modbus 功能，该参数指定了需要读取或写入的数据字数量。对于位寻址的 Modbus 功能，数量参数指定了需要读取或写入的位数量（输入或线圈）。
MAddr	WORD	Modbus 数据地址：指从终端设备（从站）读取数据时所指向的起始地址。该地址会原封不动地传输至从站，并在从站端被解释为数据地址。 但在执行诊断功能 (8) 时，此位置传输的是功能代码（即子功能码）。
cbLength	UINT	用于发送或读取动作的数据变量的大小（以字节为单位）。cbLength 必须大于或等于由数量决定的实际传输数据量。对于字访问，例如 [cbLength >= Quantity * 2]。cbLength 可以通过 SIZEOF(Modbus 数据) 计算得出。
pMemoryAddr	BYTE	PLC 中的内存地址，通过 ADR (Modbus 数据) 计算得出。对于读取操作，读取到的数据将存储在所指定的变量中。对于发送操作，数据将从指定的变量传输至终端设备。
执行	BOOL	开始信号。该操作由执行输入的上升沿触发。
超时	TIME	等待所寻址从站响应的超时时间。

输出

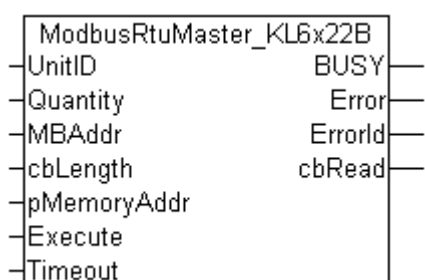
```
VAR_OUTPUT
    BUSY      : BOOL;
    Error     : BOOL;
    ErrorId   : MODBUS_ERRORS;
    cbRead    : UINT;
END_VAR
```

名称	类型	描述
繁忙	BOOL	表示功能块处于激活状态。当执行出现上升沿时，Busy 变为 TRUE；一旦启动的操作执行完毕，Busy 将重新变为 FALSE。在任何时候，只能有一项操作处于活动状态。
错误	BOOL	表示在执行操作过程中发生错误。
ErrorId	MODBUS_ERRORS	指示通信受到干扰或发生故障时的错误编号 [► 37]。
cbRead	UINT	提供读取操作的读取数据字节数。

要求

开发环境	目标平台	要包括的 PLC 库
TwinCAT v3.1.0	PC 或 CX (x86、x64、Arm®)	Tc2_ModbusRTU

5.1.1.3 ModbusRtuMaster_KL6x22B



功能块 `ModbusRtuMaster_KL6x22B` 可实现 Modbus 主站，通过串行总线端子模块 KL6031 或 KL6041 进行通信。功能块 `ModbusRtuMaster_PcCOM` [► 14] 可用于通过串行 PC 接口（COM 端口）进行通信。



硬件连接

与通信端口连接所需的数据结构已集成在该功能块中。在 PC 上，TwinCAT 系统管理器中的分配类似于 TF6340 TwinCAT 3 串行通信文档中 [串行总线端子模块章节](#) 中的描述。

该功能块不以其基本形式进行调用，而是在 PLC 程序中使用该功能块内部的单个动作。每个 Modbus 功能都是以动作的形式实现的。

支持的 Modbus 功能（动作）

- **ModbusMaster.ReadCoils**

Modbus 功能 1 = 读取线圈

从连接的从站中读取二进制输出（线圈）。数据以压缩格式（每个字节存储 8 位）从指定的地址 `pMemoryAddr` 开始存储。

- **ModbusMaster.ReadInputStatus**

Modbus 功能 2 = 读取输入状态

从连接的从站中读取二进制输入。数据以压缩格式（每个字节存储 8 位）从指定的地址 `pMemoryAddr` 开始存储。

- **ModbusMaster.ReadRegs**

Modbus 功能 3 = 读取保持寄存器

从连接的从站读取数据。

- **ModbusMaster.ReadInputRegs**

Modbus 功能 4 = 读取输入寄存器

从连接的从站中读取输入寄存器。

- **ModbusMaster.WriteSingleCoil**

Modbus 功能 5 = 写入单线圈

向连接的从站发送一个二进制输出（线圈）。数据必须以压缩形式（每字节 8 位）从指定地址 pMemoryAddr 发送。

- **ModbusMaster.WriteSingleRegister**

Modbus 功能 6 = 写入单寄存器

向连接的从站发送一个单个数据字

- **ModbusMaster.WriteMultipleCoils**

Modbus 功能 15 = 写入多个线圈

向连接的从站发送二进制输出（线圈）。数据必须以压缩形式（每字节 8 位）从指定地址 pMemoryAddr 发送。

- **ModbusMaster.WriteRegs**

Modbus 功能 16 = 预置多寄存器

向连接的从站发送数据

- **ModbusMaster.Diagnostics**

Modbus 功能 8 = 诊断

向连接的从站发送诊断请求，并附带用户指定的子功能码。由于该功能（诊断）并不访问内存地址，因此其功能代码是在数据字 MAddr 中传输。该功能所需的任何数据都包含在 pMemoryAddr 中。

输入

```
VAR_INPUT
    UnitID      : UINT;
    Quantity    : WORD;
    MAddr       : WORD;
    cbLength    : UINT;
    pMemoryAddr : POINTER TO BYTE;
    Execute     : BOOL;
    Timeout     : TIME;
END_VAR
```

名称	类型	描述
单位标识	UINT	Modbus 站地址 [► 37] (1..247)。Modbus 从站只有在接收到包含其自身站地址的报文时才会做出响应。或者，也可选择使用广播地址来回复任何请求。地址 0 保留用于广播报文，因此不是有效的站地址（见 UnitID [► 37]）
数量	WORD	对于寻址的 Modbus 功能，该参数指定了需要读取或写入的数据字数量。对于位寻址的 Modbus 功能，数量参数指定了需要读取或写入的位数量（输入或线圈）。
MAddr	WORD	Modbus 数据地址：指从终端设备（从站）读取数据时所指向的起始地址。该地址会原封不动地传输至从站，并在从站端被解释为数据地址。 但在执行诊断功能 (8) 时，此位置传输的是功能代码（即子功能码）。
cbLength	UINT	用于发送或读取动作的数据变量的大小（以字节为单位）。cbLength 必须大于或等于由数量决定的实际传输数据量。对于字访问，例如 [cbLength >= Quantity * 2]。cbLength 可以通过 SIZEOF(Modbus 数据) 计算得出。
pMemoryAddr	BYTE	PLC 中的内存地址，通过 ADR (Modbus 数据) 计算得出。对于读取操作，读取到的数据将存储在所指定的变量中。对于发送操作，数据将从指定的变量传输至终端设备。
执行	BOOL	开始信号。该操作由执行输入的上升沿触发。
超时	TIME	等待所寻址从站响应的超时时间。

输出

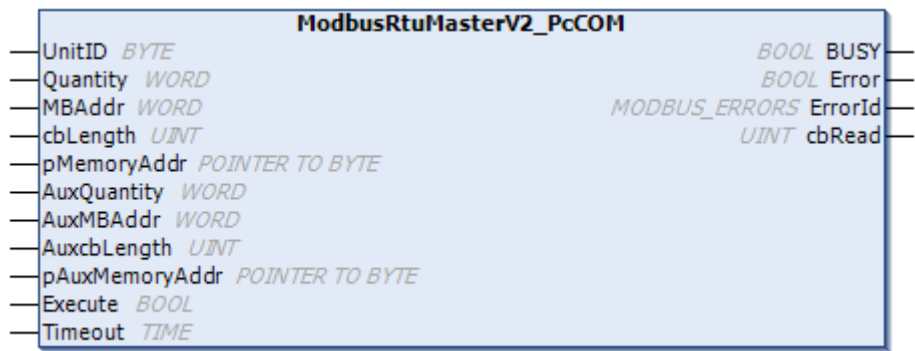
```
VAR_OUTPUT
  BUSY      : BOOL;
  Error     : BOOL;
  ErrorId   : MODBUS_ERRORS;
  cbRead    : UINT;
END_VAR
```

名称	类型	描述
繁忙	BOOL	表示功能块处于激活状态。当执行出现上升沿时，Busy 变为 TRUE；一旦启动的操作执行完毕，Busy 将重新变为 FALSE。在任何时候，只能有一项操作处于活动状态。
错误	BOOL	表示在执行操作过程中发生错误。
ErrorId	MODBUS_ERRORS	指示通信受到干扰或发生故障时的错误编号 [► 37]。
cbRead	UINT	提供读取操作的读取数据字节数。

要求

开发环境	目标平台	要包括的 PLC 库
TwinCAT v3.1.0	PC 或 CX (x86、x64、Arm®)	Tc2_ModbusRTU

5.1.2 ModbusRtuMasterV2_PcCOM



功能块 ModbusRtuMasterV2_PcCOM 实现了一个 Modbus 主站，通过 PC 串行接口（COM 端口）进行通信。可以通过串行总线端子模块进行通信的附加功能块。



硬件连接

与通信端口连接所需的数据结构已集成在该功能块中。一旦 PLC 程序集成后，它们就会显示在 TwinCAT System Manager 中，并可以与 COM 端口进行连接。操作步骤类似于 TF6340 TwinCAT 3 串行通信文档中串行 PC 接口一章的描述。

该功能块不以其基本形式进行调用，而是在 PLC 程序中使用该功能块内部的单个动作。每个 Modbus 功能都是以动作的形式实现的。

支持的 Modbus 功能（动作）

- ModbusMaster.ReadCoils**
 Modbus 功能 1 = 读取线圈
 从连接的从站中读取二进制输出（线圈）。数据以压缩格式（每个字节存储 8 位）从指定的地址 pMemoryAddr 开始存储。
- ModbusMaster.ReadInputStatus**
 Modbus 功能 2 = 读取输入状态
 从连接的从站中读取二进制输入。数据以压缩格式（每个字节存储 8 位）从指定的地址 pMemoryAddr 开始存储。

- **ModbusMaster.ReadRegs**
Modbus 功能 3 = 读取保持寄存器
从连接的从站读取数据。
- **ModbusMaster.ReadInputRegs**
Modbus 功能 4 = 读取输入寄存器
从连接的从站中读取输入寄存器。
- **ModbusMaster.WriteSingleCoil**
Modbus 功能 5 = 写入单线圈
向连接的从站发送一个二进制输出（线圈）。数据必须以压缩形式（每字节 8 位）从指定地址 pMemoryAddr 发送。
- **ModbusMaster.WriteSingleRegister**
Modbus 功能 6 = 写入单寄存器
向连接的从站发送一个单个数据字
- **ModbusMaster.WriteMultipleCoils**
Modbus 功能 15 = 写入多个线圈
向连接的从站发送二进制输出（线圈）。数据必须以压缩形式（每字节 8 位）从指定地址 pMemoryAddr 发送。
- **ModbusMaster.WriteRegs**
Modbus 功能 16 = 预置多寄存器
向连接的从站发送数据
- **ModbusMaster.Diagnostics**
Modbus 功能 8 = 诊断
向连接的从站发送诊断请求，并附带用户指定的子功能码。由于该功能（诊断）并不访问内存地址，因此其功能代码是在数据字 MBAddr 中传输。该功能所需的任何数据都包含在 pMemoryAddr 中。

MasterV2 功能块支持的 Modbus 功能（操作）

- **ModbusMaster.ReadWriteRegs**
Modbus 功能 23 = 读/写多个寄存器
将通过 Aux 参数指定的数据发送至连接的从站，并同时从该从站接收数据。接收到的数据存储在 pMemoryAddr 指定的地址中。
- **ModbusMaster.UserReadWrite**
通用用户报文
用户在指定数据 (pMemoryAddr) 的第一个字节中指定 Modbus 功能码。通过该功能，用户能够发送带有任何功能码的 Modbus 报文。从从站接收到的任何数据都存储在 pAuxMemoryAddr 指定的地址中。

输入

```
VAR_INPUT
    UnitID      : BYTE;
    Quantity    : WORD;
    MBAddr      : WORD;
    cbLength    : UINT;
    pMemoryAddr : POINTER TO BYTE;

    AuxQuantity  : WORD;
    AuxMBAddr    : WORD;
    AuxcbLength  : UINT;
    pAuxMemoryAddr : POINTER TO BYTE;

    Execute     : BOOL;
    Timeout     : TIME;
END_VAR
```

名称	类型	描述
单位标识	UINT	Modbus 站地址 [► 37] (1..247)。Modbus 从站只有在接收到包含其自身站地址的报文时才会做出响应。或者，也可选择使用广播地址来回复任何请求。地址 0 保留用于广播报文，因此不是有效的站地址（见 UnitID [► 37]）
数量	WORD	对于寻址的 Modbus 功能，该参数指定了需要读取或写入的数据字数量。对于位寻址的 Modbus 功能，数量参数指定了需要读取或写入的位数量（输入或线圈）。
MBAAddr	WORD	Modbus 数据地址：指从终端设备（从站）读取数据时所指向的起始地址。该地址会原封不动地传输至从站，并在从站端被解释为数据地址。 但在执行诊断功能 (8) 时，此位置传输的是功能代码（即子功能码）。
cbLength	UINT	用于发送或读取动作的数据变量的大小（以字节为单位）。cbLength 必须大于或等于由数量决定的实际传输数据量。对于字访问，例如 [cbLength >= Quantity * 2]。cbLength 可以通过 SIZEOF(Modbus 数据) 计算得出。
pMemoryAddr	BYTE	PLC 中的内存地址，通过 ADR (Modbus 数据) 计算得出。对于读取操作，读取到的数据将存储在所指定的变量中。对于发送操作，数据将从指定的变量传输至终端设备。
AuxQuantity	WORD	仅用于读/写功能的附加参数，参见 ReadWriteRegs/UserReadWrite [► 21]
AuxMBAAddr	WORD	仅用于读/写功能的附加参数，参见 ReadWriteRegs/UserReadWrite [► 21] 。
AuxcbLength	UINT	仅用于读/写功能的附加参数，参见 ReadWriteRegs/UserReadWrite [► 21]
pAuxMemoryAddr	BYTE	仅用于读/写功能的附加参数，参见 ReadWriteRegs/UserReadWrite [► 21]
执行	BOOL	开始信号。该操作由执行输入的上升沿触发。
超时	TIME	等待所寻址从站响应的超时时间。

输出

```

VAR_OUTPUT
    BUSY      : BOOL;
    Error     : BOOL;
    ErrorId   : MODBUS_ERRORS;
    cbRead    : UINT;
END_VAR

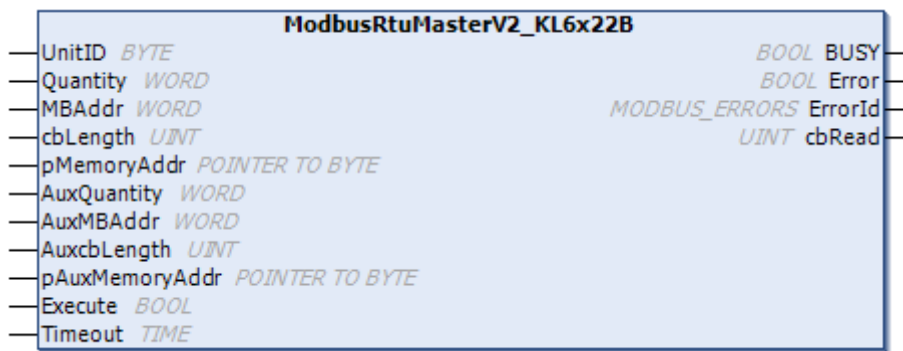
```

名称	类型	描述
繁忙	BOOL	表示功能块处于激活状态。当执行出现上升沿时，Busy 变为 TRUE；一旦启动的操作执行完毕，Busy 将重新变为 FALSE。在任何时候，只能有一项操作处于活动状态。
错误	BOOL	表示在执行操作过程中发生错误。
ErrorId	MODBUS_ERRORS	指示通信受到干扰或发生故障时的错误编号 [► 37]。
cbRead	UINT	提供读取操作的读取数据字节数。

要求

开发环境	目标平台	要包括的 PLC 库
TwinCAT v3.1.0	PC 或 CX (x86、x64、Arm®)	Tc2_ModbusRTU

5.1.3 ModbusRtuMasterV2_KL6x22B



功能块 `ModbusRtuMasterV2_KL6x22B` 可实现 Modbus 主站，通过串行总线端子模块 KL6031 或 KL6041 进行通信。也支持具有 22 字节进程映像数据的串行 EtherCAT 端子模块。功能块 `ModbusRtuMasterV2_PcCOM` [► 20] 可用于通过串行 PC 接口（COM 端口）进行通信。

i 硬件连接

与通信端口连接所需的数据结构已集成在该功能块中。在 PC 上，TwinCAT 系统管理器中的分配类似于 TF6340 TwinCAT 3 串行通信文档中[串行总线端子模块章节](#)中的描述。

该功能块不以其基本形式进行调用，而是在 PLC 程序中使用该功能块内部的单个动作。每个 Modbus 功能都是以动作的形式实现的。

支持的 Modbus 功能（动作）

- **ModbusMaster.ReadCoils**

Modbus 功能 1 = 读取线圈

从连接的从站中读取二进制输出（线圈）。数据以压缩格式（每个字节存储 8 位）从指定的地址 `pMemoryAddr` 开始存储。

- **ModbusMaster.ReadInputStatus**

Modbus 功能 2 = 读取输入状态

从连接的从站中读取二进制输入。数据以压缩格式（每个字节存储 8 位）从指定的地址 `pMemoryAddr` 开始存储。

- **ModbusMaster.ReadRegs**

Modbus 功能 3 = 读取保持寄存器

从连接的从站读取数据。

- **ModbusMaster.ReadInputRegs**

Modbus 功能 4 = 读取输入寄存器

从连接的从站中读取输入寄存器。

- **ModbusMaster.WriteSingleCoil**

Modbus 功能 5 = 写入单线圈

向连接的从站发送一个二进制输出（线圈）。数据必须以压缩形式（每字节 8 位）从指定地址 `pMemoryAddr` 发送。

- **ModbusMaster.WriteSingleRegister**

Modbus 功能 6 = 写入单寄存器

向连接的从站发送一个单个数据字

- **ModbusMaster.WriteMultipleCoils**

Modbus 功能 15 = 写入多个线圈

向连接的从站发送二进制输出（线圈）。数据必须以压缩形式（每字节 8 位）从指定地址 `pMemoryAddr` 发送。

- **ModbusMaster.WriteRegs**

Modbus 功能 16 = 预置多寄存器

向连接的从站发送数据

- **ModbusMaster.Diagnostics**

Modbus 功能 8 = 诊断

向连接的从站发送诊断请求，并附带用户指定的子功能码。由于该功能（诊断）并不访问内存地址，因此其功能代码是在数据字 MBAddr 中传输。该功能所需的任何数据都包含在 pMemoryAddr 中。

MasterV2 功能块支持的 Modbus 功能（操作）

- **ModbusMaster.ReadWriteRegs**

Modbus 功能 23 = 读/写多个寄存器

将通过 Aux 参数指定的数据发送至连接的从站，并同时从该从站接收数据。接收到的数据存储在 pMemoryAddr 指定的地址中。

- **ModbusMaster.UserReadWrite**

通用用户报文

用户在指定数据 (pMemoryAddr) 的第一个字节中指定 Modbus 功能码。通过该功能，用户能够发送带有任何功能码的 Modbus 报文。从从站接收到的任何数据都存储在 pAuxMemoryAddr 指定的地址中。

输入

```
VAR_INPUT
    UnitID      : BYTE;
    Quantity    : WORD;
    MBAddr      : WORD;
    cbLength    : UINT;
    pMemoryAddr : POINTER TO BYTE;

    AuxQuantity  : WORD;
    AuxMBAddr    : WORD;
    AuxcbLength  : UINT;
    pAuxMemoryAddr : POINTER TO BYTE;

    Execute      : BOOL;
    Timeout      : TIME;
END_VAR
```

名称	类型	描述
单位标识	UINT	Modbus 站地址 [► 37] (1..247)。Modbus 从站只有在接收到包含其自身站地址的报文时才会做出响应。或者，也可选择使用广播地址来回复任何请求。地址 0 保留用于广播报文，因此不是有效的站地址（见 UnitID [► 37]）
数量	WORD	对于寻址的 Modbus 功能，该参数指定了需要读取或写入的数据字数量。对于位寻址的 Modbus 功能，数量参数指定了需要读取或写入的位数量（输入或线圈）。
MBAAddr	WORD	Modbus 数据地址：指从终端设备（从站）读取数据时所指向的起始地址。该地址会原封不动地传输至从站，并在从站端被解释为数据地址。 但在执行诊断功能 (8) 时，此位置传输的是功能代码（即子功能码）。
cbLength	UINT	用于发送或读取动作的数据变量的大小（以字节为单位）。cbLength 必须大于或等于由数量决定的实际传输数据量。对于字访问，例如 [cbLength >= Quantity * 2]。cbLength 可以通过 SIZEOF(Modbus 数据) 计算得出。
pMemoryAddr	BYTE	PLC 中的内存地址，通过 ADR (Modbus 数据) 计算得出。对于读取操作，读取到的数据将存储在所指定的变量中。对于发送操作，数据将从指定的变量传输至终端设备。
AuxQuantity	WORD	仅用于读/写功能的附加参数，参见 ReadWriteRegs/UserReadWrite [► 24]
AuxMBAAddr	WORD	仅用于读/写功能的附加参数，参见 ReadWriteRegs/UserReadWrite [► 24] 。
AuxcbLength	UINT	仅用于读/写功能的附加参数，参见 ReadWriteRegs/UserReadWrite [► 24]
pAuxMemoryAddr	BYTE	仅用于读/写功能的附加参数，参见 ReadWriteRegs/UserReadWrite [► 24]
执行	BOOL	开始信号。该操作由执行输入的上升沿触发。
超时	TIME	等待所寻址从站响应的超时时间。

输出

```

VAR_OUTPUT
    BUSY      : BOOL;
    Error     : BOOL;
    ErrorId   : MODBUS_ERRORS;
    cbRead    : UINT;
END_VAR

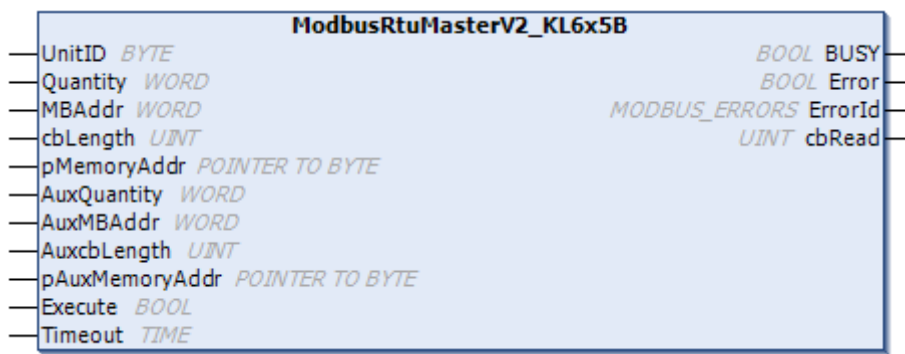
```

名称	类型	描述
繁忙	BOOL	表示功能块处于激活状态。当执行出现上升沿时，Busy 变为 TRUE；一旦启动的操作执行完毕，Busy 将重新变为 FALSE。在任何时候，只能有一项操作处于活动状态。
错误	BOOL	表示在执行操作过程中发生错误。
ErrorId	MODBUS_ERRORS	指示通信受到干扰或发生故障时的错误编号 [► 37]。
cbRead	UINT	提供读取操作的读取数据字节数。

要求

开发环境	目标平台	要包括的 PLC 库
TwinCAT v3.1.0	PC 或 CX (x86、x64、Arm®)	Tc2_ModbusRTU

5.1.4 ModbusRtuMasterV2_KL6x5B



功能块 `ModbusRtuMasterV2_KL6x5B` 可实现 Modbus 主站功能，通过串行总线端子模块 KL6001、KL6011 或 KL6021 进行通信。功能块 `ModbusRtuMasterV2_PcCOM [► 20]` 可用于通过串行 PC 接口（COM 端口）进行通信。

● 硬件连接

i 与通信端口连接所需的数据结构已集成在该功能块中。在 PC 上，TwinCAT 系统管理器中的分配类似于 TF6340 TwinCAT 3 串行通信文档中串行总线端子模块章节中的描述。

该功能块不以其基本形式进行调用，而是在 PLC 程序中使用该功能块内部的单个动作。每个 Modbus 功能都是以动作的形式实现的。

支持的 Modbus 功能（动作）

- **ModbusMaster.ReadCoils**

Modbus 功能 1 = 读取线圈

从连接的从站中读取二进制输出（线圈）。数据以压缩格式（每个字节存储 8 位）从指定的地址 `pMemoryAddr` 开始存储。

- **ModbusMaster.ReadInputStatus**

Modbus 功能 2 = 读取输入状态

从连接的从站中读取二进制输入。数据以压缩格式（每个字节存储 8 位）从指定的地址 `pMemoryAddr` 开始存储。

- **ModbusMaster.ReadRegs**

Modbus 功能 3 = 读取保持寄存器

从连接的从站读取数据。

- **ModbusMaster.ReadInputRegs**

Modbus 功能 4 = 读取输入寄存器

从连接的从站中读取输入寄存器。

- **ModbusMaster.WriteSingleCoil**

Modbus 功能 5 = 写入单线圈

向连接的从站发送一个二进制输出（线圈）。数据必须以压缩形式（每字节 8 位）从指定地址 `pMemoryAddr` 发送。

- **ModbusMaster.WriteSingleRegister**

Modbus 功能 6 = 写入单寄存器

向连接的从站发送一个单个数据字

- **ModbusMaster.WriteMultipleCoils**

Modbus 功能 15 = 写入多个线圈

向连接的从站发送二进制输出（线圈）。数据必须以压缩形式（每字节 8 位）从指定地址 `pMemoryAddr` 发送。

- **ModbusMaster.WriteRegs**

Modbus 功能 16 = 预置多寄存器

向连接的从站发送数据

- **ModbusMaster.Diagnostics**

Modbus 功能 8 = 诊断

向连接的从站发送诊断请求，并附带用户指定的子功能码。由于该功能（诊断）并不访问内存地址，因此其功能代码是在数据字 MBAddr 中传输。该功能所需的任何数据都包含在 pMemoryAddr 中。

MasterV2 功能块支持的 Modbus 功能（操作）

- **ModbusMaster.ReadWriteRegs**

Modbus 功能 23 = 读/写多个寄存器

将通过 Aux 参数指定的数据发送至连接的从站，并同时从该从站接收数据。接收到的数据存储在 pMemoryAddr 指定的地址中。

- **ModbusMaster.UserReadWrite**

通用用户报文

用户在指定数据 (pMemoryAddr) 的第一个字节中指定 Modbus 功能码。通过该功能，用户能够发送带有任何功能码的 Modbus 报文。从从站接收到的任何数据都存储在 pAuxMemoryAddr 指定的地址中。

输入

```
VAR_INPUT
    UnitID      : BYTE;
    Quantity    : WORD;
    MBAddr      : WORD;
    cbLength     : UINT;
    pMemoryAddr : POINTER TO BYTE;

    AuxQuantity  : WORD;
    AuxMBAddr    : WORD;
    AuxcbLength  : UINT;
    pAuxMemoryAddr : POINTER TO BYTE;

    Execute     : BOOL;
    Timeout      : TIME;
END_VAR
```

名称	类型	描述
单位标识	UINT	Modbus 站地址 [► 37] (1..247)。Modbus 从站只有在接收到包含其自身站地址的报文时才会做出响应。或者，也可选择使用广播地址来回复任何请求。地址 0 保留用于广播报文，因此不是有效的站地址（见 UnitID [► 37]）
数量	WORD	对于寻址的 Modbus 功能，该参数指定了需要读取或写入的数据字数量。对于位寻址的 Modbus 功能，数量参数指定了需要读取或写入的位数量（输入或线圈）。
MBAAddr	WORD	Modbus 数据地址：指从终端设备（从站）读取数据时所指向的起始地址。该地址会原封不动地传输至从站，并在从站端被解释为数据地址。 但在执行诊断功能 (8) 时，此位置传输的是功能代码（即子功能码）。
cbLength	UINT	用于发送或读取动作的数据变量的大小（以字节为单位）。cbLength 必须大于或等于由数量决定的实际传输数据量。对于字访问，例如 [cbLength >= Quantity * 2]。cbLength 可以通过 SIZEOF(Modbus 数据) 计算得出。
pMemoryAddr	BYTE	PLC 中的内存地址，通过 ADR (Modbus 数据) 计算得出。对于读取操作，读取到的数据将存储在所指定的变量中。对于发送操作，数据将从指定的变量传输至终端设备。
AuxQuantity	WORD	仅用于读/写功能的附加参数，参见 ReadWriteRegs/UserReadWrite [► 27]
AuxMBAAddr	WORD	仅用于读/写功能的附加参数，参见 ReadWriteRegs/UserReadWrite [► 27] 。
AuxcbLength	UINT	仅用于读/写功能的附加参数，参见 ReadWriteRegs/UserReadWrite [► 27]
pAuxMemoryAddr	BYTE	仅用于读/写功能的附加参数，参见 ReadWriteRegs/UserReadWrite [► 27]
执行	BOOL	开始信号。该操作由执行输入的上升沿触发。
超时	TIME	等待所寻址从站响应的超时时间。

输出

```

VAR_OUTPUT
    BUSY      : BOOL;
    Error     : BOOL;
    ErrorId   : MODBUS_ERRORS;
    cbRead    : UINT;
END_VAR

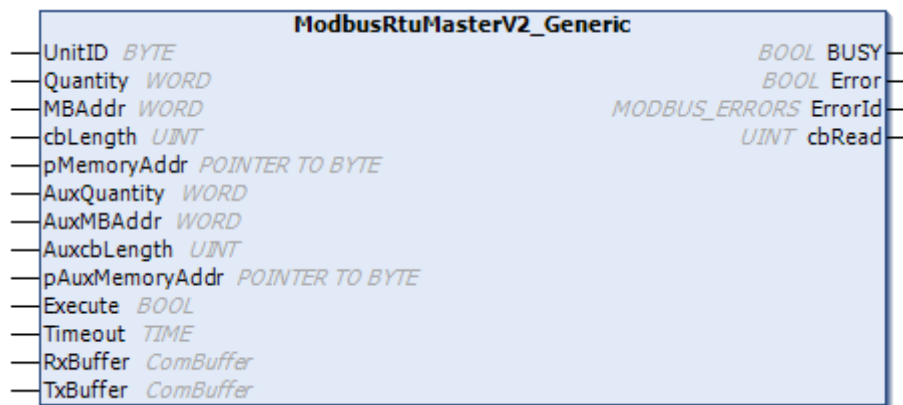
```

名称	类型	描述
繁忙	BOOL	表示功能块处于激活状态。当执行出现上升沿时，Busy 变为 TRUE；一旦启动的操作执行完毕，Busy 将重新变为 FALSE。在任何时候，只能有一项操作处于活动状态。
错误	BOOL	表示在执行操作过程中发生错误。
ErrorId	MODBUS_ERRORS	指示通信受到干扰或发生故障时的错误编号 [► 37]。
cbRead	UINT	提供读取操作的读取数据字节数。

要求

开发环境	目标平台	要包括的 PLC 库
TwinCAT v3.1.0	PC 或 CX (x86、x64、Arm®)	Tc2_ModbusRTU

5.1.5 ModbusRtuMasterV2_Generic



功能块 `ModbusRtuMasterV2_Generic` 可实现 Modbus 主站，通过各种串行接口（COM 端口、虚拟 COM 端口、EtherCAT 端子模块等）进行通信。

由于 `ModbusRtuMasterV2_Generic` 与硬件无关，其使用要比依赖硬件的功能块

`ModbusRtuMasterV2_PcCOM`, `ModbusRtuMasterV2_KL6x22B`, `ModbusRtuMasterV2_KL6x5B` 复杂一些。所有功能块都具有相同的 ModbusRTU 功能。不过，仅 `ModbusRtuMasterV2_Generic` 本身即可支持使用虚拟 COM 端口。

● 通过 TF6340 TwinCAT 3 串行通信（需要授权）连接至硬件

1 连接至通信端口所需的数据结构必须分别进行实例化。该功能块中类型为 `ComBuffer` 的数据结构，是用于解耦硬件相关后台通信的数据缓冲区。该后台通信必须通过 `Tc2_SerialCom` PLC 库中相应的功能块（如 `SerialLineControl` 或 `SerialLineControlADS`）来实现；为此，必须首先在程序中集成该 PLC 库。还需要 TF6340 TwinCAT 3 串行通信的授权。

该功能块不以其基本形式进行调用，而是在 PLC 程序中使用该功能块内部的单个动作。每个 Modbus 功能都是以动作的形式实现的。

支持的 Modbus 功能（动作）

- **ModbusMaster.ReadCoils**

Modbus 功能 1 = 读取线圈

从连接的从站中读取二进制输出（线圈）。数据以压缩格式（每个字节存储 8 位）从指定的地址 `pMemoryAddr` 开始存储。

- **ModbusMaster.ReadInputStatus**

Modbus 功能 2 = 读取输入状态

从连接的从站中读取二进制输入。数据以压缩格式（每个字节存储 8 位）从指定的地址 `pMemoryAddr` 开始存储。

- **ModbusMaster.ReadRegs**

Modbus 功能 3 = 读取保持寄存器

从连接的从站读取数据。

- **ModbusMaster.ReadInputRegs**

Modbus 功能 4 = 读取输入寄存器

从连接的从站中读取输入寄存器。

- **ModbusMaster.WriteSingleCoil**

Modbus 功能 5 = 写入单线圈

向连接的从站发送一个二进制输出（线圈）。数据必须以压缩形式（每字节 8 位）从指定地址 `pMemoryAddr` 发送。

- **ModbusMaster.WriteSingleRegister**

Modbus 功能 6 = 写入单寄存器

向连接的从站发送一个单个数据字

- **ModbusMaster.WriteMultipleCoils**

Modbus 功能 15 = 写入多个线圈

向连接的从站发送二进制输出（线圈）。数据必须以压缩形式（每字节 8 位）从指定地址 pMemoryAddr 发送。

- **ModbusMaster.WriteRegs**

Modbus 功能 16 = 预置多寄存器

向连接的从站发送数据

- **ModbusMaster.Diagnostics**

Modbus 功能 8 = 诊断

向连接的从站发送诊断请求，并附带用户指定的子功能码。由于该功能（诊断）并不访问内存地址，因此其功能代码是在数据字 MAddr 中传输。该功能所需的任何数据都包含在 pMemoryAddr 中。

MasterV2 功能块支持的 Modbus 功能（操作）

- **ModbusMaster.ReadWriteRegs**

Modbus 功能 23 = 读/写多个寄存器

将通过 Aux 参数指定的数据发送至连接的从站，并同时从该从站接收数据。接收到的数据存储在 pMemoryAddr 指定的地址中。

- **ModbusMaster.UserReadWrite**

通用用户报文

用户在指定数据 (pMemoryAddr) 的第一个字节中指定 Modbus 功能码。通过该功能，用户能够发送带有任何功能码的 Modbus 报文。从从站接收到的任何数据都存储在 pAuxMemoryAddr 指定的地址中。

输入

```
VAR_INPUT
    UnitID      : BYTE;
    Quantity    : WORD;
    MAddr       : WORD;
    cbLength    : UINT;
    pMemoryAddr : POINTER TO BYTE;

    AuxQuantity : WORD;
    AuxMAddr    : WORD;
    AuxcbLength : UINT;
    pAuxMemoryAddr : POINTER TO BYTE;

    Execute     : BOOL;
    Timeout     : TIME;
END_VAR
```

名称	类型	描述
单位标识	UINT	Modbus 站地址 [► 37] (1..247)。Modbus 从站只有在接收到包含其自身站地址的报文时才会做出响应。或者，也可选择使用广播地址来回复任何请求。地址 0 保留用于广播报文，因此不是有效的站地址（见 UnitID [► 37]）
数量	WORD	对于寻址的 Modbus 功能，该参数指定了需要读取或写入的数据字数量。对于位寻址的 Modbus 功能，数量参数指定了需要读取或写入的位数量（输入或线圈）。
MBAAddr	WORD	Modbus 数据地址：指从终端设备（从站）读取数据时所指向的起始地址。该地址会原封不动地传输至从站，并在从站端被解释为数据地址。 但在执行诊断功能 (8) 时，此位置传输的是功能代码（即子功能码）。
cbLength	UINT	用于发送或读取动作的数据变量的大小（以字节为单位）。cbLength 必须大于或等于由数量决定的实际传输数据量。对于字访问，例如 [cbLength >= Quantity * 2]。cbLength 可以通过 SIZEOF(Modbus 数据) 计算得出。
pMemoryAddr	BYTE	PLC 中的内存地址，通过 ADR (Modbus 数据) 计算得出。对于读取操作，读取到的数据将存储在所指定的变量中。对于发送操作，数据将从指定的变量传输至终端设备。
AuxQuantity	WORD	仅用于读/写功能的附加参数，参见 ReadWriteRegs/UserReadWrite [► 30]
AuxMBAAddr	WORD	仅用于读/写功能的附加参数，参见 ReadWriteRegs/UserReadWrite [► 30] 。
AuxcbLength	UINT	仅用于读/写功能的附加参数，参见 ReadWriteRegs/UserReadWrite [► 30]
pAuxMemoryAddr	BYTE	仅用于读/写功能的附加参数，参见 ReadWriteRegs/UserReadWrite [► 30]
执行	BOOL	开始信号。该操作由执行输入的上升沿触发。
超时	TIME	等待所寻址从站响应的超时时间。

输入/输出

```

VAR_IN_OUT
  RxBuffer      : ComBuffer;
  TxBuffer      : ComBuffer;
END_VAR

```

名称	类型	描述
TxBuffer	ComBuffer (Tc2_SerialCom PLC 库)	为正在使用的串行硬件提供发送数据的缓冲区。此数据缓冲区严禁由用户直接进行读写操作，它仅作为通信功能块内部使用的数据缓冲区。后台通信必须通过 Tc2_SerialCom PLC 库中相应的功能块来实现。
RxBuffer	ComBuffer (Tc2_SerialCom PLC 库)	放入接收数据的缓冲区。此数据缓冲区严禁由用户直接进行读写操作，它仅作为通信功能块内部使用的数据缓冲区。后台通信必须通过 Tc2_SerialCom PLC 库中相应的功能块来实现。

输出

```

VAR_OUTPUT
  BUSY      : BOOL;
  Error     : BOOL;
  ErrorId   : MODBUS_ERRORS;
  cbRead    : UINT;
END_VAR

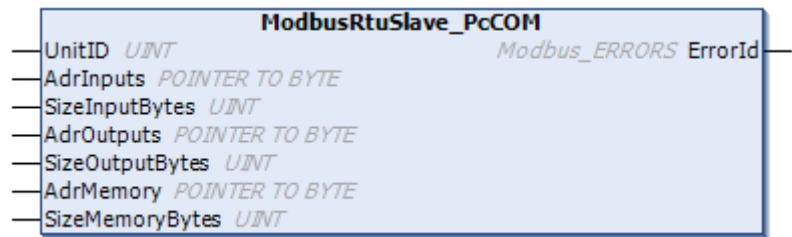
```

名称	类型	描述
繁忙	BOOL	表示功能块处于激活状态。当执行出现上升沿时，Busy 变为 TRUE；一旦启动的操作执行完毕，Busy 将重新变为 FALSE。在任何时候，只能有一项操作处于活动状态。
错误	BOOL	表示在执行操作过程中发生错误。
ErrorId	MODBUS_ERRORS	指示通信受到干扰或发生故障时的错误编号 [► 37]。
cbRead	UINT	提供读取操作的读取数据字节数。

要求

开发环境	目标平台	要包括的 PLC 库
TwinCAT v3.1.4024.0	PC 或 CX (x86、x64、Arm®)	Tc2_ModbusRTU (>= v3.5.6.0)

5.1.6 ModbusRtuSlave_PcCOM



功能块 ModbusRtuSlave_PcCOM 实现了通过串行 PC 接口（COM 端口）进行通信的 Modbus 从站。功能块 ModbusRtuSlave_KL6x5B [► 34] 可通过串行总线端子模块 KL6001、KL6011 或 KL6021 进行通信。

该功能块在接收到来自连接的 Modbus 主站的报文之前，一直处于被动状态。示例程序解释了运行原理。

● 硬件连接

i 与通信端口连接所需的数据结构已集成在该功能块中。一旦 PLC 程序集成后，它们就会显示在 TwinCAT System Manager 中，并可以与 COM 端口进行连接。操作步骤类似于 TF6340 TwinCAT 3 串行通信文档中串行 PC 接口一章的描述。

🔧 输入

```
VAR_INPUT
    UnitID      : UINT;
    AdrInputs    : POINTER TO BYTE; (* Pointer to the Modbus input area *)
    SizeInputBytes : UINT;
    AdrOutputs    : POINTER TO BYTE; (* Pointer to the Modbus output area *)
    SizeOutputBytes : UINT;
    AdrMemory     : POINTER TO BYTE; (* Pointer to the Modbus memory area *)
    SizeMemoryBytes : UINT;
END_VAR
```

名称	类型	描述
UnitID	UINT	Modbus 站地址 [► 37] (1..247)。Modbus 从站只有在接收到包含其自身站地址的报文时才会做出响应。或者，也可选择使用广播地址来回复任何请求。地址 0 预留用于广播报文，因此不是一个有效的站地址。
AdrInputs	BYTE	Modbus 输入区域 [► 12] 的起始地址。数据区通常声明为 PLC 数组，其地址可以通过 ADR 指令（输入变量）计算得出。
SizeInputBytes	UINT	Modbus 输入数组的大小，以字节为单位。大小可以通过 SIZEOF (输入变量) 计算得出。
AdrOutputs	BYTE	Modbus 输出区域 [► 12] 的起始地址。数据区通常声明为 PLC 数组，其地址可以通过 ADR 指令（输出变量）获取。
SizeOutputBytes	UINT	Modbus 输出数组的大小，以字节为单位。大小可以通过 SIZEOF (输出变量) 计算得出。
AdrMemory	BYTE	Modbus 内存区域 [► 12] 的起始地址。数据区通常声明为 PLC 数组，其地址可以通过 ADR 指令（针对内存变量）计算得出。
SizeMemoryBytes	UINT	Modbus 内存数组的大小（以字节为单位）。大小可以通过 SIZEOF (内存变量) 计算得出。

输出

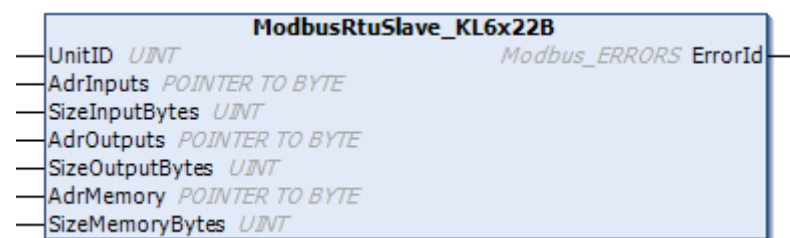
```
VAR_OUTPUT
    ErrorId : MODBUS_ERRORS;
END_VAR
```

名称	类型	描述
ErrorId	MODBUS_ERRORS	指示通信受到干扰或发生故障时的错误编号 [► 37]。

要求

开发环境	目标平台	要包括的 PLC 库
TwinCAT v3.1.0	PC 或 CX (x86、x64、Arm®)	Tc2_ModbusRTU

5.1.7 ModbusRtuSlave_KL6x22B



功能模块 `ModbusRtuSlave_KL6x22B` 可实现 Modbus 从站，通过串行总线端子模块 KL6031 或 KL6041 进行通信。也支持具有 22 字节进程映像数据的串行 EtherCAT 端子模块。功能块 `ModbusRtuSlave_PcCOM` [► 32] 可用于通过串行 PC 接口（COM 端口）进行通信。

该功能块在接收到来自连接的 Modbus 主站的报文之前，一直处于被动状态。

● 硬件连接



与通信端口连接所需的数据结构已集成在该功能块中。在 PC 上，TwinCAT 系统管理器中的分配类似于 TF6340 TwinCAT 3 串行通信文档中串行总线端子模块章节中的描述。

输入

```
VAR_INPUT
    UnitID      : UINT;
    AdrInputs   : POINTER TO BYTE; (* Pointer to the Modbus input area *)
    SizeInputBytes : UINT;
    AdrOutputs  : POINTER TO BYTE; (* Pointer to the Modbus output area *)
    SizeOutputBytes : UINT;
    AdrMemory   : POINTER TO BYTE; (* Pointer to the Modbus memory area *)
    SizeMemoryBytes : UINT;
END_VAR
```

名称	类型	描述
UnitID	UINT	Modbus 站地址 [► 37] (1..247)。Modbus 从站只有在接收到包含其自身站地址的报文时才会做出响应。或者，也可选择使用广播地址来回复任何请求。地址 0 预留用于广播报文，因此不是一个有效的站地址。
AdrInputs	BYTE	Modbus 输入区域 [► 12] 的起始地址。数据区通常声明为 PLC 数组，其地址可以通过 ADR 指令（输入变量）计算得出。
SizeInputBytes	UINT	Modbus 输入数组的大小，以字节为单位。大小可以通过 SIZEOF (输入变量) 计算得出。
AdrOutputs	BYTE	Modbus 输出区域 [► 12] 的起始地址。数据区通常声明为 PLC 数组，其地址可以通过 ADR 指令（输出变量）获取。
SizeOutputBytes	UINT	Modbus 输出数组的大小，以字节为单位。大小可以通过 SIZEOF (输出变量) 计算得出。
AdrMemory	BYTE	Modbus 内存区域 [► 12] 的起始地址。数据区通常声明为 PLC 数组，其地址可以通过 ADR 指令（针对内存变量）计算得出。
SizeMemoryBytes	UINT	Modbus 内存数组的大小（以字节为单位）。大小可以通过 SIZEOF (内存变量) 计算得出。

输出

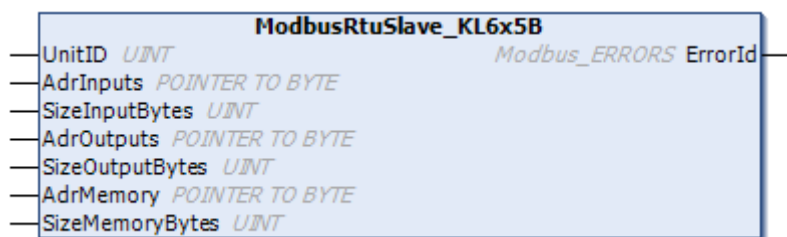
```
VAR_OUTPUT
    ErrorId : MODBUS_ERRORS;
END_VAR
```

名称	类型	描述
ErrorId	MODBUS_ERRORS	指示通信受到干扰或发生故障时的错误编号 [► 37]。

要求

开发环境	目标平台	要包括的 PLC 库
TwinCAT v3.1.0	PC 或 CX (x86、x64、Arm®)	Tc2_ModbusRTU

5.1.8 ModbusRtuSlave_KL6x5B



功能块 `ModbusRTUslave_KL6x5B` 可实现 Modbus 从站功能，通过串行总线端子模块 KL6001、KL6011 或 KL6021 进行通信。功能块 `ModbusRtuSlave_PcCOM` [► 32] 可用于通过串行 PC 接口（COM 端口）进行通信。

该功能块在接收到来自连接的 Modbus 主站的报文之前，一直处于被动状态。



硬件连接

与通信端口连接所需的数据结构已集成在该功能块中。在 PC 上，TwinCAT 系统管理器中的分配类似于 TF6340 TwinCAT 3 串行通信文档中[串行总线端子模块](#)章节中的描述。

输入

```
VAR_INPUT
  UnitID      : UINT;
  AdrInputs   : POINTER TO BYTE; (* Pointer to the Modbus input area *)
  SizeInputBytes : UINT;
  AdrOutputs  : POINTER TO BYTE; (* Pointer to the Modbus output area *)
  SizeOutputBytes : UINT;
  AdrMemory   : POINTER TO BYTE; (* Pointer to the Modbus memory area *)
  SizeMemoryBytes : UINT;
END_VAR
```

名称	类型	描述
UnitID	UINT	Modbus 站地址 [► 37] (1..247)。Modbus 从站只有在接收到包含其自身站地址的报文时才会做出响应。或者，也可选择使用广播地址来回复任何请求。地址 0 预留用于广播报文，因此不是一个有效的站地址。
AdrInputs	BYTE	Modbus 输入区域 [► 12]的起始地址。数据区通常声明为 PLC 数组，其地址可以通过 ADR 指令（输入变量）计算得出。
SizeInputBytes	UINT	Modbus 输入数组的大小，以字节为单位。大小可以通过 SIZEOF (输入变量) 计算得出。
AdrOutputs	BYTE	Modbus 输出区域 [► 12]的起始地址。数据区通常声明为 PLC 数组，其地址可以通过 ADR 指令（输出变量）获取。
SizeOutputBytes	UINT	Modbus 输出数组的大小，以字节为单位。大小可以通过 SIZEOF (输出变量) 计算得出。
AdrMemory	BYTE	Modbus 内存区域 [► 12]的起始地址。数据区通常声明为 PLC 数组，其地址可以通过 ADR 指令（针对内存变量）计算得出。
SizeMemoryBytes	UINT	Modbus 内存数组的大小（以字节为单位）。大小可以通过 SIZEOF（内存变量）计算得出。

输出

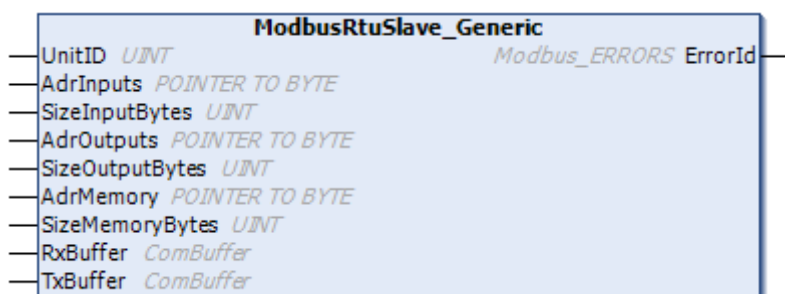
```
VAR_OUTPUT
  ErrorId : MODBUS_ERRORS;
END_VAR
```

名称	类型	描述
ErrorId	MODBUS_ERRORS	指示通信受到干扰或发生故障时的错误编号 [► 37]。

要求

开发环境	目标平台	要包括的 PLC 库
TwinCAT v3.1.0	PC 或 CX (x86、x64、Arm®)	Tc2_ModbusRTU

5.1.9 ModbusRtuSlave_Generic



功能块 `ModbusRtuSlave_Generic` 实现了一个 Modbus 从站，可通过各种串行接口（COM 端口、虚拟 COM 端口、EtherCAT 端子模块等）进行通信。

由于 `ModbusRtuSlave_Generic` 与硬件无关，其使用要比依赖硬件的功能块 `ModbusRtuSlave_PcCOM`, `ModbusRtuSlave_KL6x22B`, `ModbusRtuSlave_KL6x5B` 复杂一些。所有功能块都具有相同的 ModbusRTU 功能。不过，仅 `ModbusRtuSlave_Generic` 本身即可支持使用虚拟 COM 端口。

该功能块在接收到来自连接的 Modbus 主站的报文之前，一直处于被动状态。示例程序解释了运行原理。

● 通过 TF6340 TwinCAT 3 串行通信（需要授权）连接至硬件

i 连接至通信端口所需的数据结构必须分别进行实例化。该功能块中类型为 `ComBuffer` 的数据结构，是用于解耦硬件相关后台通信的数据缓冲区。该后台通信必须通过 `Tc2_SerialCom` PLC 库中相应的功能块（如 `SerialLineControl` 或 `SerialLineControlADS`）来实现；为此，必须首先在程序中集成该 PLC 库。还需要 TF6340 TwinCAT 3 串行通信的授权。

📁 输入

```

VAR_INPUT
    UnitID      : UINT;
    AddrInputs   : POINTER TO BYTE; (* Pointer to the Modbus input area *)
    SizeInputBytes : UINT;
    AddrOutputs  : POINTER TO BYTE; (* Pointer to the Modbus output area *)
    SizeOutputBytes : UINT;
    AddrMemory   : POINTER TO BYTE; (* Pointer to the Modbus memory area *)
    SizeMemoryBytes : UINT;
END_VAR

```

名称	类型	描述
UnitID	UINT	Modbus 站地址 [► 37] (1..247)。Modbus 从站只有在接收到包含其自身站地址的报文时才会做出响应。或者，也可选择使用广播地址来回复任何请求。地址 0 预留用于广播报文，因此不是一个有效的站地址。
AddrInputs	BYTE	Modbus 输入区域 [► 12] 的起始地址。数据区通常声明为 PLC 数组，其地址可以通过 ADR 指令（输入变量）计算得出。
SizeInputBytes	UINT	Modbus 输入数组的大小，以字节为单位。大小可以通过 SIZEOF (输入变量) 计算得出。
AddrOutputs	BYTE	Modbus 输出区域 [► 12] 的起始地址。数据区通常声明为 PLC 数组，其地址可以通过 ADR 指令（输出变量）获取。
SizeOutputBytes	UINT	Modbus 输出数组的大小，以字节为单位。大小可以通过 SIZEOF (输出变量) 计算得出。
AddrMemory	BYTE	Modbus 内存区域 [► 12] 的起始地址。数据区通常声明为 PLC 数组，其地址可以通过 ADR 指令（针对内存变量）计算得出。
SizeMemoryBytes	UINT	Modbus 内存数组的大小（以字节为单位）。大小可以通过 SIZEOF (内存变量) 计算得出。

输入/输出

```

VAR_IN_OUT
  RxBuffer      : ComBuffer;
  TxBuffer      : ComBuffer;
END_VAR

```

名称	类型	描述
TxBuffer	ComBuffer (Tc2_SerialCom PLC 库)	为正在使用的串行硬件提供发送数据的缓冲区。此数据缓冲区严禁由用户直接进行读写操作，它仅作为通信功能块内部使用的数据缓冲区。后台通信必须通过 Tc2_SerialCom PLC 库中相应的功能块来实现。
RxBuffer	ComBuffer (Tc2_SerialCom PLC 库)	放入接收数据的缓冲区。此数据缓冲区严禁由用户直接进行读写操作，它仅作为通信功能块内部使用的数据缓冲区。后台通信必须通过 Tc2_SerialCom PLC 库中相应的功能块来实现。

输出

```

VAR_OUTPUT
  ErrorId : MODBUS_ERRORS;
END_VAR

```

名称	类型	描述
ErrorId	MODBUS_ERRORS	指示通信受到干扰或发生故障时的错误编号 [► 37]。

要求

开发环境	目标平台	要包括的 PLC 库
TwinCAT v3.1.4024.0	PC 或 CX (x86、x64、Arm®)	Tc2_ModbusRTU (>= v3.5.6.0)

5.2 Datatypes

5.2.1 MODBUS_UNITID

Modbus 定义的有效站地址范围为 1 至 247。Modbus 从站只响应包含自身地址的报文。地址 0 不是有效的站地址。它用于向所有站发送广播报文。这些都没有得到答复。地址 248 至 255 被保留。

Tc2_ModbusRTU 库定义了更多的集合地址。这样，一个站就能对多个地址做出响应。

```

TYPE MODBUS_UNITID :
(
  MODBUS_UNITID_BROADCAST      := 0,
  MODBUS_UNITID_ALLVALID       := 256, (* response on address 1..247 *)
  MODBUS_UNITID_ALLBROADCAST   := 257, (* response on address 1..255 *)
  MODBUS_UNITID_ALL            := 258 (* response on address 0..255 *)
);
END_TYPE

```

要求

开发环境	目标平台	要包括的 PLC 库
TwinCAT v3.1.0	PC 或 CX (x86、x64、Arm®)	Tc2_ModbusRTU

5.2.2 MODBUS_ERRORS

```

TYPE MODBUS_ERRORS :
(
  (* Modbus communication errors *)
  MODBUSERROR_NO_ERROR, (* 0 *)
  MODBUSERROR_ILLEGAL_FUNCTION, (* 1 *)
  MODBUSERROR_ILLEGAL_DATA_ADDRESS, (* 2 *)
  MODBUSERROR_ILLEGAL_DATA_VALUE, (* 3 *)
  MODBUSERROR_SLAVE_DEVICE_FAILURE, (* 4 *)
);

```

```

MODBUSERROR_ACKNOWLEDGE, (* 5 *)
MODBUSERROR_SLAVE_DEVICE_BUSY, (* 6 *)
MODBUSERROR_NEGATIVE_ACKNOWLEDGE, (* 7 *)
MODBUSERROR_MEMORY_PARITY, (* 8 *)
MODBUSERROR_GATEWAY_PATH_UNAVAILABLE, (* A *)
MODBUSERROR_GATEWAY_TARGET_DEVICE_FAILED_TO_RESPOND, (* B *)

(* additional Modbus error definitions *)
MODBUSERROR_CHARREC_TIMEOUT := 16#20, (* 20 hex *)
MODBUSERROR_ILLEGAL_DATA_SIZE, (* 21 hex *)
MODBUSERROR_ILLEGAL_DEVICE_ADDRESS, (* 22 hex *)
MODBUSERROR_ILLEGAL_DESTINATION_ADDRESS, (* 23 hex *)
MODBUSERROR_ILLEGAL_DESTINATION_SIZE, (* 24 hex *)
MODBUSERROR_NO_RESPONSE, (* 25 hex *)

(* Low level communication errors *)
MODBUSERROR_TXBUFFOVERRUN := 102, (* 102 *)
MODBUSERROR_SENDTIMEOUT := 103, (* 103 *)
MODBUSERROR_DATASIZEOVERRUN := 107, (* 107 *)
MODBUSERROR_STRINGOVERRUN := 110, (* 110 *)
MODBUSERROR_INVALIDPOINTER := 120, (* 120 *)
MODBUSERROR_CRC := 150, (* 150 *)

(* High level PLC errors *)
MODBUSERROR_INVALIDMEMORYADDRESS := 232, (* 232 *)
MODBUSERROR_TRANSMITBUFFERTOOSMALL (* 233 *)
);
END_TYPE

```

5.3 全局常量

5.3.1 Global_Version

所有库都有特定版本。例如，版本信息会在 PLC 功能库中标注。全局常量包含有关库版本的信息：

```

VAR GLOBAL CONSTANT
stLibVersion_Tc2_Modbus_RTU : ST_LibVersion;
END_VAR

```

要检查您拥有的版本是否是所需的版本，请使用功能 F_CmpLibVersion（在 Tc2_System PLC 库中定义）。



您可能从 TwinCAT 2 中了解到的用于比较库版本的所有其他选项都已过时！

6 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

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

Arm, Arm9 and Cortex are trademarks or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere.

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

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Modbus is a registered trademark of Schneider Electric USA, Inc.

更多信息:
www.beckhoff.com/tf6255

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

