

**BECKHOFF** New Automation Technology

文件资料 | ZH

EL600x, EL602x

串行通讯接口模块





# 目录

<b>1 串口端子模块概述</b> .....	<b>7</b>
<b>2 前言</b> .....	<b>8</b>
2.1 文档说明 .....	8
2.2 文档指南 .....	9
2.3 安全说明 .....	10
2.4 文件发行状态 .....	11
2.5 EtherCAT 设备的版本标识 .....	12
2.5.1 关于标识的一般说明.....	12
2.5.2 EL 端子模块的版本标识.....	13
2.5.3 倍福识别码 (BIC) .....	14
2.5.4 BIC 电子读取 (eBIC) .....	16
<b>3 产品描述</b> .....	<b>18</b>
3.1 EL6001、EL6021 .....	21
3.1.1 简介.....	21
3.1.2 技术数据.....	22
3.2 EL6021-0021 .....	23
3.2.1 简介.....	23
3.2.2 技术数据.....	24
3.3 EL6002、EL6022 .....	25
3.3.1 简介.....	25
3.3.2 技术数据.....	26
3.4 启动 .....	27
<b>4 基本通讯</b> .....	<b>28</b>
4.1 EtherCAT 基础知识 .....	28
4.2 EtherCAT 布线 - 线缆连接 .....	28
4.3 设置看门狗的一般注意事项 .....	29
4.4 EtherCAT 状态机 .....	30
4.5 CoE 接口 .....	32
4.6 分布时钟 (Distributed Clock) .....	37
<b>5 安装和布线</b> .....	<b>38</b>
5.1 静电防护的说明 .....	38
5.2 防爆 .....	39
5.2.1 ATEX - 特殊条件 (标准温度范围).....	39
5.2.2 ATEX - 特殊条件 (扩展温度范围).....	40
5.2.3 IECEx - 特殊条件.....	41
5.2.4 ATEX 和 IECEx 的持续性文件.....	42
5.2.5 cFMus - 特殊条件.....	43
5.2.6 cFMus 防爆认证的持续性文件.....	43
5.3 UL 声明 .....	44
5.4 无通讯模块的安装位置 .....	45
5.5 增强抗振模块的安装说明 .....	46
5.6 安装位置 .....	47
5.7 注意事项 - 电源 .....	49

5.8	EL6001、EL6021-00xx .....	50
5.8.1	安装在导轨上.....	50
5.8.2	接线.....	53
5.8.3	LED 指示灯和引脚分配.....	57
5.9	EL6002、EL6022 .....	60
5.9.1	安装和拆卸 - 正面带拆卸手柄的端子模块.....	60
5.9.2	LED 指示灯 和引脚分配.....	62
5.10	处理.....	65
<b>6</b>	<b>调试.....</b>	<b>66</b>
6.1	TwinCAT 快速入门.....	66
6.1.1	TwinCAT 2.....	68
6.1.2	TwinCAT 3.....	78
6.2	TwinCAT 开发环境.....	91
6.2.1	TwinCAT real-time 实时驱动程序的安装.....	92
6.2.2	关于 ESI 设备描述文件的说明.....	98
6.2.3	TwinCAT ESI Updater.....	101
6.2.4	Online 和 Offline 之间的区别.....	101
6.2.5	创建 OFFLINE 配置.....	102
6.2.6	创建ONLINE配置.....	107
6.2.7	EtherCAT 设备的配置.....	114
6.2.8	导入/导出 EtherCAT 设备为 SCI 和 XTI 文件.....	123
6.3	EtherCAT 从站的一般调试说明.....	129
6.4	运行模式和过程数据.....	137
6.5	关于 TeVirtualComDriver 的提示信息.....	145
6.6	通信功能.....	147
6.7	EL6001 的 LIN 主站功能.....	149
6.8	示例程序.....	152
6.8.1	示例程序 1.....	152
6.8.2	示例程序 2.....	155
6.8.3	示例程序 3 (LIN).....	157
<b>7</b>	<b>EL6001、EL6021 的 CoE 对象概览.....</b>	<b>160</b>
7.1	控制字 Control 和状态字 Status.....	160
7.2	对象描述和参数化.....	163
7.2.1	用于调试的对象.....	163
7.2.2	标准对象 (0x1000-0x1FFF).....	165
7.2.3	描述文件定义的对象 (0x6000-0xFFFF) [硬件版本 03 及以上].....	181
<b>8</b>	<b>EL6002、EL6022 的 CoE 对象概览.....</b>	<b>184</b>
8.1	对象描述和参数化.....	184
8.1.1	调试对象.....	184
8.1.2	标准对象 (0x1000-0x1FFF).....	185
8.1.3	描述文件定义的对象 (0x6000-0xFFFF) [硬件版本 03 及以上].....	196
8.2	控制字和状态字.....	199
<b>9</b>	<b>附录.....</b>	<b>201</b>
9.1	EtherCAT AL 状态代码.....	201
9.2	固件兼容性.....	202

9.3	固件更新 EL/ES/EM/ELM/EPxxxx .....	204
9.3.1	设备描述 ESI 文件/XML.....	205
9.3.2	Firmware (固件) 说明.....	208
9.3.3	更新从站处理器的固件 *.efw.....	208
9.3.4	FPGA 固件 *.rbf.....	210
9.3.5	同时更新多个 EtherCAT 设备.....	214
9.4	恢复出厂状态 .....	215
9.5	技术支持和服务 .....	217



# 1 串口端子模块概述

[EL6001 \[▶ 21\]](#) (1 通道串口端子模块, RS232C)

[EL6021 \[▶ 21\]](#) (1 通道串口端子模块, RS422/RS485)

[EL6021-0021 \[▶ 23\]](#) (1 通道串口端子模块, RS422/RS485), 用于串行线路中部

[EL6002 \[▶ 25\]](#) (2 通道串口端子模块, RS232C)

[EL6022 \[▶ 25\]](#) (2 通道串口端子模块, RS422/RS485)

## 2 前言

### 2.1 文档说明

#### 目标受众

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®、TwinCAT®、TwinCAT/BSD®、TC/BSD®、EtherCAT®、EtherCAT G®、EtherCAT G10®、EtherCAT P®、Safety over EtherCAT®、TwinSAFE®、XFC®、XTS® 和 XPlanar® 是倍福自动化有限公司的注册商标并得到授权。本出版物中使用的其他名称可能是商标，第三方出于自身目的使用它们可能侵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

© 德国倍福自动化有限公司。

未经明确授权，禁止复制、分发和使用本文件以及将其内容传达给他人。  
违者将被追究赔偿责任。在专利授权、工具型号或设计方面保留所有权利。



## 2.2 文档指南

### 注意



#### 文件的其它组成部分

本文档介绍特定设备的内容。它是倍福 I/O 组件模块化文档体系的一部分。为了使用和安全操作本文档中描述的设备/装置，还需要阅读其它跨产品说明，请参见下表。

标题	描述
EtherCAT 系统文档 (PDF)	<ul style="list-style-type: none"> <li>• 系统概览</li> <li>• EtherCAT 基础知识</li> <li>• 电缆冗余</li> <li>• 热连接</li> <li>• EtherCAT 设备配置</li> </ul>
端子模块系统的防爆保护 (PDF)	根据 ATEX 和 IECEx 标准，在防爆区使用倍福端子模块系统的注意事项
控制图 I/O、CX、CPX (PDF)	接线图和防爆标记（符合 cFMus 标准）
EtherCAT/Ethernet 基础设施 (PDF)	关于设计、实施和测试的技术建议和注意事项
I/O 软件声明 (PDF)	倍福 I/O 组件的开源软件声明

可以在倍福公司网站 ([www.beckhoff.com](http://www.beckhoff.com)) 上通过以下版块查看或下载相关文档：

- 在相应产品页面的“文档和下载”区域，
- [下载中心](#)，
- [Beckhoff Information System](#)。

## 2.3 安全说明

### 安全规范

请注意以下安全说明和解释！  
可在以下页面或安装、接线、调试等区域找到产品相关的安全说明。

### 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。禁止未按文档所述修改硬件或软件配置，德国倍福自动化有限公司不对此承担责任。

### 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。

## 2.4 文件发行状态

版本	注释
5.6	<ul style="list-style-type: none"> <li>更新“对象描述和参数化”章节</li> <li>更新结构</li> <li>更新修订状态</li> </ul>
5.5	<ul style="list-style-type: none"> <li>删除 EL6002、EL6022 “推荐的安装导轨”章节</li> <li>更新“技术数据”章节</li> <li>新增“ATEX - 特殊条件（标准温度范围）”章节</li> <li>更新修订状态</li> </ul>
5.4	<ul style="list-style-type: none"> <li>新增 EL6021-0021</li> <li>更新“产品描述”章节</li> <li>更新结构</li> <li>更新修订状态</li> </ul>
5.3	<ul style="list-style-type: none"> <li>更新“调试”章节</li> <li>更新结构</li> <li>更新修订状态</li> </ul>
5.2	<ul style="list-style-type: none"> <li>更新“对象描述和参数化”章节</li> <li>更新结构</li> <li>更新修订状态</li> </ul>
5.1	<ul style="list-style-type: none"> <li>更新“EL6001 的 LIN 主站功能”章节</li> <li>更新结构</li> <li>更新修订状态</li> </ul>
5.0	<ul style="list-style-type: none"> <li>更新“技术数据”章节</li> <li>更新“传输速率优化”章节</li> <li>更新结构</li> </ul>
4.9	<ul style="list-style-type: none"> <li>更新“技术数据”章节</li> <li>更新“安装和布线”章节</li> <li>更新结构</li> <li>更新修订状态</li> </ul>
4.8	<ul style="list-style-type: none"> <li>更新“基本功能原理”章节</li> <li>更新结构</li> <li>更新修订状态</li> </ul>
4.7	<ul style="list-style-type: none"> <li>更新“对象描述和参数化”章节</li> <li>更新结构</li> <li>更新修订状态</li> </ul>
4.6	<ul style="list-style-type: none"> <li>更正 RS232 信号电平</li> <li>更新“技术数据”章节</li> <li>更新结构</li> <li>更新修订状态</li> </ul>
4.5	<ul style="list-style-type: none"> <li>更新“调试”章节</li> <li>更新修订状态</li> </ul>
0.1 - 4.4	*存档*

## 2.5 EtherCAT 设备的版本标识

### 2.5.1 关于标识的一般说明

#### 名称

一个倍福 EtherCAT 设备有一个 14 位字符编号，由以下部分组成

- 系列号
- 型号
- 版本号
- 修订版本号

示例	系列号	型号	版本	修订版本号
EL3314-0000-0016	EL 端子模块 (12 mm, 不可插拔式前连接件)	3314 (4 通道热电偶端子模块)	0000 (基本型号)	0016
ES3602-0010-0017	ES 端子模块 (12 mm, 可插拔式前连接件)	3602 (2 通道电压测量模块)	0010 (高精度版本)	0017
CU2008-0000-0000	CU 设备	2008 (8 端口高速以太网交换机)	0000 (基本型号)	0000

#### 注意

- 上述要素构成了**技术编号**。下面使用 EL3314-0000-0016来举例说明。
- EL3314-0000 是订货号，在“-0000”的情况下，通常简写为 EL3314。“-0016”是 EtherCAT 版本号。
- **订货号**由
  - 系列号 (EL、EP、CU、ES、KL、CX 等)
  - 型号 (3314)
  - 版本号 (-0000) 组成
- **修订版本号** -0016 显示技术改进的版本，例如 EtherCAT 通讯方面的功能扩展，并由倍福公司管理。原则上除非文档中另有规定，较高修订版的设备可以替换装有较低修订版的设备。每个版本通常都有一个XML文件形式的描述(ESI, EtherCAT Slave Information)，可从倍福公司网站下载。  
从 2014 年 01 月起，修订版本号显示在 IP20 端子模块的外壳上，见图“EL5021 EL 端子模块，标准 IP20 IO 设备，带有批号和修订版 ID (从 2014 年 01 月起)”。
- 型号、版本号和修订版本号在读取时当作十进制数字，但它们在存储时按十六进制数字。

## 2.5.2 EL 端子模块的版本标识

倍福 I/O 设备的序列号/数字代码通常是一个印在设备或标签上的 8 位数字。序列号表示交付状态下的配置，因此指的是整个生产批次，不区分批次中的各个模块。

序列号的结构：**KK YY FF HH**

KK - 生产周数 (CW, 日历周)

YY - 生产年份

FF - 固件版本号

HH - 硬件版本号

示例：序列号 12 06 3A 02:

12 - 生产周次为 12 周

06 - 生产年份为 2006 年

3A - 固件版本为 3A

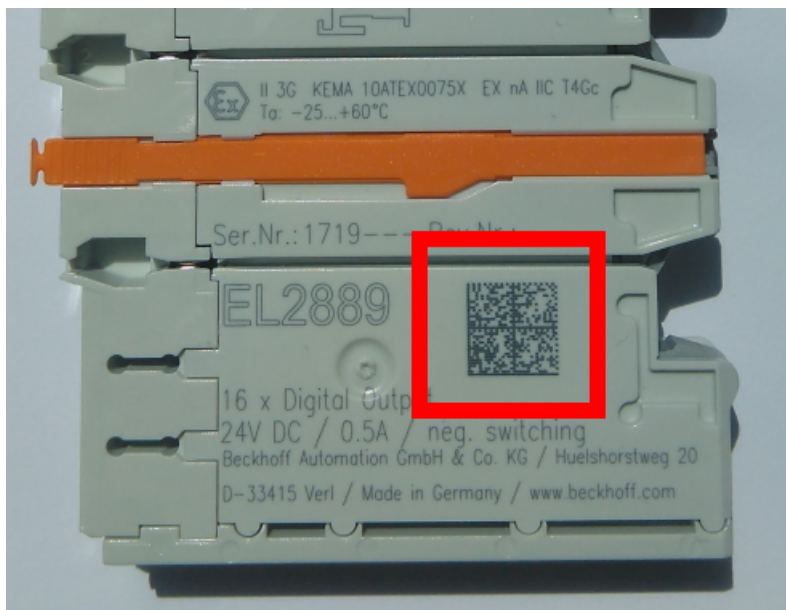
02 - 硬件版本为 02



附图 1: EL2872, 修订版本号 0022 和序列号 01200815

### 2.5.3 倍福识别码 (BIC)

倍福唯一识别码 Beckhoff Identification Code (BIC) 越来越多地应用于识别倍福产品。BIC 表示为二维码 (DMC, 编码格式 ECC200), 内容基于 ANSI 标准 MH10.8.2-2016。



附图 2: BIC 为二维码 (DMC, 编码格式 ECC200)

BIC 将在所有产品组中逐步引入。

根据不同的产品, 可以在以下地方找到:

- 在包装单元上
- 直接在产品上 (如果空间足够)
- 在包装单元和产品上

BIC 可供机器读取, 其中包含的信息客户可以用于产品管理。

每条信息都可以使用数据唯一标识符 (ANSI MH10.8.2-2016) 进行识别。数据标识符后面紧接着是一个字符串。两者加起来的最大长度如下表所示。如果信息较短, 则会以空格填充。

可能出现的信息如下, 位置 1 到 4 总是存在, 其他信息则根据生产的需要而定:

位置	信息类型	说明	数据标识符	包括数据标识符的数字位数	示例
1	倍福订单号	倍福订单号	1P	8	1P072222
2	倍福可追溯性编号 (BTN)	独特的序列号, 见以下说明	SBTN	12	SBTNk4p562d7
3	产品型号	倍福产品型号, 例如 EL1008	1K	32	1KEL1809
4	数量	包装单位的数量, 例如 1、10 等	Q	6	Q1
5	批次号	可选: 生产年份和第几周	2P	14	2P401503180016
6	ID/序列号	可选: 当前的序列号系统, 例如安全产品的序列号系统	51S	12	51S678294
7	型号扩展代码	可选: 基于标准产品的型号扩展代码	30P	32	30PF971, 2*K183
...					

倍福还使用更多类型的信息和数据标识符, 用于内部流程。

## BIC 结构

下面是包含位置 1 - 4及6 的复合信息示例。数据标识符以黑体字突出显示：

1P072222SBTNk4p562d7**1**KEL1809 **Q1** 51S678294

对应的DMC如下：



附图 3: 示例 DMC 1P072222SBTNk4p562d7**1**KEL1809 **Q1** 51S678294

## BTN

BIC 的一个重要组成部分是倍福的可追溯性编号 (BTN, 位置 2)。BTN 是由八个字符组成的唯一序列号, 从长远来看, 它将取代倍福的所有其他序列号系统 (例如, IO 组件上的批号、安全产品之前的系列序列号等)。BTN 也将被逐步引入, 所以可能会出现 BTN 还没有在 BIC 中编码的情况。

### 注意

这些资料经过精心准备, 但是所述流程还在不断优化, 我们保留随时修改流程和文档的权利, 恕不另行通知。不能依据本资料中的信息、插图和描述的修改提出任何要求。

## 2.5.4 BIC 电子读取 (eBIC)

### 电子 BIC (eBIC)

倍福识别码 (BIC) 贴在倍福产品外壳上明显可见的位置。如果可能，其应该也可以通过电子设备读出。

对产品进行电子化处理的接口对于电子读出至关重要。

### K-bus 设备 (IP20、IP67)

目前，没有计划对这些设备的信息进行电子存储和读取。

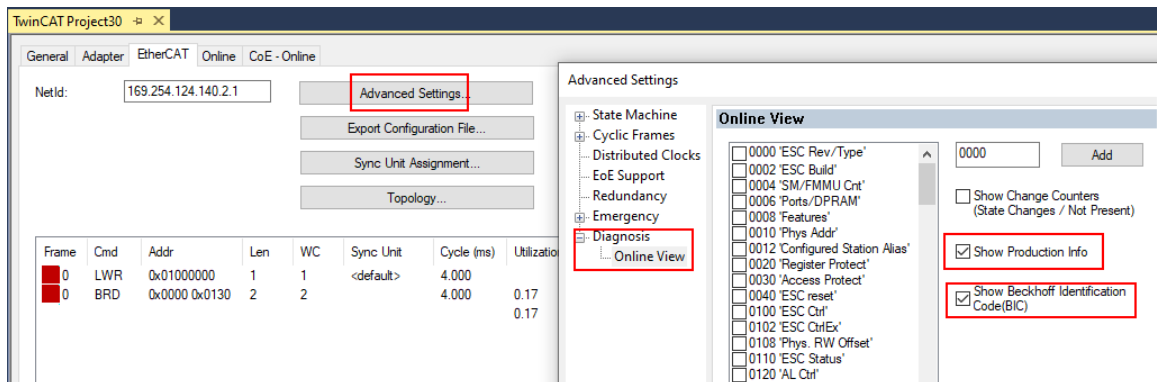
### EtherCAT 设备 (IP20、IP67)

倍福的所有 EtherCAT 设备都有一个 ESI-EEPROM，其中包含 EtherCAT 标识和修订版本号。EtherCAT 从站信息，一般也被称为 EtherCAT 主站的 ESI/XML 配置文件，储存在其中。具体关系请参见 EtherCAT 系统手册中的相应章节（[链接](#)）。

倍福还将 eBIC 存储在 ESI-EEPROM 中。eBIC 于 2020 年引入倍福 IO 生产（端子模块、盒式模块）；截至 2023 年，实施工作已基本完成。

用户可以通过电子方式访问 eBIC（如果存在），具体如下：

- 对于所有 EtherCAT 设备，EtherCAT 主站 (TwinCAT) 可以从 ESI-EEPROM 读出 eBIC
  - TwinCAT 3.1 build 4024.11 及以上版本，在线视图中可以显示 eBIC。
  - 为此，在 EtherCAT → Advanced Settings → Diagnostics 中勾选“Show Beckhoff Identification Code (BIC)”复选框：



- 然后显示 BTN 及其内容：

No	Addr	Name	State	CRC	Fw	Hw	Production Data	ItemNo	BTN	Description	Quantity	BatchNo	SerialNo
1	1001	Term 1 (EK1100)	OP	0.0	0	0	---						
2	1002	Term 2 (EL1018)	OP	0.0	0	0	2020 KW36 Fr	072222	k4p562d7	EL1809	1		678294
3	1003	Term 3 (EL3204)	OP	0.0	7	6	2012 KW24 Sa						
4	1004	Term 4 (EL2004)	OP	0.0	0	0	---	072223	k4p562d7	EL2004	1		678295
5	1005	Term 5 (EL1008)	OP	0.0	0	0	---						
6	1006	Term 6 (EL2008)	OP	0.0	0	12	2014 KW14 Mo						
7	1007	Term 7 (EK1110)	OP	0	1	8	2012 KW25 Mo						

- 注意：从图中可以看出，从 2012 年开始，生产数据包括软件版本、硬件版本和生产日期，也可以用“Show Production Info”来显示。
- 从 PLC 访问：TwinCAT 3.1. build 4024.24 及以上版本起，通过 Tc2 EtherCAT 库的 v3.3.19.0 及以上版本提供功能块 `FB_EcReadBIC` 和 `FB_EcReadBTN` 用于读取数据到 PLC。
- 带有 CoE 目录的 EtherCAT 设备还可以通过对象 `0x10E2:01` 显示自己的 eBIC，PLC 也可以轻松访问这些 eBIC：



- 设备必须处于 PREOP/SAFEOP/OP 状态下才能访问：

Index	Name	Flags	Value
1000	Device type	RO	0x015E1389 (22942601)
1008	Device name	RO	ELM3704-0000
1009	Hardware version	RO	00
100A	Software version	RO	01
100B	Bootloader version	RO	J0.1.27.0
1011:0	Restore default parameters	RO	> 1 <
1018:0	Identity	RO	> 4 <
10E2:0	Manufacturer-specific Identification C...	RO	> 1 <
10E2:01	Subindex 001	RO	1P158442SBTN0008jckp1KELM3704 Q1 2P482001000016
10F0:0	Backup parameter handling	RO	> 1 <
10F3:0	Diagnosis History	RO	> 21 <
10F8	Actual Time Stamp	RO	0x170fb277e

- 对象 0x10E2 将在批量产品的必要固件修订过程中youxian优先引入。
- 此 TwinCAT 3.1. build 4024.24 及以上版本，通过 Tc2\_EtherCAT 库的 v3.3.19.0 及以上版本提供功能块 *FB\_EcCoEReadBIC* 和 *FB\_EcCoEReadBTN* 用于读取数据到 PLC
- 为了在 PLC 中处理 BIC/BTN 数据，截至 TwinCAT 3.1 build 4024.24 版本，*Tc2\_Uutilities* 中提供了以下辅助功能
  - *F\_SplitBIC*：该函数使用已知的标识符将倍福识别代码 (BIC) *sBICValue* 分割成不同的部分，并将识别出的子字符串作为返回值存储在 *ST\_SplitBIC* 结构中
  - *BIC\_TO\_BTN*：该函数从 BIC 中提取 BTN 并将其作为返回值返回
- 注意：如果进行进一步电子处理，BTN 应作为一个字符串 (8) 来处理；标识符“SBTN”不是 BTN 的一部分。
- 技术背景  
在设备生产过程中，新的 BIC 信息被作为一个附加的类别写入 ESI-EEPROM 中。ESI 内容的结构主要由 ETG 规范决定，因此，供应商附加的特定内容是按照 ETG.2010 规定的类别存储的。ID 03 的信息表明，所有 EtherCAT 主站在 ESI 更新时，不得覆盖这些数据，也不得在 ESI 更新后恢复这些数据。该数据的结构依照 BIC 的内容，参见此处。因此，EEPROM 需要大约 50...200 字节的内存。
- 特殊情况
  - 如果一个设备中安装了多个分层排列的 ESC，则只有最上层的 ESC 携带 eBIC 信息。
  - 如果一个设备中安装了多个非分层排列的 ESC，所有 ESC 都携带 eBIC 信息。
  - 如果设备由几个具有自己身份的子设备组成，但只有最上层设备可以通过 EtherCAT 访问，则最上层设备的 eBIC 位于 CoE 对象目录 0x10E2:01，子设备的 eBIC 位于 0x10E2:nn。

## PROFIBUS; PROFINET、和 DeviceNet 设备

目前，没有计划对这些设备的信息进行电子存储和读取。

## 3 产品描述

### 技术

EL600x 和 EL602x 串口端子模块可通过 RS232（或 RS485 / RS422）接口连接设备。EL600x 以全双工模式与控制器交换数据；对于 EL602x，则还可以采用半双工模式。端子模块每个通道有一个接收缓冲区和一个发送缓冲区，参见技术数据。端子模块和控制器之间通过握手进行数据传输。

端子模块的出厂设置为：

- 波特率：9600
- 8N1：8 个数字位，1 个停止位，无奇偶校验位
- 在 EL600x 中，RTS/CTS 控制处于激活状态
- EL602x 采用全双工模式运行，点对点连接处于停用状态。

### 基本原理

在传输多字节数据时，根据编码规范（如 7E2 或 8N1），数据（共  $x$  个字节或  $8*x$  位）以包含 7 或 8 位的独立报文发送。报文包含：

- 起始位
- 数据位（7 或 8 位，从 LSB [least significant bit 最低有效位] 开始）
- 可选：奇偶校验位
  - “E” EVEN：校验位由发送方设置，偶校验
  - “O” ODD：校验位由发送方设置，奇校验
  - “N” NOT：无奇偶校验位
  - “M” MARK：发送方将奇偶校验位设置为 1
  - “S” SPACE：发送方将奇偶校验位设置为 0
- 停止位（1位 或 2位）

因此，编码规范 8N1 表示：8 个数据位、无奇偶校验位、1 个停止位。

如果选择 7 位编码，则 PLC 传输周期性过程数据到端子模块时，只发送每个数据字节的低 7 位。换句话说，如果向 EL60xx 发送 10 字节数据（每个字节包含 8 位），则会发送 10 个报文，每个报文 7 位。

如果选择 9 位编码，则必须使用 16 位过程数据接口。这样，端子模块就能使用 16 位字的低 9 位。

### 频率

发送方和接收方都必须知道数据传输的频率，并在一定的百分比范围内匹配，以确保接收方能正确检测到线路上电平的任何变化。

### 握手

发送方和接收方之间还可以再进行握手，以便接收方表明已做好接收准备。EL60xx 支持两种握手类型：

- 通过专用 RTS/CTS 数据线握手
  - 该功能必须在 CoE 中激活。
  - 仅适用于 EL6001/EL6002。
- 通过特殊数据报文握手
  - 该功能必须在 CoE 中激活。

### RS 标准

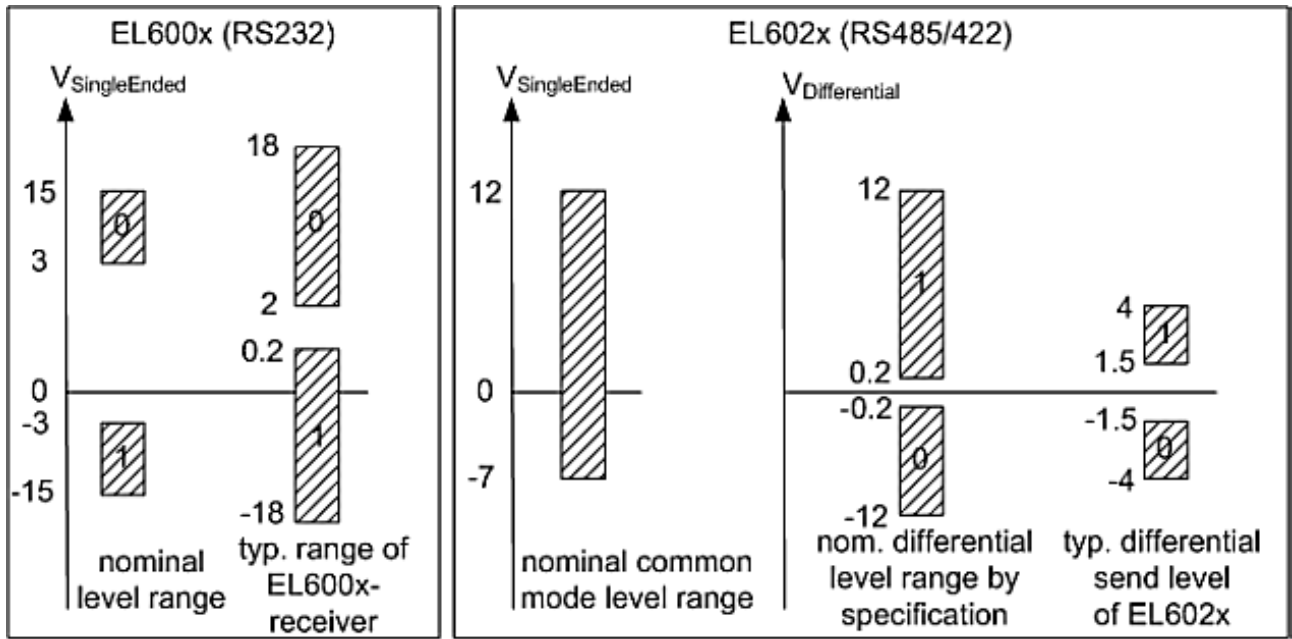
本文档中的 RS232（RS-232）是指 ANSI/EIA/TIA-232-F 标准。

本文档中的 RS422（RS-422）是指基于 ANSI/TIA/EIA-422-B-1994 的 ITU-T V.11 标准。

本文档中的 RS485（RS-485）是指 ANSI/TIA/EIA-485-A-98 标准。

电平接口

EL6001/6002 设备以参考地的 RS232 电平运行，EL6021/6022 设备以差分 RS485/422 电平运行。



voltages on wire depends on load and cabling

附图 4: 电平接口 RS232、RS485/422

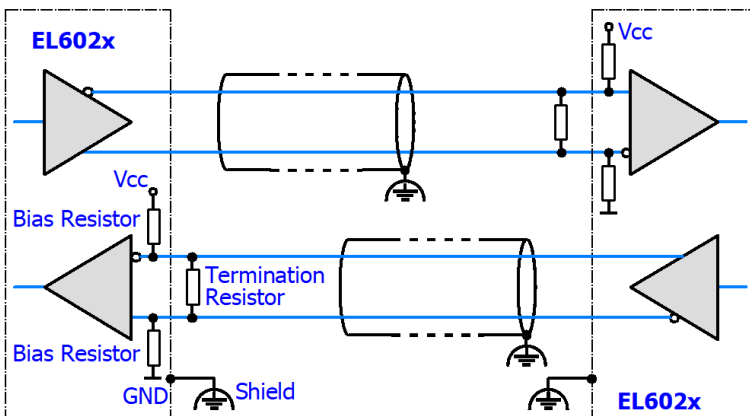
终端和拓扑结构

RS422 和 RS485 串行通信技术通过二线制线路上的电压电平运行。高电阻线路末端的反射会导致信号畸变。因此，需要在接收端安装终端电阻。对于 RS422/485，这些电阻为 120 Ω，与线路电阻一起在传输链路上产生压降。

● 允许电缆长度

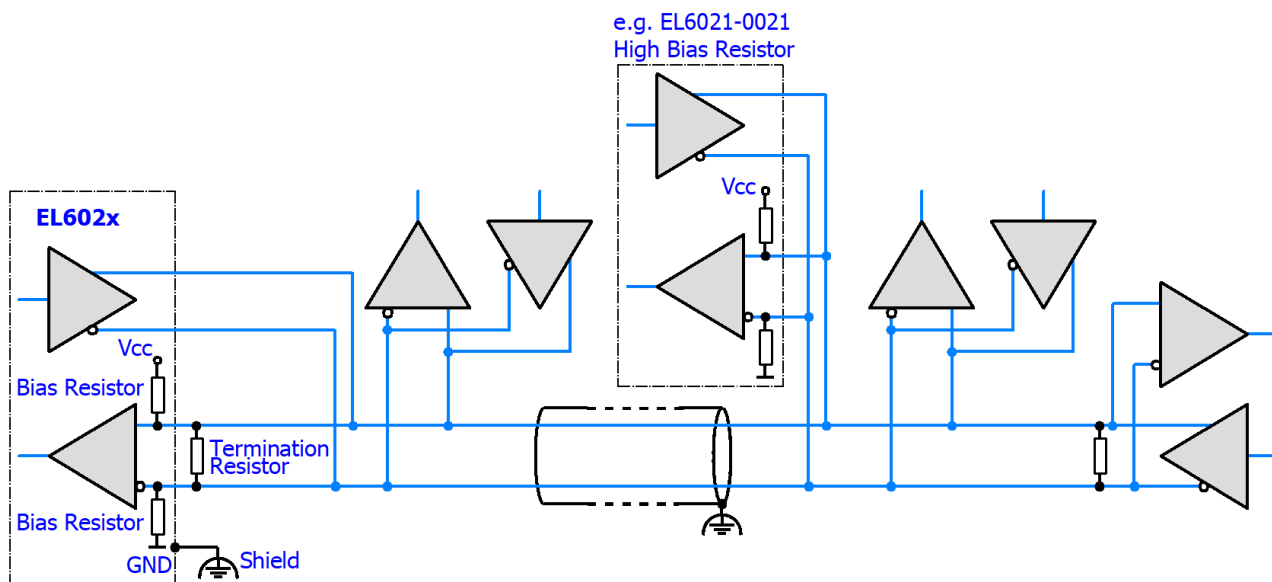
**i** 线路电阻加上终端电阻会导致传输链路上的总体电压降。终端电阻数量过多会导致信号过度衰减。系统设计应确保接收端的电压不低于 200 mV（见图），这是最低电压要求。

在 RS422 模式下，每条线路末端的 RXD 必须连接 120 Ω 的终端电阻。



附图 5: RS422 终端

在有多设备的 RS485 模式中，终端电阻只用于两个终端设备。



附图 6: RS485 终端

基本背景是 RS422/EIA-422 和 RS485/EIA-485 的设计不同:

- RS422: 1 Rx → Tx n (最多 10 台个接收设备)
- RS485: n Rx → Tx m (最多 32/128 台设备, 视总线负载情况而定)

RS485 组件通常具有高阻态输入, 因此总线负载较低。

### ● 带有 BIAS 偏置电阻的 EL602x 终端

**i** 为了在总线模式下运行, EL602x 设备没有集成终端电阻。需要的任何终端装置都必须在端子模块外部进行连接。

带有 < 1 kΩ 集成 BIAS 偏置电阻的 EL602x 设备, 即使线路断开, 也能使总线达到规定的电平。如果总线中连接了多个 EL602x 设备, 并联偏置电阻可能会影响数据通信。在这种情况下, 应将偏置电阻明显较高的 EL6021-0021 用作中间设备。

### 拓扑结构

终端电阻和偏置电阻在总线上产生负载。不过, 它们对确定总线电平至关重要, 因此必须谨慎定位。理想情况下, RS422/485 总线应配置为菊花链或线型。以下拓扑结构可能存在问题:

- 星形拓扑结构: 每个端点最好都有终端, 但这会导致总线负载过重和信号电平模糊。其它潜在问题还包括反射以及运行不稳定。
- 多分支拓扑: 没有明确的端点, 这意味着可能出现反射和环流。

### 屏蔽/屏蔽装置

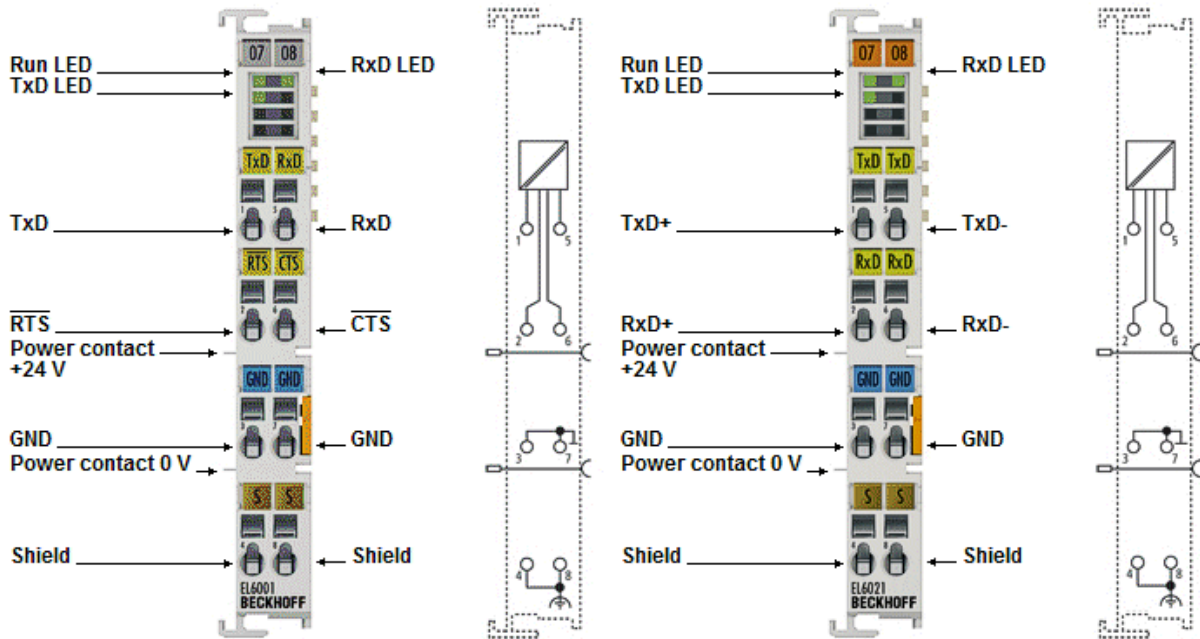
#### 注意

**不要将功能接地用于释放剩余电流或电位差!**

EL60xx 单元提供屏蔽连接, 可通过电缆屏蔽层 (FE, 即功能接地) 消除 EMC 干扰。屏蔽层不得被误用于释放剩余电流或电位差。

## 3.1 EL6001、EL6021

### 3.1.1 简介



#### 串口端子模块 (RS232C/RS422/RS485)，1 通道

串口端子模块 EL6001 和 EL6021 能够通过 RS232 或 RS422/RS485 接口连接设备。EL6001 遵循 CCITT V.28/DIN 66 259-1 标准运行。

与 EL6001 EtherCAT 端子模块连接的设备通过耦合器与自动化设备通信。这种主动的通信通道独立于上一层的总线系统，并在全双工或半双工模式 (EL6021) 下工作，其通信速率可达 115.2 kbaud。

RS232 接口通过电气隔离信号保证高抗干扰性，而 EL6021 还能通过差分信号传输保证这一点。

配上 TwinCAT Virtual Serial COM Driver 驱动程序 (参见 TwinCAT 补充说明 - 通信)，EL6001/EL6021 还可以用作为标准 Windows COM 接口。

#### 快速链接

- [EtherCAT 基础知识 \[► 28\]](#)
- [串口端子模块技术 \[► 18\]](#)
- [调试 \[► 66\]](#)
- [过程数据、常规说明 \[► 137\]](#)
- [EL60x1 的 CoE 对象描述和参数化 \[► 160\]](#)
- [EL60x1 的控制数据和状态数据 \[► 160\]](#)

### 3.1.2 技术数据

技术数据	EL6001	EL6021
数据传输通道	TxD 和 RxD, 全双工	TxD 和 RxD, 全/半双工
数据传输速率	2400...115200 baud, 默认: 9600 baud, 8 个数据位, 无奇偶校验位, 1 个停止位 固件 07 [▶ 202]及以上版本: 还提供 12000 baud 和 14400 baud 固件 11 [▶ 202]及以上版本: 1000...115200 Baud 之间的任何整数波特率	2400...115200 baud, 默认值: 9600 baud, 8 个数据位, 无奇偶校验位, 1 个停止位
数据缓冲区	864 字节接收缓冲区, 128 字节发送缓冲区 EL6001 FW08 及以上版本: 250 字节发送缓冲区	
位传输	-	差分信号
电平接口	RS232	RS485/422
位畸变	< 3 %	-
电缆长度	最长 15 m	最长 1000 m 双绞线
线路阻抗	-	120 Ω
提供外部电源	-	-
诊断	LED 状态灯	
电源	通过 E-bus 供电	
E-bus 电流消耗	典型值 120 mA	典型值 170 mA (短路时的典型值: 250 mA)
电气隔离	500 V (E-bus/RS232C)	500 V (E-bus/RS422、E-bus/RS485)
过程映像中的位宽	1 x 8 位 Control/Status, Inputs/Outputs; 3 x 8 位用户数据或 1 x 8 位 Control/Status, Inputs/Outputs; 5 x 8 位用户数据或 1 x 16 位 Control/Status, Inputs/Outputs; 22 x 8 位用户数据 (可配置)	
配置	无需设置地址 通过 TwinCAT System Manager 配置	
重量	约 55 g	
运行期间允许的环境温度范围	-25° C ... +60° C (宽温范围)	
存储期间允许的环境温度范围	-40° C ... +85° C	
允许的相对湿度	95%, 无冷凝水	
外形尺寸 (W x H x D)	约 15 mm x 100 mm x 70 mm (宽度对齐: 12 mm)	
安装 [▶ 50]	35 mm 安装导轨, 符合 EN 60715 标准	
增强抗振的能力	具备, 也可参见增强抗振模块的 <a href="#">安装说明 [▶ 46]</a>	
抗振性/耐冲击性	符合 EN60068-2-6/EN60068-2-27 标准	
EMC 抗干扰/辐射	符合 EN61000-6-2/EN61000-6-4 标准	
防护等级	IP20	
安装方向	任意	
认证/标识*	CE、cULus [▶ 44]、UKCA、EAC、 ATEX [▶ 40]、IECEX [▶ 41]、	CE、cULus [▶ 44]、UKCA、EAC、 ATEX [▶ 40]、IECEX [▶ 41]、cFMus [▶ 43]

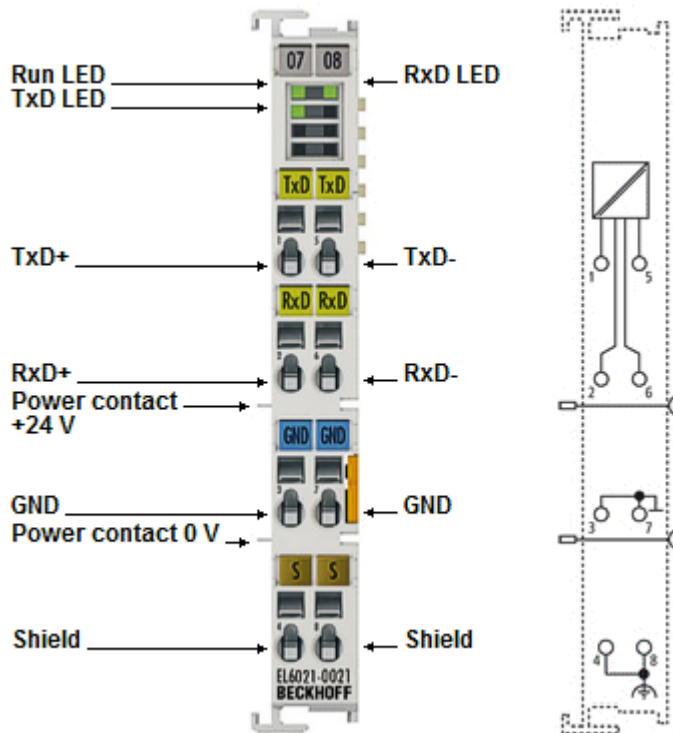
\*) 真正适用的认证/标志见侧面的型号牌 (产品标志)。

#### 其它标记

标准	标记
ATEX	II 3 G Ex nA IIC T4 Gc
IECEX	Ex nA IIC T4 Gc
cFMus	Cl. I, Div. 2, Gr. A、B、C、D Cl. I, Zone 2, AEx/Ex ec IIC T4 Gc

## 3.2 EL6021-0021

### 3.2.1 简介



串口端子模块（RS232C/RS422/RS485），1 通道，用于串行线路中部

EL6021-0021 串口端子模块可通过 RS422 或 RS485 接口连接设备。这些设备通过耦合器与自动化设备通信。这种主动的通信通道独立于上一层的总线系统，并在全双工或半双工模式下工作，其通信速率可达 115.2 kbaud。差分信号传输符合 RS422 协议规范，由于信号经过电气隔离处理，该接口具有很强的抗干扰能力。

如果总线上有多个 EL602x 相互连接，由于 EL6021-0021 配有较大的偏置电阻 [▶ 19]，应该放在线路的中部。

配上 TwinCAT Virtual Serial COM Driver 驱动程序（参见 TwinCAT 补充说明 - 通信），EL6021-0021 可用作标准 Windows COM 接口。

#### 快速链接

- [EtherCAT 基础知识 \[▶ 28\]](#)
- [串口端子模块技术 \[▶ 18\]](#)
- [调试 \[▶ 66\]](#)
- [过程数据、常规说明 \[▶ 137\]](#)
- [EL60x1 的 CoE 对象描述和参数化 \[▶ 160\]](#)
- [EL60x1 的控制数据和状态数据 \[▶ 160\]](#)

### 3.2.2 技术数据

技术数据	EL6021-0021
数据传输通道	TxD 和 RxD, 全/半双工
数据传输速率	2400...115200 baud, 默认值: 9600 baud, 8 个数据位, 无奇偶校验位, 1 个停止位
数据缓冲区	864 字节接收缓冲区, 128 字节发送缓冲区 EL6001 FW08 及以上版本: 250 字节发送缓冲区
功能特点	增大偏置电阻 [► 19]
位传输	差分信号
电平接口	RS485/422
位畸变	-
电缆长度	最长 1000 m 双绞线
线路阻抗	120 Ω
提供外部电源	-
诊断	LED 状态灯
电源	通过 E-bus 供电
E-bus 电流消耗	典型值 170 mA (短路时的典型值: 250 mA)
电气隔离	500 V (E-bus/RS422、E-bus/RS485)
过程映像中的位宽	1 x 8 位 Control/Status, Inputs/Outputs: 3 x 8 位用户数据或 1 x 8 位 Control/Status, Inputs/Outputs: 5 x 8 位用户数据或 1 x 16 位 Control/Status, Inputs/Outputs: 22 x 8 位用户数据 (可配置)
配置	无需设置地址 通过 TwinCAT System Manager 配置
重量	约 55 g
运行期间允许的环境温度范围	0° C ... +55° C
存储期间允许的环境温度范围	-25° C ... +85° C
允许的相对湿度	95%, 无冷凝水
外形尺寸 (W x H x D)	约 15 mm x 100 mm x 70 mm (宽度对齐: 12 mm)
安装 [► 50]	35 mm 安装导轨, 符合 EN 60715 标准
增强抗振的能力	具备, 可参见增强抗振模块的安装说明 [► 46]
抗振性/耐冲击性	符合 EN60068-2-6/EN60068-2-27 标准
EMC 抗干扰/辐射	符合 EN61000-6-2/EN61000-6-4 标准
防护等级	IP20
安装方向	任意
认证/标识*	CE、cULus [► 44]、UKCA、EAC、ATEX [► 39]、IECEX [► 41]

\*) 真正适用的认证/标志见侧面的型号牌 (产品标志)。

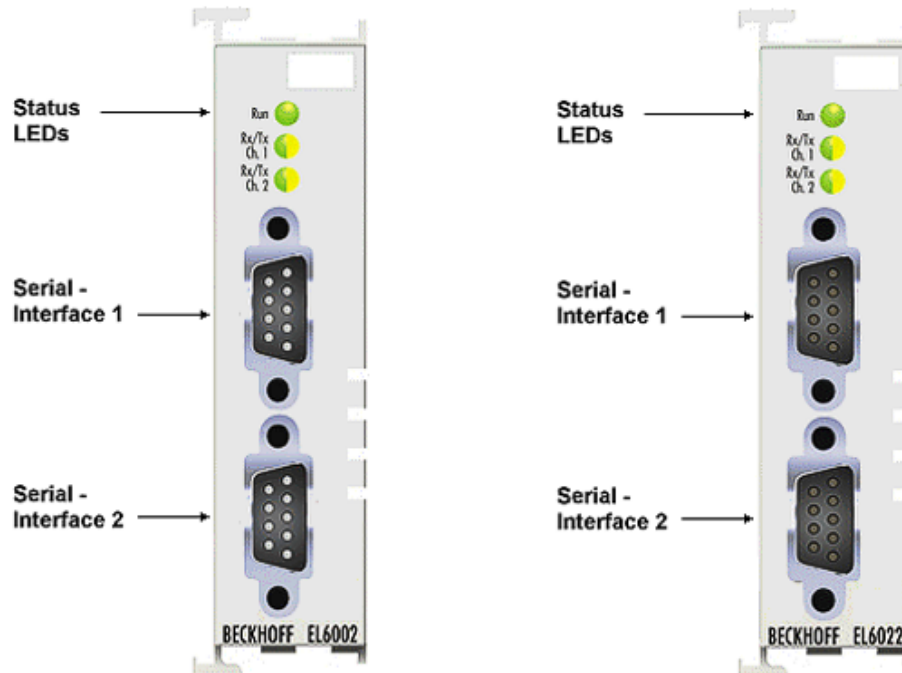
#### 其他标志

标准	标志
ATEX	II 3 G Ex nA IIC T4 Gc
IECEX	Ex nA IIC T4 Gc



### 3.3 EL6002、EL6022

#### 3.3.1 简介



#### 串口端子模块（RS232C/RS422/RS485），2 通道

串口端子模块 EL6002 和 EL6022 可通过两个 RS232 或两个 RS422/RS485 接口连接设备，每个接口带有一个 D 形 9 针连接器。接口相互电气隔离，也与 EtherCAT 电气隔离。

连接到 EL6002/EL6022 EtherCAT 端子模块的设备通过耦合器与自动化设备通信。这种主动的通信通道独立于上层 EtherCAT 系统，并在全双工模式下运行，通信速率范围为 300 baud 至 115.2 kbaud。

RS232/RS422/RS485 接口通过电气隔离信号确保高抗干扰性。EL6022 可从 E-bus 电源（电气隔离，短路保护）获取 2 路 5 V/20 mA 的电流，以向外接设备供电。

配上 TwinCAT Virtual Serial COM Driver 驱动程序，EL60xx 可用作标准的 Windows COM 接口。

#### 快速链接

- [EtherCAT 基础知识 \[► 28\]](#)
- [串口端子模块技术 \[► 18\]](#)
- [调试 \[► 66\]](#)
- [过程数据、常规说明 \[► 137\]](#)
- [EL60x1 的 CoE 对象描述和参数化 \[► 184\]](#)
- [EL60x1 的控制数据和状态数据 \[► 199\]](#)

### 3.3.2 技术数据

技术数据	EL6002	EL6022
数据传输通道	2 个, TxD 和 RxD, 全双工	2 个, TxD 和 RxD, 全/半双工
连接	2 个 D 形 9 针连接器 (DE9)	2 个 D 形 9 针连接器 (DE9)
数据传输速率	300...115200 baud 默认值: 9600 baud, 8 个数据位, 无奇偶校验位, 1 个停止位	
数据缓冲区	864 字节接收缓冲区、每通道 128 字节发送缓冲区	
电平接口	RS232	RS485/422
电缆长度	最长 15 m	最长 1000 m 双绞线
提供外部电源	-	2 路, 典型值 5 V (± 20%), 来自 E-bus 电源 (电气隔离), 最大电流 20 mA, 带短路保护
诊断	LED 状态灯	
电源	通过 E-bus 供电	
E-bus 电流消耗	典型值 170 mA	典型值 250 mA (短路时的典型值: 250 mA)
电气隔离	500 V (E-bus/RS232C)	500 V (E-bus/RS422、E-bus/RS485)
过程映像中的位宽	1 x 16 位 Control/Status, Inputs/Outputs: 22 x 8 位用户数据	
配置	无需设置地址 通过 TwinCAT System Manager 配置	
运行期间允许的环境温度范围	-25° C ... +60° C (宽温范围)	
存储期间允许的环境温度范围	-40° C ... +85° C	
允许的相对湿度	95%, 无冷凝水	
重量	约 70 g	
外形尺寸 (W x H x D)	约 26 mm x 100 mm x 52 mm (宽度对齐: 23 mm)	
安装 [▶_60]	35 mm 安装导轨, 符合 EN 60715 标准	
抗振性/耐冲击性	符合 EN60068-2-6/EN60068-2-27 标准	
EMC 抗干扰/辐射	符合 EN61000-6-2/EN61000-6-4 标准	
防护等级	IP20	
安装方向	任意	
认证/标识*	CE、cULus [▶_44]、UKCA、EAC、ATEX [▶_40]、IECEX、[▶_41]	

\*) 真正适用的认证/标志见侧面的型号牌 (产品标志)。

#### 其他标志

标准	标志
ATEX	II 3 G Ex nA IIC T4 Gc
IECEX	Ex nA IIC T4 Gc

## 3.4 启动

### 开始

如需调试：

- 按照安装和布线 [▶ 38] 章节所述，安装 EL600x / EL602x
- 按照调试 [▶ 66] 章节所述，在 TwinCAT 中配置 EL600x / EL602x

## 4 基本通讯

### 4.1 EtherCAT 基础知识

关于 EtherCAT 现场总线的基础知识，请参考 [EtherCAT 系统文档](#)。

### 4.2 EtherCAT 布线 - 线缆连接

两个 EtherCAT 设备之间的电缆长度不得超过 100 米。这源于快速以太网 (FastEthernet) 技术，首要的原因是电缆长度增加导致信号衰减。如果使用规范的电缆，则允许的最大连接长度为 5 + 90 + 5 米。另请参见关于 EtherCAT/Ethernet 基础设施的设计建议。

#### 电缆和连接器

在连接 EtherCAT 设备时，只能使用符合 EN50173 或 ISO/IEC11801 标准的 5 类 (CAT5) 及以上以太网连接件 (电缆 + 接头)。EtherCAT 使用 4 条线路进行信号传输。

例如，EtherCAT 使用 RJ45 插拔连接器。引脚分配与以太网标准 (ISO/IEC 8802-3) 兼容。

引脚	导线颜色	信号	描述
1	黄色	TD +	发送数据 +
2	橙色	TD -	发送数据 -
3	白色	RD +	接收数据 +
6	蓝色	RD -	接收数据 -

由于采用了自动电缆检测 (交叉直连自适应) 技术，在倍福的 EtherCAT 设备之间可以使用直连 (1:1) 或交叉的电缆。

#### ● 推荐的电缆

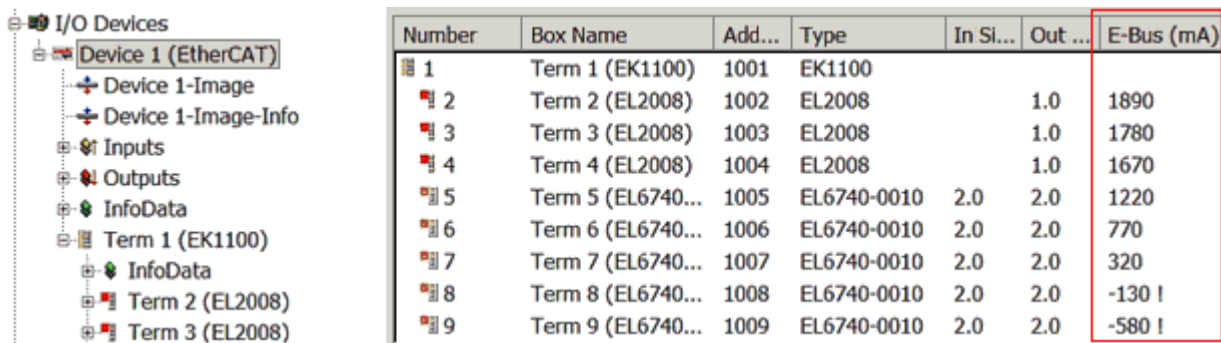
- 建议使用适当的倍福组件，例如
- 电缆组件 ZK1090-9191-xxxx
  - 相应的 RJ45 连接器、现场组件 ZS1090-0005
  - EtherCAT 电缆、现场组件 ZB9010、ZB9020
- 用于连接 EtherCAT 设备的合适电缆可参见倍福公司网站！

#### E-bus 供电

总线耦合器可以用 5 V 的 E-bus 系统电压为添加在它上面的 EL 端子模块供电；一个耦合器通常可以提供达到 2 A 的 E-Bus 电流 (详见各自的设备文件)。

关于每个 EL 端子模块需要消耗多少 E-bus 电流的信息，可参见倍福公司网站和产品目录。如果连接的端子模块需要的电流超过了耦合器可以提供的电流，则必须在整组端子模块的适当位置插入 E-Bus 电源模块 (例如 EL9410)。

在 TwinCAT System Manager 中可以显示预计的理论上最大的 E-Bus 消耗电流。如果预计 E-Bus 供电不足，剩余电流总额就会是负数，并以感叹号 (!) 标记；在这种位置前面需要插入一个 E-Bus 电源模块。



Number	Box Name	Add...	Type	In Si...	Out ...	E-Bus (mA)
1	Term 1 (EK1100)	1001	EK1100			
2	Term 2 (EL2008)	1002	EL2008		1.0	1890
3	Term 3 (EL2008)	1003	EL2008		1.0	1780
4	Term 4 (EL2008)	1004	EL2008		1.0	1670
5	Term 5 (EL6740...)	1005	EL6740-0010	2.0	2.0	1220
6	Term 6 (EL6740...)	1006	EL6740-0010	2.0	2.0	770
7	Term 7 (EL6740...)	1007	EL6740-0010	2.0	2.0	320
8	Term 8 (EL6740...)	1008	EL6740-0010	2.0	2.0	-130 !
9	Term 9 (EL6740...)	1009	EL6740-0010	2.0	2.0	-580 !

附图 7: System Manager 中的电流计算

## 注意

## 可能发生故障！

一个 I/O 站里面所有 EtherCAT 端子模块的 E-Bus 电源必须使用相同的接地电位！

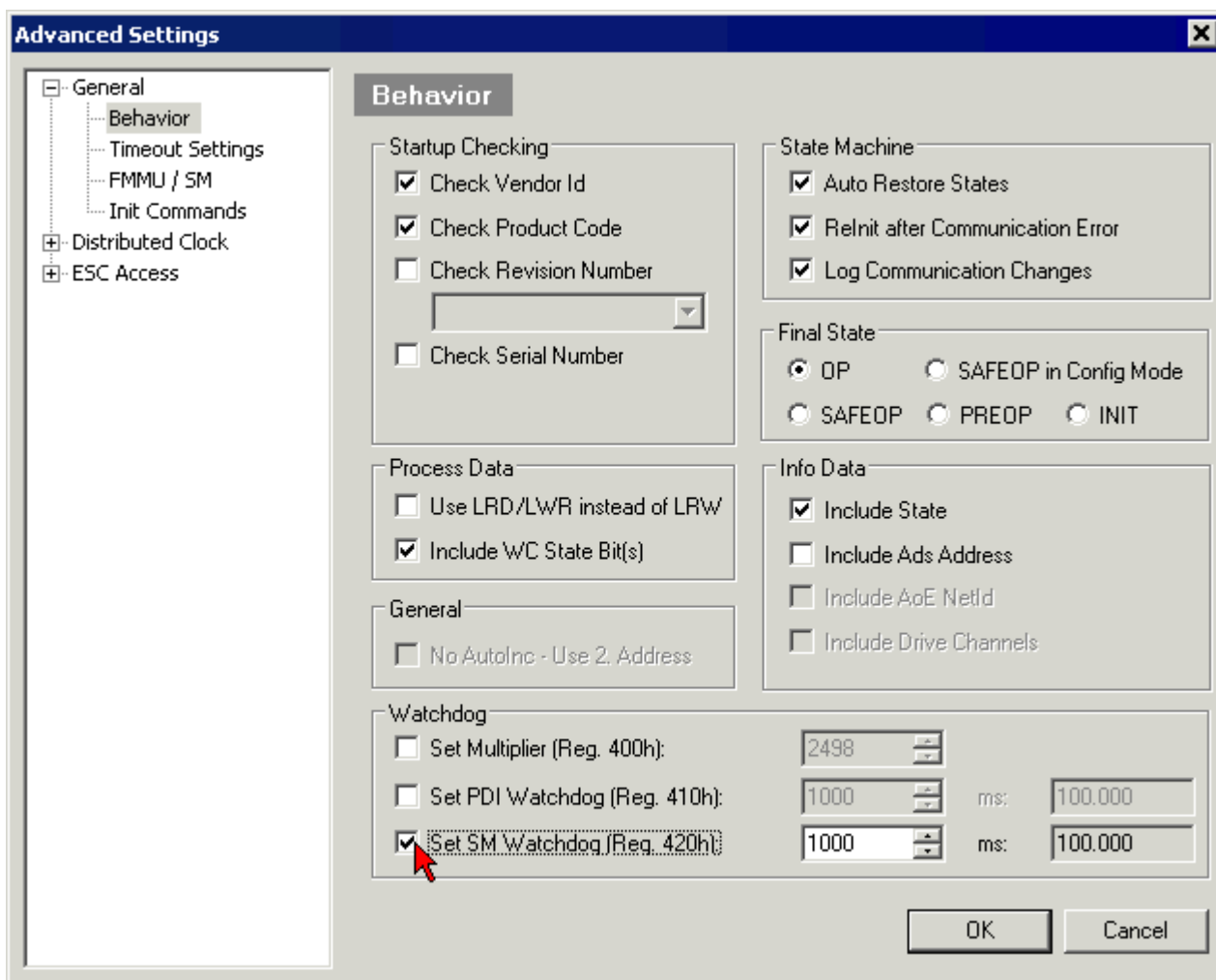
### 4.3 设置看门狗的一般注意事项

EtherCAT 端子模块配备了一个安全装置（看门狗），如果发生过程数据通讯中断的情况，就会根据设置在预定时间后将输出（如果存在）切换到预设状态，例如切换到 FALSE（关闭）或某个输出值。

EtherCAT 从站控制器（ESC）有两个看门狗：

- SM 看门狗（默认：100 ms）
- PDI 看门狗（默认：100 ms）

在 TwinCAT 中可以分别设置这两个看门狗的时间，如下所示：



附图 8: EtherCAT 选项卡 -> Advanced Settings -> Behavior-> Watchdog

注意：

- Multiplier Register（乘数寄存器）400h（十六进制，即 x0400）可用于两个看门狗。
- 每个看门狗都有自己的计时设置 410h 或 420h，与 Multiplier 相乘得到一个时间。
- 重要的是：只有勾选了前面的复选框，在 EtherCAT 启动时，乘数/计时设置才会加载到从站。
- 如果没有勾选，则不会下载任何信息，ESC 中的设置保持不变。
- 下载的数值可以在 ESC 寄存器 x0400/0410/0420 中看到：ESC Access -> Memory

### SM 看门狗 (SyncManager 看门狗)

SyncManager 看门狗在每次与端子模块成功进行 EtherCAT 过程数据通信时被重置。例如，如果由于线路中断，与端子模块的 EtherCAT 过程数据通信时间超过设定并激活的 SM 看门狗时间，则看门狗被触发。端子模块的状态（通常是 OP）不受影响。看门狗只有在 EtherCAT 过程数据访问成功后才会再次重置。

因此，从 EtherCAT 方面来看，SyncManager 看门狗可以用来监测是否与 ESC 进行正确和及时的过程数据通信。

看门狗允许的最长时间取决于设备。例如，对于“简单的”EtherCAT 从站（无固件），在 ESC 中执行看门狗通常长达 170 秒。对于“复杂的”EtherCAT 从站（带固件），SM 看门狗功能通常通过寄存器 400/420 进行参数设置。因为是通过  $\mu\text{C}$  执行，时间可以大大缩短。此外，看门狗的执行时间可能会有一定程度的波动。由于 TwinCAT 对话框允许的最大输入值为 65535，建议对所需的看门狗时间进行测试。

### PDI 看门狗 (过程数据看门狗)

如果与 EtherCAT 从站控制器 (ESC) 的 PDI 通讯丢失的时间超过了设定和激活的 PDI 看门狗时间，则该看门狗被触发。

PDI (过程数据接口) 是 ESC 的内部接口，例如与 EtherCAT 从站中本地处理器的接口。通过 PDI 看门狗，可以监测这种通信是否有故障。

因此，从应用方面来看，PDI 看门狗可以用来监测是否与 ESC 进行正确和及时的过程数据通信。

### 计算方式

Watchdog time =  $[1/25 \text{ MHz} * (\text{Watchdog multiplier} + 2)] * \text{PDI (或 SM) watchdog}$

例如：默认 Multiplier = 2498, SM watchdog = 1000  $\rightarrow$  100 ms

看门狗乘数 Multiplier + 2 的值对应于一个基数为 40ns 的看门狗刻度。

#### ⚠ 谨慎

##### 可能出现未定义的状态！

通过 SM 看门狗 = 0 来关闭 SM 看门狗的功能只在 -0016 及以上版本的端子模块中实现。在以前的版本中，不能使用这种操作方式。

#### ⚠ 谨慎

##### 可能出现设备损坏和未定义的状态！

如果 SM 看门狗被激活，并且输入了 0 值，看门狗就完全关闭。这样就会停用看门狗！如果通信中断，就不会将模块的输出设定在安全状态。

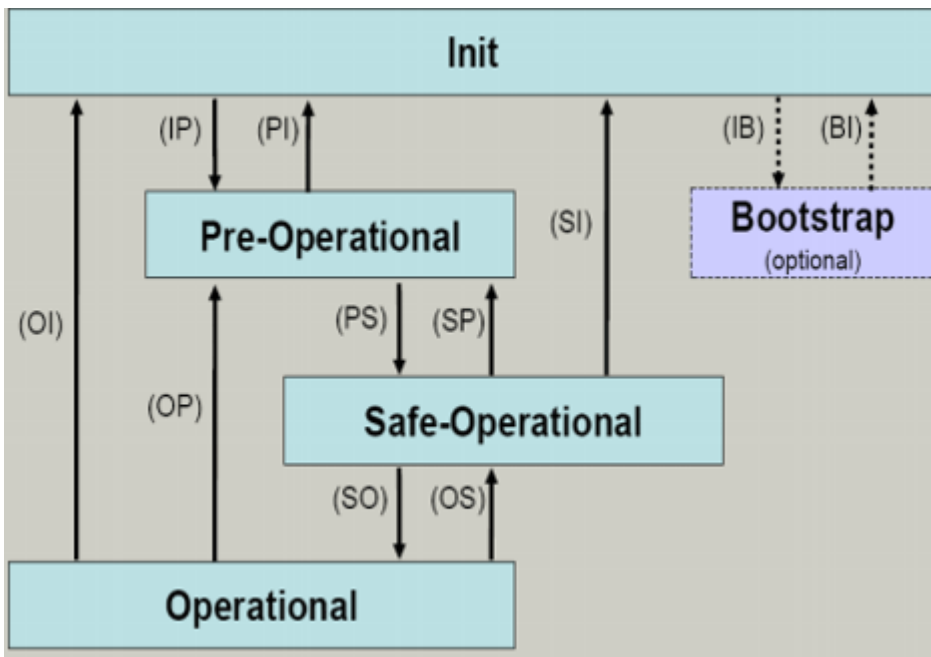
## 4.4 EtherCAT 状态机

EtherCAT 从站的状态是通过 EtherCAT 状态机 (ESM) 控制的。根据具体的状态，EtherCAT 从站可以访问或执行不同的功能。EtherCAT 主站必须在从站的不同状态下向其发送特定的命令，特别是在从站的启动期间。

以下状态之间有所区别：

- Init (初始化)
- Pre-Operational (预备运行)
- Safe-Operational (安全运行)
- Operational (运行)
- Boot (引导)

每个 EtherCAT 从站启动后的正常状态是 Operational (运行) 状态。



附图 9: EtherCAT 状态机的状态

## Init

开机后, EtherCAT 从站处于 *Init* 状态。邮箱或过程数据通信无法进行。EtherCAT 主站初始化同步管理器 (Sync Manager) 通道 0 和 1, 用于邮箱通信。

## 预备运行 (Pre-Op)

从 *Init* 切换到 *Pre-Op* 的过程中, EtherCAT 从站检查邮箱是否已正确初始化。

在 *Pre-Op* 状态下, 可以进行邮箱通信, 但不能进行过程数据通信。EtherCAT 主站对过程数据的同步管理器 (Sync Manager) 通道 (来自 Sync Manager 通道 2) 以及 FMMU 通道进行初始化。如果从站支持可配置的映射, 主站也会对 PDO 映射或同步管理器 PDO 分配进行初始化。在这个状态下, 还会传输过程数据的传输设置以及不同于默认值的模块特定参数。

## 安全运行 (Safe-Op)

从 *Pre-Op* 切换到 *Safe-Op* 的过程中, EtherCAT 从站检查用于过程数据通信的同步管理器 (Sync Manager) 是否正确, 必要时还会检查分布时钟 (Distributed Clock) 的设置是否正确。在确认状态变化之前, EtherCAT 从站将当前的输入数据复制到 EtherCAT 从站控制器 (ECSC) 的相关 DP-RAM 区域。

在 *Safe-Op* 状态下, 可以进行邮箱和过程数据通信, 但从站输出保持在安全状态, 而输入数据被周期性刷新。

### ● SAFEOP 状态下的输出

**i** 默认的看门狗 (Watchdog) 监视装置, 将模块的输出设置为 SAFEOP 和 OP 中指定的安全状态 (例如关闭状态)。如果通过停用模块中的看门狗监测来防止这种情况的发生, 那么输出也可以在 SAFEOP 状态下被切换或设置。

## 运行 (Op)

在 EtherCAT 主站将 EtherCAT 从站从 *Safe-Op* 切换到 *Op* 之前, 必须传输有效的输出数据。

在 *Op* 状态下, 从站将主站的输出数据复制到它的输出, 过程数据和邮箱通信都可以进行。

## 引导 (Boot)

在 *Boot* 状态下, 可以更新从站固件。*Boot* 状态只能通过 *Init* 状态达到。

在 *Boot* 状态下, 可以通过 *file access over EtherCAT* (FoE) 协议进行邮箱通信, 但不能进行其他邮箱通信或者过程数据通信。

## 4.5 CoE 接口

### 一般说明

CoE 接口 (CAN application protocol over EtherCAT) 用于 EtherCAT 设备的参数管理。EtherCAT 从站或 EtherCAT 主站管理固定 (只读) 或可变 (读写) 参数, 这些参数用于运行、诊断或调试。

CoE 参数的组织形式为分层表格式。原则上用户可以通过现场总线进行读取访问。EtherCAT 主站 (TwinCAT System Manager) 可以通过 EtherCAT 以 Read 或 Write 模式访问从站本地的 CoE 列表, 具体取决于 CoE 参数的属性。

CoE 参数类型可能各不相同, 包括字符串 (文本)、整数、布尔值或较长字节的字段。它们可以用来描述模块的各种特性。这些参数包括制造商 ID、序列号、过程数据设置、设备名称、模拟量测量的校准值或密码。

可以通过两层十六进制的索引号来指定参数的序号: (主) 索引 Index, 及随后的子索引 SubIndex。其数值范围是

- 索引 Index: 0x0000 ... 0xFFFF (0...65535<sub>dec</sub>)
- 子索引 SubIndex: 0x00...0xFF (0...255<sub>dec</sub>)

以这种方式定位的参数通常写成 0x8010:07, 前面的 “0x” 用于标识十六进制数字范围, 在 Index 和 SubIndex 之间用冒号分隔。

对于 EtherCAT 现场总线用户来说, 相关的索引范围是:

- 0x1000: 这是存储设备固定身份信息的地方, 包括名称、制造商、序列号等, 还有关于当前的和可用的过程数据配置的信息。
- 0x8000: 这是储存所有通道的运行和功能参数的地方, 例如滤波设置或输出频率。

其他重要的范围是:

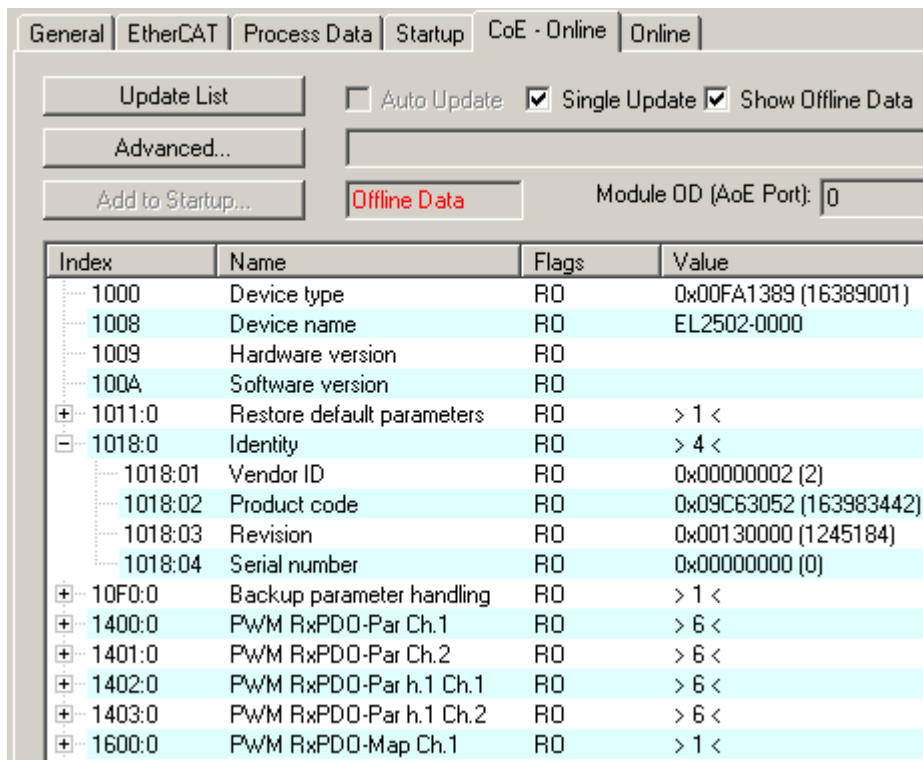
- 0x4000: 有些 EtherCAT 设备在此存储通道参数。过去, 这是在引入 0x8000 区域之前的第一个参数区。以前用 0x4000 配置参数的 EtherCAT 设备改用 0x8000 后, 出于兼容性的考虑, 这两个 CoE 索引范围都支持, 并在内部进行映射。
- 0x6000: input PDO (“input”, 是指从 EtherCAT 主站的角度看是输入)
- 0x7000: output PDO (“output”, 是指从 EtherCAT 主站的角度看是输出)

### ● 适用性

**I** 不是每个 EtherCAT 设备都有 CoE 列表。没有专用处理器的简单 I/O 模块通常没有可变参数, 因此没有 CoE 列表。

如果一个设备有 CoE 列表, 它就会在 TwinCAT System Manager 中显示为一个单独的选项卡, 并列各参数:





附图 10: “CoE Online” 选项卡

上图显示了设备“EL2502”中可用的 CoE 对象，范围从 0x1000 到 0x1600。0x1018 的子索引进行了展开显示。

### 注意

#### CoE 对象字典（CAN over EtherCAT）的修改，通过程序访问。

当使用/操作 CoE 参数时，请注意 EtherCAT 系统文档中“[CoE 接口](#)”章节中的一般 CoE 注意事项：

- 如果需要更换组件，请保留 Startup List。
- 在线字典和离线字典之间的区别
- 当前最新的 XML 描述文件（从倍福公司网站下载），
- “CoE-Reload”用于重置所做的更改。
- 系统运行期间通过 PLC 程序访问（参见 [TwinCAT3 | PLC Library: Tc2 EtherCAT](#) 和 [Example program R/W CoE](#)）

#### 数据管理和 “NoCoeStorage” 功能

有的参数，特别是从站的设置参数，是可配置的和可写入的。这可以在 Write 或 Read 模式下进行

- 通过 System Manager 直接修改（图“CoE Online”选项卡）  
这个方法在系统/从站调试时非常有用。点击修改参数的索引（Index）行，在“SetValue”对话框中输入一个值。
- 通过控制系统（PLC）的 ADS 通讯，例如通过 TcEtherCAT.lib 库中的功能块进行修改。  
这个方法推荐用于系统运行时修改 CoE，或者暂时无法打开 System Manager 亦或是没有操作人员的情况下使用。

## ● 数据管理

如果从站的 CoE 参数被在线修改，倍福设备会将任何修改以掉电保持的方式存储在 EEPROM 中，也就是说，重新启动后，修改后的 CoE 参数仍然可用。但其它制造商的设备则可能情况有所不同。

EEPROM 在写入操作方面的使用寿命是有限的。通常写入 100,000 次以后，就不能保证新的（修改的）数据能被可靠地保存或仍然可读。这不会影响正常调试。然而，如果在机器运行时 CoE 参数通过 ADS 不断被修改，就很有可能达到极限使用寿命限。通过 NoCoeStorage 功能可以禁止保存修改后 CoE 值，但是否支持该功能取决于固件版本。关于这个功能是否适用于相应的设备，请参考本文件中的技术数据。

- 如果支持该功能：通过在 CoE 0xF008 中一次性输入代码 0x12345678 来激活该功能，只要代码不被改变，该功能就一直有效。开启设备后，保存 CoE 值的功能就处于非活动状态。改变后的 CoE 值不会保存到 EEPROM 中，因此修改次数不受限制。
- 不支持该功能：考虑到使用寿命限值，不允许连续改变 CoE 值。

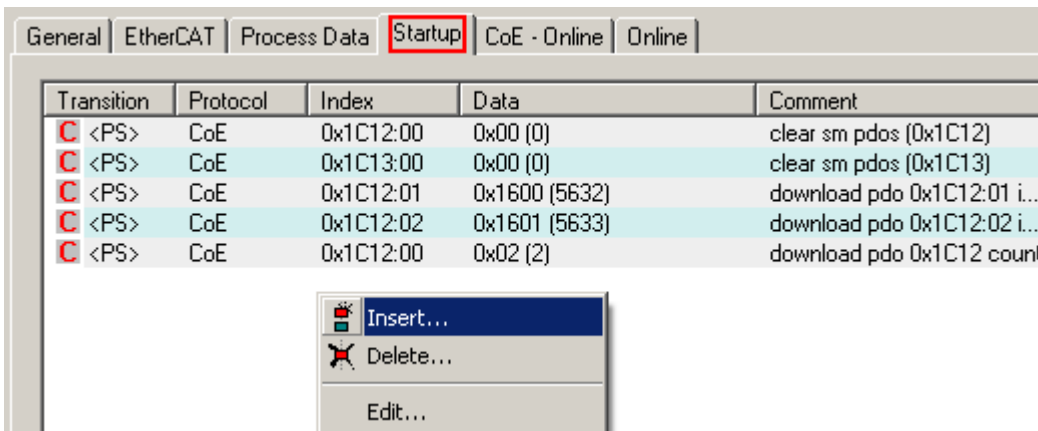
## ● Startup List

如果更换了端子模块，端子模块的本地 CoE 列表中的修改会丢失。如果一个端子模块被更换成新的倍福端子模块，新模块具有默认设置。因此，建议将 EtherCAT 从站所有的 CoE 修改项放到它的 Startup List，因为 EtherCAT 总线启动时会自动处理这个列表中的各项。通过这种方式，一个 EtherCAT 从站更换后可以自动按照用户的定义进行参数设置。

如果使用的 EtherCAT 从站不能在本地永久存储 CoE 值，则必须使用 Startup List。

### 手动修改 CoE 参数的推荐方法

- 在 System Manager 中进行必要的更改  
数据存储在 EtherCAT 从站本地，
- 如果要永久保存该值，请在 Startup List 中输入。  
Startup 中的条目顺序通常无关紧要。



附图 11: TwinCAT System Manager 中的 Startup List

Startup List 会事先包含那些 System Manager 基于 ESI 的定义配置的 CoE 参数值。也可以另外创建应用特定的条目。

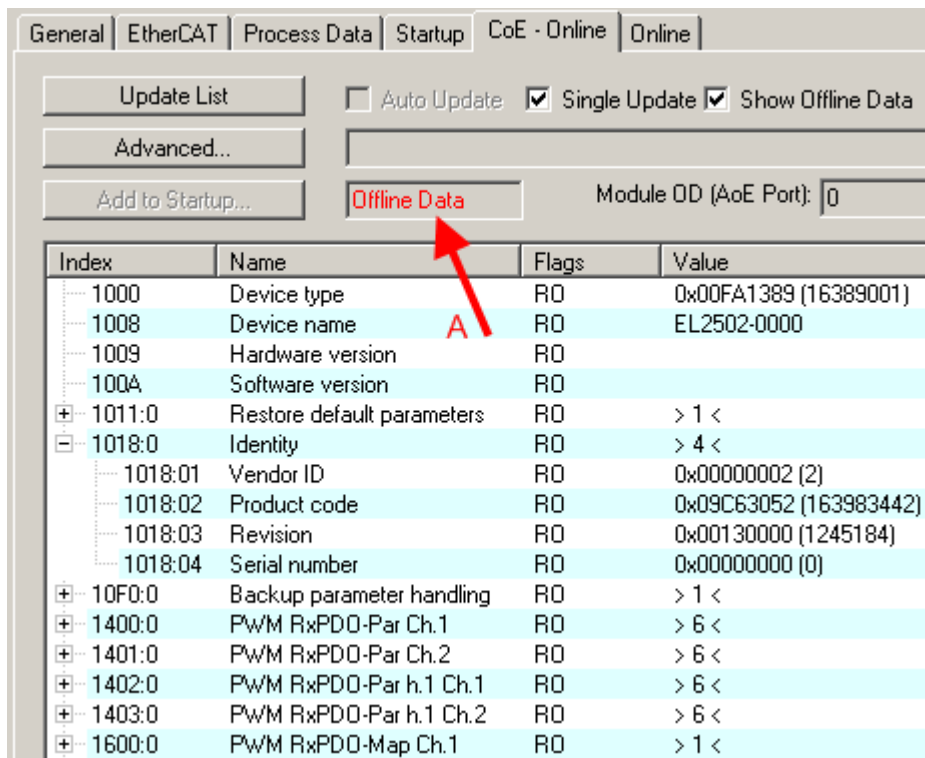
### online/offline列表

在使用 TwinCAT System Manager 时，必须区分 EtherCAT 设备是否“可用”，即已经上电并连接到 EtherCAT 从而处于**在线 (Online)** 状态，或者是在从站未连接的情况下创建了一个**离线 (Offline)** 配置。

在这两种情况下，都会显示一个 CoE 列表，如图“CoE Online 选项卡”所示。连接状态显示为 offline/online。

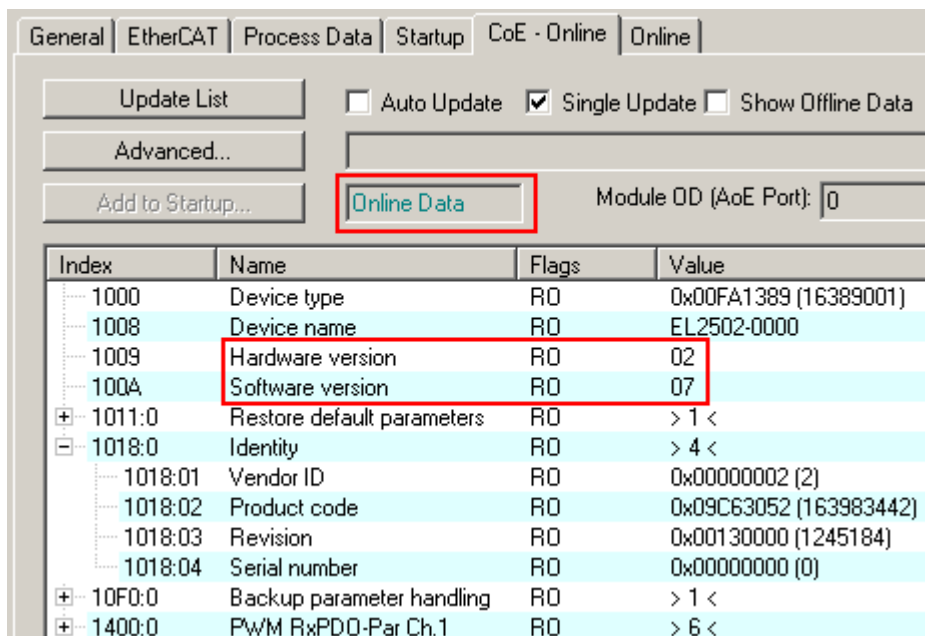
- 如果从站处于离线状态
  - 显示 ESI 文件中的离线列表。此时修改 CoE 参数是没有意义的，也无法进行。
  - 配置的状态显示在“Identity”下。
  - 不显示固件或硬件版本，因为只有实际在线的设备才有这些特征参数。

- Offline Data显示为红色。



附图 12: 离线列表

- 如果从站是在线状态
  - 读取实际的当前从站列表。这可能需要几秒钟，具体取决于数据大小和周期时间。
  - 显示的是实际身份信息
  - 根据电子信息显示设备的固件和硬件版本
  - Online Data显示为绿色。



附图 13: 在线列表

## 基于通道的顺序

通常包含几个相同功能通道的EtherCAT 设备都具有CoE 列表。例如，一个 4 通道模拟量 0...10 V 输入端子模块也有4条逻辑通道，因此有4套相同的通道参数。为了避免在文件中列出每个通道，往往用占位符“n”来表示各个通道的编号。

在 CoE 系统中，16 个 Index（每个 Index 有 255 个 SubIndex）通常足以表示所有通道参数。因此，基于通道的顺序是以  $16_{\text{dec}}/10_{\text{hex}}$  的间隔排列的。以参数范围 0x8000 为例：

- 通道 0: 参数范围 0x8000:00 ... 0x800F:255
- 通道 1: 参数范围 0x8010:00 ... 0x801F:255
- 通道 2: 参数范围 0x8020:00 ... 0x802F:255
- ...

这种情况一般写成 0x80n0。

关于 CoE 接口的详细信息，可参见倍福公司网站 [EtherCAT 系统文档](#)。

## 4.6 分布时钟 (Distributed Clock)

分布时钟表示 EtherCAT 从站控制器 (ESC) 中的一个本地时钟，具有以下特点：

- 单位  $1\text{ ns}$
- 零点  $1.1.2000\ 00:00$
- 大小  $64$  位 (足够未来 584 年使用；但是，一些 EtherCAT 从站只提供 32 位支持，即变量在大约 4.2 秒后溢出)
- EtherCAT 主站自动将本地时钟与 EtherCAT 总线中的主站时钟同步，精度  $<100\text{ ns}$ 。

详细信息请参见 [EtherCAT 系统描述](#)。

## 5 安装和布线

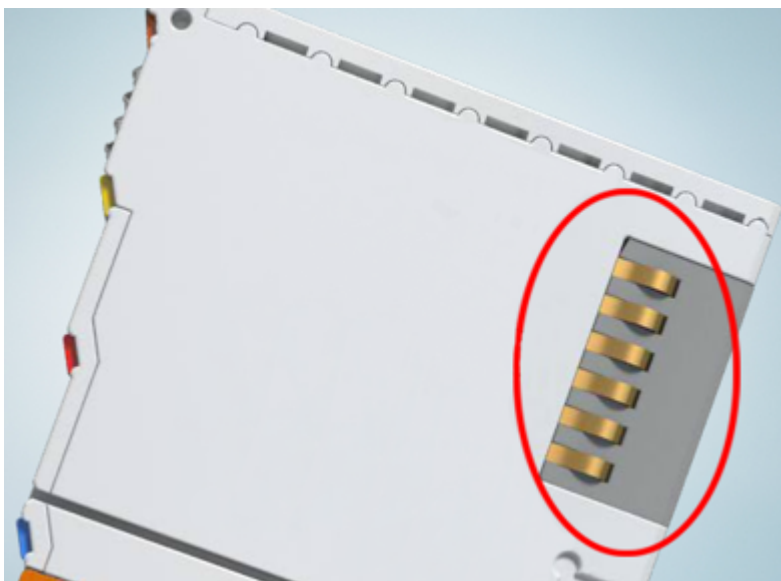
### 5.1 静电防护的说明

#### 注意

#### 静电放电可能会破坏设备！

这些设备含有因处理不当而导致静电放电风险的部件。

- 请确保已进行静电放电，避免直接接触设备的触点。
- 避免与高度绝缘的材料（合成纤维、塑料薄膜等）接触。
- 在处理该设备时，周围环境（工作场所、包装和人员）应恰当接地。
- 每个 I/O 站必须在最末端使用 [EL9011](#) 或 [EL9012](#) 端子盖板，以确保达到保护等级和 ESD 静电保护。



附图 14: 倍福 I/O 组件的弹簧触点

## 5.2 防爆

### 5.2.1 ATEX – 特殊条件（标准温度范围）

#### ⚠ 警告

在潜在爆炸性区域使用具有标准温度范围的 Beckhoff 现场总线组件，请遵守防爆 ATEX 指令 (2014/34/EU) 的特别规定！

- 经认证的组件应当安装在一个合适的外壳中，保证按照 EN 60079-15 标准至少达到 IP54 的防护等级！应当按此标准考虑使用过程中的环境条件！
- 关于防尘（仅指证书编号为 KEMA 10ATEX0075 X，第 9 版的现场总线组件）：考虑到设备使用的环境条件，设备应安装在一个合适的外壳中，对于 IIIA 组或 IIIB 组提供符合 EN 60079-31 标准的 IP54 防护等级，对于 IIIC 组则提供 IP6X 的防护等级！
- 如果在额定运行期间，电缆、线路或管道的进线点的温度高于 70° C，或电线分支点的温度高于 80° C，那么必须选择耐受温度数据满足实际测量温度值的线缆！
- 在潜在爆炸性区域使用具有标准温度范围的 Beckhoff 现场总线组件，请遵守相关标准允许的环境温度范围 0 至 55° C！
- 必须采取措施，防止因瞬时干扰电压而超过额定工作电压的 40% 以上！
- 只有在关闭电源电压或确保非爆炸性环境的情况下，才可以从总线端子模块系统中拔出或拆除单个模块！
- 只有在关闭电源电压或确保非爆炸性环境的情况下，才可以连接或断开经认证部件的接线！
- 只有在关闭电源电压或确保非爆炸性环境的情况下，才可以更换 KL92xx/EL92xx 馈电端子模块的保险丝！
- 只有在关闭电源电压或确保非爆炸性环境的情况下，才能调整地址拨码和 ID 开关！

#### 标准

符合下列标准规定，满足基本健康和安全要求：

- EN 60079-0:2012+A11:2013
- EN 60079-15:2010
- EN 60079-31:2013（仅适用于证书编号 KEMA 10ATEX0075 X，第 9 版）

#### 标志

经过 ATEX 指令认证适用于潜在爆炸性区域的标准温度范围 Beckhoff 现场总线组件带有以下标记之一：



II 3G KEMA 10ATEX0075 X Ex nA IIC T4 Gc Ta: 0 ... +55° C

II 3D KEMA 10ATEX0075 X Ex tc IIIC T135° C Dc Ta: 0 ... +55° C  
(仅适用于证书编号 KEMA 10ATEX0075 X，第 9 版的现场总线组件)

或



II 3G KEMA 10ATEX0075 X Ex nA nC IIC T4 Gc Ta: 0 ... +55° C

II 3D KEMA 10ATEX0075 X Ex tc IIIC T135° C Dc Ta: 0 ... +55° C  
(仅适用于证书编号 KEMA 10ATEX0075 X，第 9 版的现场总线组件)

## 5.2.2 ATEX – 特殊条件（扩展温度范围）

### ⚠ 警告

在潜在爆炸性区域使用具有扩展温度范围（ET）的 Beckhoff 现场总线组件，请遵守防爆 ATEX 指令（2014/34/EU）的特别规定！

- 经认证的组件应当安装在一个合适的外壳中，保证按照 EN 60079-15 标准至少达到 IP54 的防护等级！应当按此标准考虑使用过程中的环境条件！
- 关于防尘（仅指证书编号为 KEMA 10ATEX0075 X，第 9 版的现场总线组件）：考虑到设备使用的环境条件，设备应安装在一个合适的外壳中，对于 IIIA 组或 IIIB 组提供符合 EN 60079-31 标准的 IP54 防护等级，对于 IIIC 组则提供 IP6X 的防护等级！
- 如果在额定运行期间，电缆、线路或管道的进线点的温度高于 70° C，或电线分支点的温度高于 80° C，那么必须选择耐受温度数据满足实际测量温度值的线缆！
- 在潜在的爆炸性区域使用具有扩展温度范围（ET）的 Beckhoff 现场总线组件时，请遵守相关标准允许的环境温度范围 -25 至 60° C ！
- 必须采取措施，防止因瞬时干扰电压而超过额定工作电压的 40% 以上！
- 只有在关闭电源电压或确保非爆炸性环境的情况下，才可以从总线端子模块系统中拔出或拆除单个模块！
- 只有在关闭电源电压或确保非爆炸性环境的情况下，才可以连接或断开经认证部件的接线！
- 只有在关闭电源电压或确保非爆炸性环境的情况下，才可以更换 KL92xx/EL92xx 馈电端子模块的保险丝！
- 只有在关闭电源电压或确保非爆炸性环境的情况下，才能调整地址拨码和 ID 开关！

### 标准

符合下列标准规定，满足基本健康和安要求：

- EN 60079-0:2012+A11:2013
- EN 60079-15:2010
- EN 60079-31:2013（仅适用于证书编号 KEMA 10ATEX0075 X，第 9 版）

### 标志

经过 ATEX 指令认证的适用于潜在爆炸性区域的扩展温度范围（ET）Beckhoff 现场总线组件带有以下标记：



II 3G KEMA 10ATEX0075 X Ex nA IIC T4 Gc Ta: -25 ... +60° C  
 II 3D KEMA 10ATEX0075 X Ex tc IIIC T135° C Dc Ta: -25 ... +60° C  
 （仅适用于证书编号 KEMA 10ATEX0075 X，第 9 版的现场总线组件）

或



II 3G KEMA 10ATEX0075 X Ex nA nC IIC T4 Gc Ta: -25 ... +60° C  
 II 3D KEMA 10ATEX0075 X Ex tc IIIC T135° C Dc Ta: -25 ... +60° C  
 （仅适用于证书编号 KEMA 10ATEX0075 X，第 9 版的现场总线组件）



### 5.2.3 IECEX – 特殊条件

#### ⚠ 警告

**在潜在爆炸性区域使用 Beckhoff 现场总线组件，请遵守相关标准的特别规定！**

- 关于气体：考虑到设备使用的环境条件，设备应安装在合适的外壳中，保证按照 EN 60079-15 标准至少达到 IP54 的防护等级！
- 关于防尘（仅指证书编号为 IECEX DEK 16.0078X，第 3 版的现场总线组件）：考虑到设备使用的环境条件，设备应安装在一个合适的外壳中，对于 IIIA 组 或 IIIB 组提供符合 EN 60079-31 标准的 IP54 防护等级，对于 IIIC 组则提供 IP6X 的防护等级！
- 本设备只能在 IEC 60664-1 规定的污染等级不超过 2 级的区域（Zone 2）内使用！
- 应作出规定，防止因瞬时干扰造成超过额定电压 119V！
- 如果在额定运行期间，电缆、线路或管道的进线点的温度高于 70° C，或电线分支点的温度高于 80° C，那么必须选择耐受温度数据满足实际测量温度值的线缆！
- 在潜在的爆炸性区域内使用 Beckhoff 现场总线组件时，请遵守相关标准允许的环境温度范围！
- 只有在关闭电源电压或确保非爆炸性环境的情况下，才可以从总线端子模块系统中拔出或拆除单个模块！
- 只有在关闭电源电压或确保非爆炸性环境的情况下，才可以连接或断开经认证部件的接线！
- 只有在关闭电源电压或确保非爆炸性环境的情况下，才能调整地址拔码和 ID 开关！
- 只有在关闭电源电压或确保非爆炸性环境的情况下，才能打开经认证的设备的前盖！

#### 标准

符合下列标准规定，满足基本健康和 safety 要求：

- EN 60079-0:2011
- EN 60079-15:2010
- EN 60079-31:2013（仅适用于证书编号 IECEX DEK 16.0078X，第 3 版）

#### 标志

经过 IECEX 认证可在有爆炸危险区域使用的 Beckhoff 现场总线组件带有以下标记：

现场总线组件的标记，证书编号为 IECEX DEK 16.0078X，第 3 版：

**IECEX DEK 16.0078 X**  
**Ex nA IIC T4 Gc**  
**Ex tc IIIC T135° C Dc**

具有新版本证书的现场总线组件的标记：

**IECEX DEK 16.0078 X**  
**Ex nA IIC T4 Gc**

## 5.2.4 ATEX 和 IECEx 的持续性文件

### 注意



符合 ATEX 和 IECEx 的关于防爆的持续性文件

请注意持续性文件

**防爆 端子模块系统的防护**

根据 ATEX 和 IECEx 标准，在危险区域使用倍福端子模块系统的注意事项

可以在倍福公司主页 [www.beckhoff.com](http://www.beckhoff.com) 的产品下载区下载！

## 5.2.5 cFMus – 特殊条件

### 警告

在潜在爆炸性区域使用 Beckhoff 现场总线组件，请遵守相关标准的特别规定！

- 设备应安装在防护等级不低于 IP54 的外壳中，确保符合 ANSI/UL 60079-0（美国）或 CSA C22.2 No. 60079-0（加拿大）标准要求。
- 设备只能在 IEC 60664-1 规定的污染等级不超过 2 级的区域（Zone 2）内使用。
- 应提供瞬时保护，确保设备供电端子处的电压不超过额定峰值电压的 140%。
- 电路应满足 IEC 60664-1 规定的 II 类过电压保护要求。
- 只有在系统电源和现场电源均切断，或者确认当前场所为非危险区时，才可拆除或插入现场总线组件。
- 只有在系统电源切断，或者确认当前场所为非危险区时，才可断开或连接现场总线组件的接线。

### 标准

符合下列标准规定，满足基本健康和 safety 要求：

M20US0111X（美国）

- FM Class 3600:2018
- FM Class 3611:2018
- FM Class 3810:2018
- ANSI/UL 121201:2019
- ANSI/ISA 61010-1:2012
- ANSI/UL 60079-0:2020
- ANSI/UL 60079-7:2017

FM20CA0053X（加拿大）：

- CAN/CSA C22.2 No. 213-17:2017
- CSA C22.2 No. 60079-0:2019
- CAN/CSA C22.2 No. 60079-7:2016
- CAN/CSA C22.2 No. 61010-1:2012

### 标志

通过 cFMus 防爆认证、适合在防爆危险区中使用的倍福现场总线组件带有以下标志：

FM20US0111X（美国）：  
Cl. I, Div. 2, Gr. A, B, C, D  
Cl. I, Zone 2, AEx ec IIC T4 Gc

FM20CA0053X（加拿大）：  
Cl. I, Div. 2, Gr. A, B, C, D  
Ex ec T4 Gc

## 5.2.6 cFMus 防爆认证的持续性文件

### 注意



依据 cFMus 的关于防爆的持续性文件




请注意持续性文件

I/O、CX、CPX 控制图

接线图和 Ex 防爆标志，

可以在倍福公司主页 [www.beckhoff.com](http://www.beckhoff.com) 的产品下载区下载！

## 5.3 UL 声明

⚠ 谨慎	
	<p><b>应用</b></p> <p>倍福 EtherCAT 模块只适用于与具备 UL 认证的倍福 EtherCAT 系统一起使用。</p>
⚠ 谨慎	
	<p><b>检查</b></p> <p>关于 cULus 检查，仅对倍福 I/O 系统的火灾和电击风险进行了调查（符合 UL508 和 CSA C22.2 No.142 标准）。</p>
⚠ 谨慎	
	<p><b>带有以太网连接器的设备</b></p> <p>不可用于连接通信电路（telecommunication circuits）。</p>

## 基本原则

符合 UL508 的 UL 认证。有这种认证的设备带有此标志：

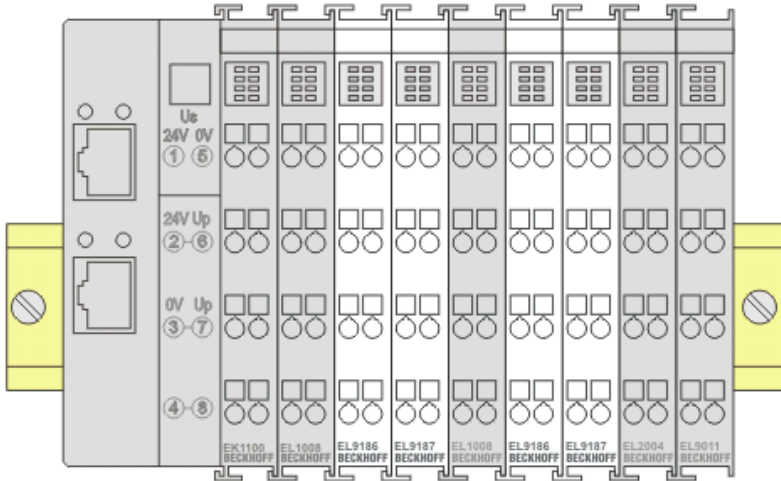


## 5.4 无通讯模块的安装位置

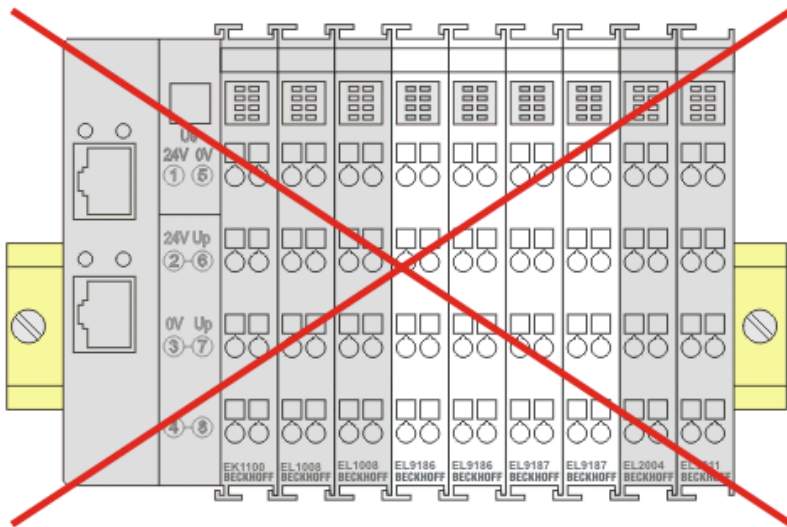
**i** 关于总线端子 I/O 站中无通讯模块安装位置的提示

那些在总线端子 I/O 站中不参与数据传输的 EtherCAT 端子模块 (ELxxxx / ESxxxx)，即所谓的无通讯模块。无通讯模块不消耗 E-Bus 的电流。为了确保最佳的数据传输，不能直接把超过两个的无通讯模块连续并列装在一起！

无通讯模块安装位置示例（高亮显示）



附图 15: 正确的安装位置



附图 16: 错误的安装位置

## 5.5 增强抗振模块的安装说明

### ⚠ 警告

#### 有触电和损坏设备的危险！

在开始安装、拆卸或连接总线端子模块之前，应将总线端子模块系统置于安全、断电的状态！

#### 额外检查

这些端子模块经过了以下额外测试：

验证	说明
振动	在 3 个坐标轴方向上各 10 次
	6 Hz < f < 60 Hz, 位移 0.35 mm, 恒定振幅
	60.1 Hz < f < 500 Hz, 加速度 5 g, 恒定振幅
冲击	在 3 个坐标轴的每个方向上各 1000 次冲击
	25 g, 6 ms

#### 附加安装说明

对于增强抗振模块，适用以下附加的安装说明：

- 增强抗振能力适用于所有允许的安装位置
- 使用符合 EN 60715 TH35-15 标准的安装导轨
- 用机械固定装置将一组端子模块从安装导轨的两端固定住，例如接地端子或加固的端部堵头
- 一个端子模块 I/O 站的最大模块数量（不含耦合器）是：  
64 个端子模块（12 mm）或 32 个端子（24 mm）
- 在安装导轨的加工和安装的过程中，应避免变形、扭曲、受挤压和弯曲
- 安装导轨的固定点必须以 5 厘米的间隔设置
- 使用沉头螺钉固定安装导轨
- 电缆的固定环和连接导线之间的自由长度应保持尽可能短。与线槽应保持约 10 厘米的距离。

## 5.6 安装位置

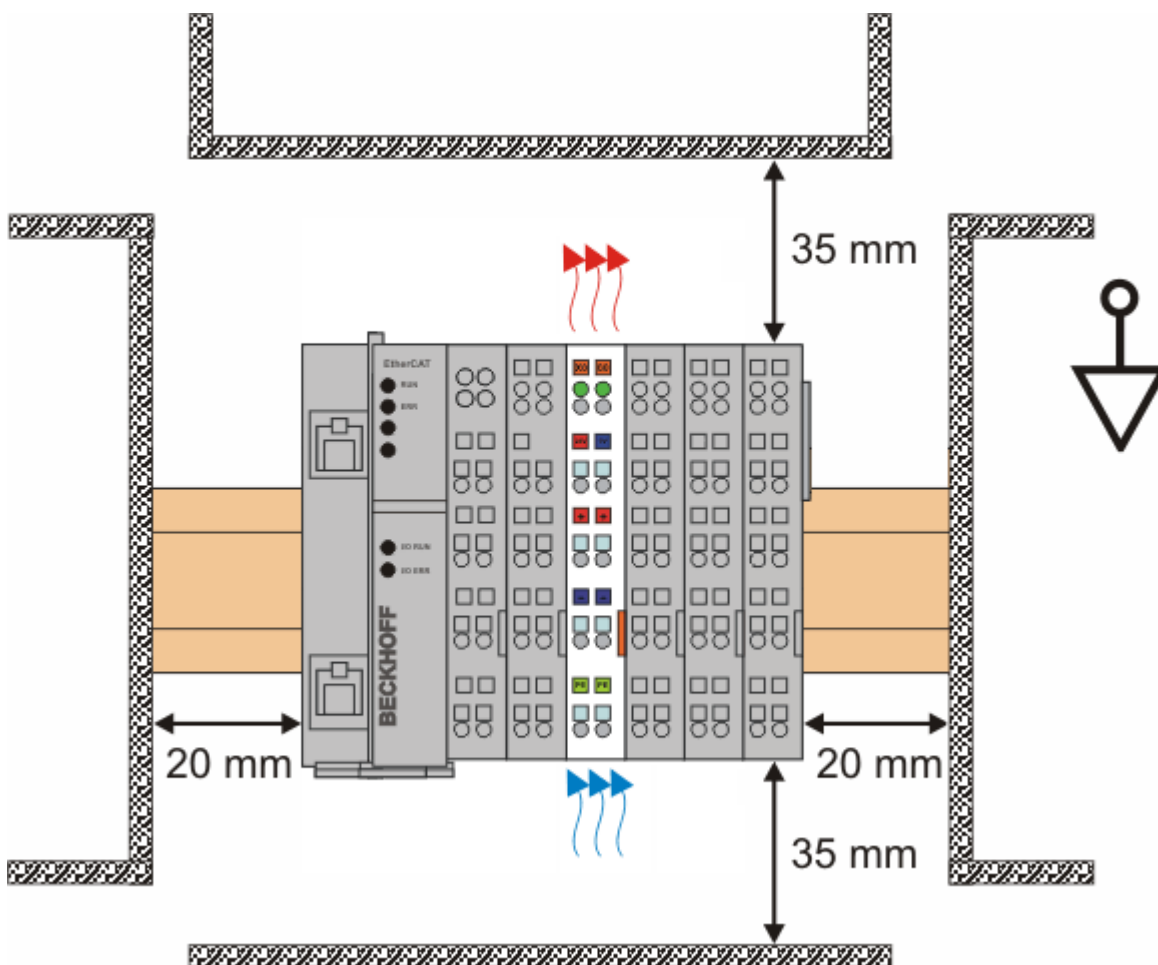
### 注意

#### 关于安装位置和工作温度范围的限制

请参考端子模块的技术数据，以确定是否规定了关于安装位置和/或工作温度范围的任何限制。在安装高功率耗散的端子模块时，确保在端子模块上方和下方的其他部件之间保持足够的间距，以保证充分的通风！

#### 最佳安装位置（标准）

最佳的安装位置是安装导轨水平安装，EL/KL 端子模块接线的一面朝前（见图标准安装位置的推荐距离）。从端子模块的下面通风，通过对流实现电子元件的最佳冷却。“从下面”是指相对于重力方向而言。



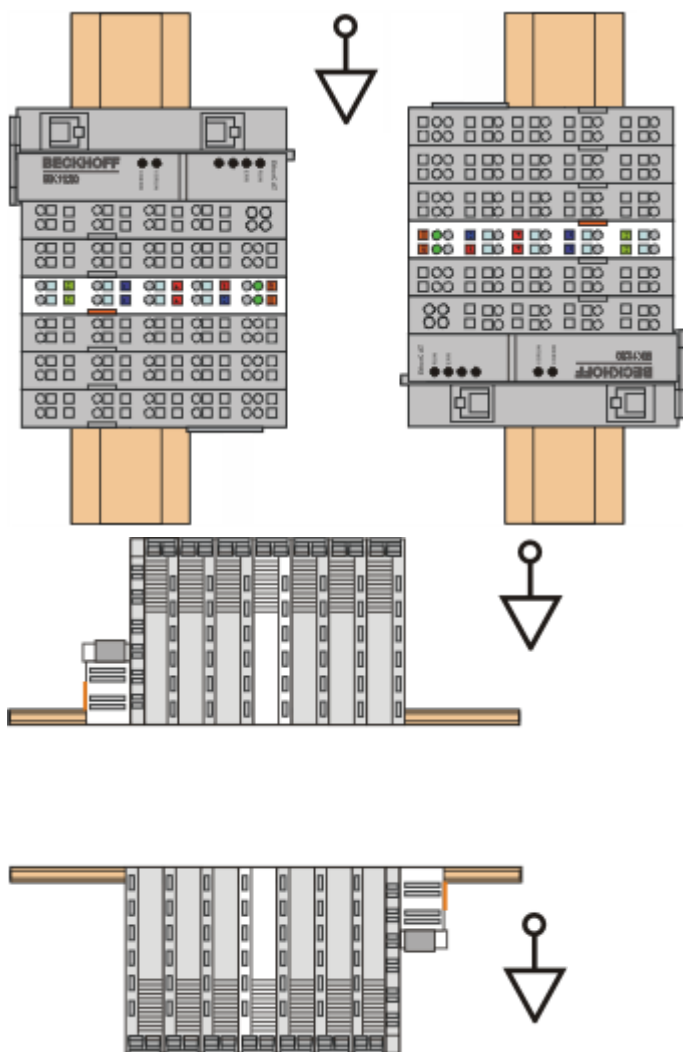
附图 17: 标准安装位置的推荐距离

建议遵守图 标准安装位置的推荐距离 中所示的距离。

#### 其他安装位置

所有其他安装位置的特点是安装导轨的空间布局不同，参见图 其他安装位置。

上面规定的与其它部件的最小距离也适用于这些安装位置。



附图 18: 其他安装位置



## 5.7 注意事项 - 电源

### 警告

#### 从 SELV/PELV 电源单元供电!

必须使用符合 IEC 61010-2-201 的 SELV/PELV 电路（安全超低电压 Safety Extra Low Voltage，保护超低电压 Protective Extra Low Voltage）为本设备供电。

注意事项：

- SELV/PELV 电路可能会引起 IEC 60204-1 等标准的进一步要求，例如关于电缆间距和绝缘。
- SELV（安全超低电压）电源提供安全的电气隔离和电压限制，而不需要连接到保护导体，PELV（保护性超低电压）电源也需要安全连接到保护导体。

## 5.8 EL6001、EL6021-00xx

### 5.8.1 安装在导轨上

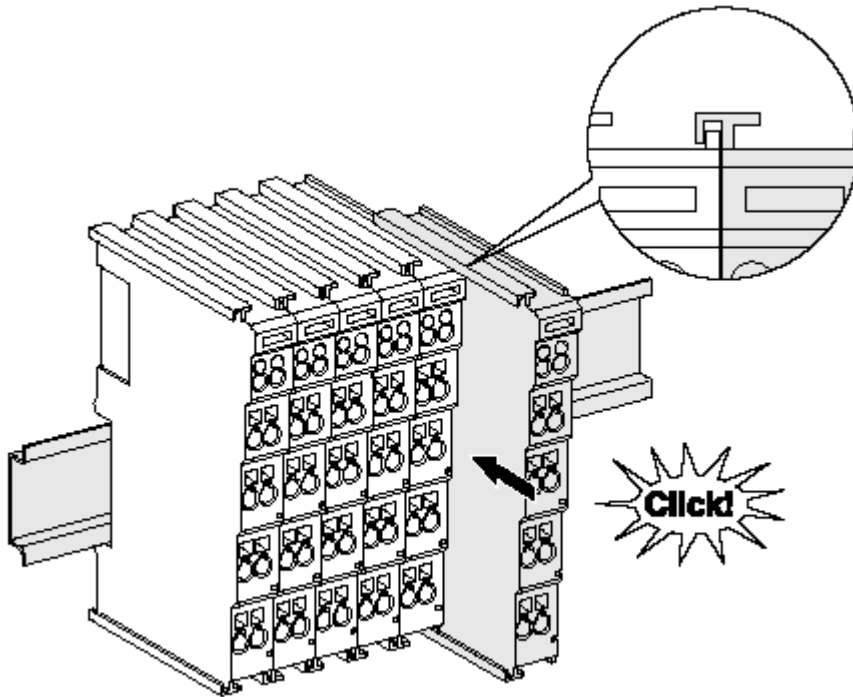
#### ⚠ 警告

**有触电和损坏设备的危险！**

在开始安装、拆卸或连接总线端子模块之前，请将总线端子模块系统带入一个安全的、断电的状态！

总线端子模块系统设计用于安装在控制柜或接线盒中。

#### 组装



附图 19：安装在安装轨道上

总线耦合器和总线端子模块通过施加轻微压力安装到市售 35 毫米安装导轨（符合 EN 60715 标准的 DIN 导轨）上：

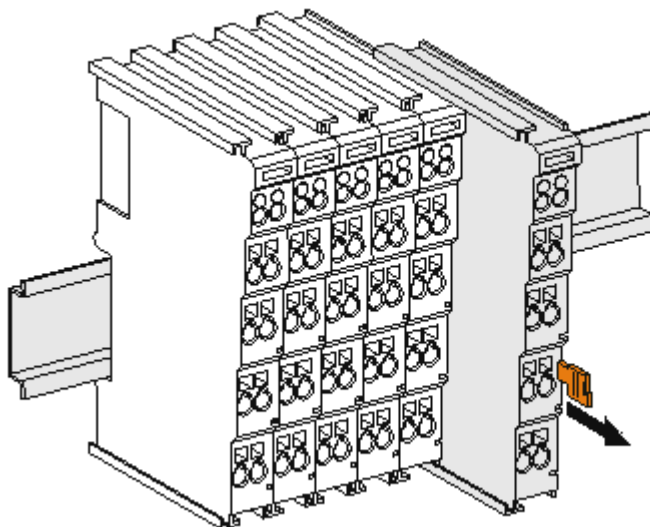
1. 首先将现场总线耦合器安装在安装导轨上。
2. 现在，总线端子模块安装在现场总线耦合器的右侧。用榫卯连接组件，将端子模块推到安装轨道上，直到锁扣卡在安装轨道上。  
如果端子模块先卡在安装轨道上，然后在没有榫头的情况下推到一起，那么连接将无法运行！正确组装后，外壳之间不应看到明显的间隙。

#### ● 安装导轨的固定

**i**

端子模块和耦合器的锁定机制延伸至安装轨道的轮廓。在安装时，组件的锁定机制不能与安装轨道的固定螺栓发生冲突。为了在端子模块和耦合器下面安装高度为 7.5 毫米的安装导轨，应该使用平坦的安装连接（如沉头螺钉或盲铆钉）。

## 拆卸



附图 20: 端子模块的拆卸

每个端子模块都由安装轨道上的锁扣固定，拆卸时必须松开锁扣：

1. 用橙色的接线柱拉动端子模块，使其离开安装轨道约 1 厘米。在这样做的时候，该端子模块的安装导轨锁扣会自动松开，您可以轻松地将该端子模块从总线端子排中拉出来，而不需要过度用力。
2. 用拇指和食指同时抓住松开的端子模块的上、下凹槽外壳表面，将端子模块从总线端子排中拉出。

## 一个总线端子 I/O 站内的连接

总线耦合器和总线端子模块之间的电气连接通过连接部件自动实现：

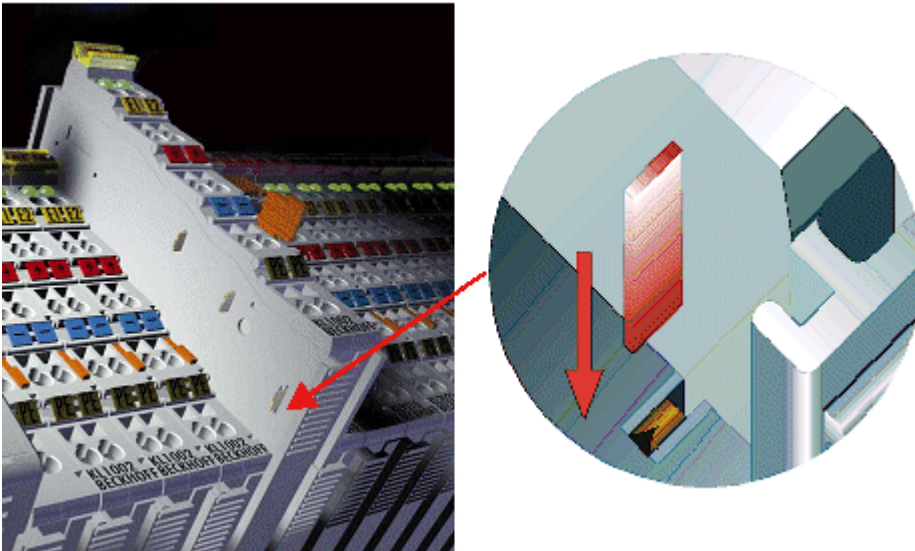
- 通过 K-Bus/E-Bus 的六个弹簧触点实现数据的传输和总线端子模块供电。
- 电源触点给现场电子设备供电，因此形成了总线端子 I/O 站内的一个电源母线。电源触点通过总线耦合器上的端子供电（最高 24V），或者对于更高的电压则通过电源馈电端子模块供电。

### ● 电源触点

**i** 在设计总线端子 I/O 站时，必须考虑到各个总线端子模块的引脚分配，因为有些类型（如模拟量总线端子模块或数字量 4 通道总线端子模块）没有或没有完整的通过电源触点的回路。电源馈电端子模块（KL91xx、KL92xx 或 EL91xx、EL92xx）中断了前面电源触点形成的母线，而提供了一个新的电源母线的起点。

### PE 电源触点

标有 PE 的电源触点可作为安全接地使用。出于安全考虑，在两个相邻模块插接时 PE 触点会首先对接连通。PE 触点可以将高达 125 A 的短路电流导入大地。



附图 21: 左侧的电源触点

**注意****设备可能的损坏**

请注意，出于电磁兼容性的考虑，PE 触点与安装导轨是电容耦合的。这可能会导致在绝缘测试中出现错误的结果，或者导致端子模块损坏（例如，在对额定电压为 230V 的用电设备进行绝缘测试时，对 PE 线进行破坏性放电）。在进行绝缘测试时，请在总线耦合器或电源馈电端子模块处断开 PE 电源线！为了使关联的馈电点都脱离后再进行测试，这些电源馈电端子模块可以松开，并从端子组中拉出至少 10 mm。

**警告****有触电的危险！**

PE 电源接点不得用于其他电位！

## 5.8.2 接线

### 5.8.2.1 接线系统

#### ⚠ 警告

#### 有触电和损坏设备的危险!

在开始安装、拆卸或连接总线端子模块之前，请将总线端子模块系统带入一个安全的、断电的状态!

#### 概述

总线端子模块系统提供不同的连接方式，以便根据各个具体的应用场合进行最佳选择：

- ELxxxx 和 KLxxxx 系列，带标准接线端子模块的外壳中集成了各种电子元件和接线装置。
- ESxxxx 和 KSxxxx 系列，具有可插拔的接线座，并能在模块更换时避免拆除和重新接线。
- High Density Terminals 高密度端子模块（HD 端子模块），在一个外壳中集成了各种电子元件和接线装置，具有较高的封装密度。

#### 标准接线（ELxxxx / KLxxxx）



附图 22：标准接线

ELxxxx 和 KLxxxx 系列端子模块经过多年的使用和测试。特征是集成了免螺钉弹簧动力技术，可以快速和简单地接线。

#### 可插拔接线（ESxxxx / KSxxxx）



附图 23：可插拔接线

ESxxxx 和 KSxxxx 系列端子模块的特征是有一个可插拔的接线座。组装和接线过程与 ELxxxx 和 KLxxxx 系列相同。可插拔的接线座使得维护时可以把全部接线作为一个插拔连接器从外壳顶部拆卸下来。通过拉动解锁片，可以将模块下半部从 I/O 站中拆出来。装入新的组件并插入带接线的连接器。这样一来，可大大减少安装时间，避免接线错误。

常见端子模块的尺寸只有一点点变动。新的接线座在深度方向增加了约 3 mm。而端子模块的最大高度仍保持不变。

电缆的固定环可简化很多应用中的装配工作，防止在拆除接线座时发生连接线缠结在一起的现象。

截面积为 0.08 mm<sup>2</sup> 至 2.5 mm<sup>2</sup> 之间的导线仍采用弹簧连接技术。

ESxxxx 和 KSxxxx 系列整体保留了 ELxxxx 和 KLxxxx 系列产品的命名。

### 高密度端子模块 (HD 端子模块)



附图 24: 高密度端子模块

该系列端子模块有 16 个接线点，特点是设计特别紧凑，因为其封装密度是标准 12 毫米模块的两倍。大线径导线和带管型端子的导线可以直接插入弹簧式接线点，不需要工具。

---

#### ● 高密度端子模块的接线



ELx8xx 和 KLx8xx 系列高密度端子模块不支持可插拔式接线。

---

### 超声“粘合”（超声焊接）导线

---

#### ● 超声“粘合”导线



标准模块和高密度端子模块也可以使用超声“粘合”（超声焊接）导体来接线。此时请注意以下有关导线尺寸宽度 [► 55] 的表格！

---

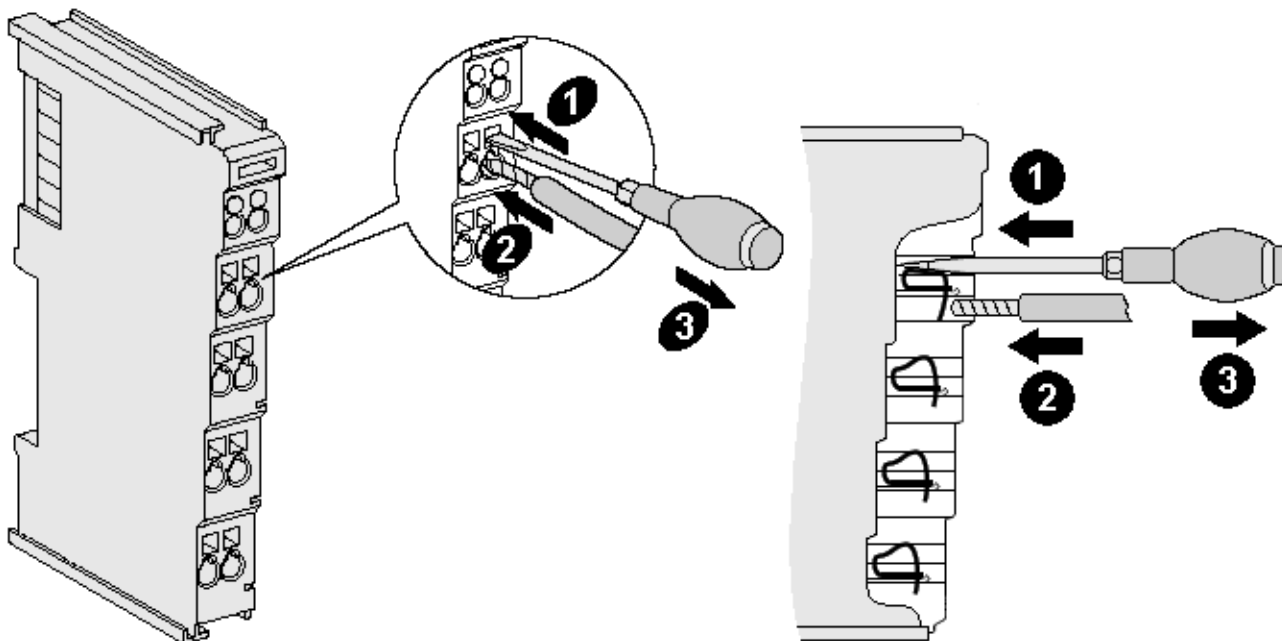
## 5.8.2.2 接线

## ⚠ 警告

有触电和损坏设备的危险！

在开始安装、拆卸或连接总线端子模块之前，请将总线端子模块系统带入一个安全的、断电的状态！

用于标准接线的端子模块 ELxxxx/KLxxxx 和用于可插拔接线的端子模块 ESxxxx/KSxxxx



附图 25: 在一个接线点上连接线缆

总线模块上最多提供 8 个接线点，用于连接单芯线缆或细绞线。接线点采用弹簧动力技术。按以下方式连接导线：

1. 将螺丝刀插入接线点上方的方形开口，一直插到底，使接线点张开。螺丝刀不要转动或上下移动（不要撬动）。
2. 然后将导线插入端子模块的圆形开口，不需要用力。
3. 拔出螺丝刀，接线点会自动闭合，永久地牢牢固定住接线。

端子模块适合的导线规格见下表。

端子模块外壳	ELxxxx, KLxxxx	ESxxxx, KSxxxx
导线规格（单芯线）	0.08 ... 2.5 mm <sup>2</sup>	0.08 ... 2.5 mm <sup>2</sup>
导线规格（细导线）	0.08 ... 2.5 mm <sup>2</sup>	0.08 ... 2.5 mm <sup>2</sup>
导线规格（带管型端子的导线）	0.14 ... 1.5 mm <sup>2</sup>	0.14 ... 1.5 mm <sup>2</sup>
剥线长度	8 ... 9 mm	9 ... 10 mm

高密度端子模块（HD Terminals [► 54]）有 16 个接线点

如果是单芯导线，则 HD 端子模块的接线采用直接插入的方式，不需要工具，即剥线后只需将其插入接线点。需要松开导线时，也像标准模块一样，用螺丝刀插入接线点上方的方形开口，直插到底，即可拔出电缆。端子模块适合的导线规格见下表。

端子模块外壳	高密度外壳
导线规格 (单芯线)	0.08 ... 1.5 mm <sup>2</sup>
导线规格 (细导线)	0.25 ... 1.5 mm <sup>2</sup>
导线规格 (带管型端子的导线)	0.14 ... 0.75 mm <sup>2</sup>
导线规格 (超声“粘合”导线)	仅 1.5 mm <sup>2</sup> (见注意事项 [▶ 54])
剥线长度	8 ... 9 mm

### 5.8.2.3 屏蔽

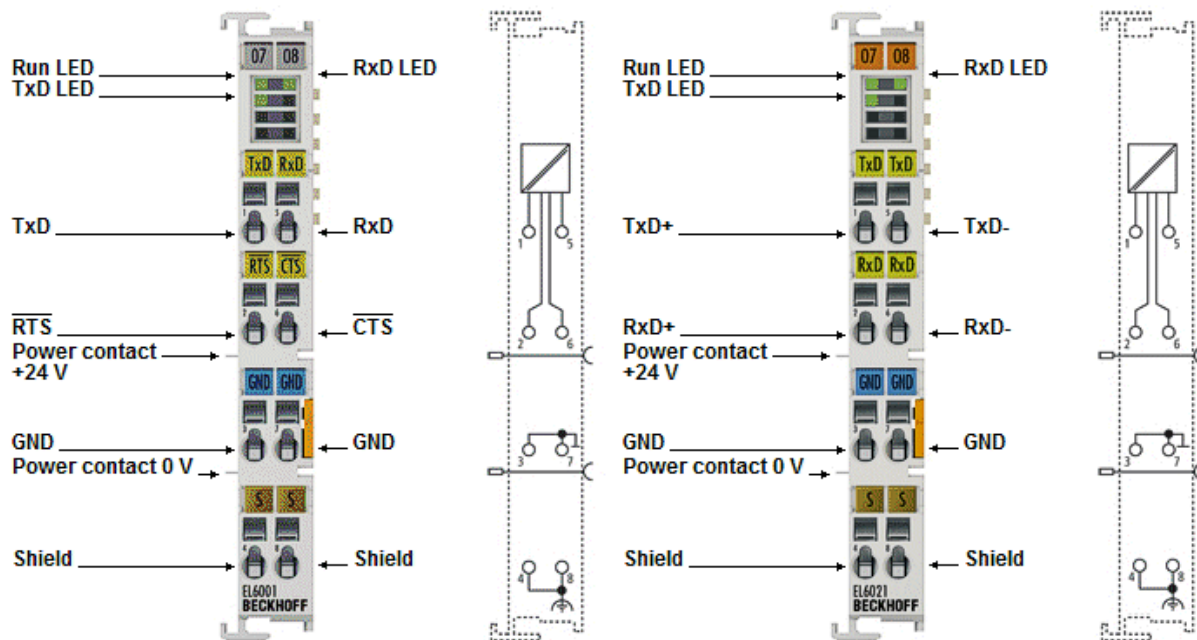


#### 屏蔽

编码器、模拟量传感器和执行器的接线应当始终使用屏蔽双绞线。



### 5.8.3 LED 指示灯和引脚分配



附图 26: EL6001、EL6021 - LED 指示灯和引脚分配

#### LED 指示灯

LED	颜色	含义	
RUN	绿色	该 LED 指示端子模块的工作状态:	
		熄灭	EtherCAT 状态机 [▶ 30] 的状态: INIT = 端子模块的初始化或 BOOTSTRAP = 用于端子模块固件更新 [▶ 204] 的功能
		闪烁	EtherCAT 状态机的状态: PREOP = 执行邮箱通信和设置各种参数
		单次闪烁	EtherCAT 状态机的状态: SAFEOP = 验证 Sync Manager [▶ 115] 通道和分布时钟。输出保持安全状态
	常亮	EtherCAT 状态机的状态: OP = 正常运行状态; 可以进行邮箱和过程数据通信	
TxD	绿色	发送信号线的状态 (常亮: 发送信号线上的信号为高电平)	
RxD	绿色	接收信号线的状态 (常亮: 接收信号线上的信号为高电平)	

#### EL6001 端子模块连接器的分配

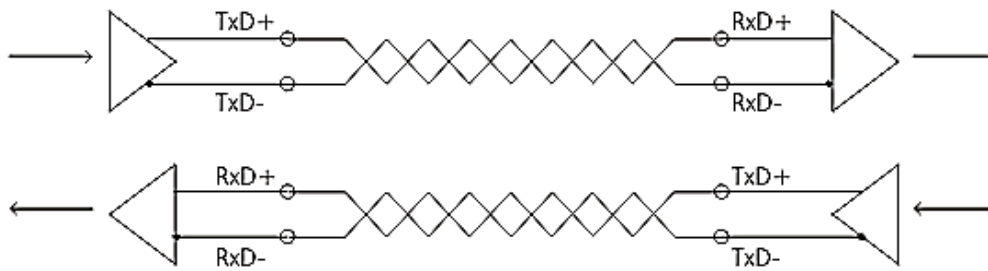
端子模块接点	名称	信号
1	TxD	信号线 (发送数据)
5	RxD	信号线 (接收数据)
2	RTS	控制线 (请求发送)
6	CTS	控制线 (清除发送)
3	GND	接地 (与端子 7 内部短接)
7	GND	接地 (与端子 3 内部短接)
4	Shield	屏蔽 (与端子 8 内部短接)
8	Shield	屏蔽 (与端子 4 内部短接)

## EL6021-00xx 端子模块连接器的分配

端子模块接点	名称	信号
1	TxD+	信号线 + (发送数据)
5	TxD-	信号线 - (发送数据)
2	RxD+	信号线 + (接收数据)
6	RxD-	信号线 - (接收数据)
3	GND	接地 (与端子 7 内部短接)
7	GND	接地 (与端子 3 内部短接)
4	Shield	屏蔽 (与端子 8 内部短接)
8	Shield	屏蔽 (与端子 4 内部短接)

### RS422 传输的连接

在 RS422 模式下，数据可以全双工模式传输。



附图 27: RS422 传输的连接

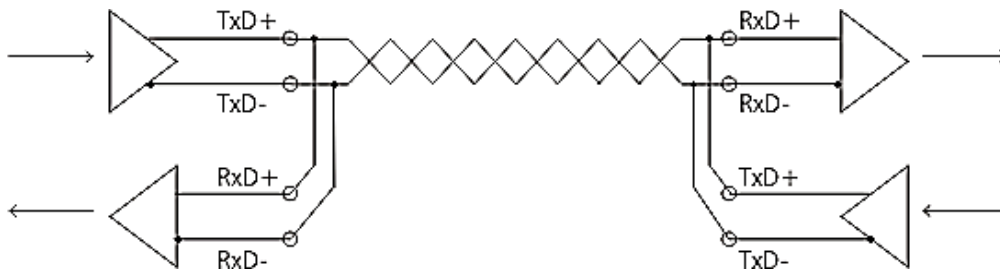
建议 RS422 线路只使用一个接收设备，但多达 10 个接收设备也很常见。网络越复杂（设备、分支、插头连接），调试就越复杂，因为必须确定适当的参数才能实现可靠的通信：终端设置（位置和距离）、波特率、线路长度（如有必要）。

强烈建议采用具有短分支的清晰线型拓扑（MultiDrop），必要时将距离最远的两个端点设为终端。

### RS485 传输的连接

在 RS485 运行模式下，数据交换采用半双工传输方式。在这种运行模式下，可以建立一个由多个相互通信的站点组成的总线结构。

建议采用分支较短的清晰线型拓扑。



附图 28: RS485 传输的连接

在 RS485 运行模式下，传输和接收线路相互连接。因此，端子模块不仅能接收来自其它设备的数据，还能接收自己发送的数据。可以通过 Settings 对象中的 Index 0x8000:06 “Enable half duplex” 来抑制该功能。

在 RS485 运行模式下，只有在传输完成后才能接收新数据。

“Enable half duplex”	“Enable point to point connection (RS422)”	模式
0	0	RS485: 端子模块接收自身数据和来自其它设备的数据
0	1	RS422: 正常运行模式；端子模块以全双工模式运行。
1	0	RS485: 端子模块仅接收来自其它设备的数据
1	1	RS422: 只有在最后一个数据传输完毕后，接收设备才会启用。

## 5.9 EL6002、EL6022

### 5.9.1 安装和拆卸 – 正面带拆卸手柄的端子模块

端子模块借助 35 mm 安装导轨（如安装导轨 TH 35-15）固定在装配面上。

#### ● 安装导轨的固定

**i** 端子模块和耦合器的锁紧部件延伸至安装导轨。在安装时，模块的锁紧部件不能顶住安装导轨的固定螺栓。如果导轨的安装孔位刚好在端子模块和耦合器的正下方，应该使用安装面平齐的方式（如沉头螺钉或盲孔铆钉）。

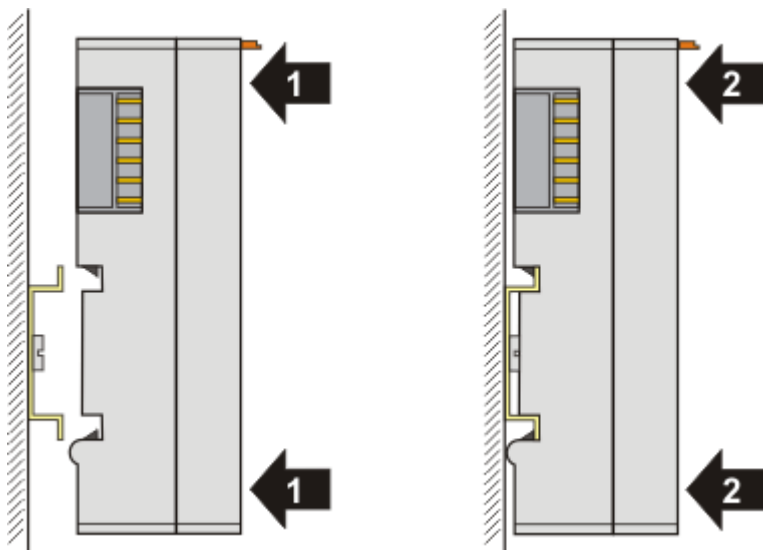
#### ⚠ 警告

**有触电和损坏设备的危险！**

在开始安装、拆卸或给总线端子模块接线之前，总线端子模块系统应当处于安全、断电的状态！

#### 安装

- 将安装导轨安装到预定的装配位置。

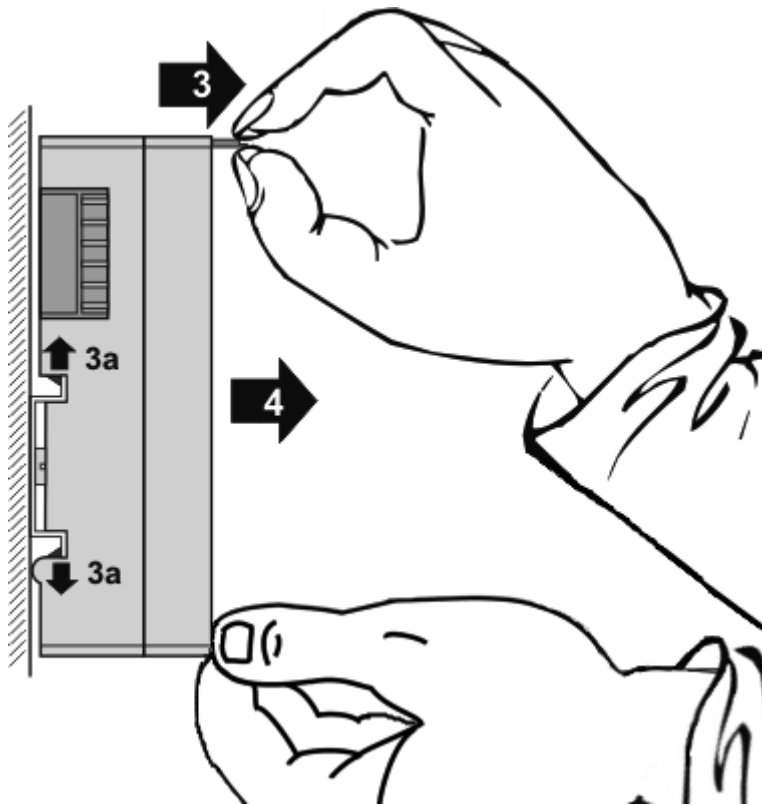


然后将接线端子模块向安装导轨上按压（1），直至模块在安装导轨上锁紧（2）。

- 连接电缆。

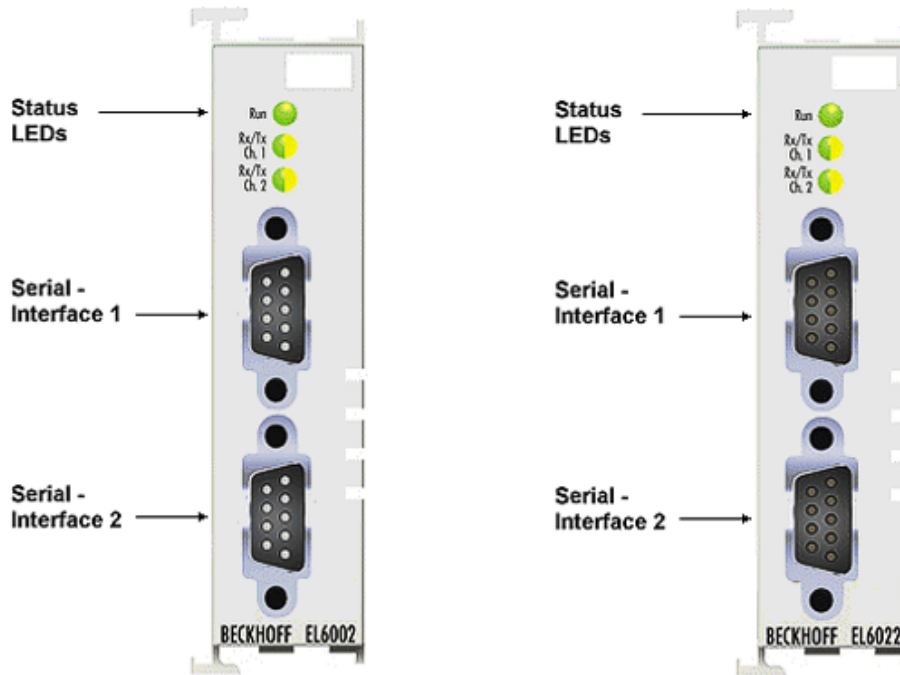
#### 拆卸

- 拆除所有电缆。
- 用拇指和食指抽出解锁钩（3）。内部机构将两个锁耳从顶帽导轨拉回接线端子模块（3a）。



- 从安装面上拔下端子模块（4）。  
避免模块倾斜；如有必要，应以另一只手稳住模块。

### 5.9.2 LED 指示灯 和引脚分配



附图 29: E16002、EL6022 - LED

#### LED 指示灯

LED	颜色	含义	
RUN	绿色	该 LED 指示端子模块的工作状态:	
		熄灭	EtherCAT 状态机 [▶ 30] 的状态: INIT = 端子模块的初始化或 BOOTSTRAP = 用于端子模块固件更新 [▶ 204] 的功能
		闪烁	EtherCAT 状态机的状态: PREOP = 执行邮箱通信和设置各种参数
		单次闪烁	EtherCAT 状态机的状态: SAFEOP = 验证 Sync Manager [▶ 115] 通道和分布时钟。 输出保持安全状态
		常亮	EtherCAT 状态机的状态: OP = 正常运行状态; 可以进行邮箱和过程数据通信
TxCh. 1	橙色	该端子模块的串口正在发送数据 (通道 1)	
RxCh. 1	绿色	该端子模块的串口正在接收数据 (通道 1)	
TxCh. 2	橙色	该端子模块的串口正在发送数据 (通道 2)	
RxCh. 2	绿色	该端子模块的串口正在接收数据 (通道 2)	

EL6002 引脚分配

2 个 D 形 9 脚 针连接器

D 形 针连接器 (俯视图)	通道 1 引脚分配	通道 2 引脚分配
	<b>引脚 RS232</b> 1 DCD (内部短接引脚 4 和 6) 2 RxCH1 3 TxCH1 4 DTR (内部短接引脚 1 和 6) 5 GND 6 DSR (内部短接引脚 1 和 4) 7 RTS CH1 8 CTS CH1 9 -	<b>引脚 RS232</b> 1 DCD (内部短接引脚 4 和 6) 2 RxCH2 3 TxCH2 4 DTR (内部短接引脚 1 和 6) 5 GND 6 DSR (内部短接引脚 1 和 4) 7 RTS CH2 8 CTS CH2 9 -

**i** GND 连接

两个通道的 GND 通过高阻态 RC 组合进行内部连接

EL6022 引脚分配

2 个 D 形 9 脚 针连接器 (公头)

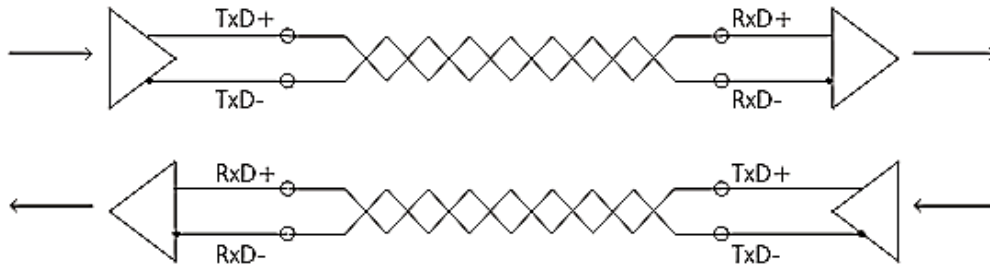
D 形 针连接器 (俯视图)	通道 1 引脚分配	通道 2 引脚分配
	<b>引脚 RS485/RS422</b> 1 - 2 Tx+ CH1 3 Rx+ CH1 4 - 5 GND 6 +5 V 7 Tx- CH1 8 Rx- CH1 9 -	<b>引脚 RS485/RS422</b> 1 - 2 Tx+ CH2 3 Rx+ CH2 4 - 5 GND 6 +5 V 7 Tx- CH2 8 Rx- CH2 9 -

**i** GND 连接

两个通道的 GND 通过高阻态 RC 组合进行内部连接

### RS422 传输的连接

在 RS422 模式下，数据可以全双工模式传输。



附图 30: RS422 传输的连接

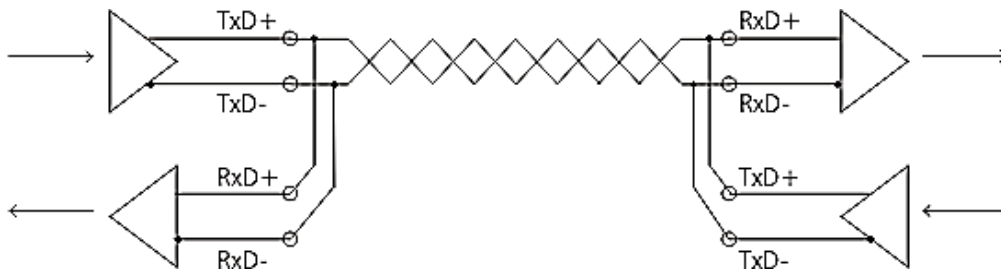
建议 RS422 线路只使用一个接收设备，但多达 10 个接收设备也很常见。网络越复杂（设备、分支、插头连接），调试就越复杂，因为必须确定适当的参数才能实现可靠的通信：终端设置（位置和距离）、波特率、线路长度（如有必要）。

强烈建议采用具有短分支的清晰线型拓扑（MultiDrop），必要时将距离最远的两个端点设为终端。

### RS485 传输的连接

在 RS485 运行模式下，数据交换采用半双工传输方式。在这种运行模式下，可以建立一个由多个相互通信的站点组成的总线结构。

建议采用分支较短的清晰线型拓扑。



附图 31: RS485 传输的连接

在 RS485 运行模式下，传输和接收线路相互连接。因此，端子模块不仅能接收来自其它设备的数据，还能接收自己发送的数据。可以通过 Settings 对象中的 Index 0x8000:06 “Enable half duplex” 来抑制该功能。

在 RS485 运行模式下，只有在传输完成后才能接收新数据。

“Enable half duplex”	“Enable point to point connection (RS422)”	模式
0	0	RS485: 端子模块接收自身数据和来自其它设备的数据
0	1	RS422: 正常运行模式；端子模块以全双工模式运行。
1	0	RS485: 端子模块仅接收来自其它设备的数据
1	1	RS422: 只有在最后一个数据传输完毕后，接收设备才会启用。



## 5.10 处理



标有带叉轮式垃圾桶的产品不得与普通垃圾一起丢弃。该设备被认为是废弃的电气和电子设备。必须遵守国家废弃电气和电子设备的处理规定。

## 6 调试

### 6.1 TwinCAT 快速入门

TwinCAT 是实时 (real-time) 控制器的开发环境, 包括多 PLC 系统、NC 轴控制系统的编程和操作。通过 TwinCAT 可以进行整个系统的映射, 并能够访问控制器的编程环境 (包括编译)。也可以直接读取或写入单个数字量/模拟量的输入或输出, 例如为了验证其功能。

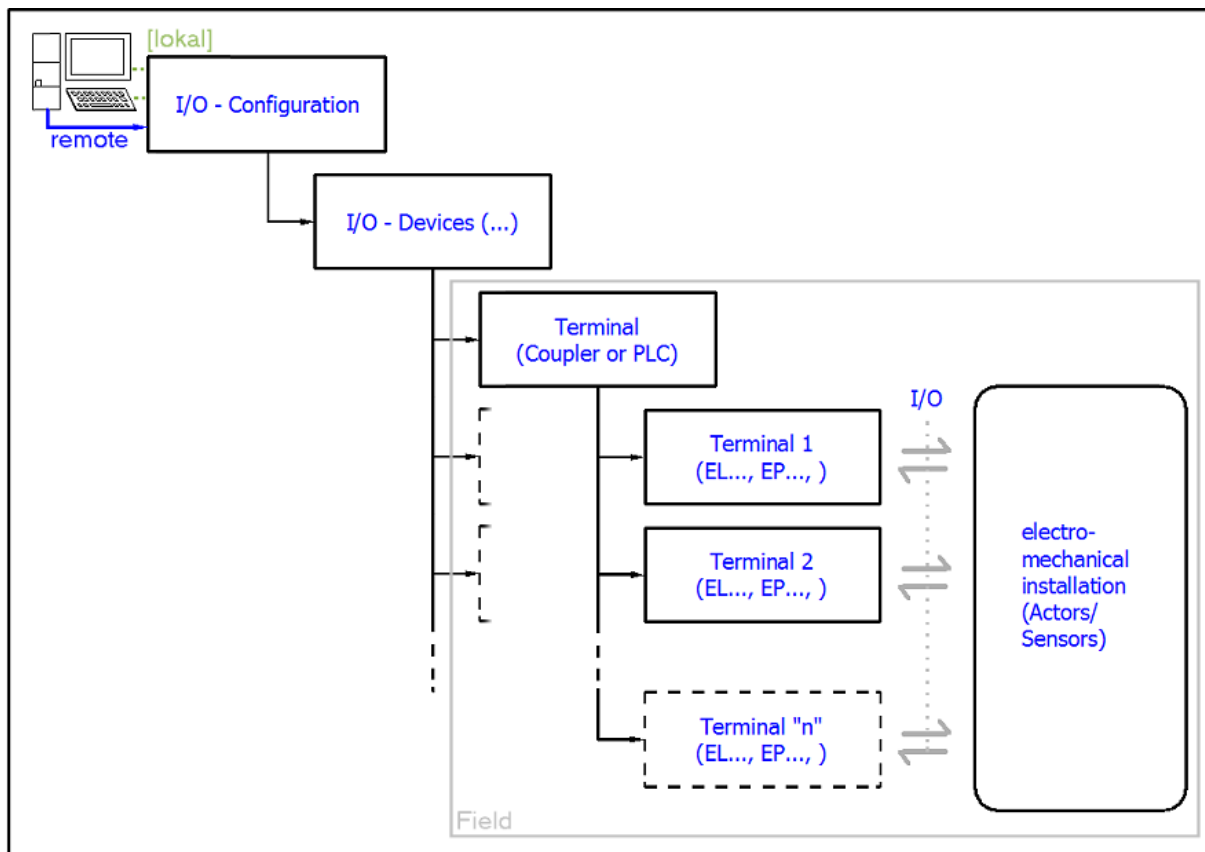
更多信息请参考 <http://infosys.beckhoff.com>:

- **EtherCAT 系统手册:**  
Fieldbus Components → EtherCAT Terminals → EtherCAT System Documentation → Setup in the TwinCAT System Manager
- **TwinCAT 2** → TwinCAT System Manager → I/O Configuration
- 特别是 TwinCAT 驱动程序的安装:  
**Fieldbus components** → Fieldbus Cards and Switches → FC900x - PCI Cards for Ethernet → Installation

Devices 下包含了实际配置的相关端子模块。所有配置数据的输入可以直接通过编辑功能 (离线) 或通过 “扫描 (Scan)” 功能 (在线):

- “**offline**”: 通过添加和放置单个组件来自定义配置, 可以从一个目录中选择并配置。
  - 关于离线模式的步骤, 请参见 <http://infosys.beckhoff.com>:  
**TwinCAT 2** → TwinCAT System Manager → IO Configuration → Add an I/O device
- “**online**”: 读取现有的硬件配置
  - 另请参见 <http://infosys.beckhoff.com>:  
**Fieldbus components** → Fieldbus Cards and Switches → FC900x - PCI Cards for Ethernet → Installation → Searching for devices

从用户的配置电脑到控制器及下属组件的层次关系如下:



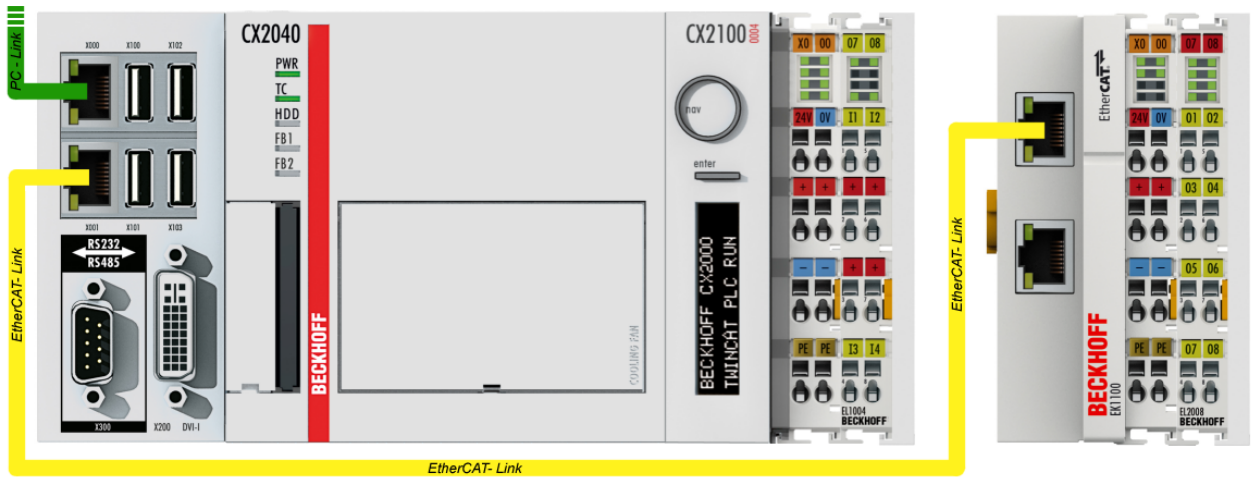
附图 32: 用户侧 (调试) 和实际安装组件之间的关系

在 TwinCAT 2 和 TwinCAT 3 中，用户插入某些组件（I/O device, terminal, box...）的方式相同。下列描述仅涉及在线操作过程。

**配置示例（实际配置）**

基于下面的配置示例，后面的小节描述了 TwinCAT 2 和 TwinCAT 3 的操作过程：

- CX2040 控制系统（PLC）包括 CX2100-0004 电源单元
- 连接到 CX2040 的右边（E-bus）：  
EL1004（4 通道数字量输入端子模块 24 V<sub>DC</sub>）
- 通过 X001端口（RJ-45）连接：**EK1100** EtherCAT 耦合器
- 连接到 EK1100 EtherCAT 耦合器的右边（E-bus）：  
EL2008（8 通道数字量输出端子模块 24 V<sub>DC</sub>; 0.5 A）
- （X000 可选连接外部 PC，提供用户接口）



附图 33：系统配置，含嵌入式控制器、输入（EL1004）和输出（EL2008）

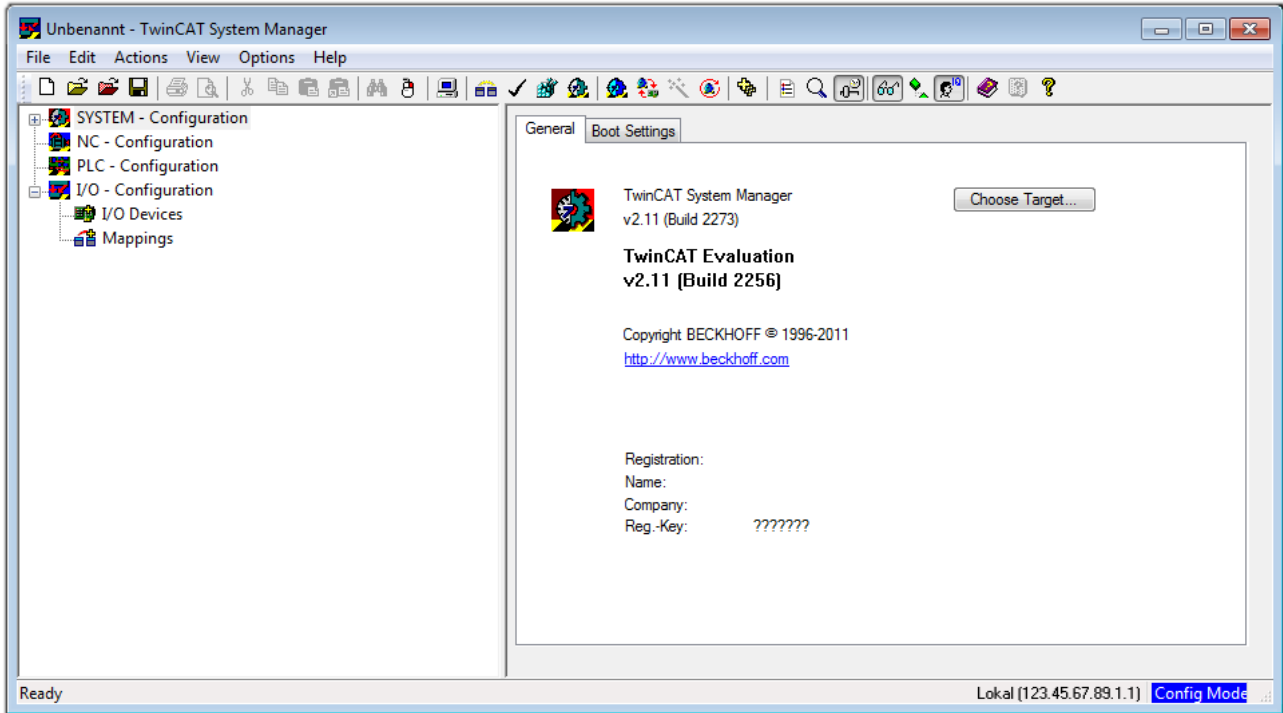
请注意，一个配置可能有多个任意组合；例如，EL1004 端子模块可以接在耦合器之后，或者 EL2008 端子模块可以附加到 CX2040 的右侧而无须 EK1100 耦合器。

## 6.1.1 TwinCAT 2

### 启动

TwinCAT 2 有两个用户界面：TwinCAT System Manager 用于硬件配置；TwinCAT PLC Control 用于控制程序的开发和编译。项目开发通常从 TwinCAT System Manager 开始。

在开发 PC 上成功安装 TwinCAT 系统后，启动 TwinCAT 2 System Manager，显示用户界面如下：



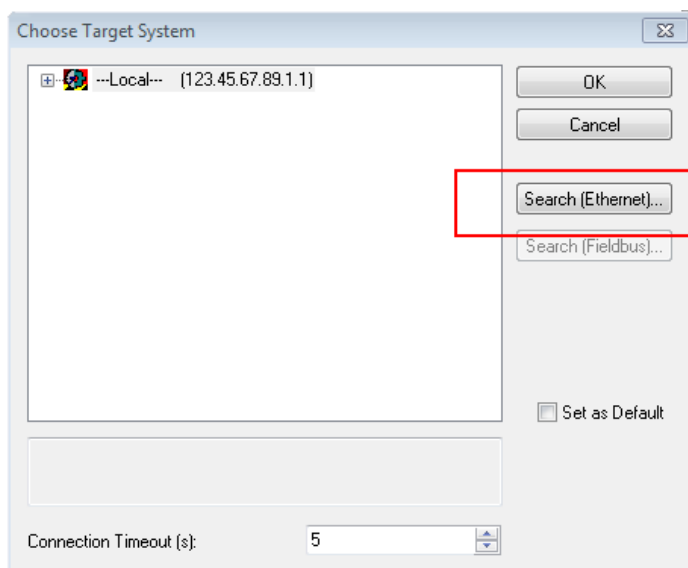
附图 34: TwinCAT 2 的初始用户界面

一般来说，TwinCAT 可以工作在本地或远程模式。如果目标 PLC 上安装了 TwinCAT 系统（含用户界面的标准开发环境），TwinCAT 就可以工作在本地模式。这种情况下，下一个步骤是“Insert Device (插入设备) [▶\_70]”。

如果要从 TwinCAT 开发环境连接到另一个安装在远程 PLC 上的 TwinCAT Runtime 运行环境，就得先识别到 Target System(目标系统)。在菜单中的“Actions(行动)”→“Choose Target System...”项下，通过符号



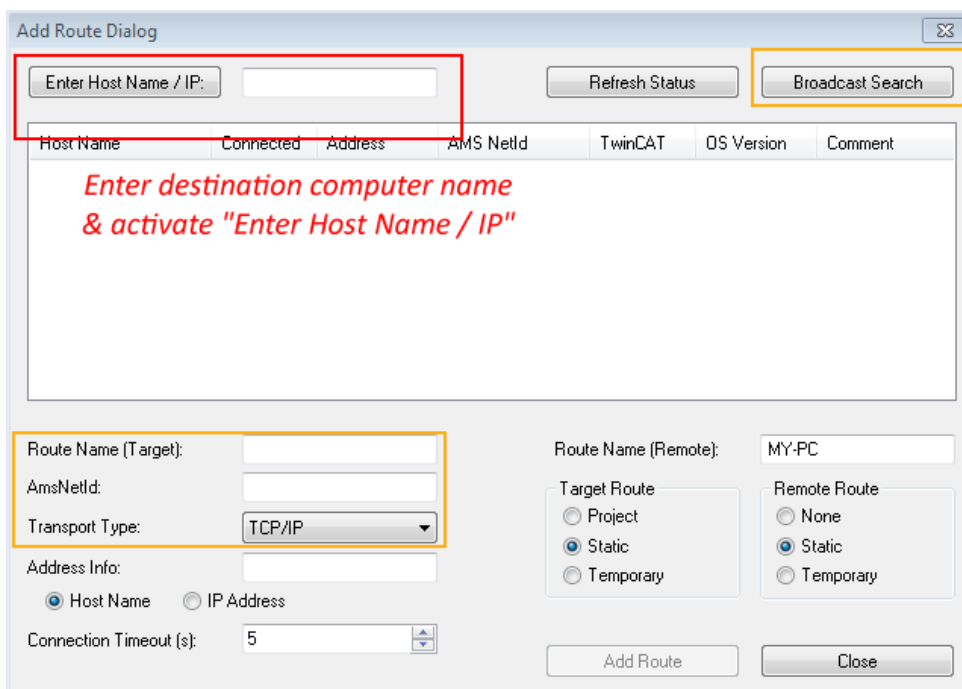
“ ”或“F8”键，打开以下窗口：



附图 35: 选择目标系统

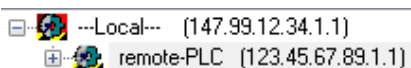
使用“Search (Ethernet)…”进入目标系统。弹出下一个对话框，在此可以选择：

- 在“Ether Host Name/IP: (输入主机名称/IP)”处输入已知的计算机名称（如红框所示）
- 执行“Broadcast Search (广播搜索)”（如果不知道确切的计算机名称）
- 输入已知的控制器 IP 或 AmsNetID



附图 36: 指定通过 TwinCAT System Manager 访问的 PLC: 选择目标系统



进入目标系统后，就可以选择以下方式（可能需要输入正确密码）：



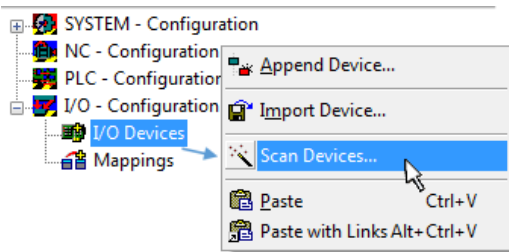
按“OK”确认，就可以通过 TwinCAT System Manager 访问目标系统。

## 添加设备

在 TwinCAT 2 System Manager 用户界面左侧的配置树中，选择“ I/O Devices”，然后右键单击，打开右键

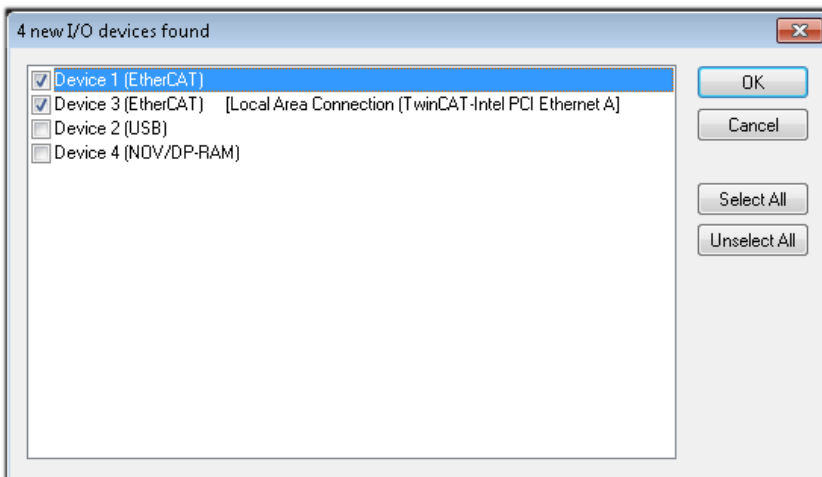
菜单，选择“Scan Devices...”，或者通过  在菜单栏中启动该操作。首先，可能需要点击  或通过菜单

“Actions” → “Set/Reset TwinCAT Config Mode...” TwinCAT System Manager (Shift + F4) 将 TwinCAT System Manager 设置为“Config Mode(配置模式)”。



附图 37: 选择“Scan Devices...”

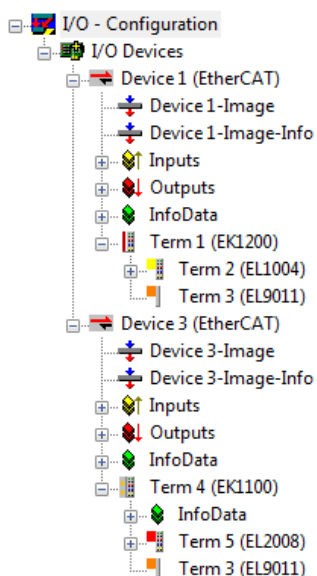
确认接下来的警告信息，并在对话框中选择“EtherCAT”设备：



附图 38: 自动检测 I/O 设备：选择要集成的设备

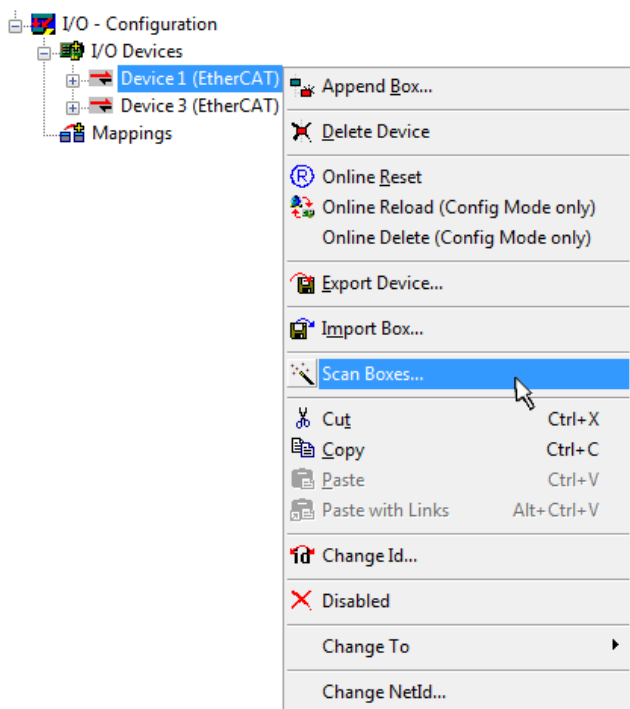
确认“Find new boxes”信息，以确定连接到设备的端子模块。“Free Run”自由运行功能允许在“Config Mode”配置模式下操作输入和输出值，这个功能也需要确认才能启用。

根据本节开头描述的示例配置 [▶ 67]，结果如下：



附图 39: TwinCAT 2 System Manager 中的配置映射

上述整个过程包括两个步骤，可以独立进行（首先确定设备，然后确定每个设备连接的元件，如端子盒、端子模块等）。此外，也可以从“Device...”的右键菜单中选择“Scan Box”（搜索功能），读取目标设备 Device 下面连接的元件（从站）：



附图 40: 读取连接到 Device 的各个端子模块

这可功能可以用于快速发现实际配置的变动。

## PLC 的编程和集成

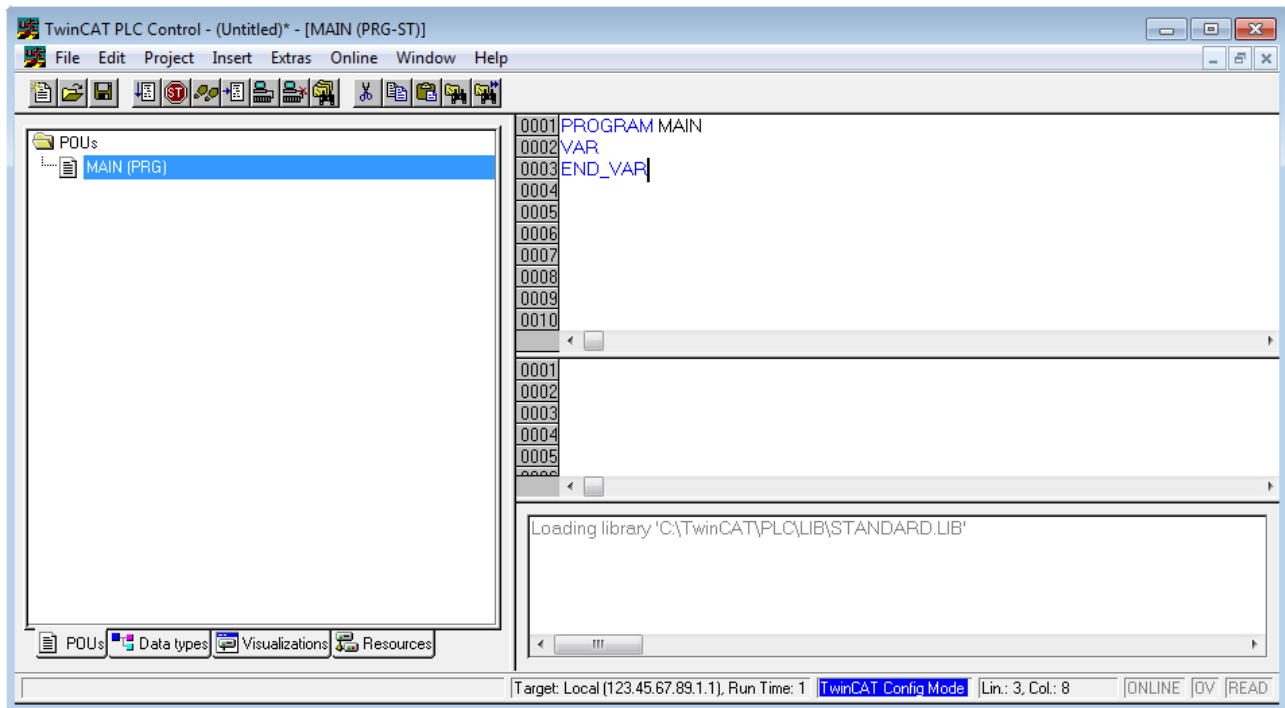
TwinCAT PLC Control 开发环境可以用不同的语言创建控制程序：TwinCAT PLC Control 支持 IEC 61131-3 中描述的所有 5 种语言：包括两种基于文本的语言和三种图形化的语言：

- 基于文本的语言
  - 指令表（IL）
  - 结构化文本（ST）
- 图形化语言

- 功能块图 (FBD)
- 梯形图 (LD)
- 连续功能块图 (CFC)
- 顺序流程功能图 (SFC)

下面的内容只用到结构化文本 (ST)。

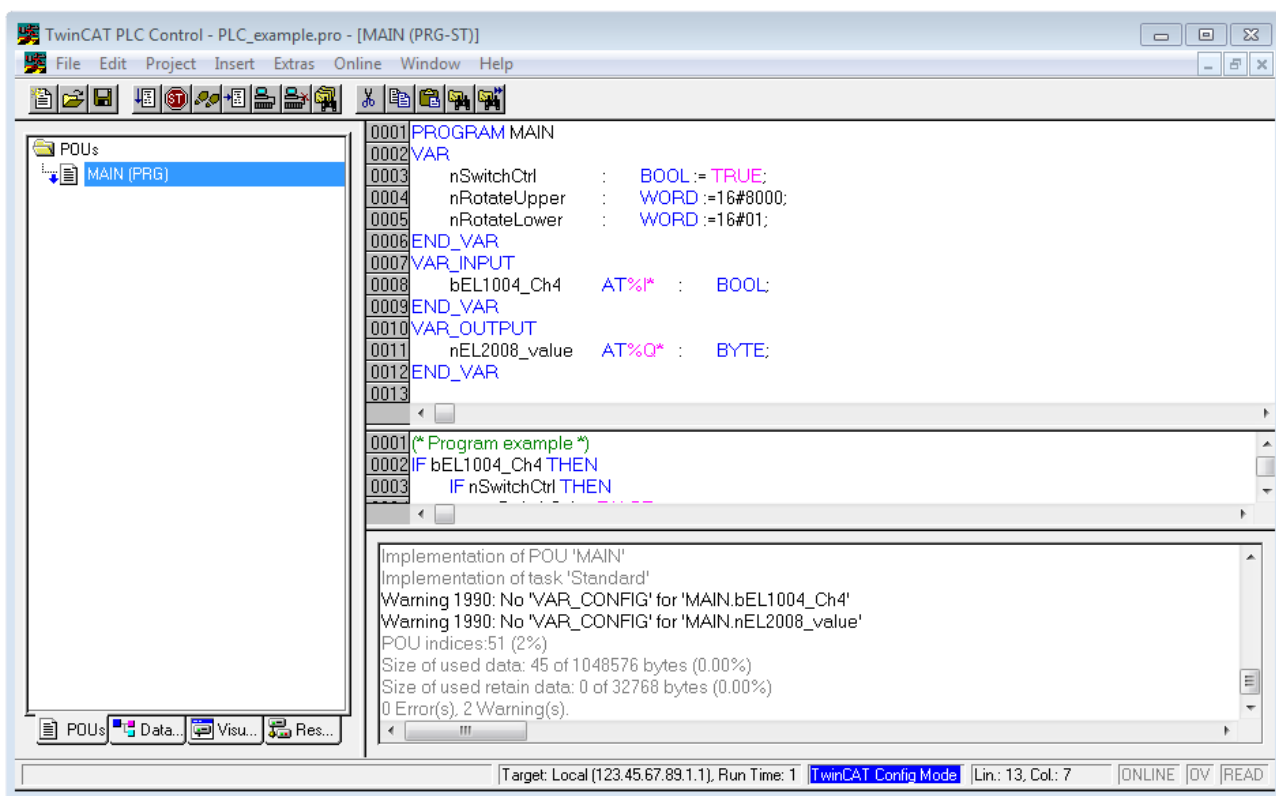
启动 TwinCAT PLC Control 后，新建一个项目，显示以下用户界面：



附图 41: 刚启动的 TwinCAT PLC Control

在示例项目中创建变量和程序，并保存为名称 “PLC\_example.pro”：

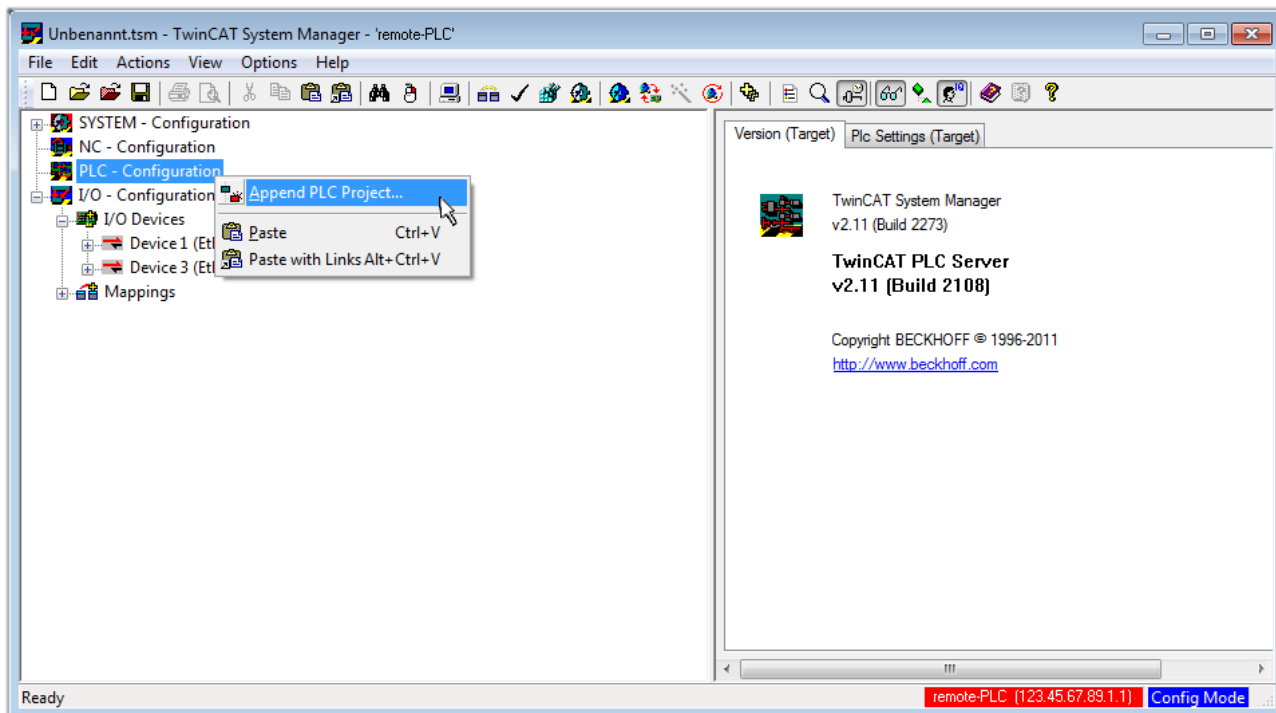




附图 42: 示例程序编译后，包含未分配地址的变量

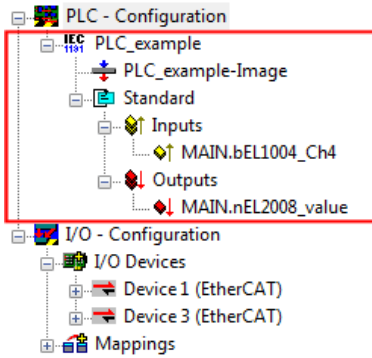
编译过程后的警告 Warning 1990 (missing “VAR\_CONFIG”)，表示定义的外部变量（含标志符 “AT%I\*” 或 “AT%Q\*”）还没有分配地址。编译成功后，TwinCAT PLC Control 会在项目路径下创建一个 “\*.tpy” 文件，该文件包含了指定的 IO 变量，但 System Manager 还没有为其分配地址，因此出现了警告。只要在 System Manager 中引入该 .tpy 文件并保存，再次编译时警告就不会再出现了。

首先，在 System Manager 中导入 TwinCAT PLC Control 项目。从 PLC configuration 的右键菜单（右击）选择 “Append PLC Project...”：



附图 43: 添加 TwinCAT PLC Control 项目

在弹出的浏览窗口中选择 PLC 配置文件 “PLC\_example.tpy”。这样 System Manager 的 System configuration 中就集成进了这个PLC项目，其中包含两个用“AT”标识的变量：

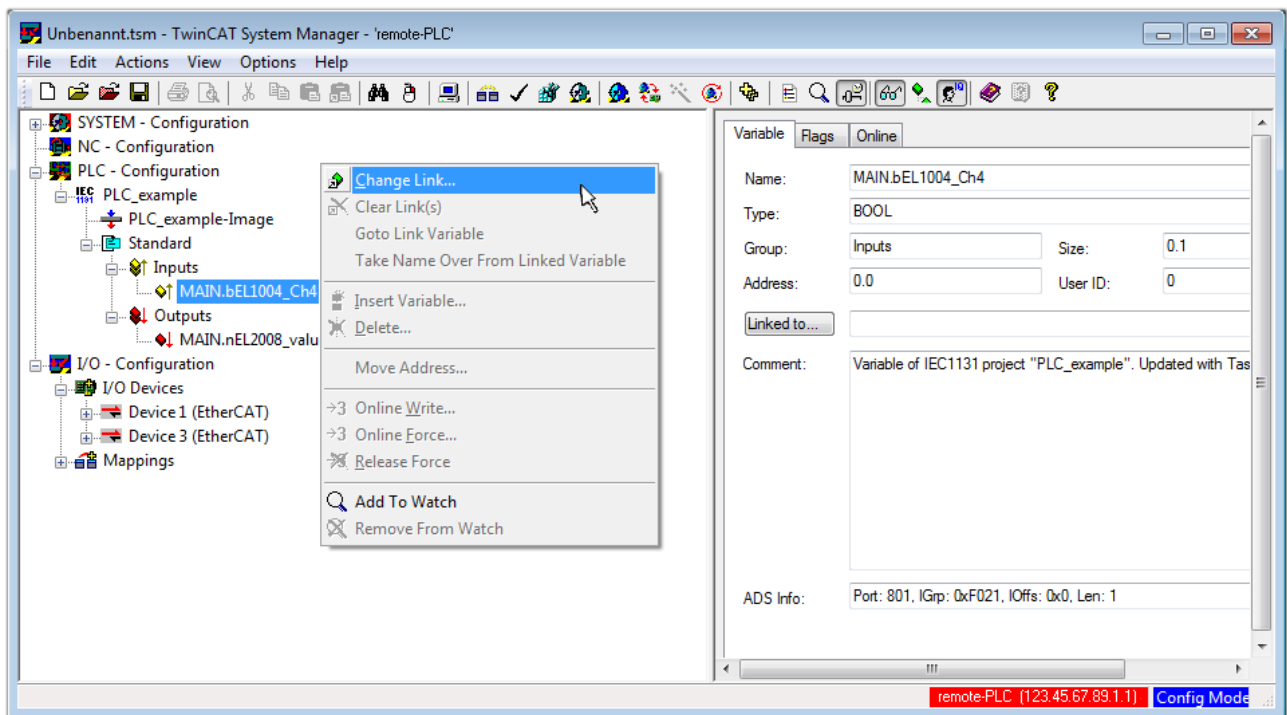


附图 44: 在System Manager 的 PLC Configuration 下导入 PLC 项目

现在可以将两个变量 “bEL1004\_Ch4” 和 “nEL2008\_value” 分配给 I/O configuration 下的某些过程对象了。

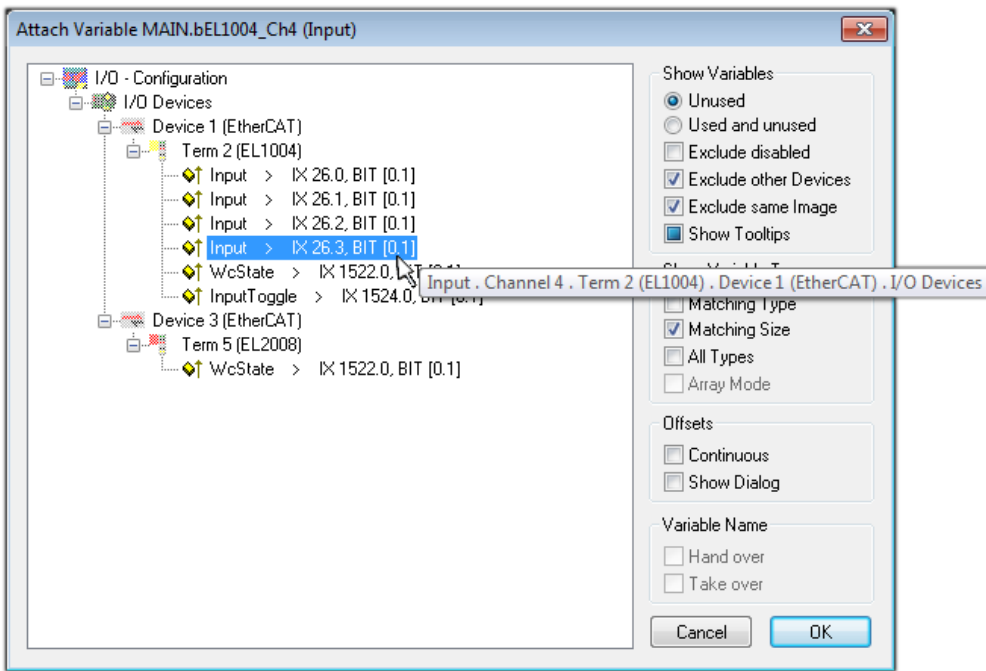
### 变量分配

通过集成的 “PLC\_example” 项目中某个变量的右键菜单 “Change Link...” 打开一个窗口，选择合适的过程对象（PDO） “Standard”：



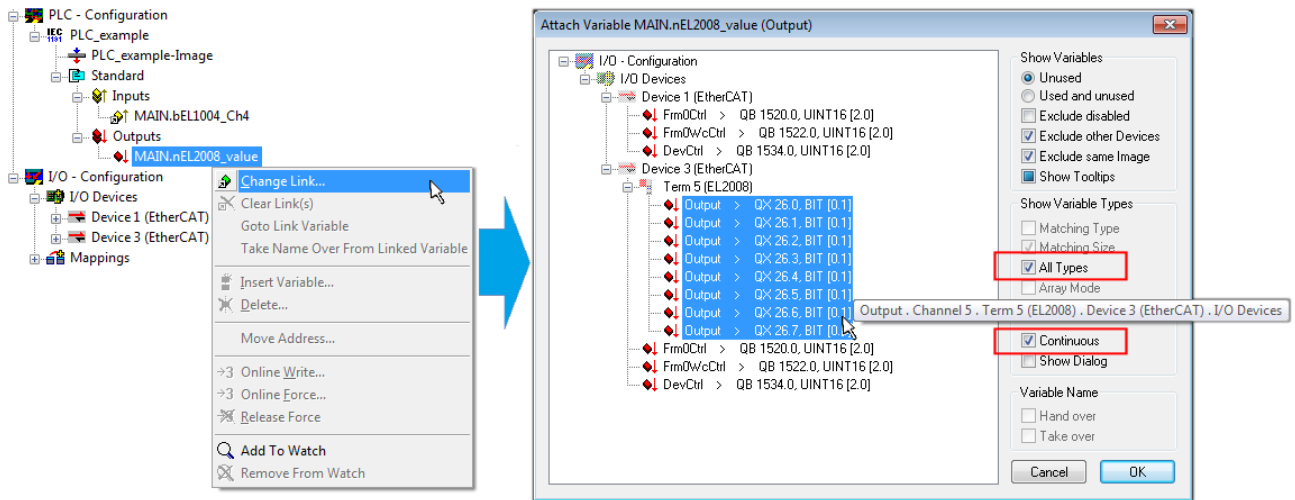
附图 45: 在 PLC 变量和过程对象之间建立链接

在弹出的窗口中，可以为 PLC configuration 中的 BOOL 类型变量 “bEL1004\_Ch4” 选择过程对象：




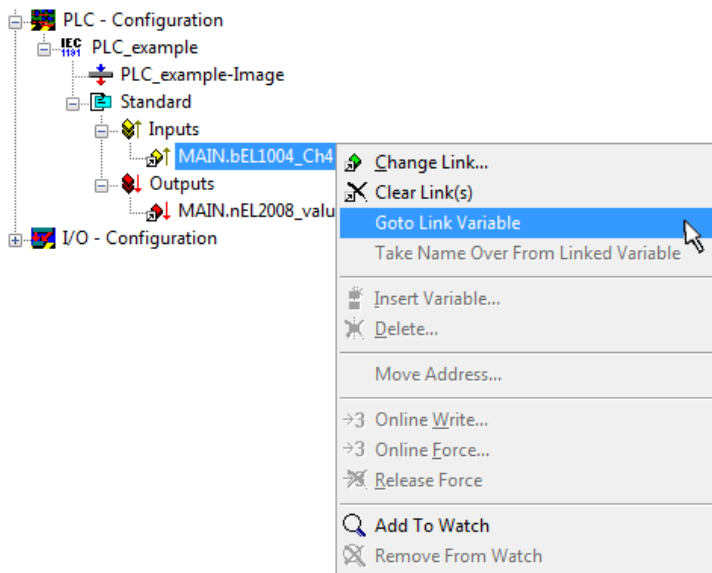
附图 46: 选择 BOOL 类型的 PDO

根据默认设置，只有部分PDO 对象可供选择。本例中，选择 EL1004 端子模块的通道 4 的 input 用于链接。否则，如果要为一个输出的字节变量分配一组八个独立的输出位，那么在创建链接时，必须勾选“**All types**”复选框。下图显示整个过程：



附图 47: 同时选择几个 PDO: 勾选“Continuous”和“All types”

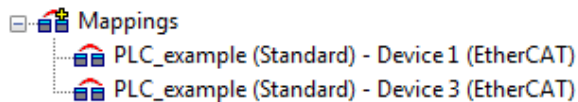
请注意，“Continuous”复选框也要勾选。这种设计旨在将变量“nEL2008\_value”的字节中包含的位按顺序分配给 EL2008 端子模块的所有八个选定的输出位。这样就可以在PLC程序中用一个字节对应端子模块的所有8个输出，字节的第0到7位分别对应模块的第1到8通道。在变量的黄色或红色对象处有一个特殊符号（），表示变量已链接。也可以通过从变量的右键菜单中选择“Goto Link Variable（转到链接变量）”来检查链接。此时，链接的对方（在这种情况下是 PDO）被自动选中：



附图 48: “Goto Link Variable” 的应用, 以 “MAIN.bEL1004\_Ch4” 为例

为 PDO 分配变量的过程通过菜单选项 “Actions” → “Create assignment” 或者通过  来完成。



在配置文件中可以直观地查看变量的分配结果:



建立链接的过程也可以反向进行, 即从 PDO 链接到变量。但在本例中, 不可能为 EL2008 选择所有输出位, 因为这个端子模块只提供单个数字量输出。如果一个端子模块有一个 byte, word, int 之类的 PDO, 就有可能为其分配一套标准位宽的变量。在这里, “Goto Link Variable” 也可以反向执行, 以选择相应的 PLC 实例。

### 激活配置

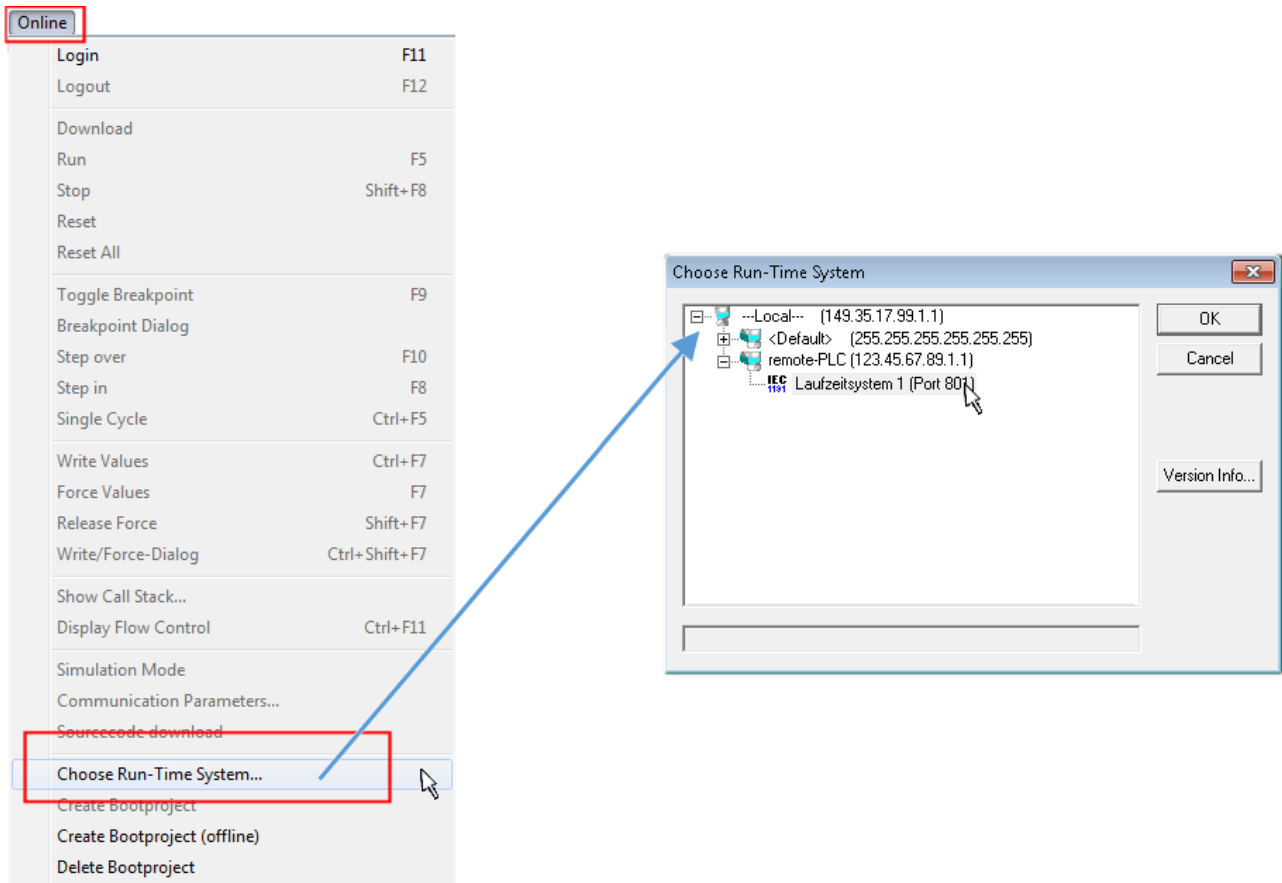
PDO 到 PLC 变量的分配过程建立了从控制器到端子模块的输入和输出的连接。下面激活该配置。首先, 可以

通过  (或通过 “Actions” → “Check Configuration”) 来检查配置。如果没有错误, 可以通过  (或通过 “Actions” → “Activate Configuration”) 激活配置, 将 System Manager 的设置传输至 TwinCAT runtime 系统。确认此时弹出的信息 “Old configurations will be overwritten! (以前的配置将被覆盖)” 点击 “OK” 按钮, 确认 “Restart TwinCAT system in Run mode (重启 TwinCAT 系统至运行模式)”。


几秒钟后, TwinCAT real-time (实时核) 的状态  显示在 TwinCAT System Manager 的右下方。这样就可以按以下方法启动 PLC 系统了。

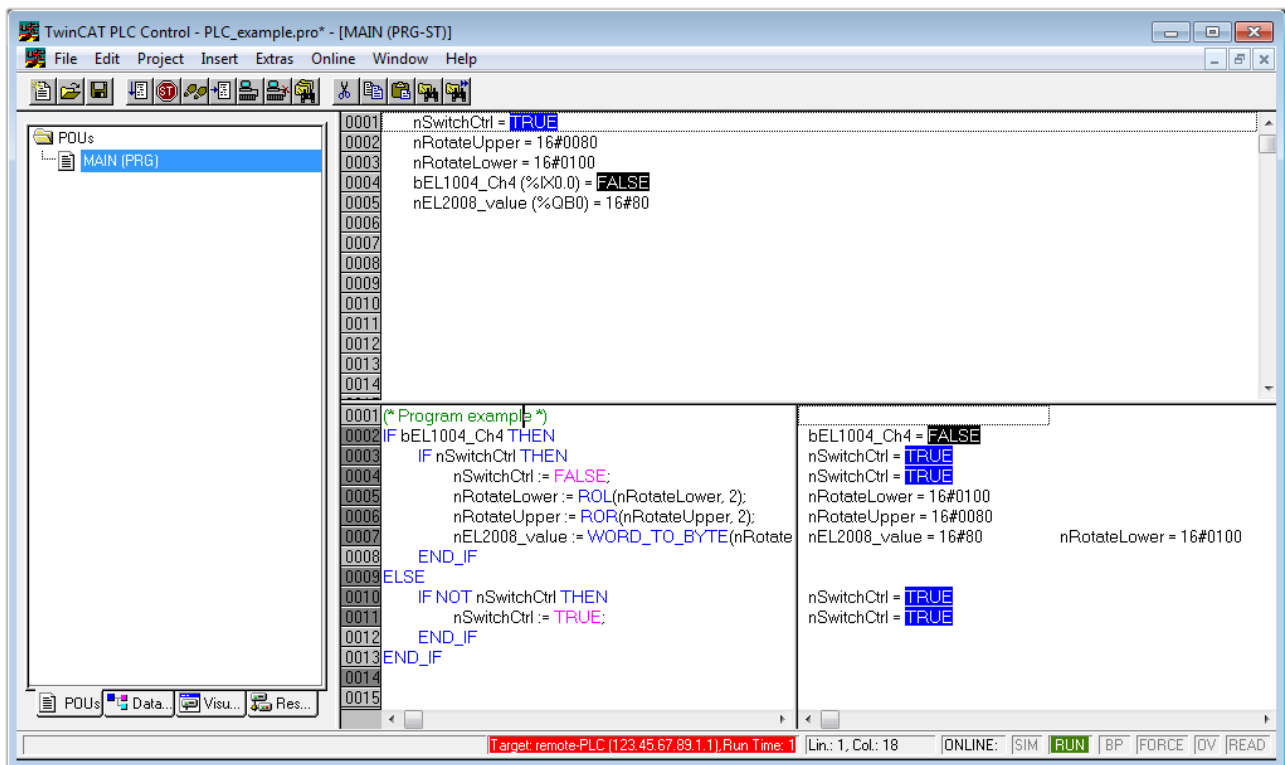
### 启动控制器

从远程系统操作控制器, 必须先在 PLC Control 中通过菜单 “Online” → “Choose Runtime System...”, 以便连接到 IPC/EPC:



附图 49: 选择目标系统（远程）

在此例中，选择了“Runtime system 1 (port 801)”并确认。通过菜单选项“Online” → “Login”，F11 键或通过点击符号 ，将 PLC 与 TwinCAT real-time 实时系统链接起来。然后就可以加载控制程序并运行。系统弹出信息“*No program on the controller! Should the new program be loaded?*”，应点击“*Yes*”确认。TwinCAT runtime 运行环境已经做好程序启动的准备：



附图 50: PLC Control 登录，做好程序启动准备

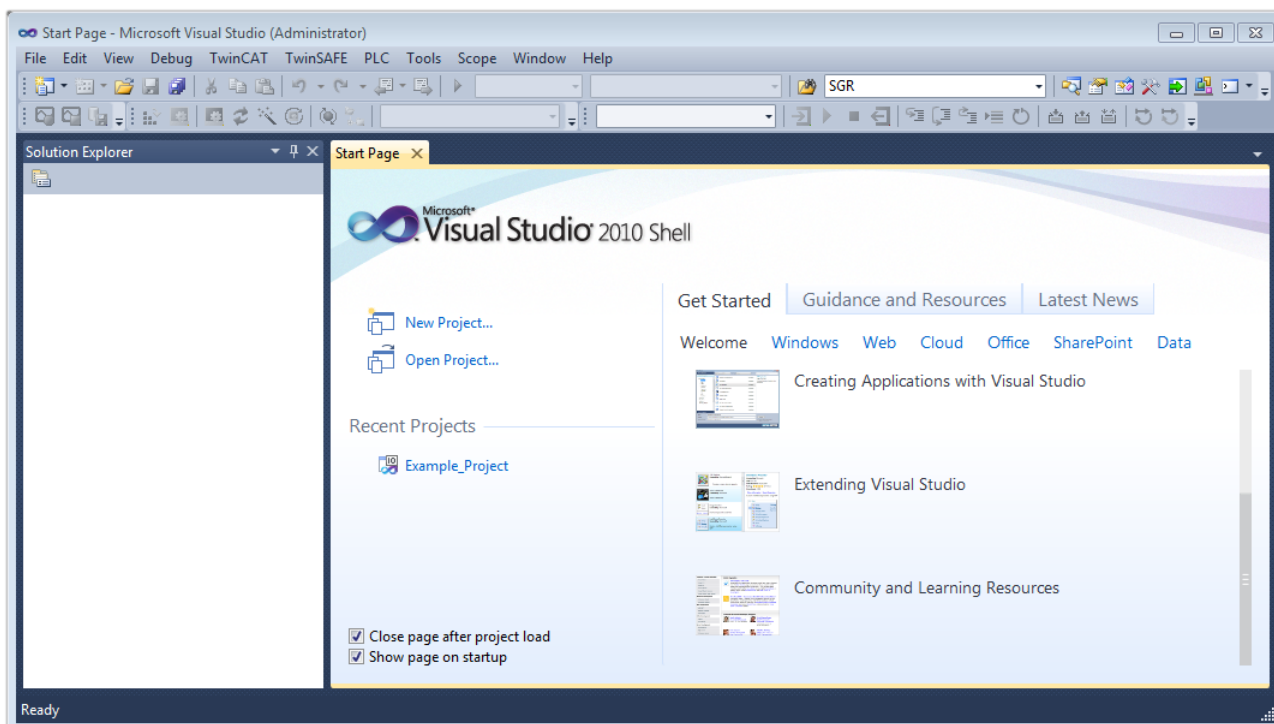
然后就可以通过“Online” → “Run”，F5 键或  启动 PLC。

## 6.1.2 TwinCAT 3


### 启动

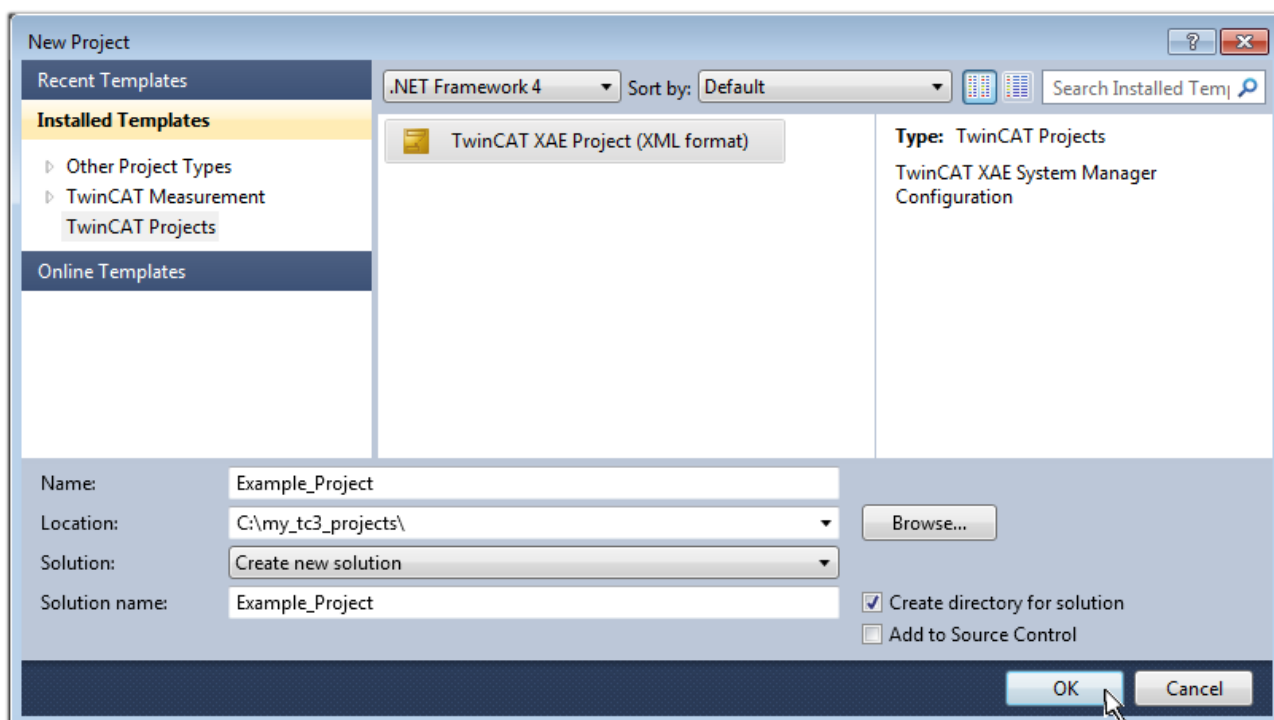
TwinCAT 3 集成于 Microsoft Visual Studio，所有功能都包含在一个开发环境中：启动后，项目文件浏览器显示在通用窗口区域的左侧（参见 TwinCAT 2 的“TwinCAT System Manager”），用于与电气组件进行通信。

用于开发的 PC 上成功安装 TwinCAT 系统后，TwinCAT 3 (shell) 在启动后显示以下用户界面：



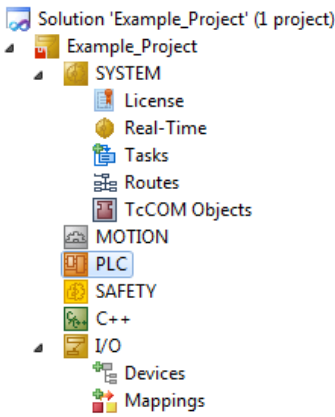
附图 51: TwinCAT 3 的初始用户界面

首先通过  **New TwinCAT Project...** (或“File” → “New” → “Project...”) 创建一个新项目。在下面的对话框中，输入需要的信息 (如图所示)：



附图 52: 新建 TwinCAT 3 项目

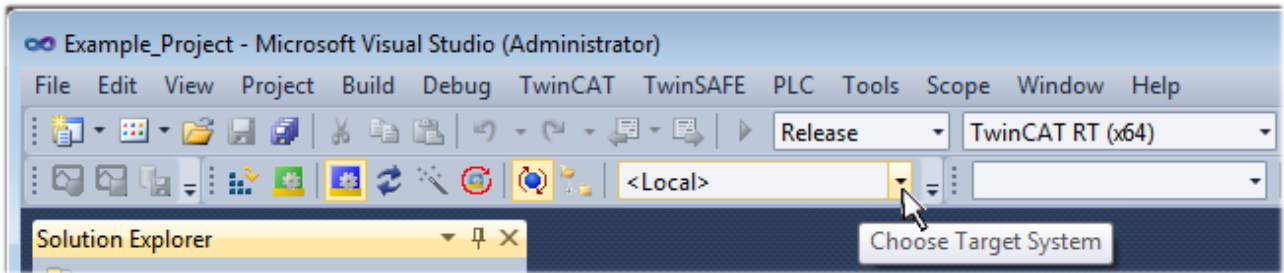
然后，在项目文件浏览器中就会显示新建的项目：



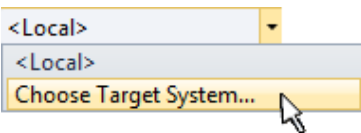
附图 53: 项目文件浏览器中的 TwinCAT 3 新建项目

一般来说, TwinCAT 可以工作在本地或远程模式。如果 TwinCAT 系统 (包括标准的用户开发界面) 安装在相应的 PLC 上 (本地), TwinCAT 就可以在本地模式下使用, 这种情况下, 可以执行下一个步骤 “[Insert Device \[ 81 \]](#)”。

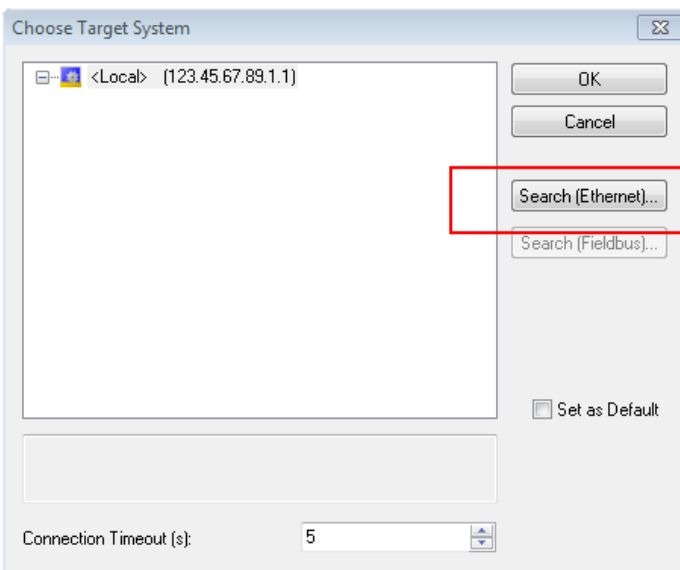
如果要从 TwinCAT 开发环境连接到另一个安装在远程 PLC 上的 TwinCAT Runtime 运行环境, 就得先识别到 Target System (目标系统)。通过菜单栏中的符号:



展开下拉菜单:



并打开以下窗口:

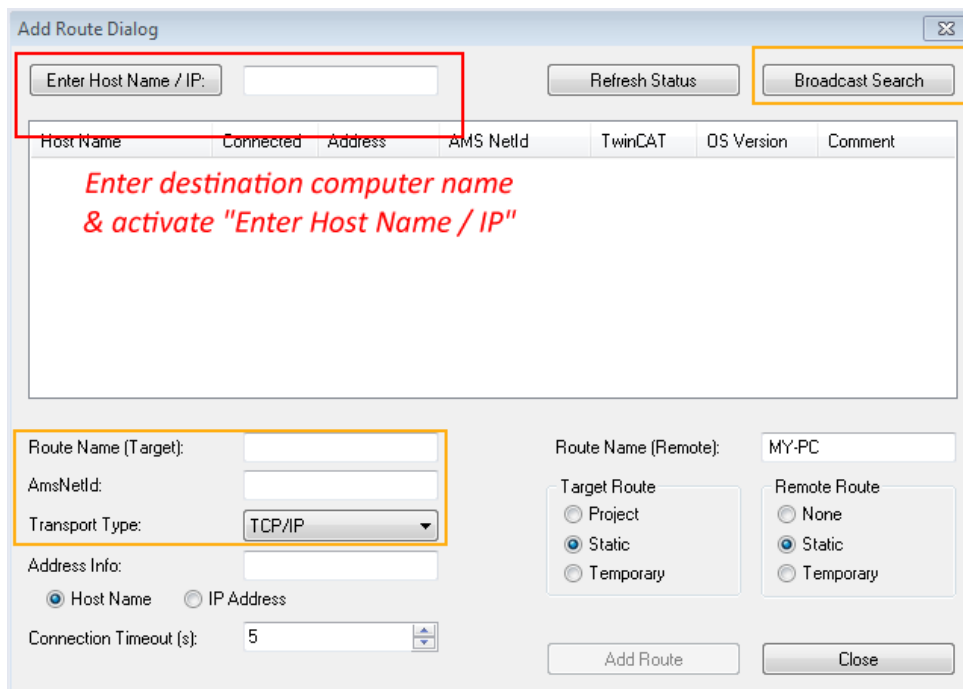


附图 54: 选择对话框: Choose the target system

使用 “Search (Ethernet)…” 进入目标系统。弹出下一个对话框, 在此可以选择:

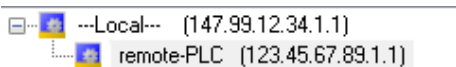


- 在“Ether Host Name/IP: (输入主机名称/IP)”处输入已知的计算机名称（如红框所示）
- 执行“Broadcast Search (广播搜索)”（如果不知道确切的计算机名称）
- 输入已知的控制器 IP 或 AmsNetID



附图 55: 指定通过 TwinCAT System Manager 访问的 PLC: 选择目标系统

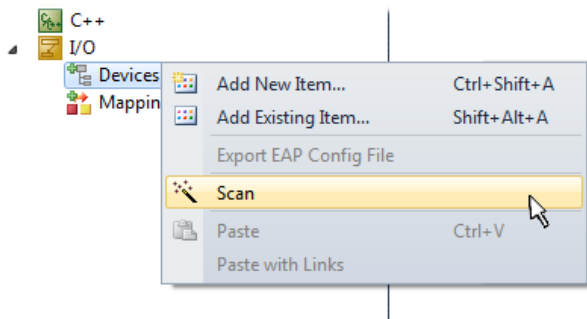
进入目标系统后, 按以下方式进行选择 (可能需要输入正确密码):



点击“OK”按钮确认, 就可以通过 Visual Studio shell 访问目标系统。

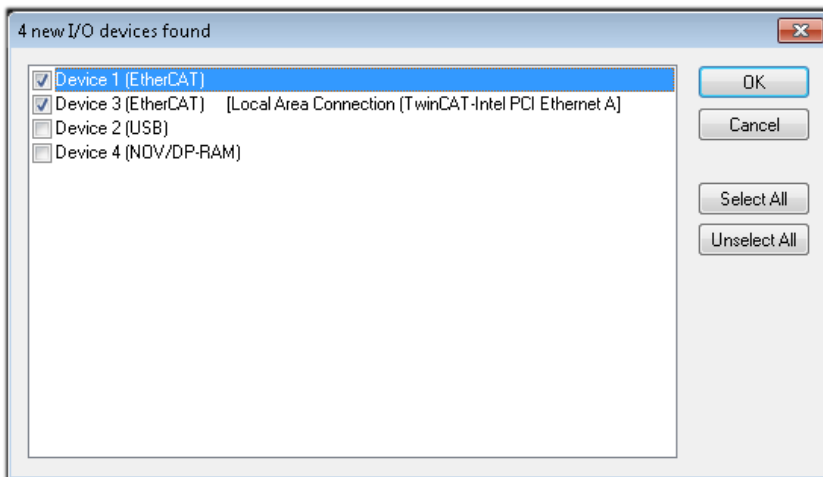
### 添加设备

在 Visual Studio shell 用户界面左侧的项目文件浏览器中, 选择“I/O”节点下的“Device”, 然后右键单击打开右键菜单, 选择“Scan”或通过菜单栏中的  开始操作。首先, TwinCAT System Manager 可能需要通过  或通过菜单“TwinCAT”→“Restart TwinCAT (Config Mode)”设置成“Config Mode”。



附图 56: 选择“Scan(扫描)”

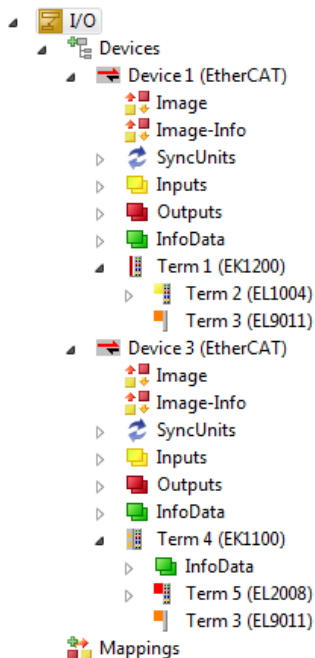
确认接下来的警告信息, 并在对话框中选择“EtherCAT”设备:



附图 57: 自动检测 I/O 设备: 选择要集成的设备

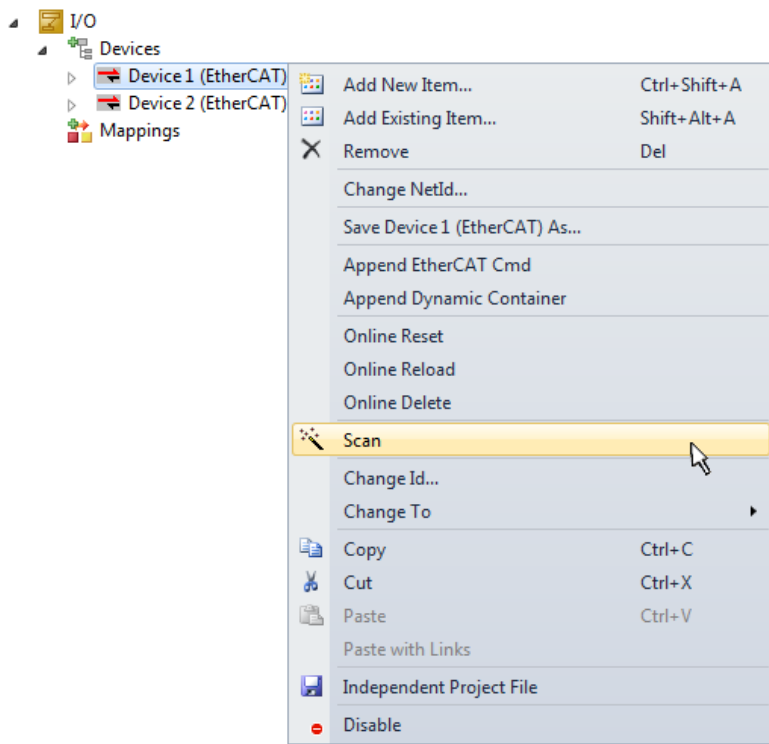
确认“Find new boxes”信息，以确定连接到设备的端子模块。“Free Run”自由运行功能允许在“Config Mode”配置模式下操作输入和输出值，这个功能也需要确认才能启用。

根据本节开头描述的示例配置 [▶ 67]，结果如下：



附图 58: 在集成于 VS shell 的TwinCAT 3 环境中配置映射

上述整个过程包括两个步骤，可以独立进行（首先确定设备，然后确定每个设备连接的元件，如端子盒、端子模块等）。此外，也可以从“Device...”的右键菜单中选择“Scan Box”（搜索功能），读取目标设备 Device 下面连接的元件（从站）：



附图 59: 读取连接到 Device 的各个端子模块

这功能可以用于快速发现实际配置的变动。

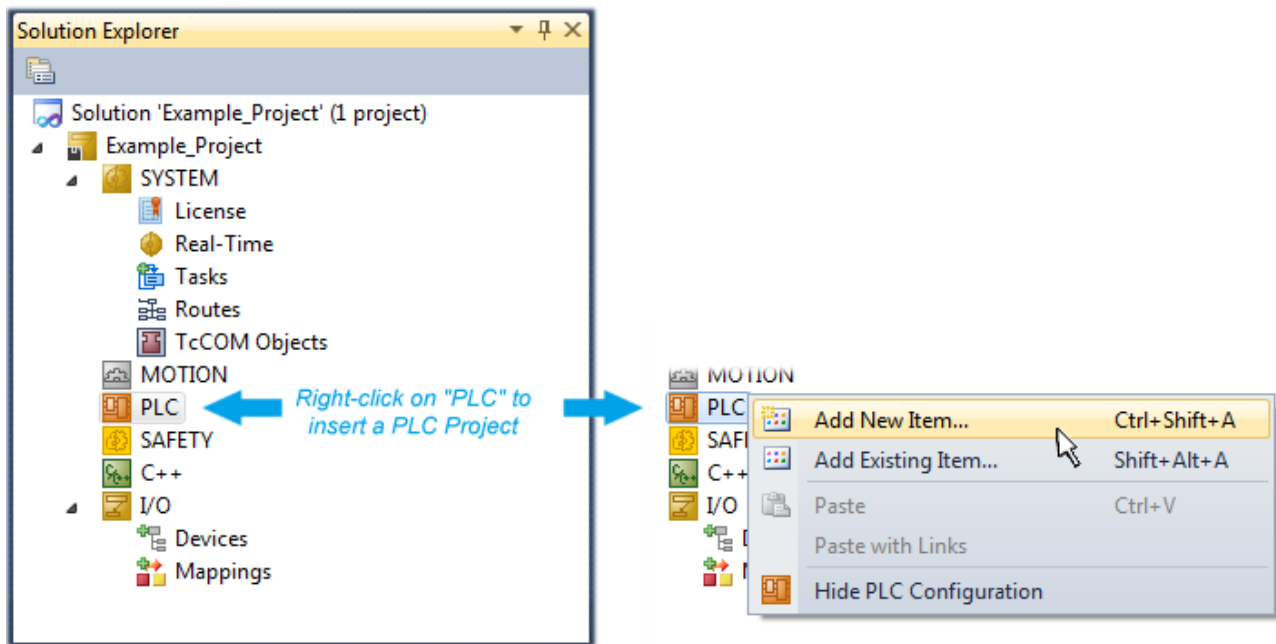
### 对 PLC 进行编程

TwinCAT PLC Control 开发环境可以用不同的语言创建控制程序：TwinCAT PLC Control 支持 IEC 61131-3 中描述的所有 5 种语言：包括两种基于文本的语言和三种图形化的语言：

- **基于文本的语言**
  - 指令表 (IL)
  - 结构化文本 (ST)
- **图形化语言**
  - 功能块图 (FBD)
  - 梯形图 (LD)
  - 连续功能块图 (CFC)
  - 顺序流程功能图 (SFC)

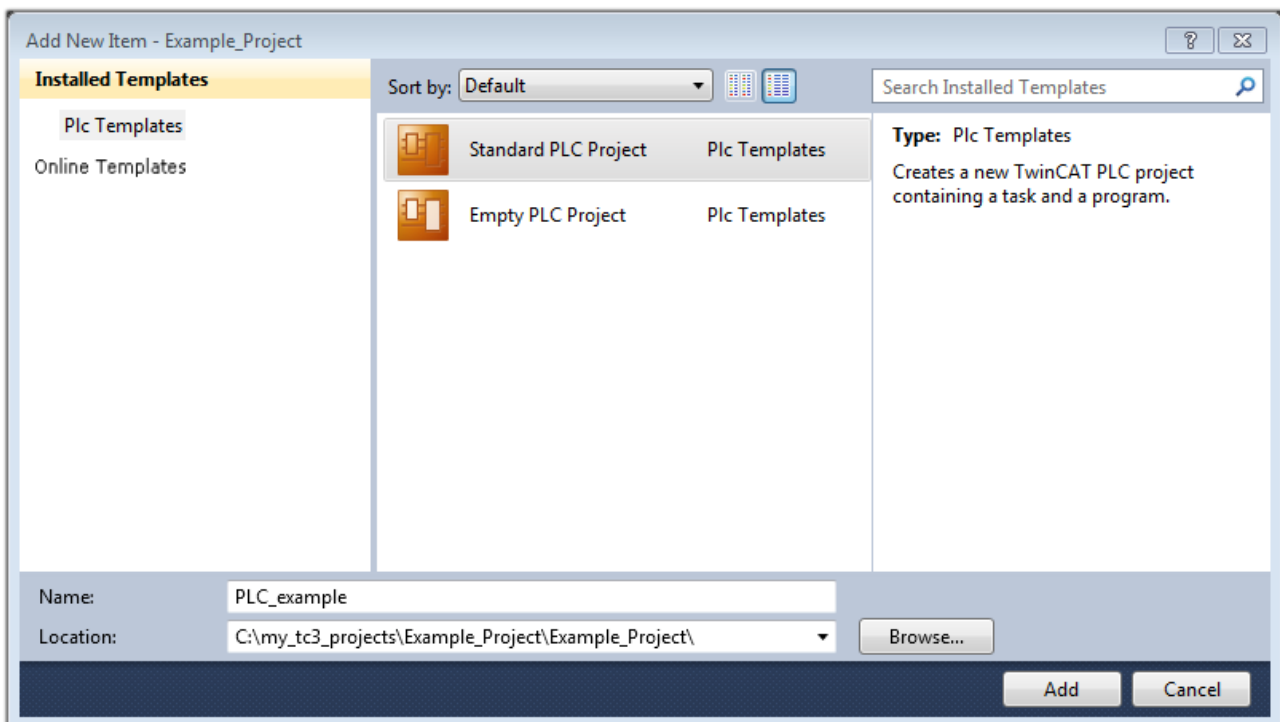
下面的内容只用到结构化文本 (ST)。

为了创建一个编程环境，通过项目文件浏览器中“PLC”的右键菜单，选择“Add New Item...”，将一个 PLC 子项目添加到示例项目中：



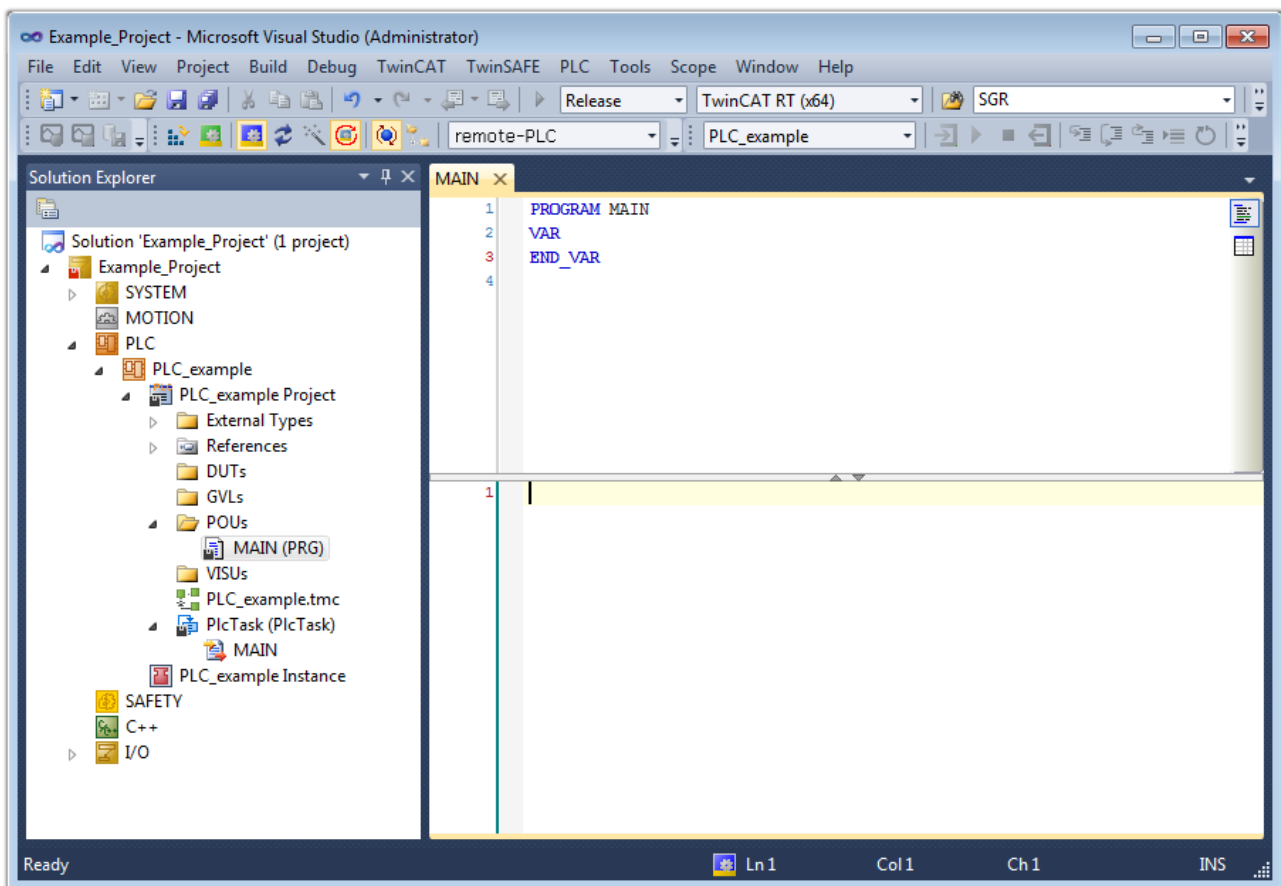
附图 60: 在节点“PLC”下添加编程环境

在弹出的对话框中选择“Standard PLC project”，并输入“PLC\_example”作为项目名称（示例），然后选择一个相应的目录：



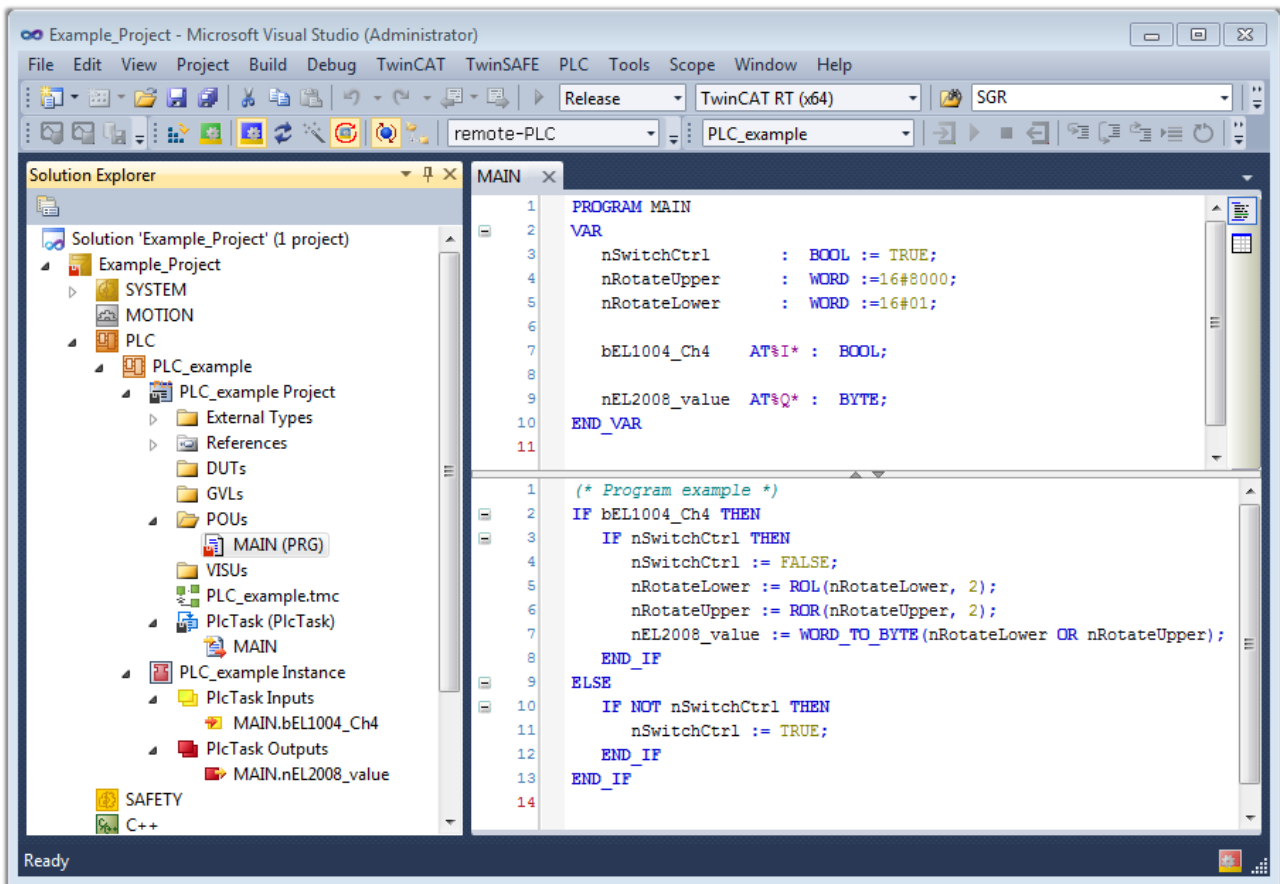
附图 61: 为 PLC 编程环境指定名称和目录

选择“Standard PLC project”已经存在的“Main”程序，可以通过双击“POUs”中的“PLC\_example\_project”打开。以下是一个初始项目的用户界面：



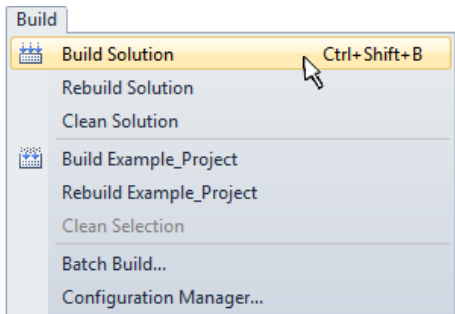
附图 62: 标准 PLC 项目的初始程序“Main”

现在，已经为下一阶段的工作创建了示例变量和示例程序。



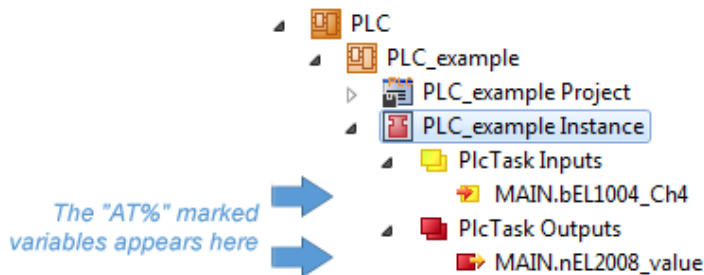
附图 63: 示例程序编译后, 包含未分配地址的变量

现在, 控制程序被创建一个项目文件夹, 接下来是编译过程:



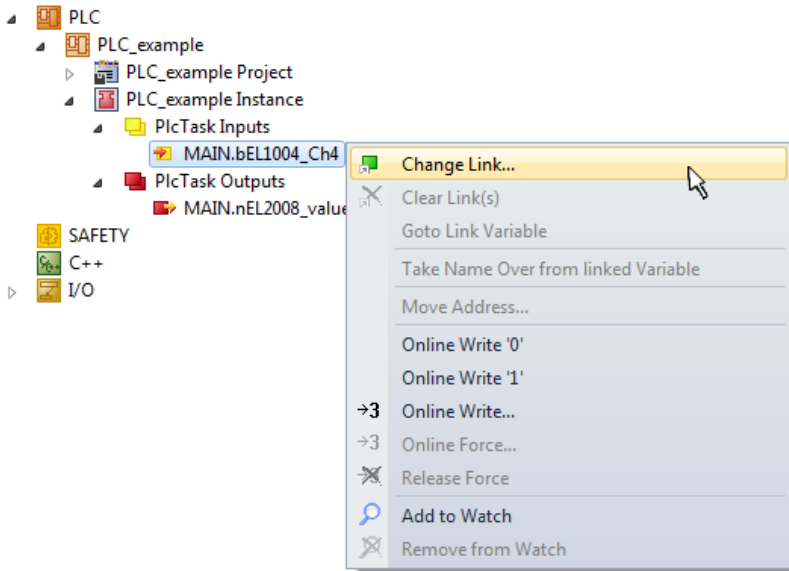
附图 64: 开始编译程序

下列变量在 ST/PLC 程序中以“AT%”标识, 所以在项目文件浏览器的“Instance”中可为其分配硬件:



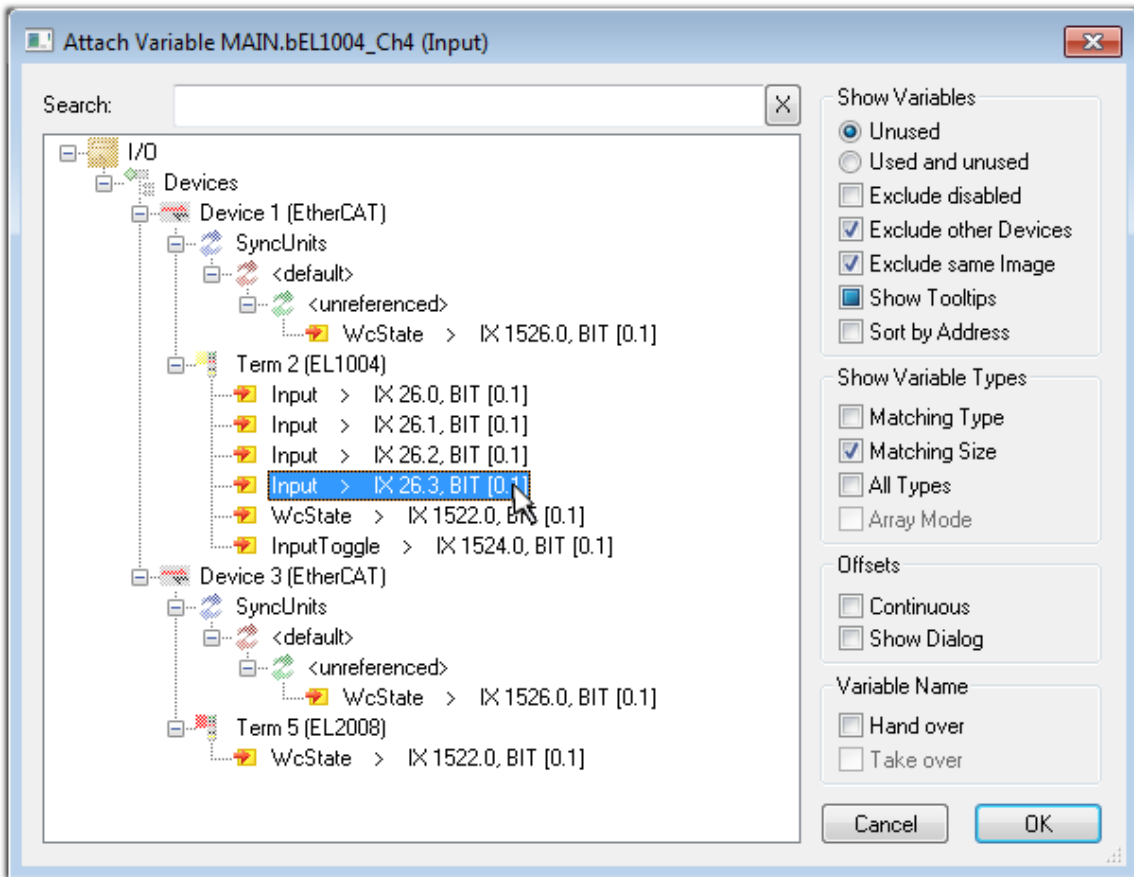
### 变量分配

通过 PLC instance 中一个变量的右键菜单, 使用“Change Link...”打开一个窗口, 选择合适的过程对象 (PDO) 进行链接:



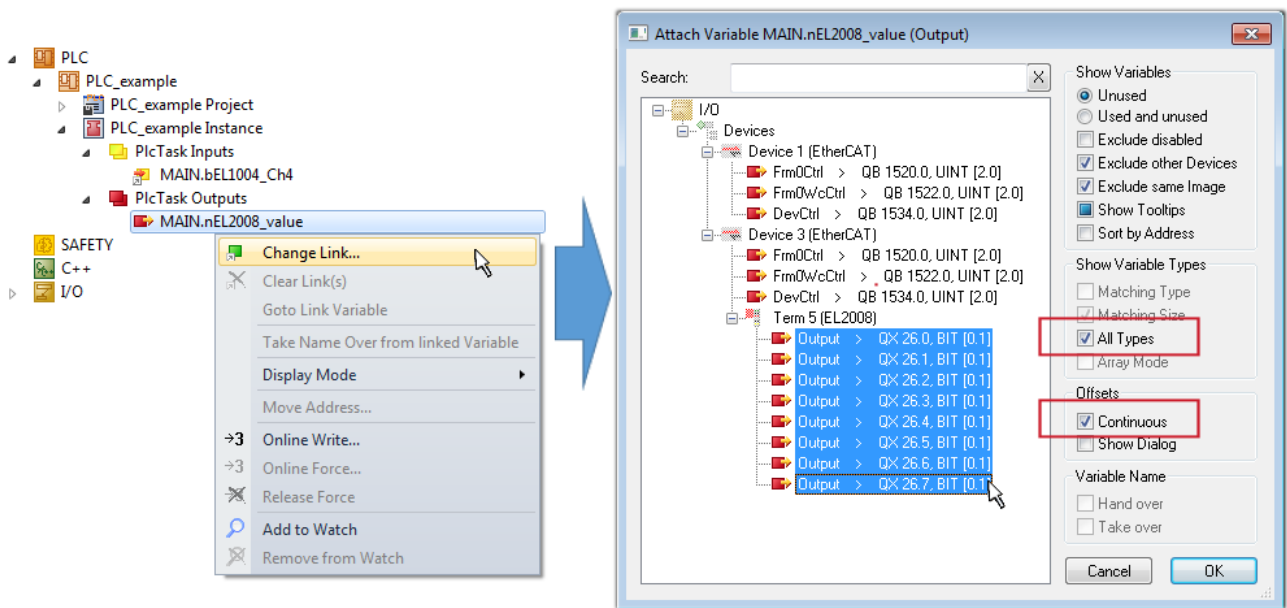
附图 65: 在 PLC 变量和过程对象之间建立链接

在弹出的窗口中，可以为 PLC configuration 中的 BOOL 类型变量“bEL1004\_Ch4” 选择过程对象：




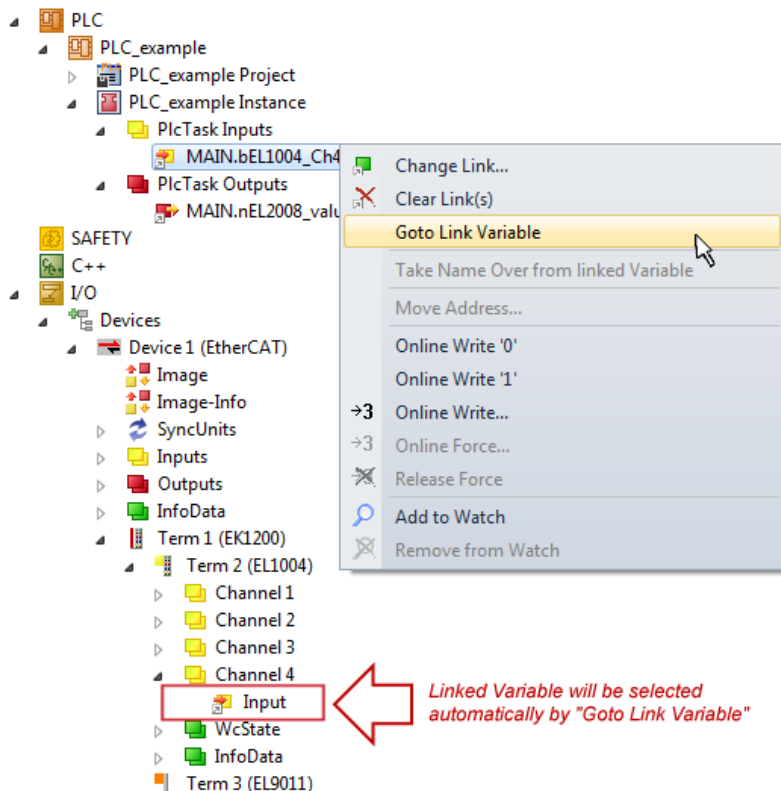
附图 66: 选择 BOOL 类型的 PDO

根据默认设置，只有部分PDO 对象可供选择。本例中，选择 EL1004 端子模块的通道 4 的 input 用于链接。否则，如果要为一个输出的字节变量分配一组八个独立的输出位，那么在创建链接时，必须勾选“*All types*”复选框。下图显示整个过程：



附图 67: 同时选择几个 PDO: 勾选“Continuous”和“All types”

请注意，“Continuous”复选框也要勾选。这种设计旨在将变量“nEL2008\_value”的字节中包含的位按顺序分配给 EL2008 端子模块的所有八个选定的输出位。这样就可以在 PLC 程序中用一个字节对应端子模块的所有 8 个输出，字节的第 0 到 7 位分别对应模块的第 1 到 8 通道。在变量的黄色或红色对象处有一个特殊符号 (  )，表示变量已链接。也可以通过从变量的右键菜单中选择“Goto Link Variable (转到链接变量)”来检查链接。此时，链接的对方 (在这种情况下是 PDO) 被自动选中:



附图 68: “Goto Link Variable”的应用，以“MAIN.bEL1004\_Ch4”为例

建立链接的过程也可以反向进行，即从 PDO 链接到变量。但在本例中，不可能为 EL2008 选择所有输出位，因为这个端子模块只提供单个数字量输出。如果一个端子模块有一个 byte, word, int 之类的 PDO，就有可能为其分配一套标准位宽的变量。在这里，“Goto Link Variable”也可以反向执行，以选择相应的 PLC 实例。



## ● 关于变量分配类型的说明

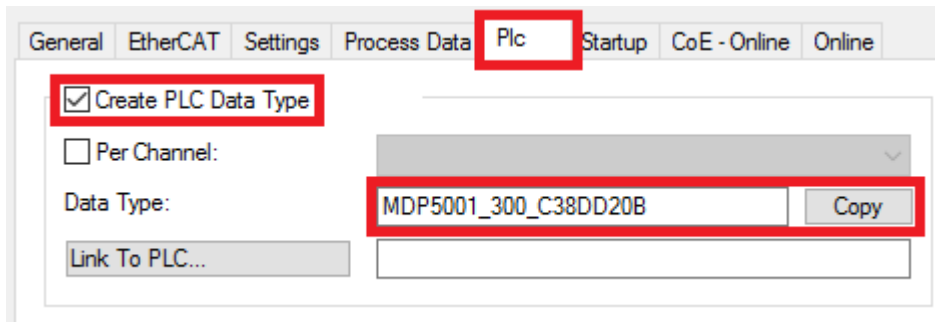


以下类型的变量分配方式只适用于 TwinCAT V3.1.4024.4 及以上版本，并且只能用于带微处理器的端子模块。

在 TwinCAT 中，可以根据一个端子模块的过程数据来创建一个结构体。然后在 PLC 中创建该结构体的一个实例，从而直接从 PLC 中访问过程数据，而无需自行声明变量。

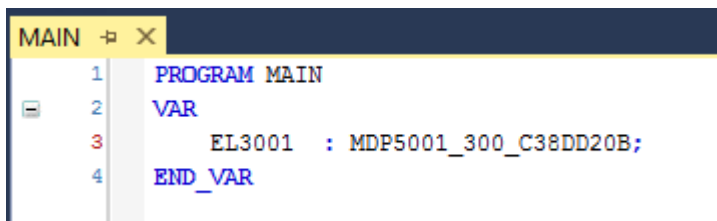
下面以 EL3001 单通道模拟量输入端子模块  $-10\dots+10\text{ V}$  为例，说明其操作步骤：

1. 首先，必须在 TwinCAT 的“Process data”选项卡中选择所需的过程数据。
2. 之后，必须通过复选框在“PLC”选项卡中生成 PLC 数据类型。
3. 然后可以用“Copy”按钮来复制“Data Type”文本框中的数据类型。



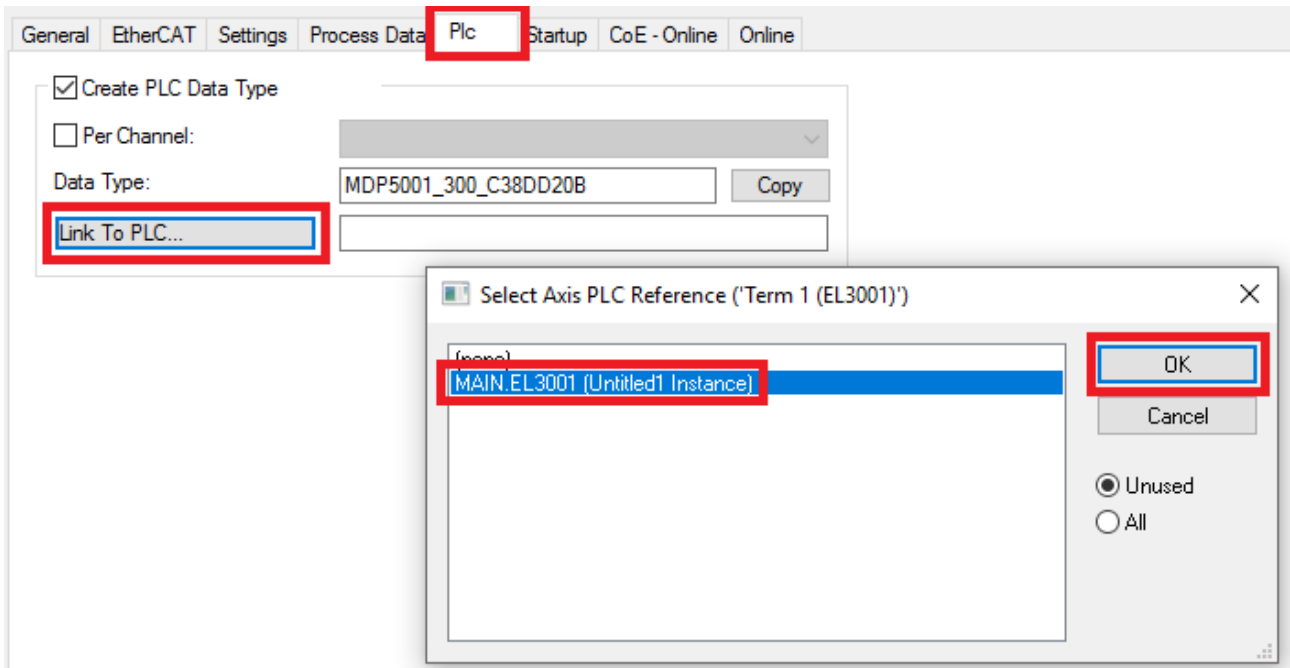
附图 69：创建一个 PLC 数据类型

4. 接下来，必须在 PLC 中创建一个数据结构的实例，类型为上一步复制的数据类型。



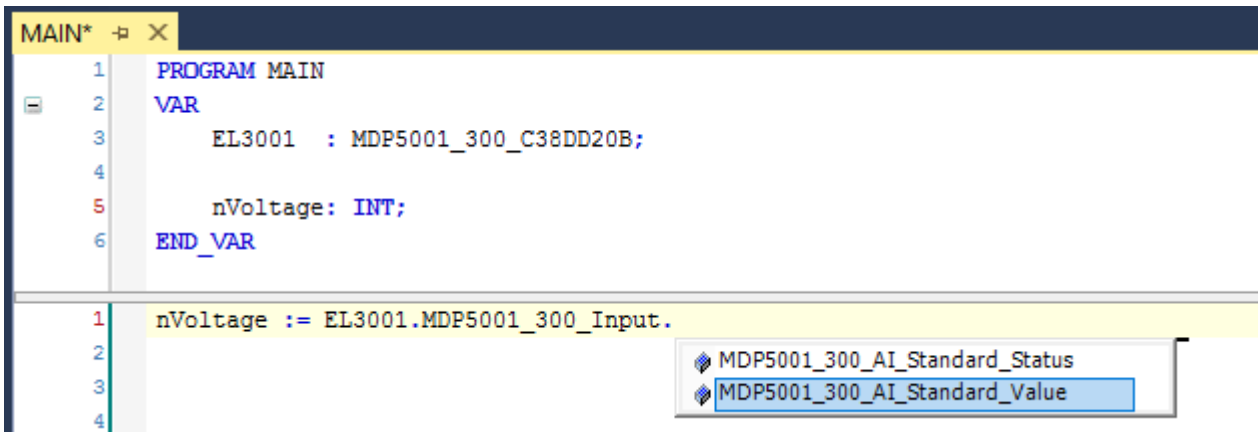
附图 70：结构体的实例

5. 然后，还必须创建项目文件夹。这可以通过组合键“CTRL + Shift + B”或通过 TwinCAT 的“Build”菜单来完成。
6. 然后，必须将端子模块的“PLC”选项卡中的结构体链接到刚才创建的 PLC 结构体实例。




附图 71: 结构体的链接

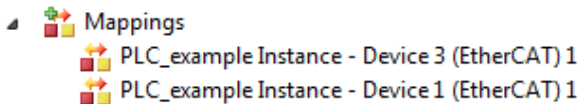
7. 这样就可以在 PLC 的程序代码中通过结构体读取或写入过程数据。



附图 72: 从过程数据的结构体中读取一个变量


**激活配置**


PDO 到 PLC 变量的分配过程建立了从控制器到端子模块的输入和输出的连接。现在可以用  或通过菜单“TwinCAT”下的选项激活配置，以便将开发环境中的配置传送到 TwinCAT runtime 运行系统中。确认此时弹出的信息“Old configurations will be overwritten! (以前的配置将被覆盖)” 点击“OK”按钮，确认“Restart TwinCAT system in Run mode (重启 TwinCAT 系统至运行模式)”。在项目文件浏览器中可以看到对应的变量分配结果：

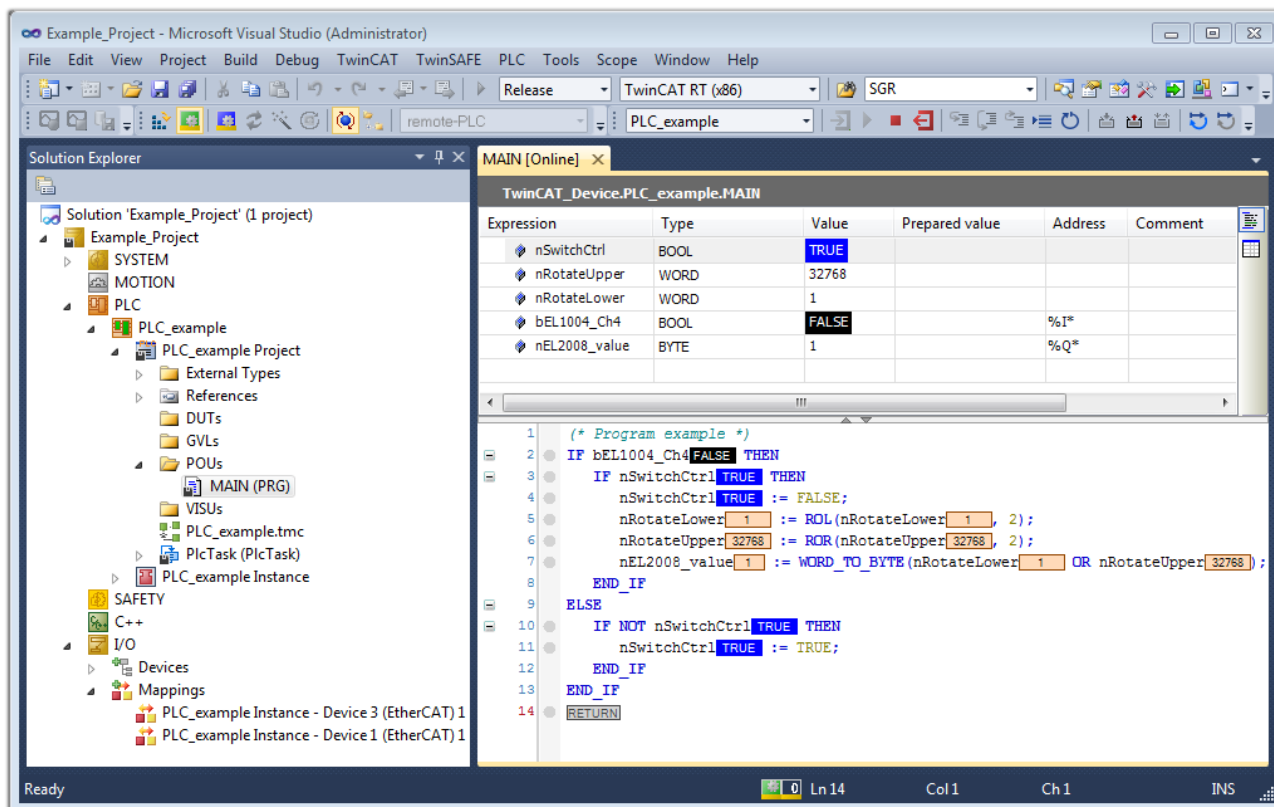


几秒钟后，运行模式的相应状态会以不停旋转的符号  显示在 VS shell 开发环境的右下方。这样就可以按以下方法启动 PLC 系统了。



## 启动控制器

选择菜单选项“PLC”→“Login”或点击，将 PLC 与 TwinCAT real-time 实时系统连接起来，并加载控制程序，准备运行。系统弹出信息“*No program on the controller! Should the new program be loaded?*”，应点击“*Yes*”确认。TwinCAT runtime 运行环境已经准备好启动 PLC 程序，通过点击符号

、“F5”键或通过菜单“PLC”中的“Start”即可启动。程序启动后，编程环境会在线显示 PLC runtime 中各个变量的实时值：



附图 73: TwinCAT 3 开发环境 (VS shell) : Login 后，程序已运行

用于停止和退出 (Logout)的两个操控按键会产生对应的动作 (“Shift + F5”也可用于停止，或者可以通过菜单“PLC”选择这两个动作)。

## 6.2 TwinCAT 开发环境

自动化软件 TwinCAT (The Windows Control and Automation Technology) 分为两种：

- TwinCAT 2: System Manager (用于配置) 和 PLC Control (用于编程)
- TwinCAT 3: TwinCAT 2 的增强版 (在同一个开发环境进行编程和配置)

### 详细信息:

- **TwinCAT 2:**
  - 以面向变量的方式将 I/O 设备与任务连接起来
  - 以面向变量的方式将任务与任务连接起来
  - 支持 Bit 级别的数据单位
  - 支持同步或异步映射关系
  - 支持连贯的数据区和过程映像交互
  - Datalink on NT - 程序符合开放式微软标准 (OLE、OCX、ActiveX、DCOM+ 等)

- 在 Windows NT/2000/XP/Vista、Windows 7、NT/XP Embedded、CE 中集成 IEC 61131-3 软 PLC、软 NC 和软 CNC。
- 可连接所有常见的现场总线
- [更多...](#)

#### 其他特点:

- **TwinCAT 3 (eXtended Automation) :**
  - 集成 Visual Studio®
  - 可以选择多种编程语言
  - 支持 IEC 61131-3 的面向对象扩展功能
  - 支持使用 C/C++ 语言编写实时应用程序
  - 可以连接 MATLAB®/Simulink®
  - 使用开放式接口，具有良好的扩展性
  - 灵活的 run-time (运行时) 环境
  - 支持多核 CPU 和 64 位操作系统
  - 提供 TwinCAT Automation Interface (自动化编程接口)，可以自动生成代码和创建项目
  - [更多...](#)

在下面的章节中，将介绍在 PC 系统上通过 TwinCAT 开发环境进行控制系统的调试，以及特定控制组件的基本功能。

关于 TwinCAT 2 和 TwinCAT 3 的更多信息，请参见<http://infosys.beckhoff.com>。

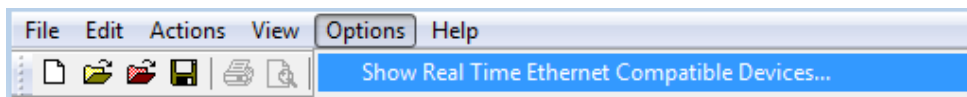
### 6.2.1 TwinCAT real-time 实时驱动程序的安装

为了使 IPC 控制器的标准以太网端口具备实时功能，必须在 Windows 下为该端口安装倍福 real-time 实时驱动程序。

可以通过几种方式进行：

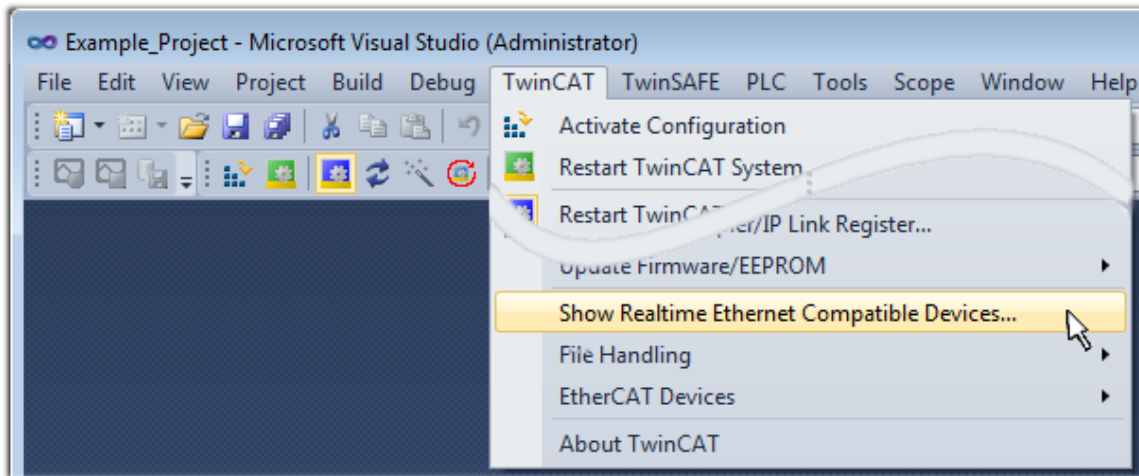
#### A: 通过 TwinCAT 适配器对话框

在 System Manager 中，通过“Options → Show Real Time Ethernet Compatible Devices”，调出本地以太网接口的 TwinCAT 概览。



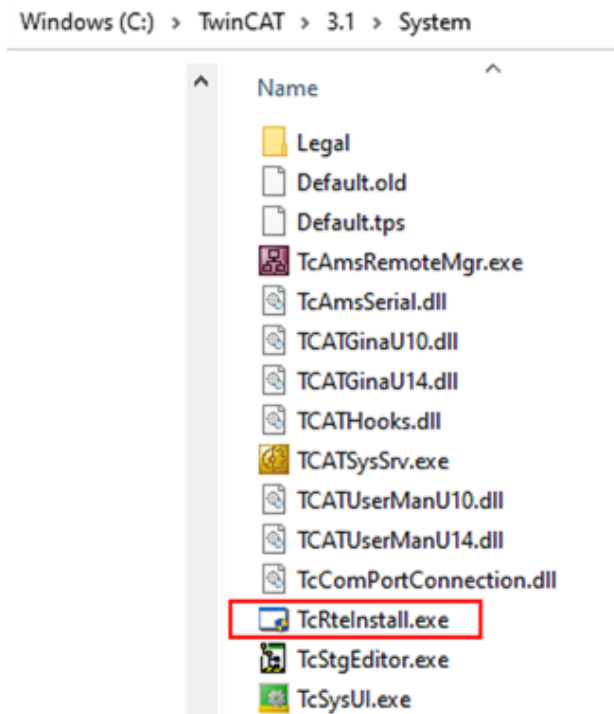
附图 74: System Manager “选项” (TwinCAT 2)

在 TwinCAT 3 环境中，这个功能需要通过菜单“TwinCAT”来调用：



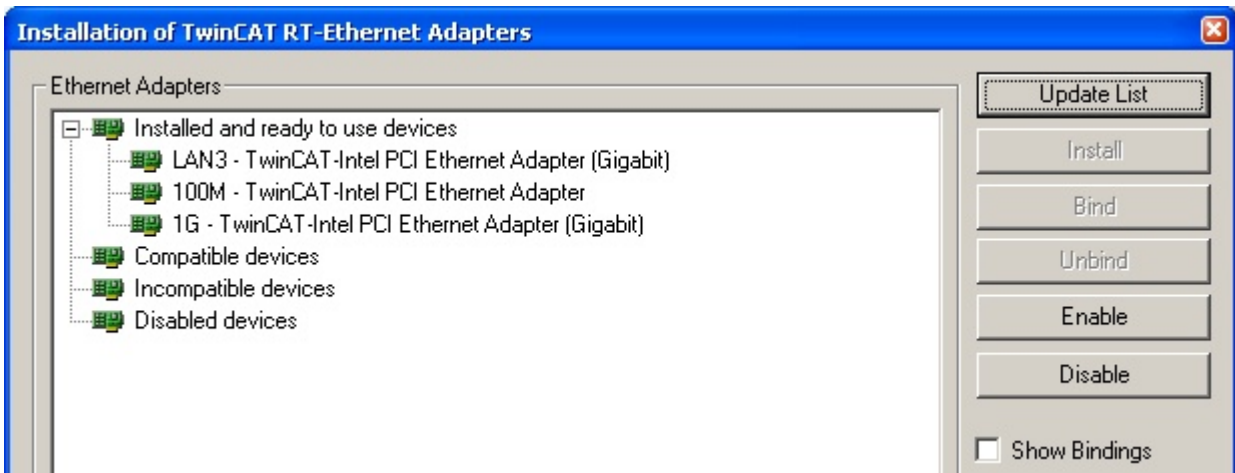
附图 75: 在 VS Shell 下调用 (TwinCAT 3)

B: 通过 TwinCAT 目录下的 TcRteInstall.exe



附图 76: TwinCAT 目录下的 TcRteInstall

两种情况下均出现以下对话框:



附图 77: 网络接口概览

在“Compatible devices（兼容设备）”下列出的接口可以通过“Install”按钮指定一个驱动程序。驱动程序只应安装在兼容的设备上。

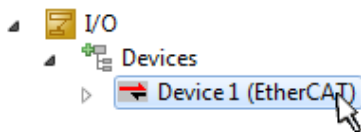
关于未签名驱动程序的 Windows 警告可以忽略。

或者，首先可以插入一个 EtherCAT 设备，如离线配置创建 [▶ 102] 章节所述，以便通过其 EtherCAT 属性（“Adapter”选项卡上的“Compatible Devices...”按钮）查看兼容的以太网端口：

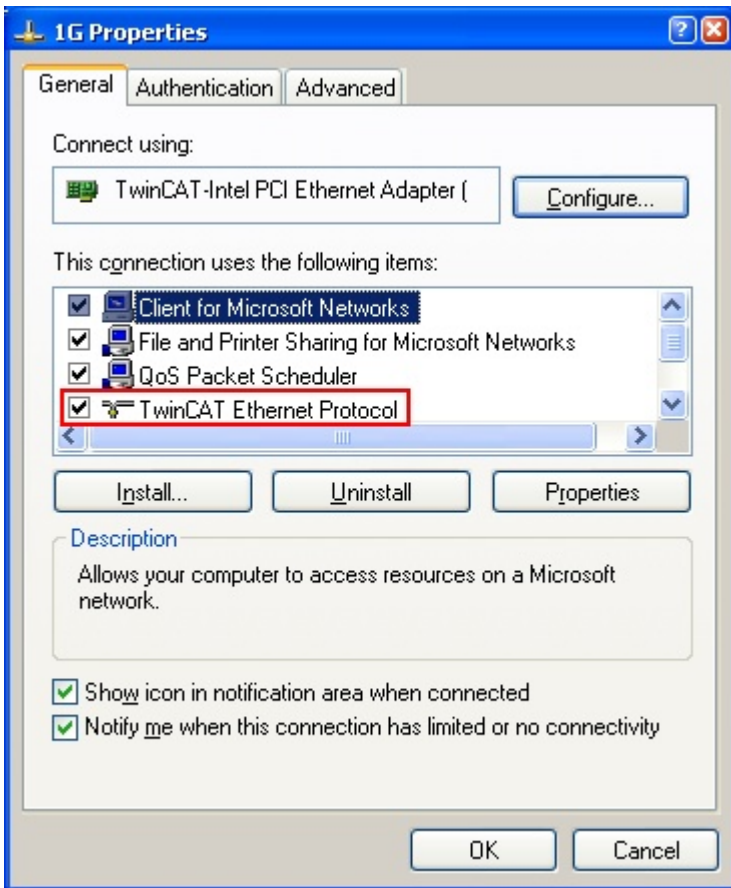


附图 78: EtherCAT 设备属性 (TwinCAT 2)：点击“Adapter”选项卡的“Compatible Devices...”

TwinCAT 3: EtherCAT设备的属性可以通过双击“I/O”下解决方案资源管理器中的“设备 (EtherCAT)”打开：

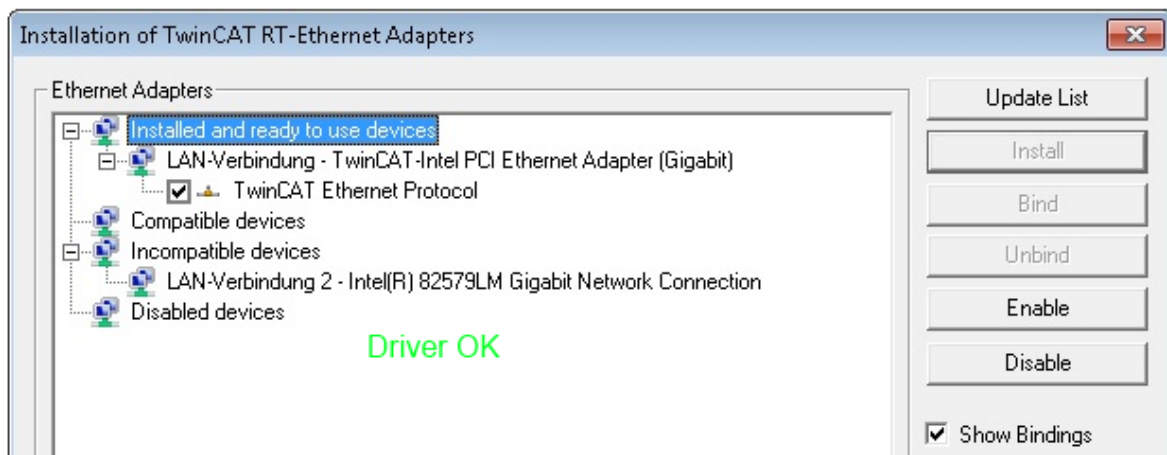


安装后，Windows 的网络接口概览中显示驱动程序已激活（Windows 开始→系统属性→网络）



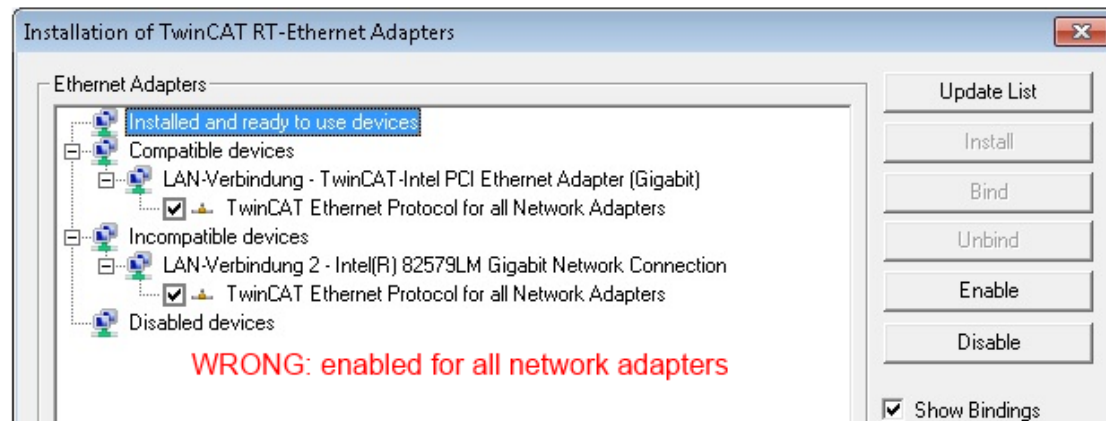
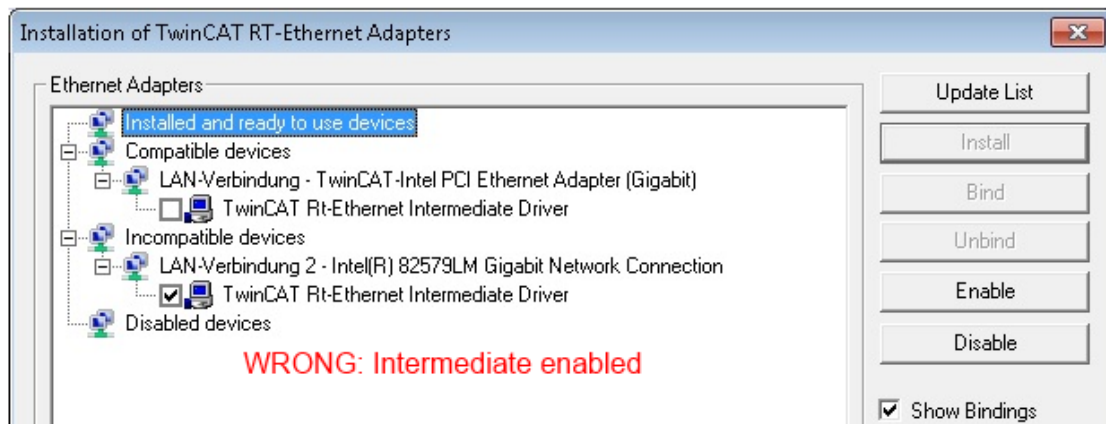
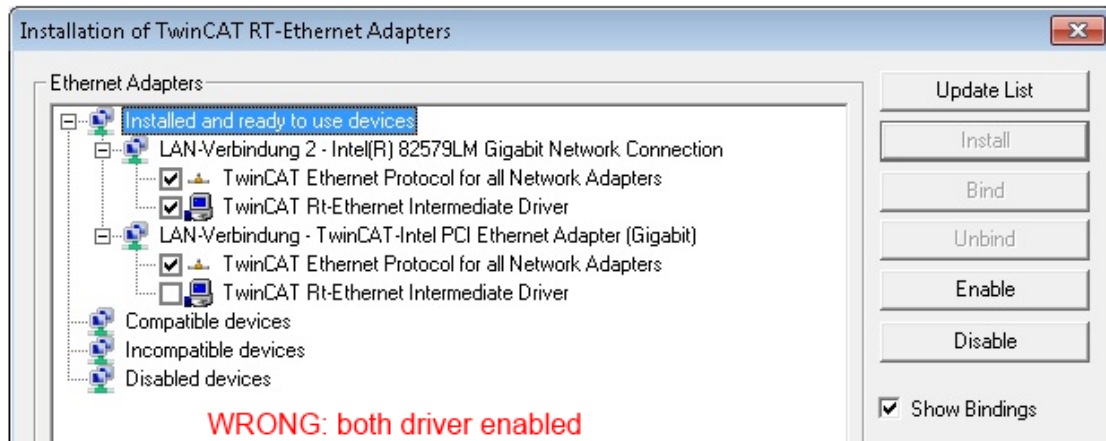
附图 79: Windows 的网络接口属性

驱动程序的正确设置如下:



附图 80: 以太网端口驱动程序的正确设置示例

必须避免下面几种可能的设置:



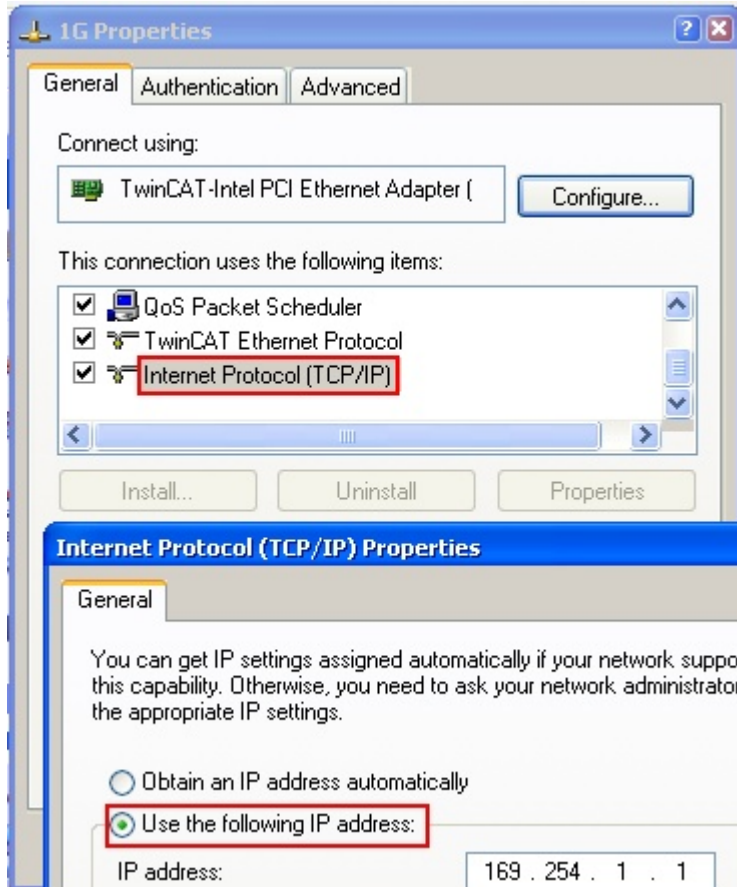
附图 81: 以太网端口驱动程序的错误设置



## 所用端口的 IP 地址

**i** IP 地址/DHCP

在大多数情况下，被配置为 EtherCAT 设备的以太网端口不会传输一般的 IP 数据包。因此，在使用 EL6601 或类似设备时，最好是通过“Internet Protocol TCP/IP”驱动设置为该端口指定一个固定的 IP 地址并禁用 DHCP。这样就避免了在没有 DHCP 服务器的情况下，以太网端口的 DHCP 客户端为自己分配默认 IP 地址所带来的延迟。例如，一个合适的地址空间是 192.168.x.x。



附图 82：以太网端口的 TCP/IP 设置

## 6.2.2 关于 ESI 设备描述文件的说明

### 最新 ESI 设备描述文件的安装说明

TwinCAT EtherCAT 主站/System Manager需要所使用设备的设备描述文件，以便在在线或离线模式下生成配置。设备描述包含在 XML 格式的 ESI 文件（EtherCAT Slave Information）中。这些文件可以向各个从站的制造商索取。一个 \*.xml 文件可能包含几个设备描述。

倍福 EtherCAT 设备的 ESI 文件可从[倍福公司网站](#)获取。

ESI 文件应存放在 TwinCAT 安装目录下。

默认设置：

- **TwinCAT 2:** C:\TwinCAT\IO\EtherCAT
- **TwinCAT 3:** C:\TwinCAT\3.1\Config\Io\EtherCAT

如果 ESI 文件在上次 System Manager 窗口打开后发生了变化，当打开一个新的 System Manager 窗口时，则会重新装载（一次）这些文件。

TwinCAT 的安装包括倍福 ESI 文件集，而该文件集是创建 TwinCAT build 版本时的最新 ESI 版本。

对于 TwinCAT 2.11/TwinCAT 3 及以上版本，如果编程 PC 连接到互联网，就可以通过以下方式从 System Manager 中更新 ESI 目录：

- **TwinCAT 2:** Option → “Update EtherCAT Device Descriptions”
- **TwinCAT 3:** TwinCAT → EtherCAT Devices → “Update Device Descriptions (via ETG Website)…”

也可以通过[TwinCAT ESI Updater \[► 101\]](#) 更新 ESI 目录。



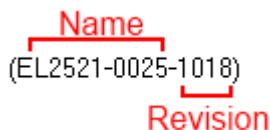
#### ESI

\*.xml 文件与 \*.xsd 文件关联，后者描述了 ESI XML 文件的结构。因此，如需更新 ESI 设备描述，这两种文件类型都应更新。

### 设备的识别

EtherCAT 设备/从站由四个属性来区分，它们决定了完整的设备标识符。例如，设备标识符 EL2521-0025-1018 由以下部分组成：

- 系列号 “EL”
- 型号 “2521”
- 子版本号 “0025”
- 修订版本 “1018”

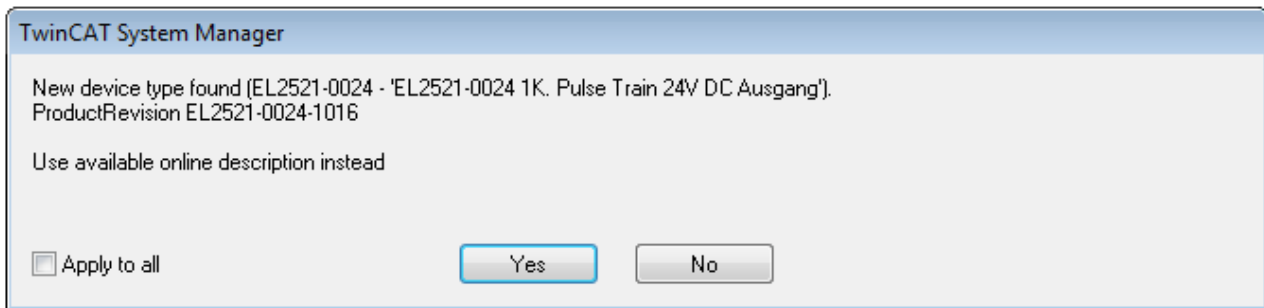


附图 83: 标识符结构

名称 + 类型组成的订货号（此处：EL2521-0025）描述了设备功能。修订版本表示技术上的升级，并由倍福公司进行管理。原则上，一个较高版本的设备可以替换一个较低版本的设备，除非在文件中另有规定。每个修订版都有自己的 ESI 描述。参见详细说明 [\[► 12\]](#)。

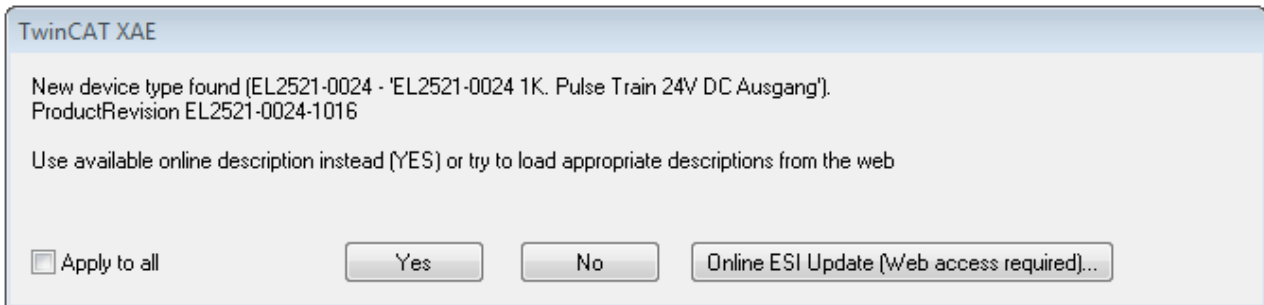
### 在线描述

如果 EtherCAT 配置通过扫描实际设备而在线创建的（参见在线设置部分），并且没有找到可用的从站 ESI 描述（由名称和修订版本号指定），System Manager 会询问是否应使用存储在设备中的描述。在任何情况下，System Manager 需要这些信息来正确设置与从站的周期性和非周期性通信。



附图 84: 在线描述信息窗口 (TwinCAT 2)

在 TwinCAT 3 中, 会出现一个类似的窗口, 它也提供网络更新:



附图 85: 在线描述信息窗口 (TwinCAT 3)

尽可能不要选择 Yes, 而是向从站设备制造商索取所需 ESI。安装完 XML/XSD 文件后, 应重新配置。

### 注意

#### 扫描设备时, 修改“推荐”配置

- ✓ 如果扫描发现了 TwinCAT 未知的设备, 必须对以下两种情况区别处理。这里以 EL2521-0000 的修订版 1019 为例
- a) 根本没有 EL2521-0000 设备的 ESI, 无论是 1019 版本还是更早版本。所以必须向制造商 (这种情况下是倍福) 申请 ESI。
- b) 存在 EL2521-0000 设备的 ESI, 但版本比实际扫描到的更旧, 例如 1018 或 1017。  
此时应首先进行内部检查, 以确定库存的备件是否可以配置为高版本。一个新的/更高的修订版通常也会带来新的功能。如果不使用这些功能, 可以毫不犹豫地配置中使用以前的修订版 1018 继续工作。这也是倍福兼容性规则所声明的。

请特别参阅“[关于使用倍福 EtherCAT IO 组件的一般注意事项](#)”一章。关于手动配置请参考“[离线配置创建 \[102\]](#)”一章。

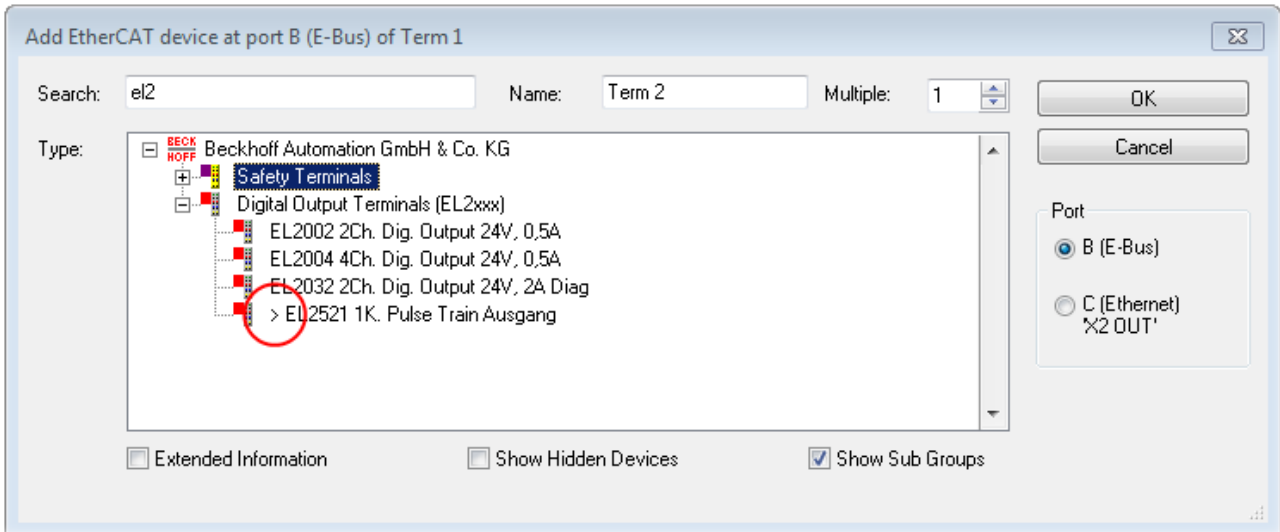
如果使用在线描述, System Manager 会从 EtherCAT 从站的 EEPROM 中读取一份设备描述。在复杂的从站中, EEPROM 的大小可能不足以容纳完整的 ESI, 此时配置中的 ESI 就会不完整。因此, 建议这种情况下优先使用离线 ESI 文件。

System Manager 在其 ESI 目录下为在线扫描找到的设备创建一个新的描述文件“OnlineDescription0000...xml”, 其中包含所有在线读取的 ESI 描述。

`OnlineDescriptionCache000000002.xml`

附图 86: System Manager 创建的文件 OnlineDescription.xml

也可以稍后再向该配置中手动添加一个从站。在线创建的从站在选择列表中以前缀“>”表示 (参见图以 EL2521 的在线记录 ESI 为例进行说明)。



附图 87: 以EL2521为例说明用在线 ESI 文件创建的从站

如果使用了这样的在线 ESI 文件，而后来又拿到了制造商的 ESI 文件，应按以下方式删除 OnlineDescription.xml 文件：

- 关闭所有 System Manager 窗口
- 在 Config Mode 下重启 TwinCAT
- 删除 “OnlineDescription0000...xml”
- 重新启动 TwinCAT System Manager (System Manager)

在此过程后，该文件不再显示。如有必要，请按 <F5> 更新

### ● TwinCAT 3.x 的在线描述

**i** 除了上述 “OnlineDescription0000...xml” 文件外，TwinCAT 3.x 还创建了一个 EtherCAT 缓存，其中包含新发现的设备，例如在 Windows 7 下：

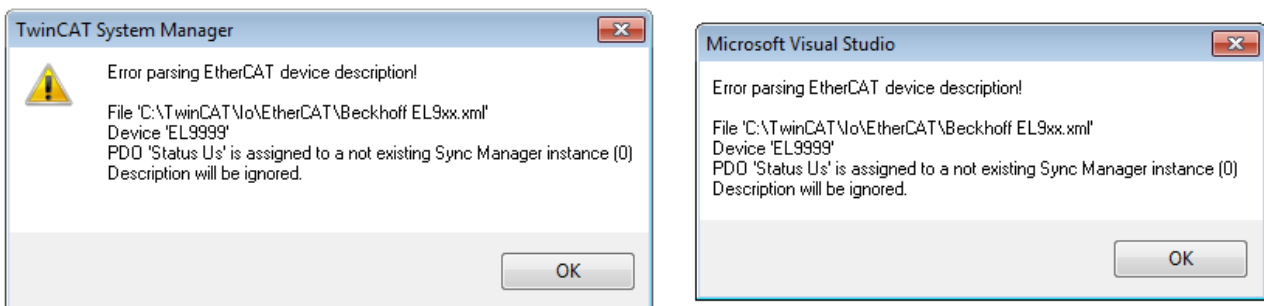
`C:\User\[USERNAME]\AppData\Roaming\Beckhoff\TwinCAT3\Components\Base\EtherCATCache.xml`

（请注意操作系统的语言设置！）

该文件也必须删除。

### ESI 文件出错

如果某个 ESI 文件出错，System Manager 无法读取，则 System Manager 会弹出一个信息窗口。



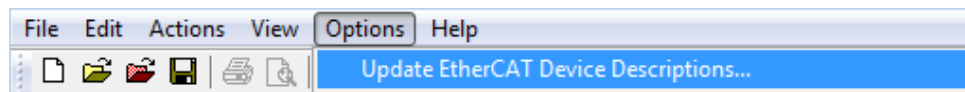
附图 88: 错误 ESI 文件的信息窗口（左：TwinCAT 2；右：TwinCAT 3）

可能的原因包括：

- \*.xml 的结构与相关的 \*.xsd 文件不一致 → 检查原理图
- 内容不能被翻译成设备描述 → 联系从站的制造商

### 6.2.3 TwinCAT ESI Updater

对于 TwinCAT 2.11 及以上版本，如果存在在线外网连接，System Manager 可以自动搜索当前的倍福 ESI 文件：

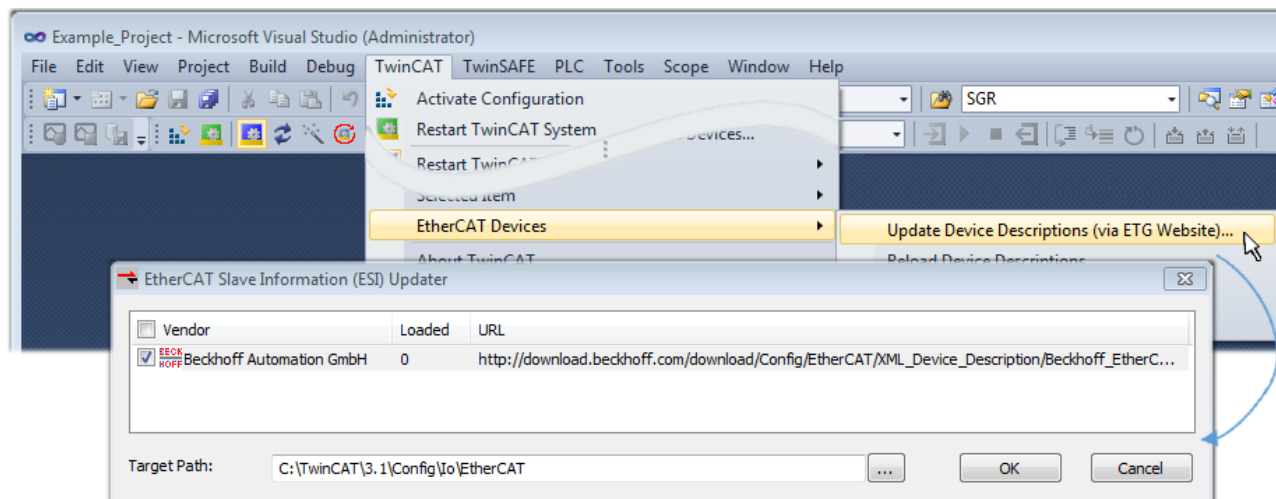


附图 89：使用 ESI Updater (>= TwinCAT 2.11)

调用方法：

“Options” → “Update EtherCAT Device Descriptions”

在 TwinCAT 3 中，则选择：



附图 90：使用 ESI Updater (TwinCAT 3)

ESI Updater (TwinCAT 3) 选项非常方便，可将 EtherCAT 制造商通过互联网提供的 ESI 数据自动下载到 TwinCAT 目录中 (ESI = EtherCAT Slave Information)。TwinCAT 访问存储在 ETG 的集中 ESI ULR 目录列表；然后可以在 Updater 对话框中查看这些条目，但是无法在此进行更改。

调用方式：

“TwinCAT” → “EtherCAT Devices” → “Update Device Description (via ETG Website)…”。

### 6.2.4 Online 和 Offline 之间的区别

Online 和 Offline (在线和离线) 之间的区别是针对实际存在物理 I/O (驱动器、端子模块、EJ-模块等) 而言的。如果需要提前在编程电脑上进行系统配置，例如在笔记本电脑上，则只能在“Offline configuration (离线配置)”模式下进行。此时所有组件都必须在配置中手动输入，例如根据电气设计图纸。

如果目标控制器已经连接到 EtherCAT 系统，所有设备都已通电，且网络基础设施已准备就绪，就可以简单地通过目标控制器的“scanning (扫描)”来生成 TwinCAT 配置。这就叫做在线配置。

在任何情况下，每次启动过程中，EtherCAT 主站都会检查所发现的从站是否与当前配置相符。这个检查的规则可以在从站的扩展设置界面中进行选择。请参考最新 ESI 设备描述文件的安装说明 [▶ 98]。

#### 配置前的准备工作：

- 实际 EtherCAT 硬件 (设备、耦合器、驱动器) 必须存在并安装完成
- 所有设备/模块必须通过 EtherCAT 电缆连接或者在 I/O 站中以设计的顺序组装连接
- 所有设备/模块接上电源，做好通信准备
- 目标系统上的 TwinCAT 必须处于 Config Mode (配置模式)。

#### 在线扫描过程包括：

- 检测 EtherCAT 设备 [▶ 107] (IPC 的以太网端口)

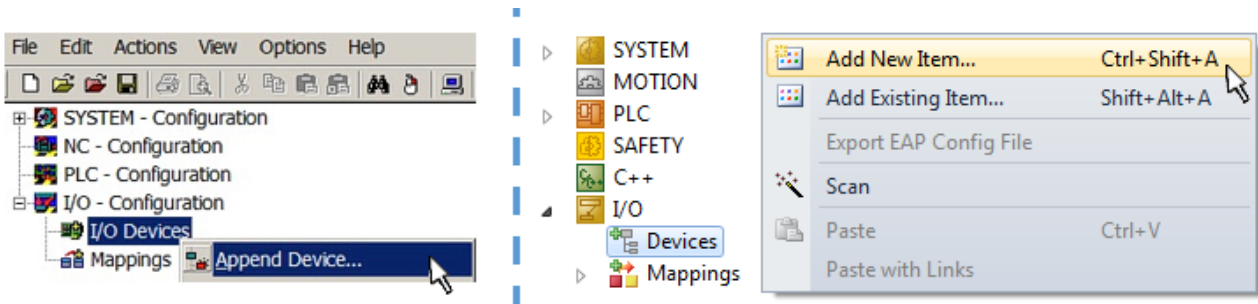
- 检测连接的 EtherCAT 从站 [▶ 108]。这一步骤可独立于前一步骤单独进行。
- 故障排除 [▶ 111]

也可以通过现有配置扫描 [▶ 111] 以进行比较。

## 6.2.5 创建 OFFLINE 配置

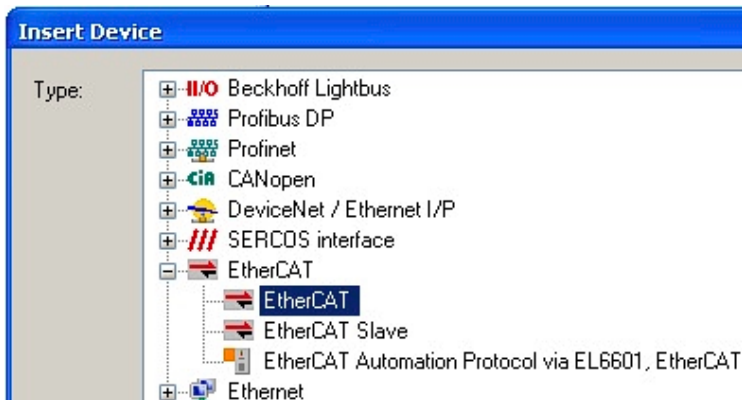
### 创建 EtherCAT 设备

在一个空白的 System Manager 窗口中创建一个 EtherCAT 设备。



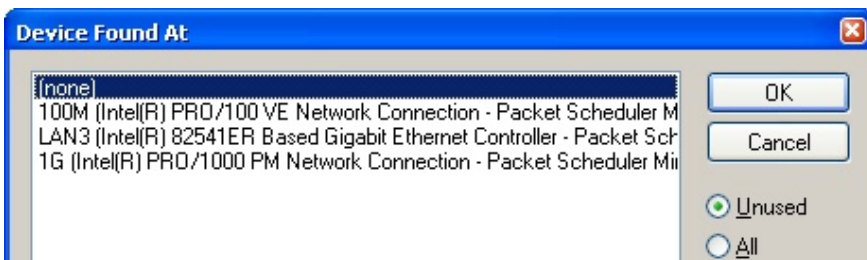
附图 91: 添加 EtherCAT 设备 (左: TwinCAT 2; 右: TwinCAT 3)

对于带有 EtherCAT 从站的 EtherCAT I/O 应用, 选择类型“EtherCAT”。对于目前通过 EL6601/EL6614 实现的 publisher/subscriber (发布/订阅) 服务, 选择“EtherCAT Automation Protocol via EL6601”。



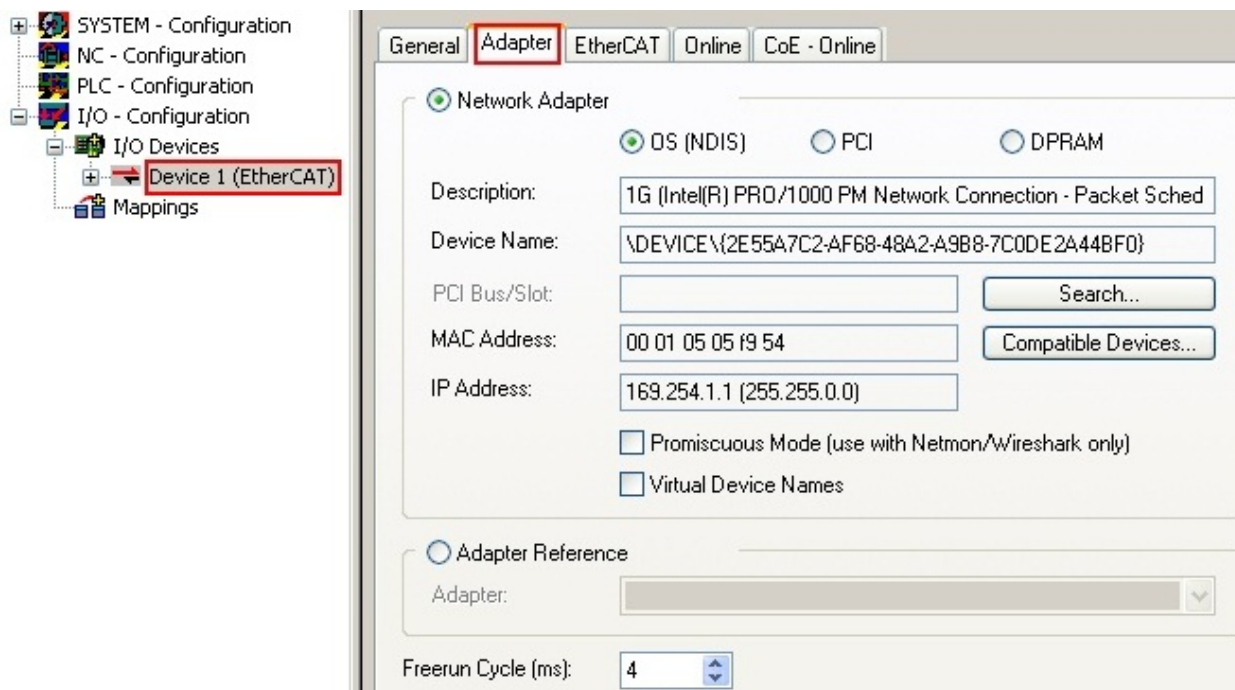
附图 92: 选择 EtherCAT 连接 (TwinCAT 2.11, TwinCAT 3)

然后在 TwinCAT runtime 运行系统中为这个虚拟设备分配一个实际的以太网端口。



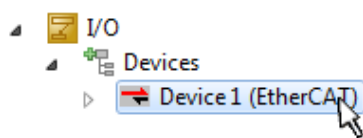
附图 93: 选择以太网端口

可以在创建 EtherCAT 设备时自动弹出的窗体中进行选择, 也可以将来在属性对话框中进行设置/修改; 参见图“EtherCAT 设备属性 (TwinCAT 2)”。



附图 94: EtherCAT 设备属性 (TwinCAT 2)

TwinCAT 3: EtherCAT设备的属性可以通过双击 “I/O” 下解决方案资源管理器中的“设备 (EtherCAT)” 打开:

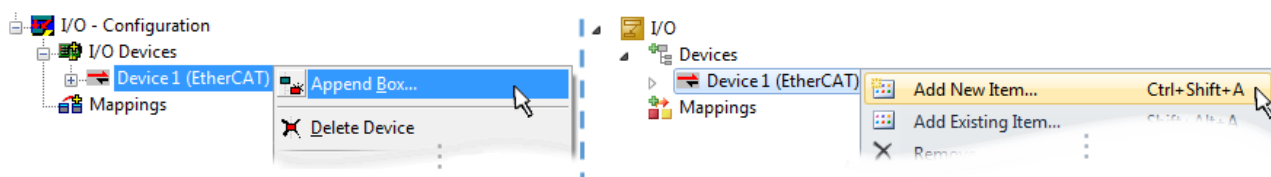


### ● 选择以太网端口

**I** 在安装了TwinCAT实时驱动程序的EtherCAT设备上才能选择以太网端口。这必须为每个端口单独进行。请参考各自的安装页面 [▶ 92]。

### 配置 EtherCAT 从站

选中配置树中的一个设备并右键单击，可以进一步添加其它设备。



附图 95: 添加 EtherCAT 设备 (左: TwinCAT 2; 右: TwinCAT 3)

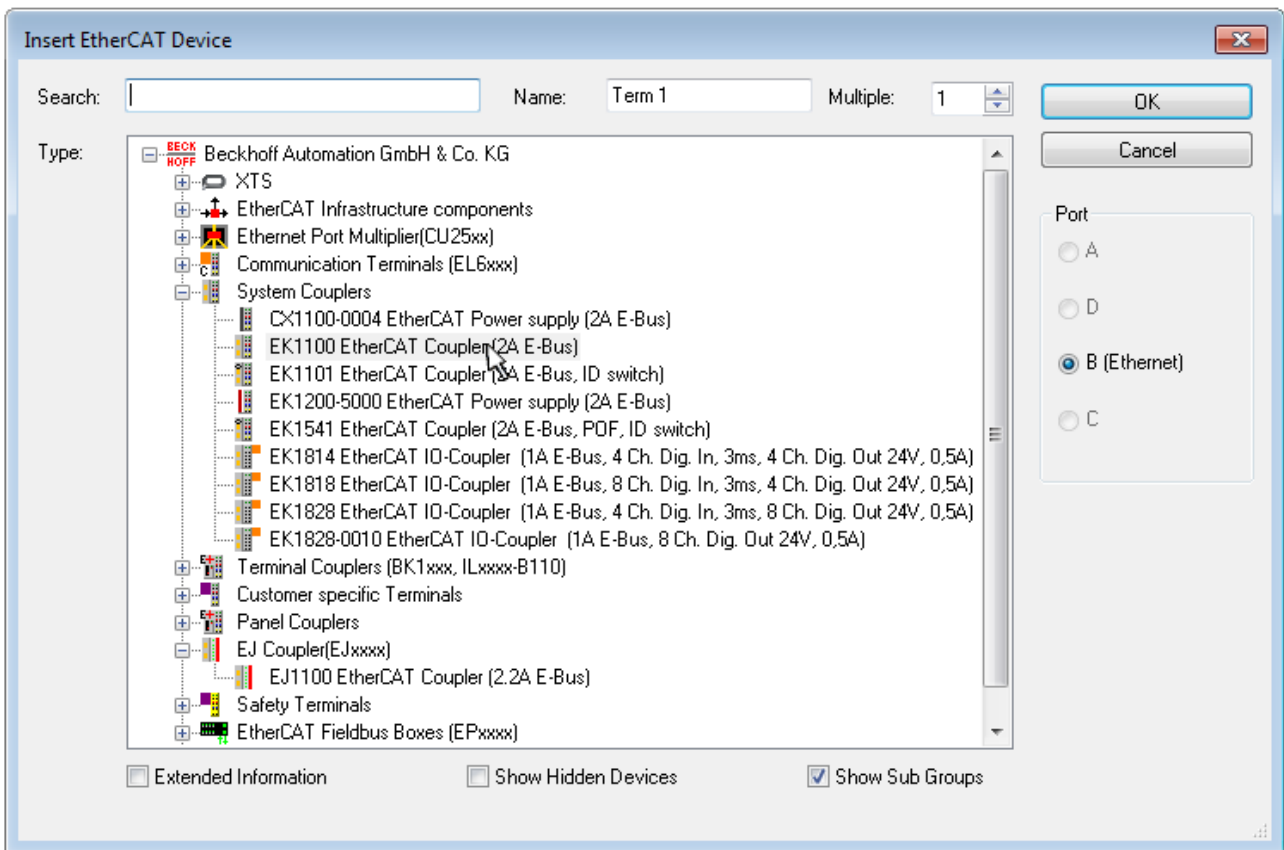
弹出选择新设备的对话框 对话框中只显示已有 ESI 文件的设备。

只显示可以添加到上一步选中项之后的设备，以供选择。也会显示端口可用的物理层（图“新增 EtherCAT 设备的选择对话框”）。如果是基于电缆的带PHY传输的 Fast-Ethernet（快速以太网）物理层，那么也只能选择基于电缆的设备，如图“新增 EtherCAT 设备的选择对话框”所示。如果上一个设备有多个空闲的端口（例如EK1122 或 EK1100），可以在右边选择需要的端口（A）。

### 物理层概述

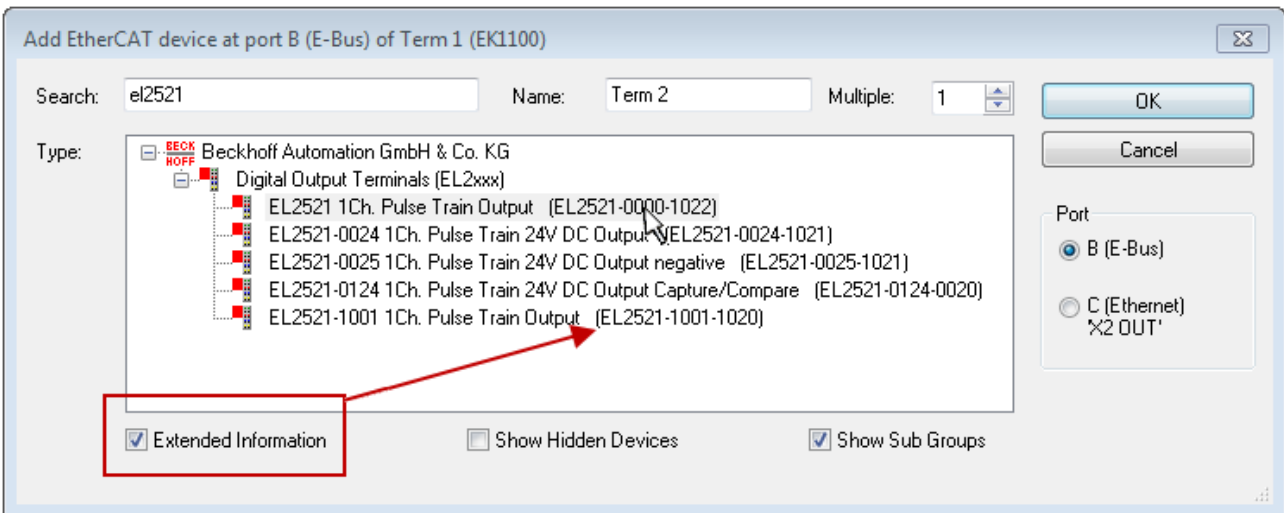
- “Ethernet”：基于电缆的 100BASE-TX；耦合器、盒模块、带 RJ45/M8/M12 连接器的设备
- “E-Bus”：LVDS “端子模块总线”，EtherCAT 插拔式模块 (EJ)，EtherCAT 端子模块 (EL/ES)，各种模块化模块

Search 搜索框用于查找指定的设备（自 TwinCAT 2.11 或 TwinCAT 3 起）。



附图 96: 新增 EtherCAT 设备的选择对话框

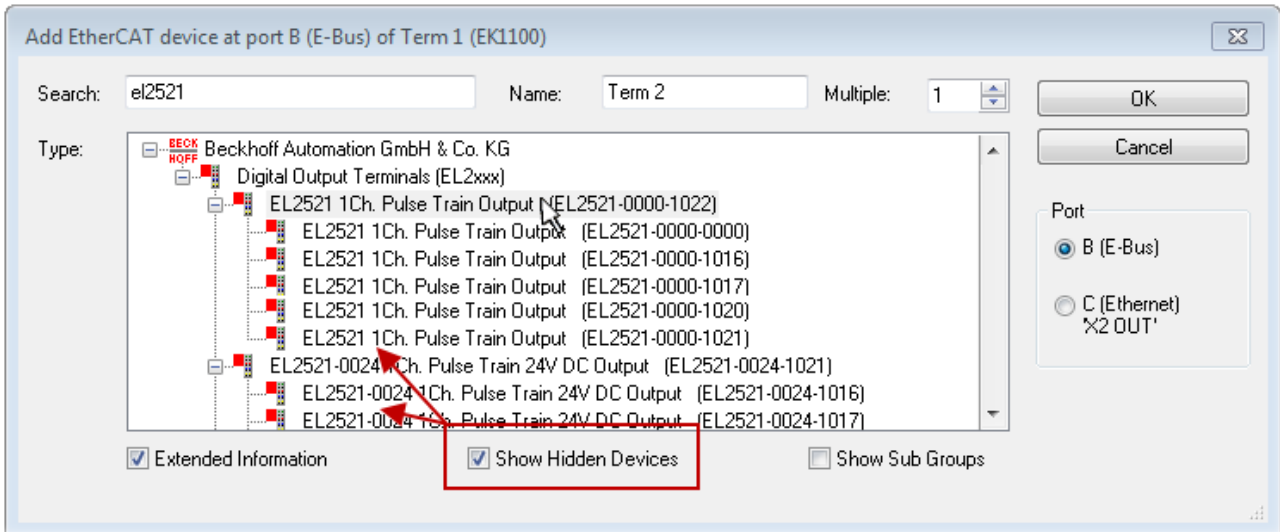
默认情况下，只要根据名称/设备类型进行选择。如果要选择设备的特定版本，可以勾选“Extended Information”，把版本信息也显示出来。



附图 97: 显示设备版本

很多时候，由于历史原因或增加功能，例如进行了技术升级，一个设备可能存在多个版本。为简化起见（见图“新增 EtherCAT 设备的选择对话框”），在倍福设备的选择对话框中只显示最近（即最高）的修订版本，从而也是最新出厂的设备版本。如需以 ESI 描述显示系统可用的所有设备版本，请勾选“Show Hidden Devices（显示隐藏设备）”复选框，见图“显示以前的版本”。





附图 98: 显示以前的版本

**● 修订版本的设备选择 – 兼容性**

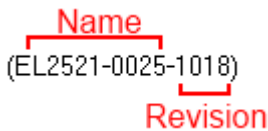
**i** ESI描述还定义了过程图像、主站和从站/设备之间的通信类型以及设备功能（如果适用）。物理设备（固件，如果适用）必须支持主站的通信查询/设置。这是向后兼容的，也就是说，如果EtherCAT主站将其视为较早版本，那么应支持较新设备（较高版本）。对于Beckhoff的EtherCAT端子/端子盒/EJ模块，应符合以下兼容性规则。

**系统中的设备版本 >= 配置中的设备版本**

这也使得后续更换设备时无需改变配置（驱动器可能存在不同规格）。

**示例**

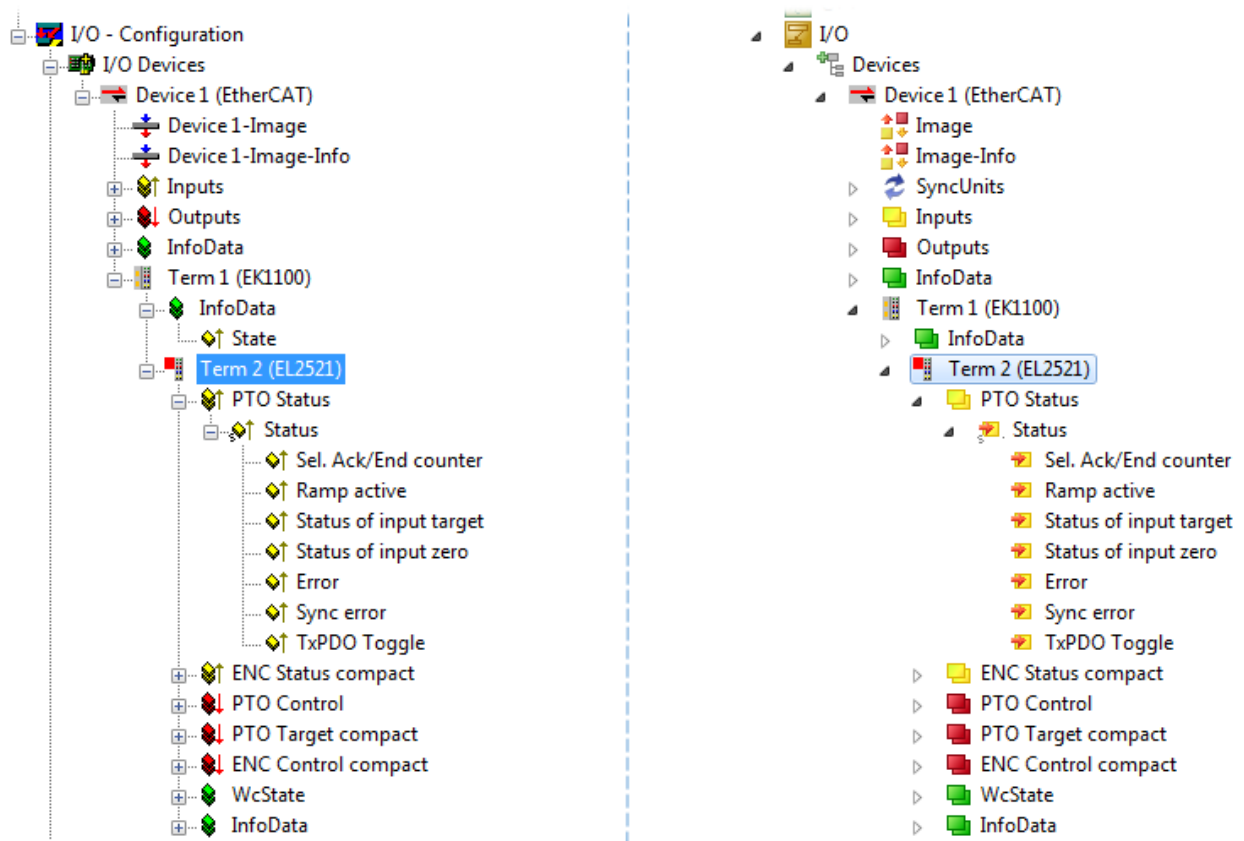
如果在配置中指定了EL2521-0025-1018，则在实践中可以使用EL2521-0025-1018或更高版本（-1019, -1020）。



附图 99: 终端的名称/修订版本

如果TwinCAT系统中存在当前ESI描述，则选择对话框中提供的最新修订版本与Beckhoff的生产状态相符。如果在实际应用中使用了当前Beckhoff设备，建议在创建新配置时使用最近的设备版本。在应用中使用库存的较早设备时，方才应使用较早的修订版本。

在这种情况下，设备的过程图像显示在配置树中，并可以进行如下参数化：与任务的链接、CoE/DC设置、插件定义、启动设置...




附图 100: TwinCAT 树中的 EtherCAT 端子模块 (左: TwinCAT 2; 右: TwinCAT 3)



## 6.2.6 创建ONLINE配置

### 检测/扫描 EtherCAT 设备

如果 TwinCAT 系统处于CONFIG模式，则可以使用在线设备搜索。这可以通过下方信息栏中的符号表示：

- 在 TwinCAT 2 上，通过 TwinCAT System Manager 窗口中蓝色显示的 **Config Mode** 来表示“Config Mode”。
- 在 TwinCAT 3 上，通过开发环境用户界面中的符号  表示。

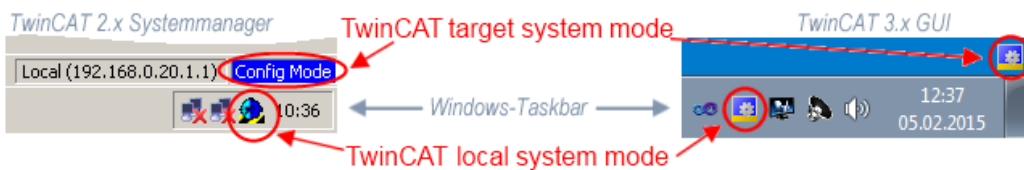
以下方法可以将TwinCAT 设置成配置模式：

- TwinCAT 2: 通过选择菜单栏中的  或通过“Actions”→“Set/Reset TwinCAT to Config Mode...”
- TwinCAT 3: 通过选择菜单栏中的  或通过“TwinCAT”→“Restart TwinCAT (Config Mode)”

### 配置模式下的在线扫描

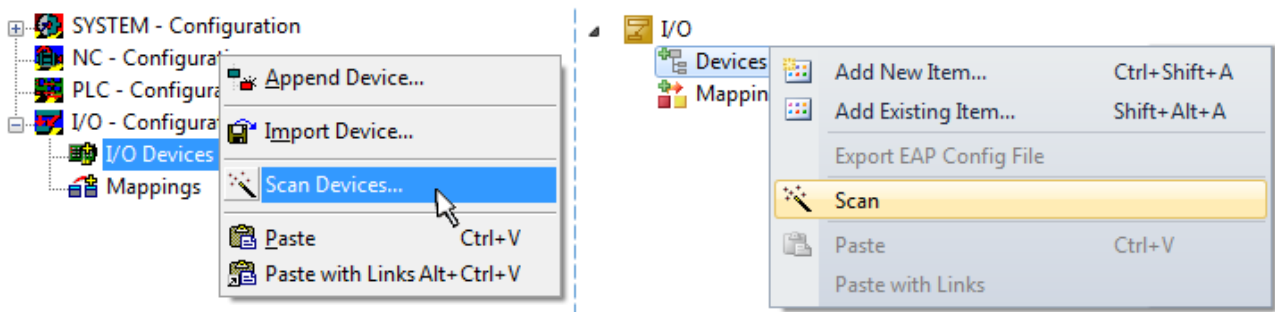
**i** 在 RUN 模式（生产运行）下，在线搜索不可用。注意 TwinCAT 编程系统和 TwinCAT 目标系统之间的区别。

Windows 任务栏中的 TwinCAT 2 图标 () 或 TwinCAT 3 图标 () 始终显示本地 IPC 的 TwinCAT 模式。与此相对，TwinCAT 2 的 System Manager 窗口或 TwinCAT 3 的用户界面会显示目标系统的状态。



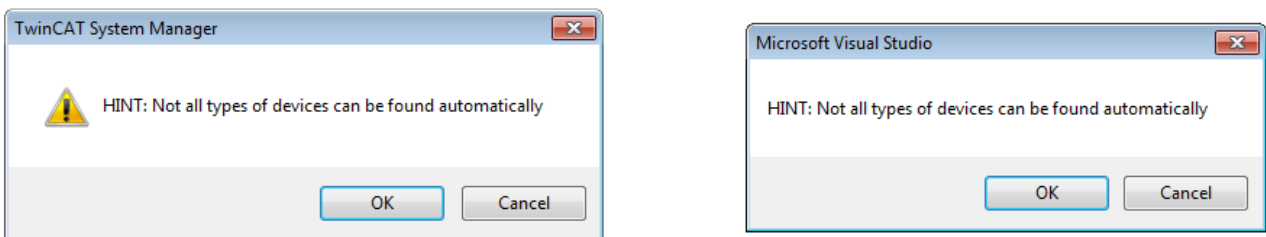
附图 101: 本地/目标系统差异 (左: TwinCAT 2; 右: TwinCAT 3)

右键单击配置树中的“I/O Devices”可以打开搜索对话框。



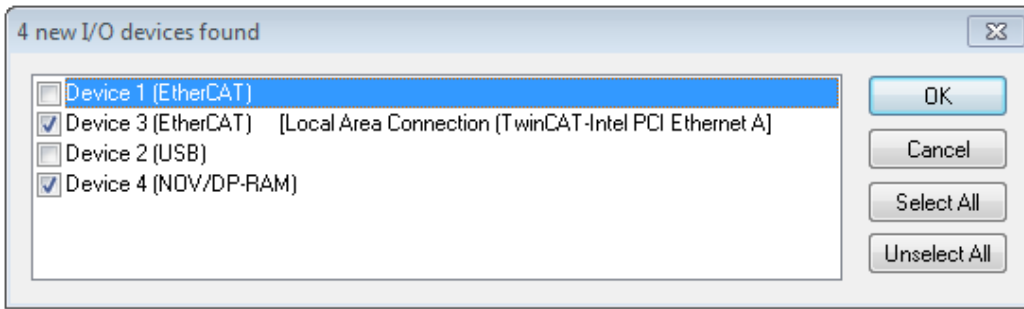
附图 102: Scan Devices (扫描设备) (左: TwinCAT 2; 右: TwinCAT 3)

这种扫描模式不仅试图找到 EtherCAT 设备（或可作为 EtherCAT 设备使用的以太网端口），而且还试图找到 NOVDRAM、现场总线卡、SMB 等。然而，并非所有设备都能自动找到。



附图 103: 自动设备扫描的注意事项 (左: TwinCAT 2; 右: TwinCAT 3)

安装了 TwinCAT real-time 实时驱动程序的以太网端口被显示为“RT Ethernet”设备。为测试目的，一个 EtherCAT 帧被发送到这些端口。如果扫描过程从响应中检测到已连接一个 EtherCAT 从站，该端口将立即显示为“EtherCAT Device”。



附图 104: 检测到的以太网设备

通过各自的复选框可以选择设备（如图“检测到的以太网设备”所示，例如图中设备 3 和设备 4 被选中）。在通过“OK”按钮进行确认后，建议对所有选定的设备进行设备扫描，见图“自动创建 EtherCAT 设备后的扫描”。

### ● 选择以太网端口



在安装了TwinCAT实时驱动程序的EtherCAT设备上才能选择以太网端口。这必须为每个端口单独进行。请参考各自的安装页面 [▶ 92]。

## 检测/扫描 EtherCAT 设备

### ● 在线扫描功能



在扫描过程中，主站在从站的 EEPROM 中查询 EtherCAT 从站的身份信息。名称和修订版本号用于确定类型。从存储的 ESI 数据中找到各个设备，并以其 ESI 文件定义的默认设置集成到当前配置。

**Name**  
(EL2521-0025-1018)  
**Revision**

附图 105: 默认设置示例

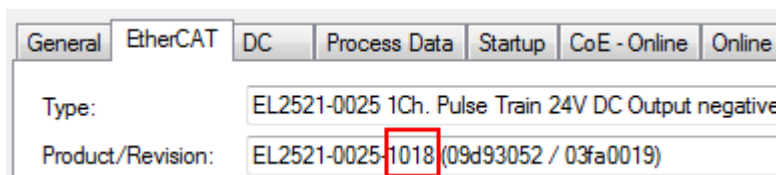
## 注意

### 在量产机型上使用从站扫描功能

扫描功能应谨慎使用。它是一个实用和快速的工具，用于创建一个初始配置，作为调试的基础。然而，扫描功能不应用于设备量产或重复生产时创建配置，而是仅在必要时用于和已定义的初始配置进行比较 [▶ 111]。背景：由于倍福出于产品维护的原因，已交付产品还会继续更新修订版本。通过在线扫描可以临时创建配置，根据设备清单，在线扫描的配置与初始配置是完全相同的（在机器结构相同的情况下）；但是，个别设备的修订版本可能与初始配置不同。

### 示例：

A 公司制造了一台机器 B 的原型机，该机器以后将被批量生产。为此，制造了原型机。在 TwinCAT 中对 I/O 设备进行了扫描，并创建了初始配置“B. TSM”。修订版本为 1018 的 EL2521-0025 EtherCAT 端子模块装在某处。于是，它就这样创建到了 TwinCAT 配置文件中：



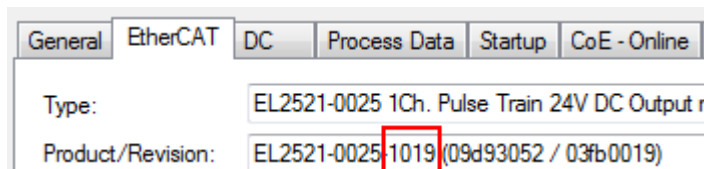
附图 106: 安装修订版本-1018的EtherCAT 端子模块，

同样，在原型机测试阶段，该端子模块的功能和属性由程序员/调试工程师进行测试完成以后就可以随时投入使用，比如通过 PLC “B.pro” 或 NC 寻址访问。（这也同样适用于TwinCAT 3 解决方案）。

原型开发完成以后，机器 B 开始批量生产，倍福继续为该机器提供 EL2521-0025-0018。如果机器批量生产部门的调试工程师总是进行扫描，那么每台机器都会再次产生一个内容相同的 B 配置。同样，A 公司可能会在全球范围内为即将批量生产的带有 EL2521-0025-1018 端子模块的机器创建备件仓库。

一段时间后，倍福对 EL2521-0025 进行了升级，新增了功能 C。因此更改了固件，在外观上标注了更高的固件版本和**新的修订版本-1019**。尽管如此，新设备自动支持前一版本的功能和界面；因此，没有必要对“B. TSM”甚至“B. pro”进行调整。量产机器可以继续用“B. tsm”和“B. pro”来生产；为了检查生产的机器，需要对照初始配置“B. tsm”进行**比较扫描** [► 111]。

然而，如果现在机器批量生产部门不使用“B. tsm”，而是进行扫描来创建生产用的配置，那么修订版本-1019 将被自动检测并创建到配置中：



附图 107: 检测修订版本 1019 的 EtherCAT 端子模块

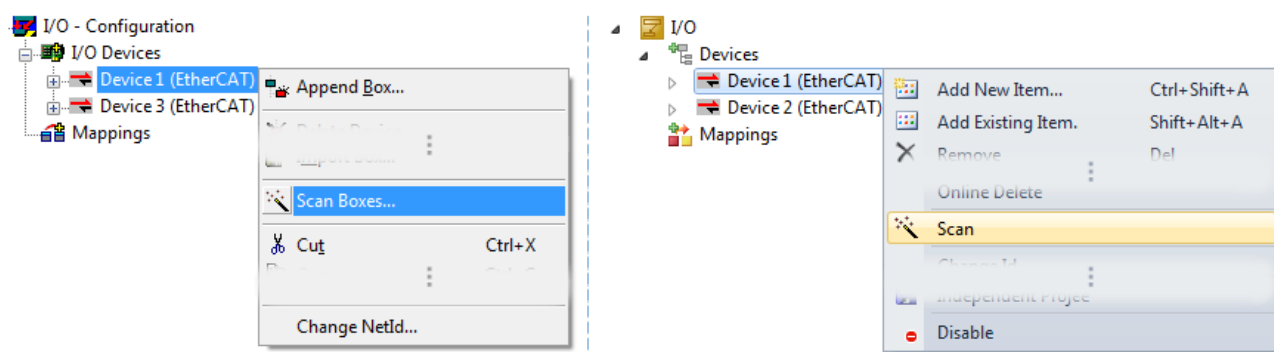
调试工程师通常不会注意到这一点。TwinCAT 也不会发出任何信号，因为实际上是从头创建了一个新的配置。然而，根据兼容性规则，这意味着不应该将 EL2521-0025-1018 的备件安装到这台机器上（即使这在绝大多数情况下还是可以使用的）。

此外，还可能发生的情况是，由于 A 公司的开发及生产，EL2521-0025-1019 的新功能 C（例如，改进的模拟量滤波器或用于诊断的额外过程数据）被发现并使用了，而无需经过内部审核。以前的备件库存就不能再用于以这种方式创建的新配置“B2. TSM”。如果机器已经开始批量生产，扫描就应该只是为了提供信息，以便和定义的初始配置进行比较。更改配置务必小心！

如果在配置中创建了 EtherCAT 设备（手动或通过扫描），则可以在 I/O 区域扫描设备/从站。



附图 108: 自动创建 EtherCAT 设备后的扫描（左：TwinCAT 2；右：TwinCAT 3）



附图 109: 手动扫描特定 EtherCAT 主站上的设备（左：TwinCAT 2；右：TwinCAT 3）

在 System Manager (TwinCAT 2) 或用户界面 (TwinCAT 3) 中，可以通过状态栏底部的进度条监控扫描过程。



附图 110: TwinCAT 2 的扫描进度示例

配置已建立，然后可以切换到在线状态 (OPERATIONAL)。



附图 111: Config/FreeRun 查询 (左: TwinCAT 2; 右: TwinCAT 3)

在 Config/FreeRun 模式下, System Manager 在蓝色和红色之间交替显示, 而 EtherCAT 设备继续以 4ms 的空转周期时间 (默认设置) 运行, 即使没有活动任务 (NC, PLC) 也不例外。



附图 112: 显示在状态栏下方的“Free Run”和“Config Mode”来回切换



附图 113: TwinCAT 也可以通过一个按钮切换到这种状态 (左: TwinCAT 2; 右: TwinCAT 3)

然后 EtherCAT 系统应处于功能循环状态, 如图在线显示示例所示。

The screenshot shows the 'EtherCAT Online' window with the following data:

No	Addr	Name	State	CRC
1	1001	Term 1 (EK1100)	OP	0, 0
2	1002	Term 2 (EL2008)	OP	0, 0
3	1003	Term 3 (EL3751)	SAFEOP	0, 0
4	1004	Term 4 (EL2521-0024)	OP	0

Counter	Cyclic	Queued
Send Frames	47718	+ 6791
Frames / sec	499	+ 31
Lost Frames	0	+ 0
Tx/Rx Errors	0	/ 0

附图 114: 在线显示示例

请注意:

- 所有从站应处于 OP 状态
- EtherCAT 主站的“Actual State”应处于 OP 状态
- “frames/sec”应与周期时间相匹配, 同时将 Sent Frames 纳入考量。
- 不应出现过多的“Lost Frames”或 CRC 错误

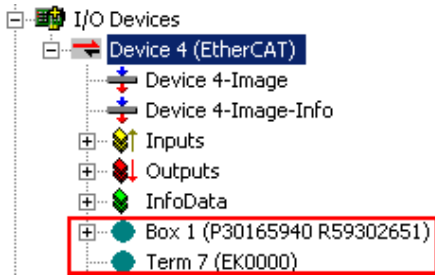
至此, 配置工作就完成了。该配置可以按照手动流程 [▶ 102] 中的描述进行修改。

故障排除

在扫描过程中可能会出现各种状况：

- 检测到一个**unknown device (未知设备)**，即没有 ESI XML 描述的 EtherCAT 从站。  
此时，System Manager 可以读取该设备中存储的任何 ESI。这种情况在“关于 ESI 设备描述的说明”一章中进行了描述。
- **Device are not detected properly (设备未被正确检测到)**  
可能的原因包括：
  - 数据链路出现故障，导致扫描过程中数据丢失
  - 从站的设备描述无效

应有针对性地检查接线和设备，例如通过 emergency scan (紧急扫描) 进行检查。然后重新执行扫描。



附图 115：识别错误

在System Manager中，这种情况下的设备可能被识别为 EK0000 或 unknown devices (未知设备)。无法操作或操作无效。

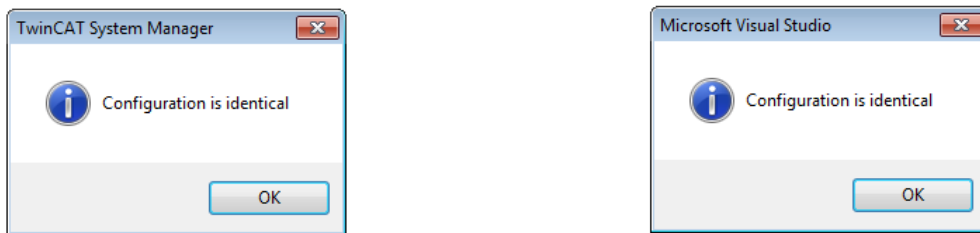
扫描现有配置

**注意**

**比较后修改配置**

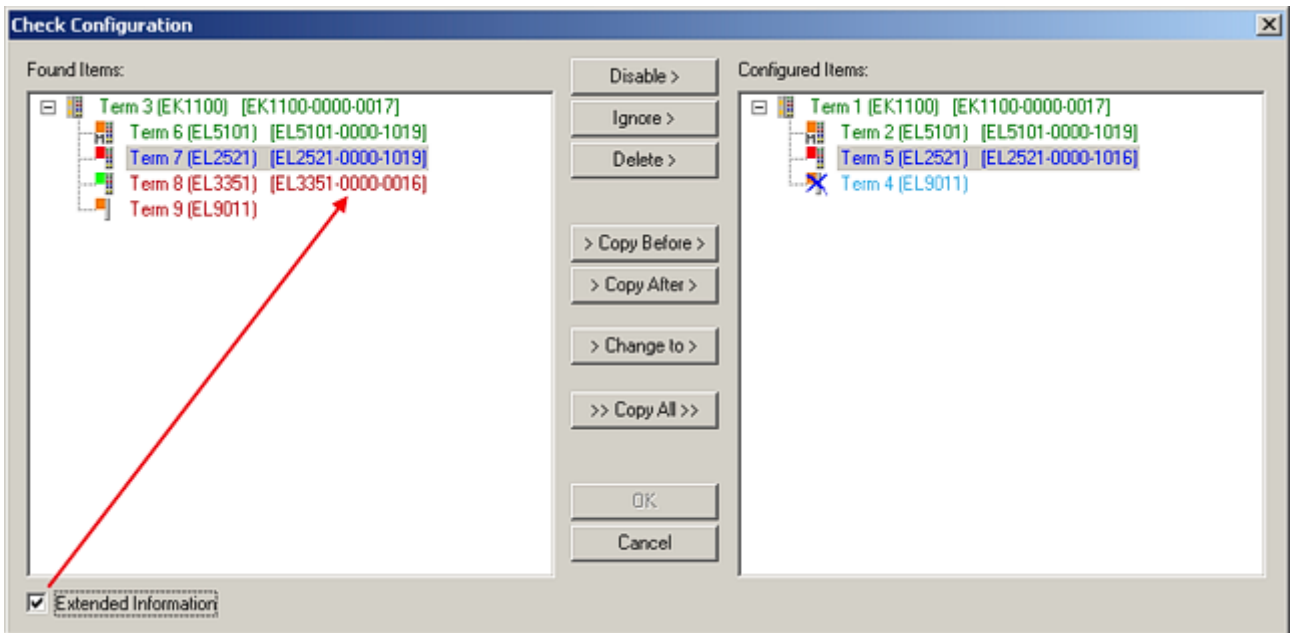
通过这种扫描 (TwinCAT 2.11 或 3.1)，目前只对设备属性中的供应商 (制造商)、设备名称和修订版本进行比较！务必谨慎执行“ChangeTo (更改为)”和“Copy (复制)”操作，认真考虑倍福 IO 兼容性规则 (见前文)。然后，原来配置的设备版本被扫描发现的修订版本所取代；这可能会影响设备支持的过程数据和功能。

如果对现有配置进行扫描，实际的 I/O 环境可能与配置完全一致，也可能有所不同。这样就可以比较两个配置了。



附图 116：相同配置 (左：TwinCAT 2；右：TwinCAT 3)

如果检测到有改动，差异会显示在更正对话框中，从而让用户就可以根据需要修改配置。



附图 117: 更正对话框

建议勾选“Extended Information”复选框，以显示修订版本的差异。

颜色	说明
绿色	此 EtherCAT 从站与另一侧的条目相匹配。类型和修订版本均匹配。
蓝色	此 EtherCAT 从站在另一侧也存在，但其版本不同。其他修订版本可能具有过程数据和其他/附加功能的其他默认数值。 如果找到的修订版本高于配置的修订版本，只要考虑到兼容性问题，就可以使用该从站。 如果找到的修订版低于配置的修订版，很可能无法使用从站。找到的设备可能并不支持主站基于较高修订版本所期望的所有功能。
淡蓝色	此 EtherCAT 从站被忽略（“忽略”按钮）
红色	<ul style="list-style-type: none"> <li>此 EtherCAT 从站在另一侧不存在。</li> <li>存在但版本不同，且属性也与指定版本不同。 兼容性原则也适用于此处：如果找到的版本高于配置的版本，只要考虑到兼容性问题，就可以使用，因为后继设备应该支持前代设备的功能。 如果找到的修订版低于配置的修订版，很可能无法使用从站。找到的设备可能并不支持主站基于较高修订版本所期望的所有功能。</li> </ul>

### ● 修订版本的设备选择 - 兼容性

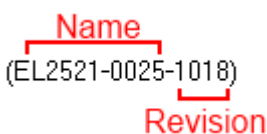
**i** ESI描述还定义了过程图像、主站和从站/设备之间的通信类型以及设备功能（如果适用）。物理设备（固件，如果适用）必须支持主站的通信查询/设置。这是向后兼容的，也就是说，如果EtherCAT主站将其视为较早版本，那么应支持较新设备（较高版本）。对于Beckhoff的EtherCAT端子/端子盒/EJ模块，应符合以下兼容性规则。

**系统中的设备版本 >= 配置中的设备版本**

这也使得后续更换设备时无需改变配置（驱动器可能存在不同规格）。

### 示例

如果在配置中指定了EL2521-0025-1018，则在实践中可以使用EL2521-0025-1018或更高版本（-1019，-1020）。

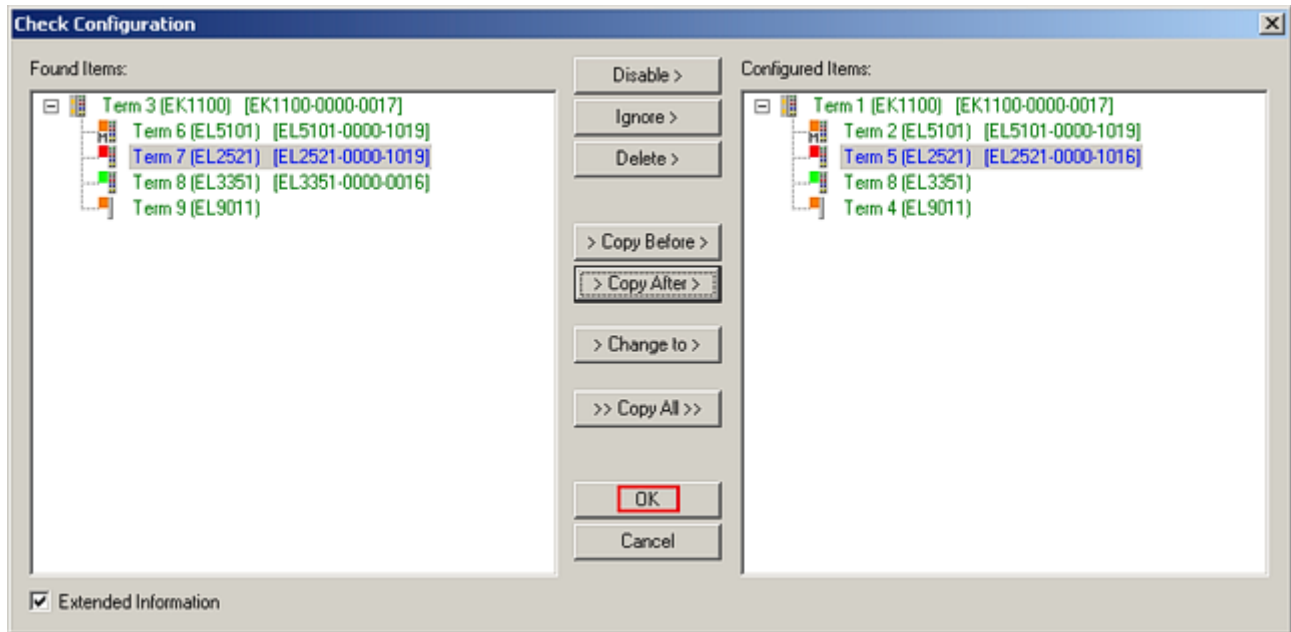


附图 118: 终端的名称/修订版本



如果TwinCAT系统中存在当前ESI描述，则选择对话框中提供的最新修订版本与Beckhoff的生产状态相符。如果在实际应用中使用了当前Beckhoff设备，建议在创建新配置时使用最近的设备版本。在应用中使用库存的较早设备时，方才应使用较早的修订版本。

在这种情况下，设备的过程图像显示在配置树中，并可以进行如下参数化：与任务的链接、CoE/DC设置、插件定义、启动设置…

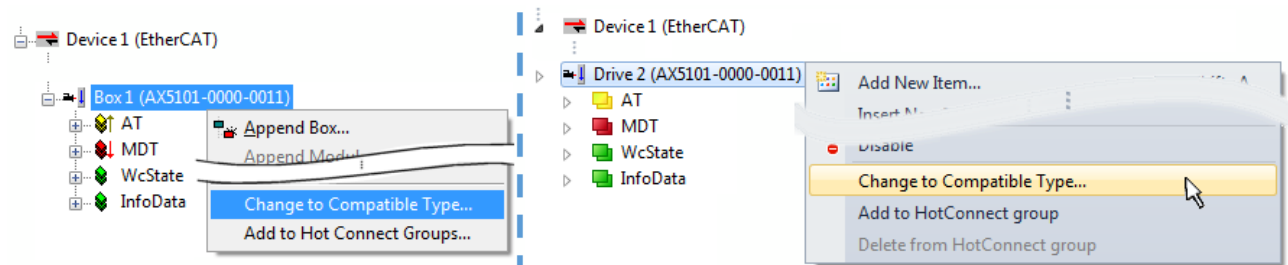


附图 119: 更正对话框，有修改项

一旦所有的修改被保存或接受，点击“确定”将它们传输到实际的 \*.tsm 配置。

### 更改为兼容类型 (Change to Compatible Type)

TwinCAT 提供一个功能 *Change to Compatible Type...* 用于切换到另一个设备版本，同时保留任务中的链接。



附图 120: 对话框“Change to Compatible Type...” (左: TwinCAT 2; 右: TwinCAT 3)

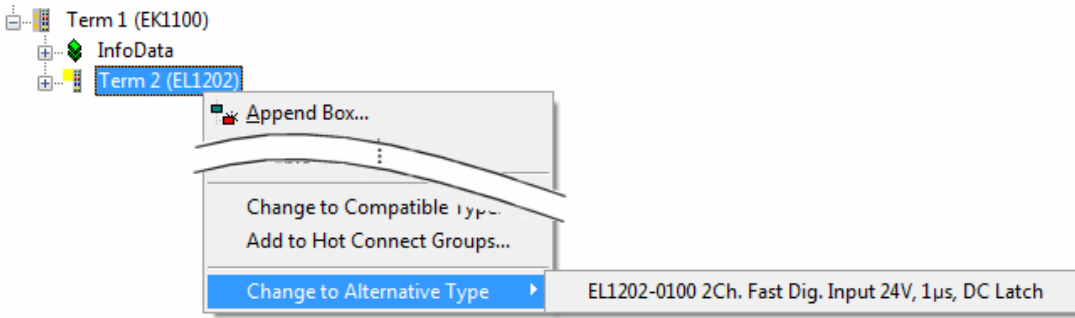
TwinCAT 对 EtherCAT 设备的 ESI 中下列元素进行了比较，并假定它们是相同的，以决定一个设备是否被表示为“兼容”：

- 物理层 (例如RJ45、Ebus...)
- FMMU (允许实际数量比配置的多)
- SyncManager (SM, 允许实际数量比配置的多)
- EoE (属性 MAC, IP)
- CoE (属性 SdoInfo, PdoAssign, PdoConfig, PdoUpload, CompleteAccess)
- FoE
- PDO (过程数据: Sequence, SyncUnit SU, SyncManager SM, EntryCount, Entry.Datatype)

这个功能最好是在 AX5000 设备上使用。

### 更改为替代类型 (Change to Alternative Type)

TwinCAT System Manager 提供用于切换设备的功能: Change to Alternative Type

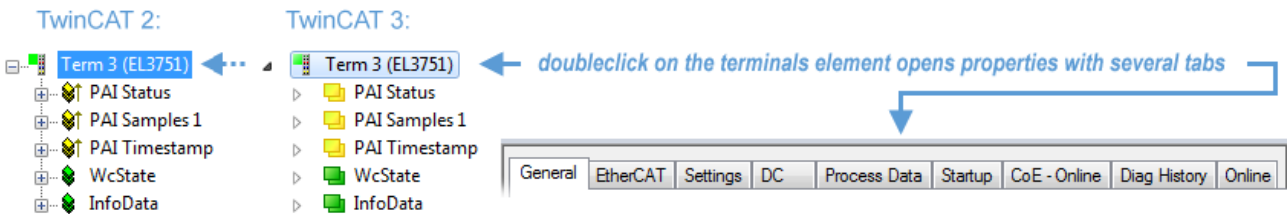


附图 121: TwinCAT 2 对话框 Change to Alternative Type

如果调用 Change to Alternative Type, System Manager 会在本地的设备 ESI (在此例中: EL1202-0000) 中搜索其中包含的兼容设备的详细信息。配置被更改, 且 ESI-EEPROM 也同时被覆盖, 因此这个过程只有在在线状态 (ConfigMode) 下才能执行。

## 6.2.7 EtherCAT 设备的配置

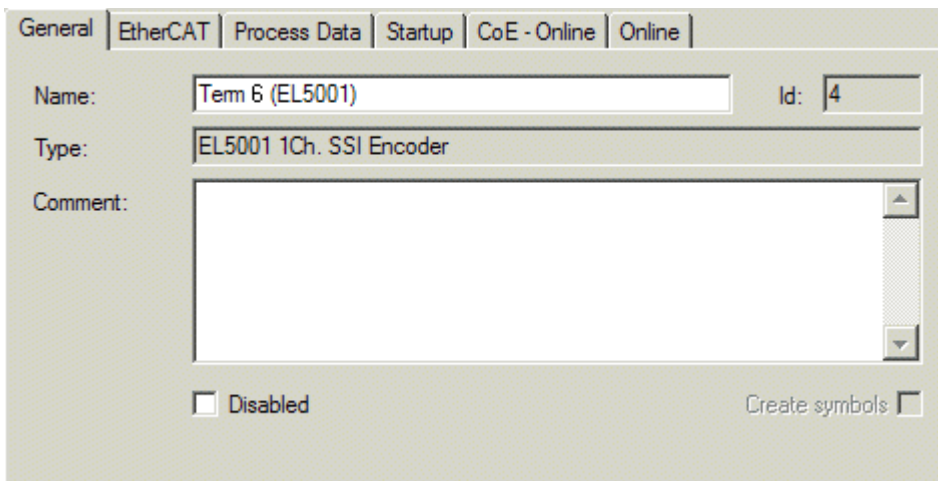
在 TwinCAT 2 System Manager 的左侧窗口或 TwinCAT 3 开发环境的 Solution Explorer (解决方案浏览器) 中, 分别点击树结构中希望配置的端子模块 (在示例中: EL3751 Term 3)。



附图 122: 树形结构的分支, 端子模块 EL3751

在 TwinCAT System Manager (TwinCAT 2) 或开发环境 (TwinCAT 3) 的右侧窗口中, 有各种用于配置端子模块的选项卡, 而具体提供哪些选项卡则取决于从站设备的复杂程度。因此, 如上面的例子所示, 端子模块 EL3751 提供许多设置选项, 也提供相应数量的选项卡。相反, 对于端子模块 EL1004, 就只提供 “General”、“EtherCAT”、“Process Data” 和 “Online” 选项卡。有的端子模块 (例如 EL6695) 通过一个带有自己名称的选项卡提供特殊功能, 本例中的选项卡名称就是 “EL6695”。此外, 还有一些端子模块提供一个特定的 “Settings” 选项卡, 其中包括诸多设置选项 (例如 EL3751)。

#### “General (常规)” 选项卡



附图 123: “General (常规)” 选项卡

Name	EtherCAT 设备的名称
Id	EtherCAT 设备的编号
Type	EtherCAT 设备类型
Comment	注释（例如关于系统的注释）。
Disabled	可以在此停用 EtherCAT 设备。
Create symbols	选中此复选框，才能通过 ADS 访问该 EtherCAT 从站。

#### “EtherCAT”选项卡

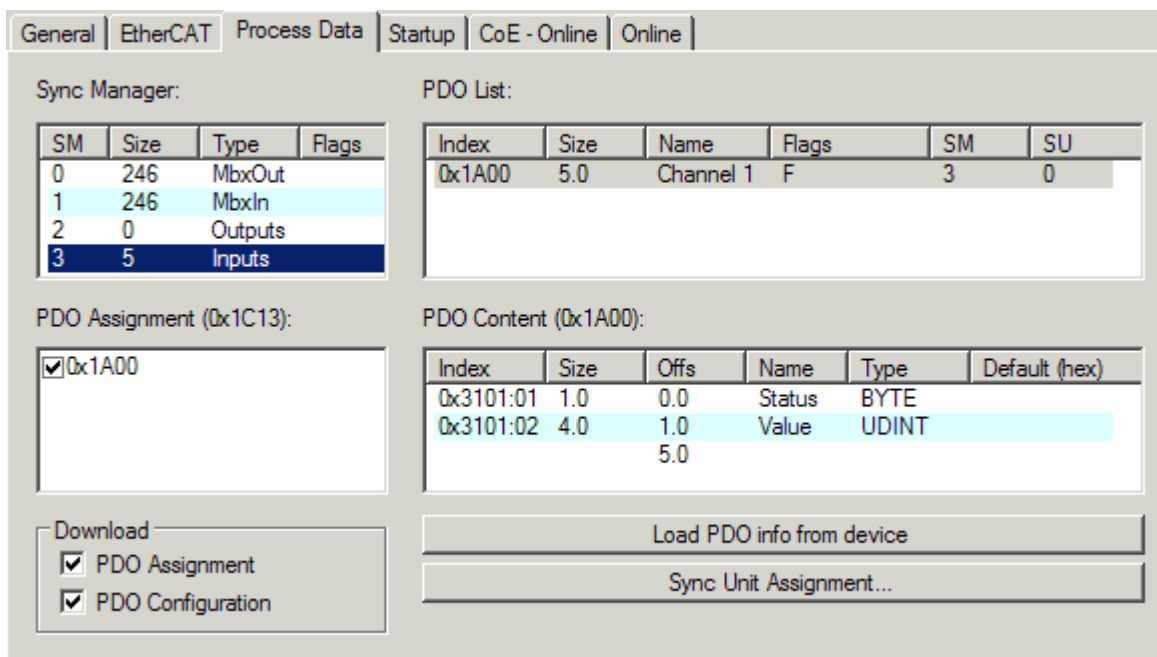
附图 124: “EtherCAT”选项卡

Type	EtherCAT 设备类型
Product/Revision	EtherCAT 设备的产品编号和修订版本号
Auto Inc Addr.	EtherCAT 设备的自动增量寻址功能。自动增量寻址用于通过物理位置对通信环中的每个 EtherCAT 设备进行寻址。在启动阶段，当 EtherCAT 主站为 EtherCAT 设备分配地址时，将使用自动增量寻址。进行自动增量寻址时，通信链路上的第一个 EtherCAT 从站的地址为 $0000_{\text{hex}}$ 。每增加一个从站，地址就减 1 ( $FFFF_{\text{hex}}$ 、 $FFFE_{\text{hex}}$ ... )。
EtherCAT Addr.	一个 EtherCAT 从站的固定地址。该地址由 EtherCAT 主站在启动阶段分配。勾选输入字段左边的复选框，以修改默认值。
Previous Port	该设备连接的 EtherCAT 设备的名称和端口。如果可以在不改变通信环中 EtherCAT 设备顺序的情况下将该设备与另一个设备进行连接，则该组合字段被激活，可以选择该设备所连接的 EtherCAT 设备。
Advanced Settings	点击该按钮打开高级设置对话框。

标签底部的链接指向该 EtherCAT 设备对应的产品主页。

#### “Process Data (过程数据)”选项卡

用于过程数据配置。EtherCAT从站的输入和输出数据表示为CANopen过程数据对象（ProcessDataObjects, PDO）。如果EtherCAT从站支持该功能，用户可以通过 PDO 分配选择一个PDO，并通过该对话框修改各个PDO的内容。



附图 125: “Process Data (过程数据)” 选项卡

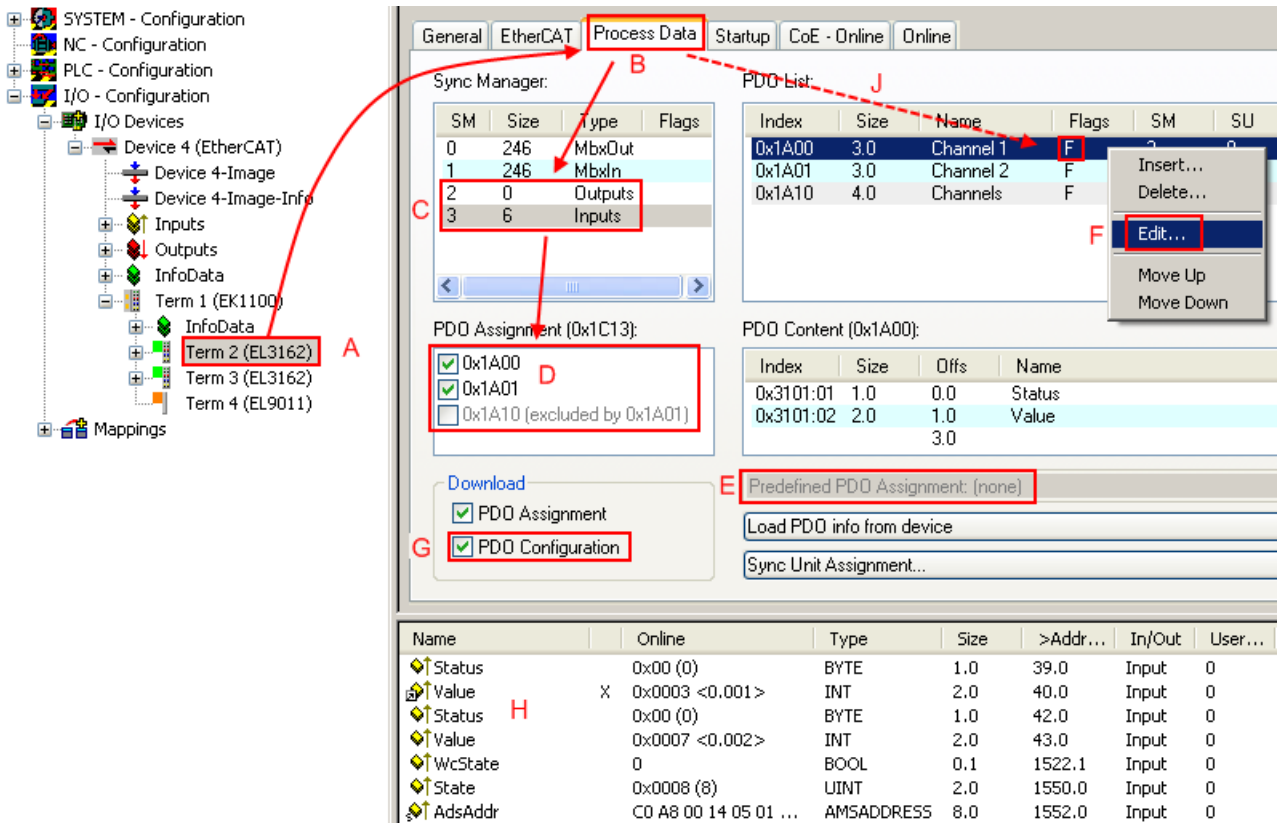
EtherCAT从站在每个周期内传输的过程数据 (PDO) 是应用程序期望周期性更新的用户数据, 或者是被发送到从站的用户数据。为此, EtherCAT 主站 (Beckhoff TwinCAT) 在启动阶段对每个EtherCAT从站进行了参数设置, 以定义其希望传输到该从站或从该从站传输的过程数据 (位/字节大小、数据源位置、传输类型)。如果配置错误, 将会使从站启动失败。

对于Beckhoff EtherCAT EL、ES、EM、EJ和EP从站, 一般情况下适用以下规定:

- 设备支持的输入/输出过程数据由制造商在 ESI/XML 描述中定义。TwinCAT EtherCAT 主站使用 ESI 描述来正确配置从站。
- 过程数据可以在System Manager (系统管理器) 中修改。参见设备文件。  
修改示例包括: 屏蔽一个通道、显示额外的循环信息、16位显示代替8位数据大小等等。
- 在所谓的“智能”EtherCAT 设备中, 过程数据信息也被存储在 CoE 目录中。CoE 目录中任何导致不同 PDO 设置的更改都会使从站启动失败。不建议修改模块出厂配置的过程数据, 因为设备固件 (如有) 与这些 PDO 组合是配套的。

如果设备文件允许修改过程数据, 请按以下步骤操作 (见图配置过程数据)。

- A: 选择需要配置的设备
- B: 在“Process Data”标签中选择Sync Manager 同步管理器下的输入或输出 (C)
- D: 可以选择或取消选择 PDO
- H: 新的过程数据在System Manager (系统管理器) 中作为可链接的变量可见  
一旦配置被激活且TwinCAT被重新启动 (或EtherCAT主站被重新启动), 新的过程数据就会激活。
- E: 如果从站支持, 可以通过选择一个所谓的 PDO 记录 (“predefined PDO settings”) 来同时修改输入和输出的 PDO 。



附图 126: 配置过程数据

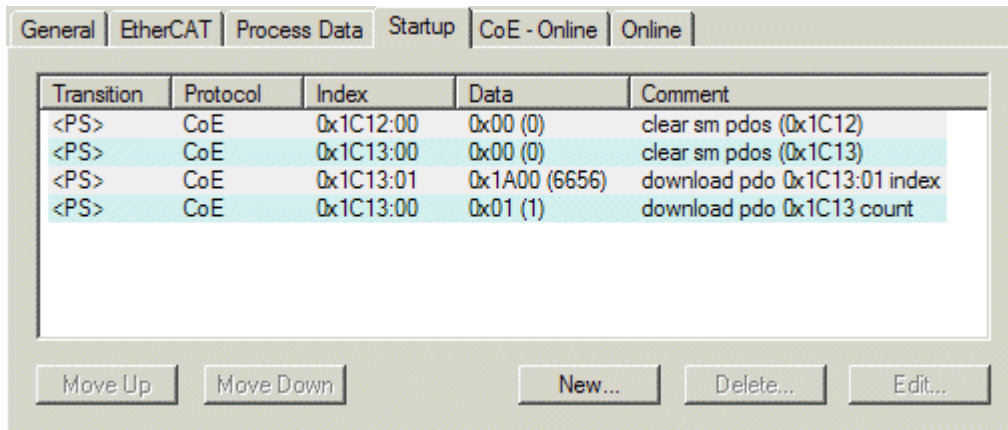
### ● 手动修改过程数据

根据ESI的描述，一个 PDO 可以在PDO概述中以标志“F”标为“固定”（图配置过程数据，J）。即使 TwinCAT 提供相关对话框（“Edit”），也不能改变此类 PDO 的配置。特别是，CoE内容不能作为循环过程数据显示。这通常也适用于设备支持下载 PDO 配置的情况，“G”。在配置不正确的情况下，EtherCAT从站通常会拒绝启动，并改变为OP状态。System Manager（系统管理器）显示“invalid SM cfg”记录器信息：这个错误信息（“invalid SM IN cfg”或“invalid SM OUT cfg”）也提示了启动失败的原因。

此外，还可在本节末尾查看详细说明。

### “Startup（启动）”选项卡

如果EtherCAT从站配有邮箱并支持CANopen over EtherCAT (CoE) 或 Servo drive over EtherCAT协议，则显示Startup（启动）选项卡。这个选项卡显示了在启动期间哪些下载请求被发送到邮箱。另外，也可以在列表显示中添加新的邮箱请求。下载请求会按照它们在列表中显示的不同顺序发送到从站。



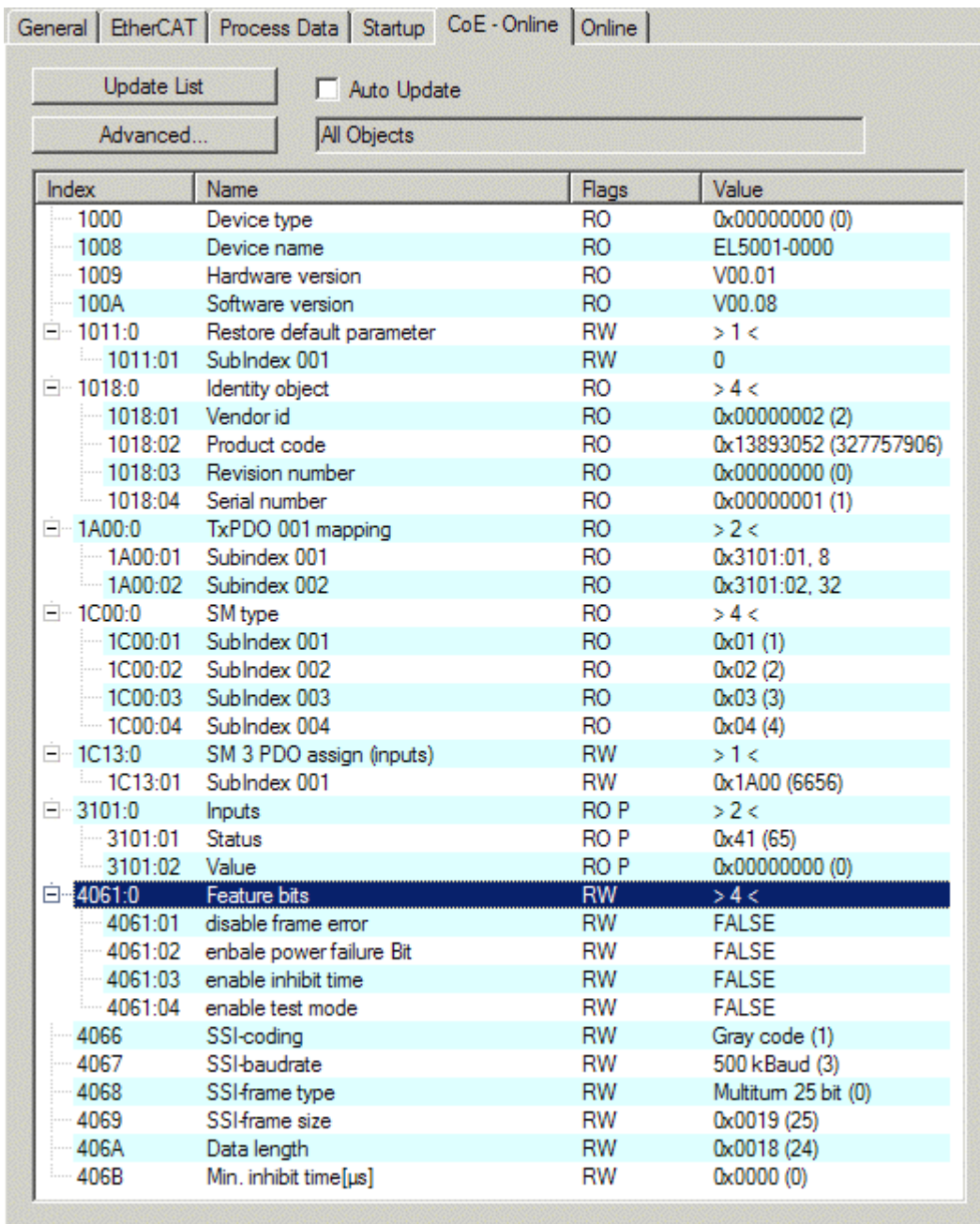
附图 127: “Startup（启动）”选项卡

列	Description
Transition	发送请求的过渡期。这可以是 <ul style="list-style-type: none"> <li>• 从Pre-OP到Safe-OP (PS) 的过渡, 或</li> <li>• 从Safe-OP到运行 (SO) 的过渡。</li> </ul> 如果过渡用 “<>” 括起来 (如<PS>), 则这种邮箱请求是固定的, 用户不能修改或删除。
Protocol	邮箱协议类型
Index	对象的索引
Data	该对象要下载的数据。
Comment	将被发送到邮箱的请求的描述

<b>Move Up</b>	该按钮可将所选请求在列表中向上移动一个位置。
<b>Move Down</b>	该按钮可将所选请求在列表中向下移动一个位置。
<b>New</b>	该按钮可添加一个新的邮箱下载请求, 将在启动时发送。
<b>Delete</b>	该按钮可以删除选定的条目。
<b>Edit</b>	该按钮可编辑当前的邮箱请求内容。

### “CoE - Online” 选项卡

如果EtherCAT从站支持*CANopen over EtherCAT* (CoE) 协议, 则会显示额外的*CoE - Online*选项卡。该对话框列出了从站对象列表的内容 (SDO上传), 并让用户能够从这个列表中修改对象的内容。关于各个EtherCAT设备对象的详细信息, 可参见设备特定的对象描述。



附图 128: “CoE - Online” 选项卡

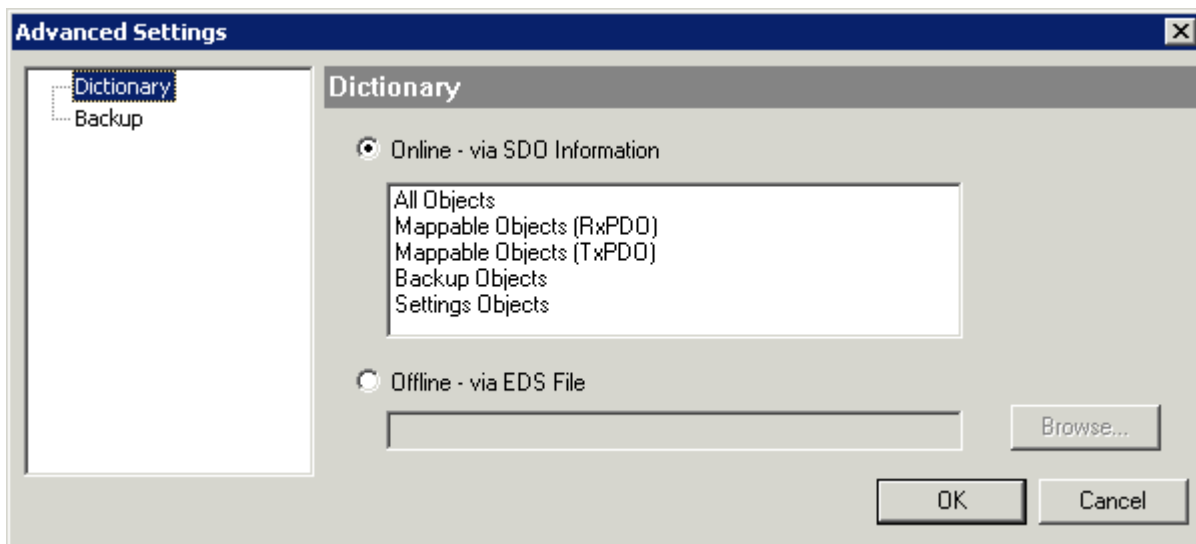
对象列表显示

列	Description
Index	对象的索引和子索引
Name	对象的名称
Flags	RW 该对象可以被读取，且数据可被写入对象（读/写）。
	RO 该对象可以被读取，但不能向该对象写入数据（只读）。
	P 附加P将对象标识为过程数据对象。
Value	对象数值

**Update List** *Update List* 按钮可更新显示列表中的所有对象。

**Auto Update** 如果选择了这个复选框，对象的内容会自动更新。

**Advanced** *Advanced* 按钮可打开 *Advanced Settings* 对话框。在这里，你可以指定哪些对象会显示在列表中。

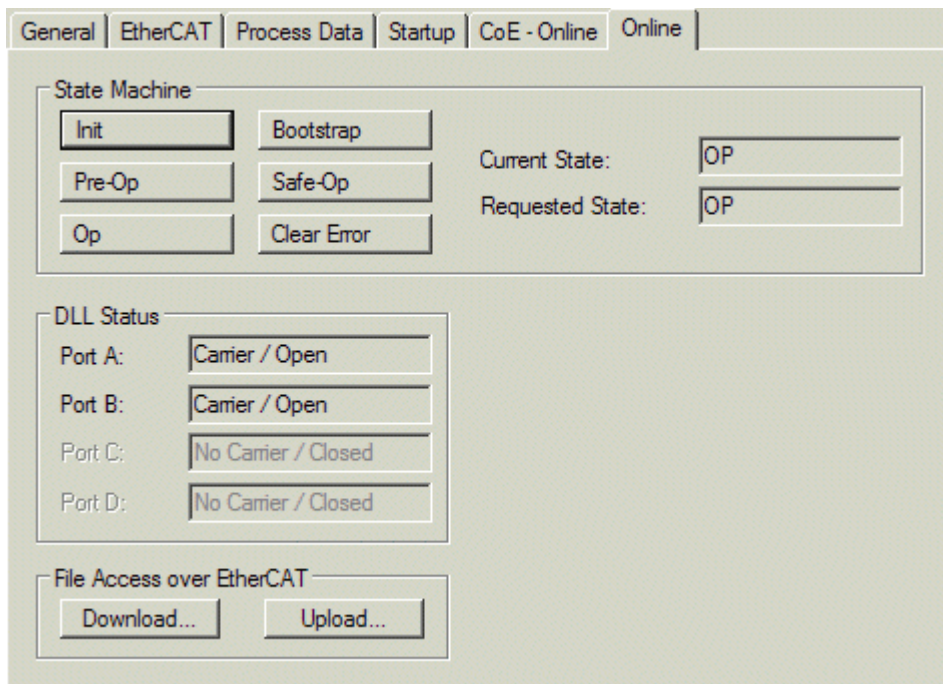


附图 129: “Advanced settings (高级设置)” 对话框

**Online - via SDO Information** 如果选择了这个选项按钮，就会通过SDO信息从从站上传包含在从站对象列表中的对象列表。下面的列表可以用来指定哪些对象类型要被上传。

**Offline - via EDS File** 如果选择了这个选项按钮，将从用户提供的EDS文件中读取对象列表中包含的对象列表。

### “Online (在线)” 选项卡



附图 130: “Online (在线)” 选项卡



State Machine 状态机

- Init** 点击该按钮将 EtherCAT 设备设置为 *Init* 状态。
- Pre-Op** 点击该按钮将 EtherCAT 设备设置为 *Pre-OP* 状态。
- Op** 点击该按钮将 EtherCAT 设备设置为 *OP* 状态。
- Bootstrap** 点击该按钮将 EtherCAT 设备设置为 *Bootstrap* 状态。
- Safe-Op** 点击该按钮将 EtherCAT 设备设置为 *Safe-OP* 状态。
- Clear Error** 点击该按钮删除故障显示。如果 EtherCAT 从站在状态改变期间出现故障，将会设置错误标志。  
 示例：EtherCAT 从站处于 PREOP 状态（预运行）。主站现在请求 SAFEOP 状态（安全运行）。如果从站在状态改变期间出现故障，将设置错误标志。目前状态显示为 ERR PREOP。在按下 *Clear Error* 按钮后，错误标志将被清除，且当前状态再次显示为 PREOP。
- Current State** 指示 EtherCAT 设备的当前状态。
- Requested State** 指示 EtherCAT 设备请求的状态。

DLL Status

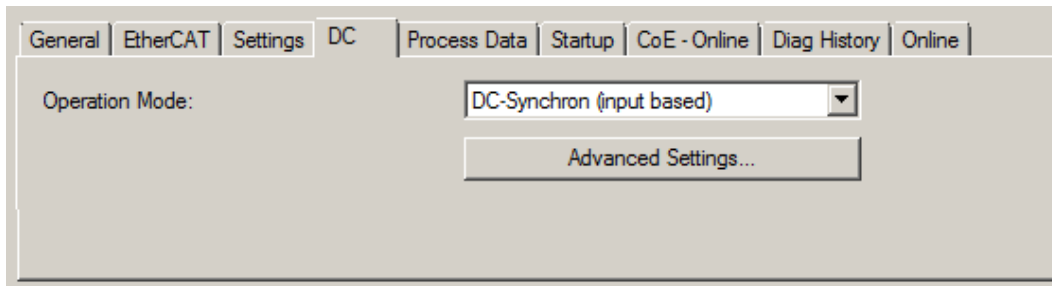
指示 EtherCAT 从站各个端口的 DLL 状态（数据链路层状态）。DLL 状态分为以下四种：

Status	Description
No Carrier / Open	端口没有通讯信号，但端口处于打开状态。
No Carrier / Closed	端口没有通讯信号，且端口处于关闭状态。
Carrier / Open	端口有通讯信号，且端口处于打开状态。
Carrier / Closed	端口有通讯信号，但端口处于关闭状态。

通过 EtherCAT 进行文件访问

- Download** 通过这个按钮，文件可以被写入 EtherCAT 设备中。
- Upload** 通过这个按钮，可以从 EtherCAT 设备中读取一个文件。

“DC” 选项卡（分布时钟）



附图 131: “DC” 选项卡（分布时钟）

- Operation Mode** 选项（可选）：
  - FreeRun
  - SM-Synchron
  - DC-Synchron (Input based)
  - DC-Synchron
- Advanced Settings...** 用于重新调整 TwinCAT 时钟的高级设置，这是 EtherCAT 从站实时特性的决定性因素

关于分布时钟的详细信息，请参见 <http://infosys.beckhoff.com>：

Fieldbus Components → EtherCAT Terminals → EtherCAT System documentation → EtherCAT basics → Distributed Clocks

还请参阅有关此

📖 Process Data (过程数据) 选项卡的详细描述 [▶ 122]

### 6.2.7.1 Process Data (过程数据) 选项卡的详细描述

#### Sync Manager (同步管理器)

列出了Sync Manager 同步管理器 (SM) 的配置。

如果EtherCAT设备有一个邮箱, SM0用于邮箱输出 (MbxOut), SM1用于邮箱输入 (MbxIn)。

SM2用于输出过程数据 (输出), SM3 (输入) 用于输入过程数据。

如果选择了一个输入, 相应的 PDO 分配会显示在下面的 PDO 分配列表中。

#### PDO 分配

所选Sync Manager 同步管理器的 PDO 分配。所有为该Sync Manager 同步管理器类型定义的 PDO 都在这里列出:

- 如果在Sync Manager 同步管理器列表中选择了输出Sync Manager 同步管理器 (输出), 则显示所有的 RxPDO。
- 如果在Sync Manager 同步管理器列表中选择了输入Sync Manager 同步管理器 (输入), 则显示所有的 TxPDO。

所选条目是参与过程数据传输的 PDO。在System Manager (系统管理器) 的树状图中, 这些 PDO 被显示为 EtherCAT 设备的变量。变量名称与 PDO 的 *Name* 参数相同, 如 PDO 列表中所示。如果 PDO 分配列表中的一个条目被停用 (未被选中且呈灰色), 这表明该输入被排除在PDO分配之外。为了能够选择一个灰色的 PDO, 必须先取消选择当前选定的PDO。

#### ● 激活 PDO 分配



✓ 如果改变 PDO 分配以激活新的PDO分配,

a) EtherCAT从站必须运行一次PS状态转换周期 (从Pre-OP到Safe-OP) (见[Online\(在线\) 选项卡 \[▶ 120\]](#)),

b) 且System Manager (系统管理器) 必须重新加载EtherCAT从站

(  TwinCAT 2按钮或  TwinCAT 3按钮)

#### PDO list (PDO 列表)

该 EtherCAT 设备支持的所有 PDO 列表。所选 PDO 的内容显示在 *PDO Content* 列表中。PDO 配置可通过双击条目进行修改。

列	Description
Index	PDO 索引。
Size	PDO 大小 (单位: 字节)。
Name	PDO名称。 如果这个 PDO 被分配给一个Sync Manager 同步管理器, 它将作为从站的一个变量出现, 并以这个参数作为名称。
Flags	F 固定内容: 该 PDO 内容固定, System Manager (系统管理器) 无法更改。 M 必须填写的 PDO 内容。该 PDO 为必填项, 因此必须分配给一个Sync Manager 同步管理器! 因此, 该 PDO 不能从 <i>PDO Assignment</i> 列表中删除。
SM	被分配 PDO 的 Sync Manager 同步管理器。如果该条目为空, 则该 PDO 不参与过程数据通信。
SU	被分配 PDO 的同步单元。

#### PDO Content (PDO 内容)

显示当前选中的 PDO 内容。如果 PDO 的标志F (固定内容) 没有被设置, 表示其内容可以被修改。

## Download (下载)

对于具备 Mailbox 邮箱功能的智能设备，PDO Configuration (配置) 和 PDO Assignment (分配) 都可以下载到设备上。这是一个可选的功能，并非所有 EtherCAT 从站都支持。

## PDO 分配

如果选择这个复选框，在 PDO 分配列表中配置的 PDO 分配会在启动时下载到设备。发送给设备的请求命令可以在 [Startup \[▶ 117\]](#) 选项卡中查看。

## PDO 配置

如果选择了该复选框，各 PDO 的配置（如 PDO 列表和 PDO 内容显示中所示）将被下载到 EtherCAT 从站。

## 6.2.8 导入/导出 EtherCAT 设备为 SCI 和 XTI 文件

### SCI 和 XTI 导出/导入 - 处理用户定义/修改的 EtherCAT 从站

#### 6.2.8.1 基本原则

EtherCAT 从站一般通过以下要素进行参数化：

- 周期性过程数据 (PDO)
- 同步特性 (分布时钟 DC、FreeRun、SM-Synchron)
- CoE 参数 (非周期性对象字典)

注意：根据从站的类型，这三个要素可能不会全部出现。

为了更好地理解导出/导入功能，暂时只考虑 IO 配置的常规流程：

- 用户/编程人员在 TwinCAT 系统环境中进行 IO 配置。这涉及到所有输入/输出设备，例如连接到所用现场总线的驱动器。  
注意：在以下章节中，仅说明 TwinCAT 系统环境中的 EtherCAT 配置。
- 用户手动将设备添加到配置中，或通过在线系统执行扫描。
- 然后配置 IO 系统。
- 插入一个从站时，系统配置中出现从站供应商提供的默认配置，包括默认 PDO、默认同步方法和 ESI (XML 设备描述文件) 中定义的 CoE StartUp 参数。
- 如果有必要，可以根据相应的设备文档修改从站的配置元素，例如 PDO 配置或同步方法。

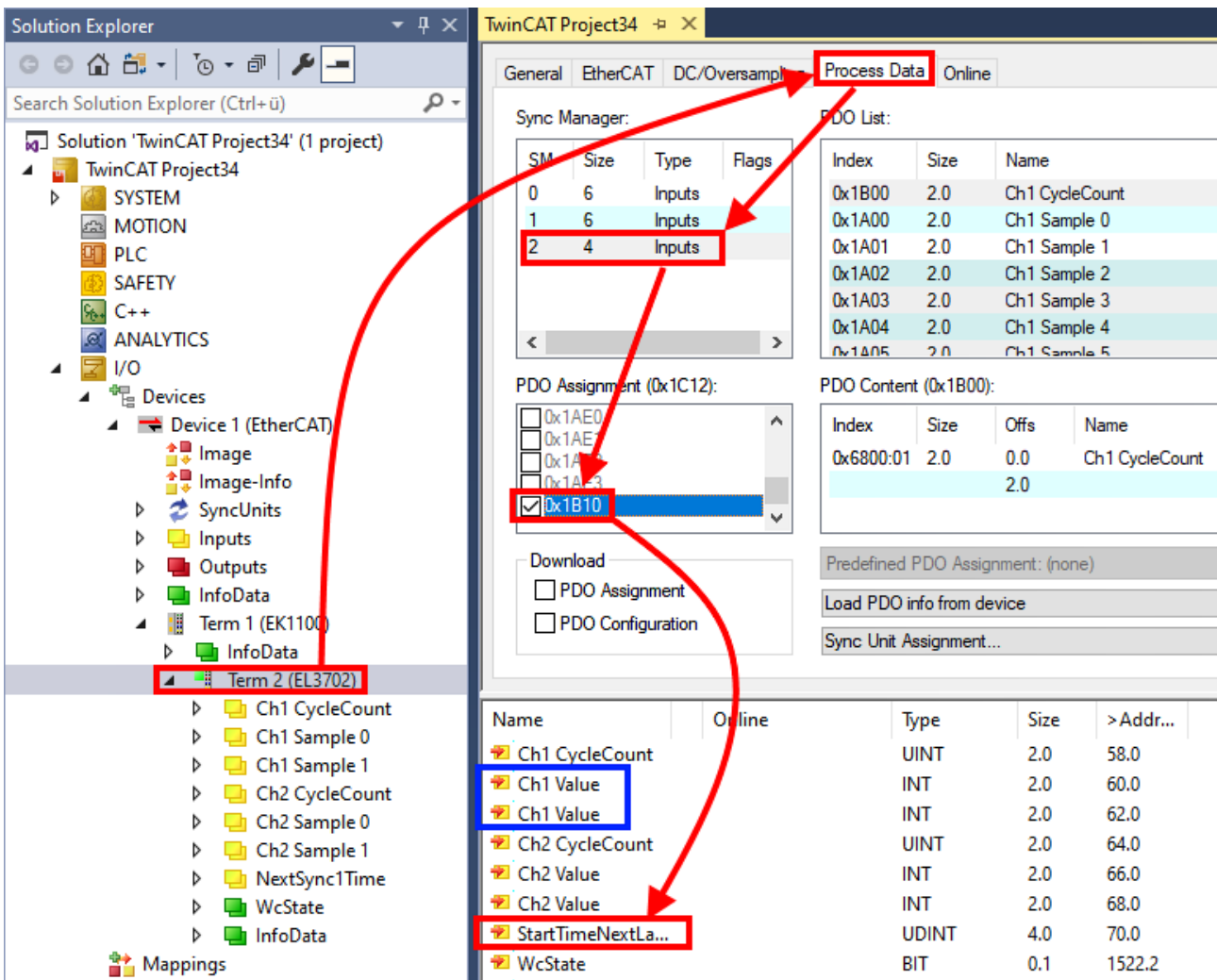
有时需要在其他项目中重复使用参数完全相同的从站，而不必在从站默认配置的基础上进行重复修改。为了实现这个功能，需要执行如下步骤：

- 从项目中导出从站配置，
- 以文件形式存储和传输，
- 导入到另一个 EtherCAT 项目中。

TwinCAT 为此提供了两种方法：

- 用于 TwinCAT 环境中：Export/Import as **x**ti 文件 或
- 外部使用，即用于非 TwinCAT 的 EtherCAT 主站：Export/Import as **s**ci 文件。

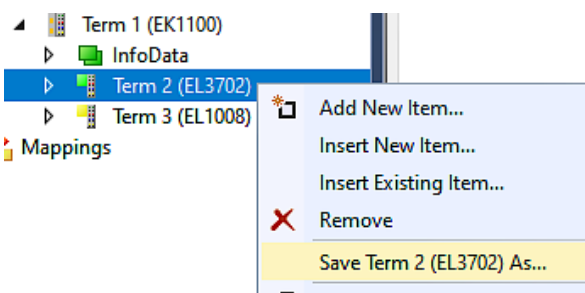
举例说明：一个标准的 EL3702 端子模块被设置为 2 倍超采样 (蓝色)，并添加了选项 PDO “StartTimeNextLatch” (红色)：



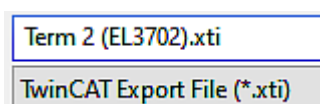
下面演示上述两种导出和导入修改过配置的端子模块的方法。

### 6.2.8.2 导出/导入 xti 文件的步骤（用于 TwinCAT 环境）

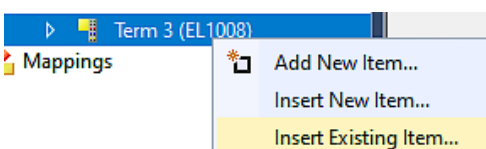
每个 IO 设备都可以单独导出/保存：



xti 文件可以存储：



并通过“Insert Existing item（插入现有项目）”在另一个 TwinCAT 系统中再次导入：



### 6.2.8.3 导出/导入 sci 文件的步骤（用于 TwinCAT 和第三方EtherCAT 主站）

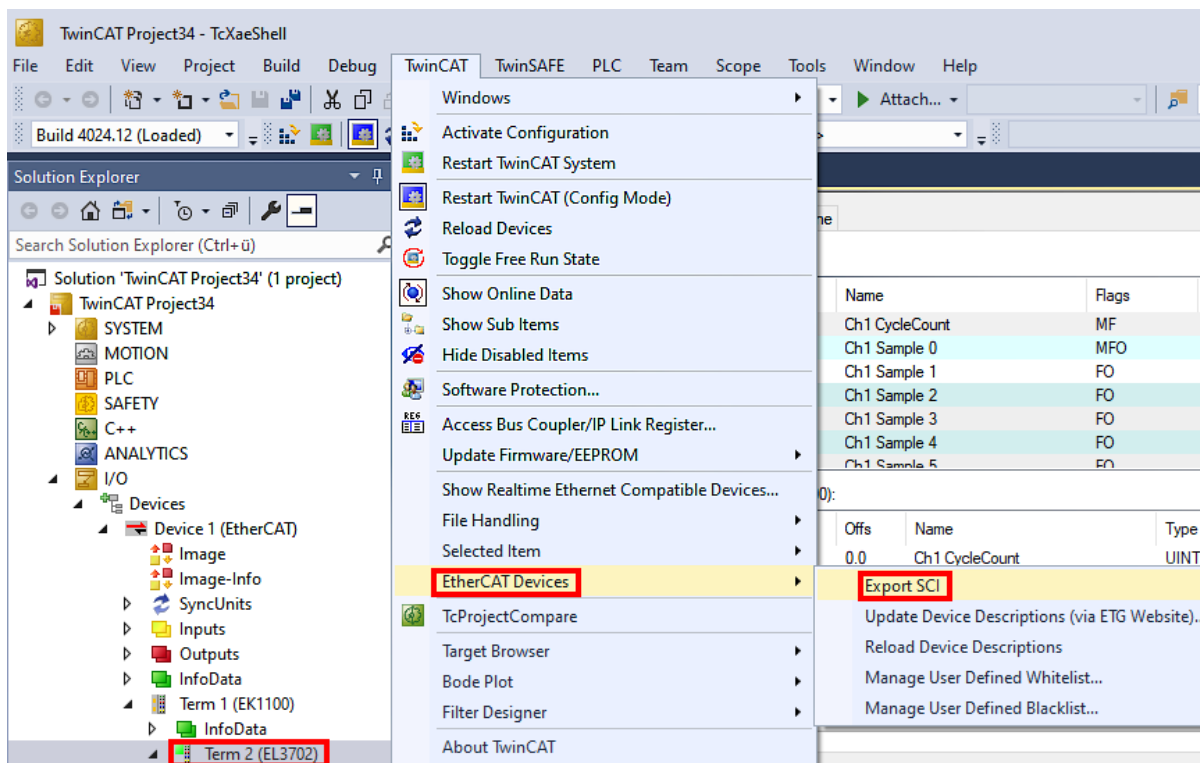
关于适用性的说明（2021/01）

TwinCAT 3.1 build 4024.14 及以上版本才支持 SCI 方法。

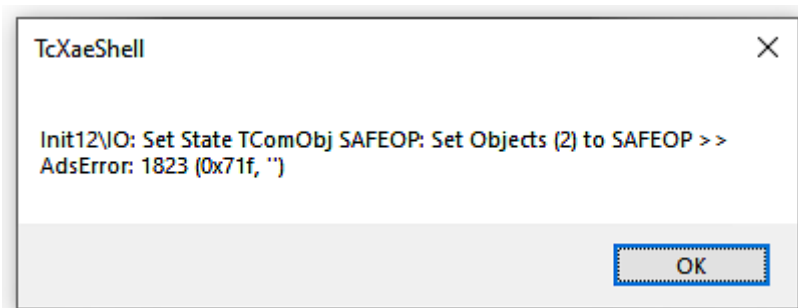
SCI 文件（Slave Configuration Information）基于 ESI 文件（EtherCAT Slave Information）的设置选项，描述了 EtherCAT 从站（端子模块、端子盒、驱动器...）具体完整的配置。也就是说，它包括 PDO、CoE 和同步特性。

#### Export:

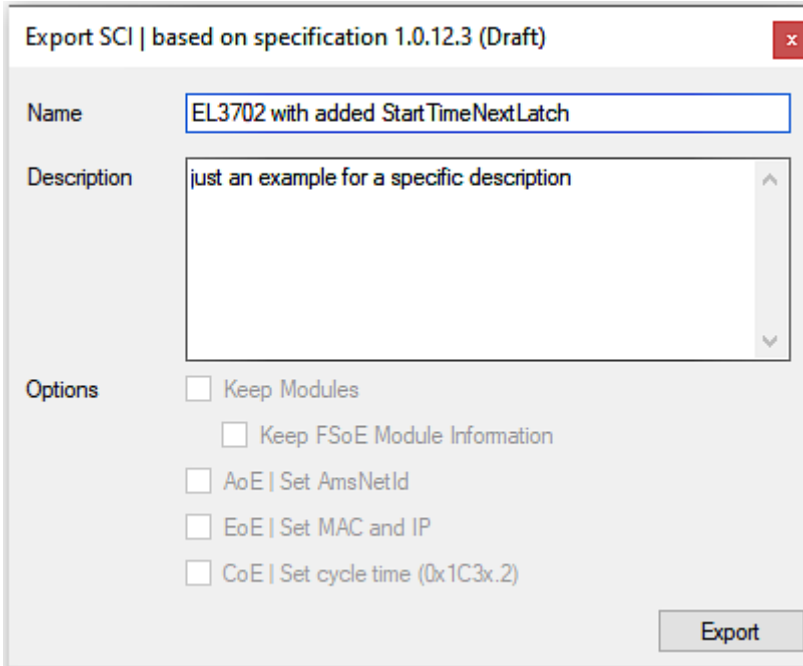
- 通过菜单选择单个设备（也可进行多选）：  
TwinCAT → EtherCAT Devices → Export SCI.



- 如果 TwinCAT 处于离线状态（即没有连接到一个实际运行的控制器），可能会出现一个警告信息，因为在执行该功能后，系统会尝试重新加载 EtherCAT 网络。但是，这并不会影响到结果，可以通过点击 OK 来确认：



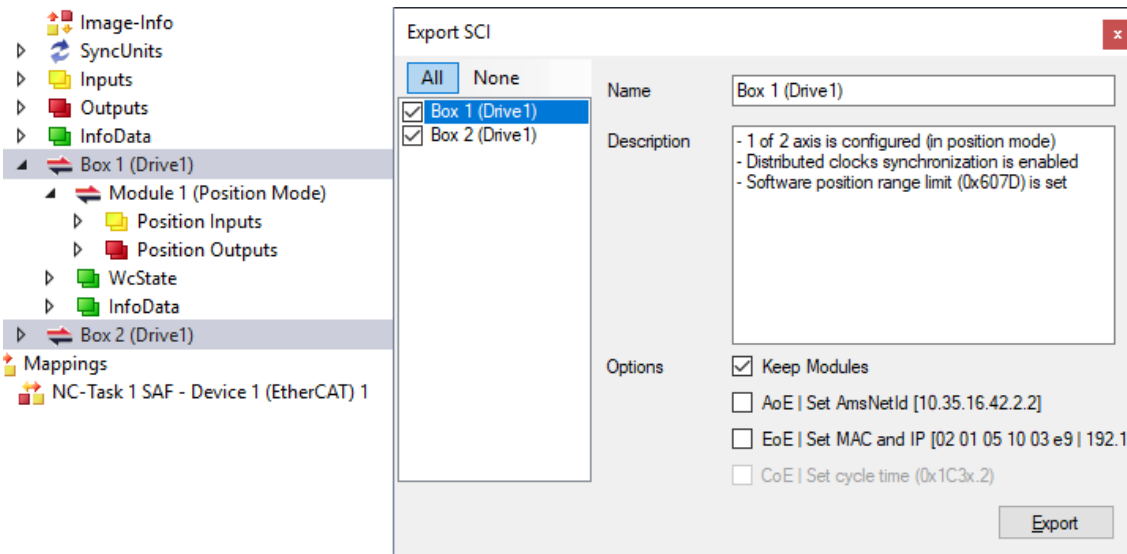
- 还可以提供描述信息:



- 对话框的说明:

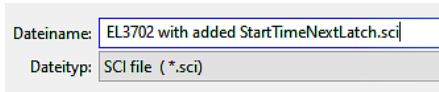
Name	SCI 的名称，由用户指定。	
描述	实际应用的从站配置描述，由用户指定。	
Options	Keep modules (保留模块)	如果一个从站支持 modules/slots (模块/插槽)，用户可以决定是否导出这些模块/插槽，或者在导出时将模块和设备数据合并。
	AoE   Set AmsNetId	配置的 AmsNetId 被导出。通常情况下，这取决于网络，不能总是事先确定。
	EoE   Set MAC and IP	配置的虚拟 MAC 和 IP 地址存储在 SCI 中。通常情况下，这些都取决于网络，不能总是事先确定。
	CoE   Set cycle time (0x1C3x.2)	配置的周期时间被导出。通常情况下，这取决于网络，不能总是事先确定。
ESI	参考原始 ESI 文件。	
Export	保存 SCI 文件。	

- 系统提供下述列表视图用于多重选择 (*Export multiple SCI files* 导出多个 SCI 文件):

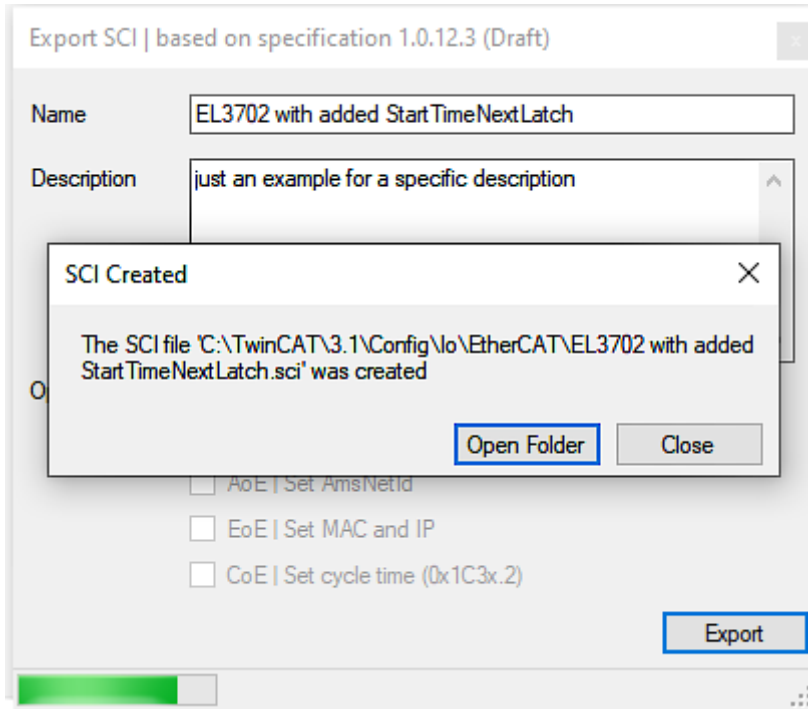


- 选择要导出的从站:
  - All: 所有从站都被选中进行导出。
  - None: 所有从站都被取消选择。

- sci 文件可以保存在本地:



- 执行导出:



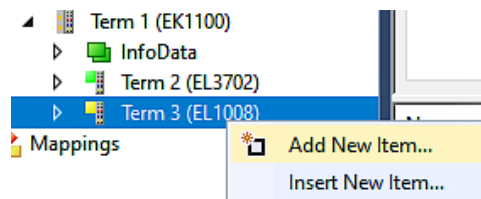
**Import**

- 一个 sci 描述可以像任何普通的倍福设备描述一样，手动插入到 TwinCAT 配置中。
- sci 文件必须位于 TwinCAT ESI 路径中，通常在：  
C:\TwinCAT\3.1\Config\Io\EtherCAT

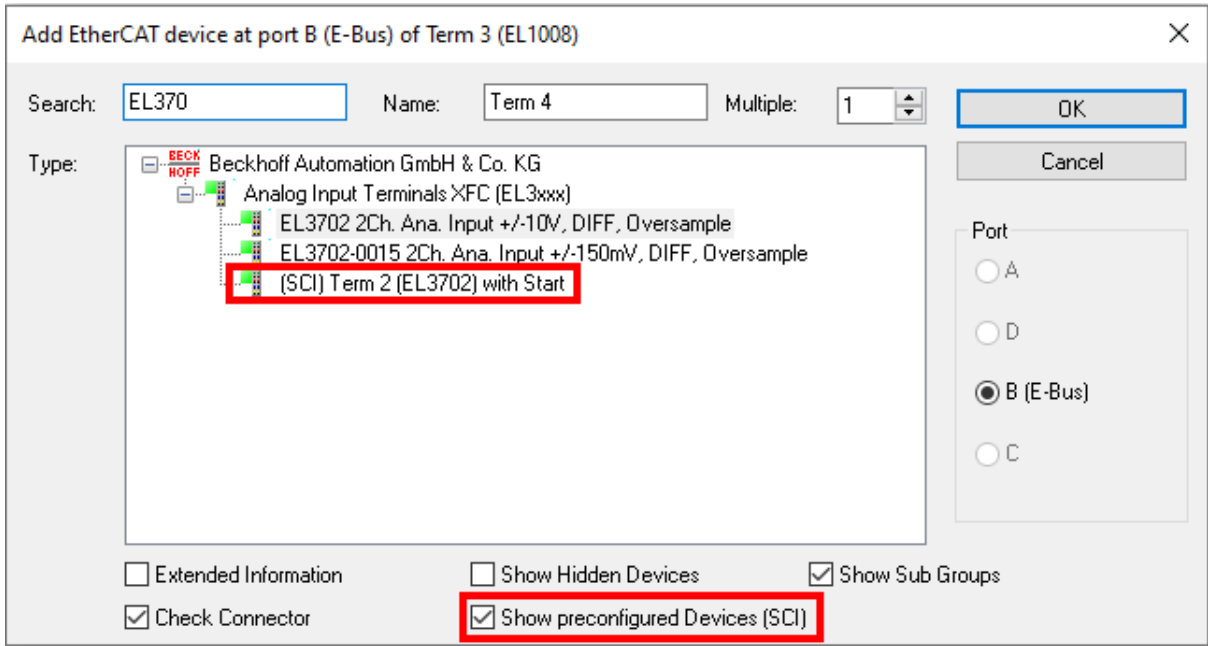
EL3702 with added StartTimeNextLatch.sci	11.01.2021 13:29	SCI-Datei	6 KB
--	------------------	-----------	------

下

- 打开选择对话框:

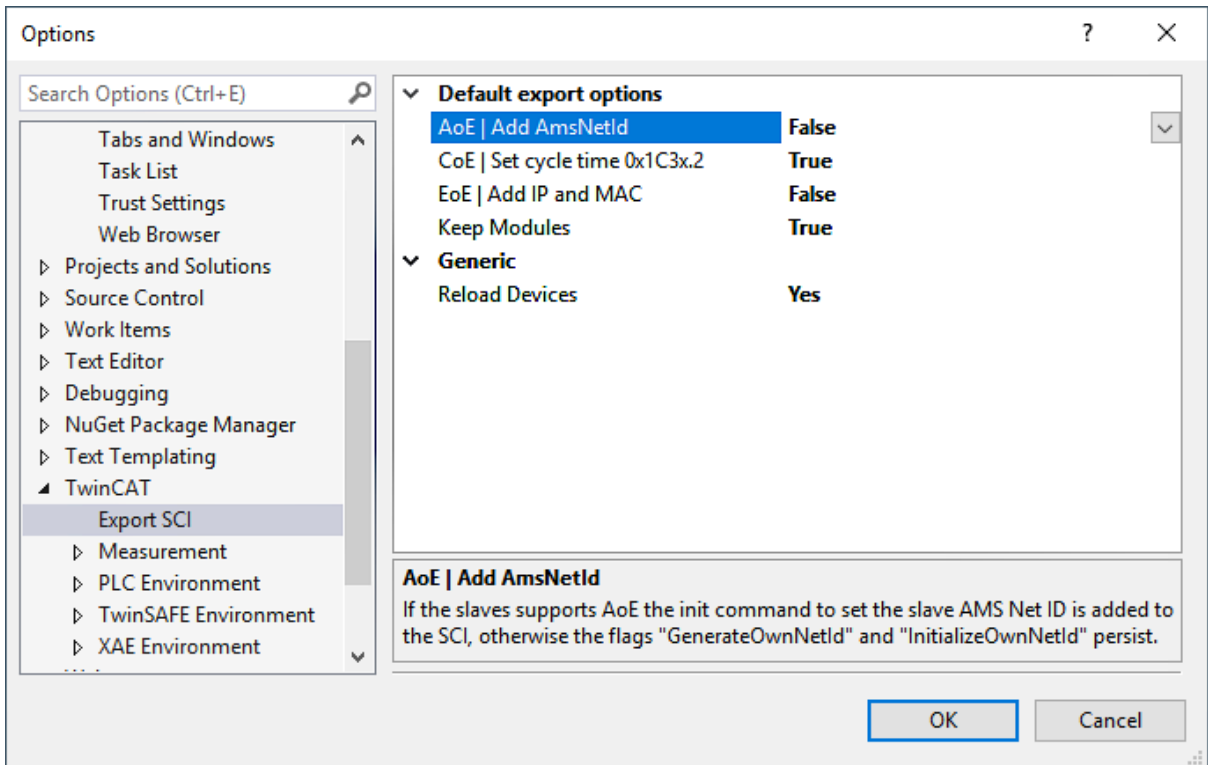


- 显示 SCI 设备，选择并插入所需设备：



补充说明

- 可以通过常规选项对话框 (Tools → Options → TwinCAT → Export SCI) 对 SCI 功能进行设置：

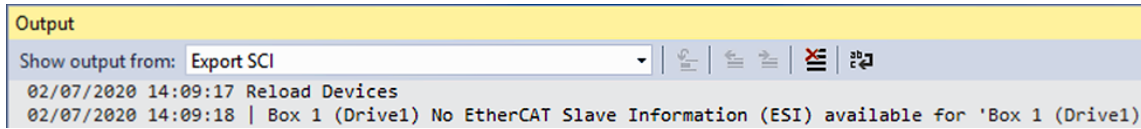


设置说明：

Default export options (默认的导出选项)	AoE   设置 AmsNetId	默认设置：是否导出配置的 AmsNetId。
	CoE   Set cycle time (0x1C3x.2)	默认设置：是否导出配置的周期时间。
	EoE   Set MAC and IP	默认设置：是否导出配置的 MAC 和 IP 地址。
	Keep modules (保留模块)	默认设置：模块是否持续存在。
Generic (通用选项)	Reload Devices (重新载入设备)	设置是否在 SCI 导出前执行重新载入设备命令。强烈建议这样做，以确保从站配置的一致性。



如果需要，SCI 错误信息会显示在 TwinCAT 日志的输出窗口：



### 6.3 EtherCAT 从站的一般调试说明

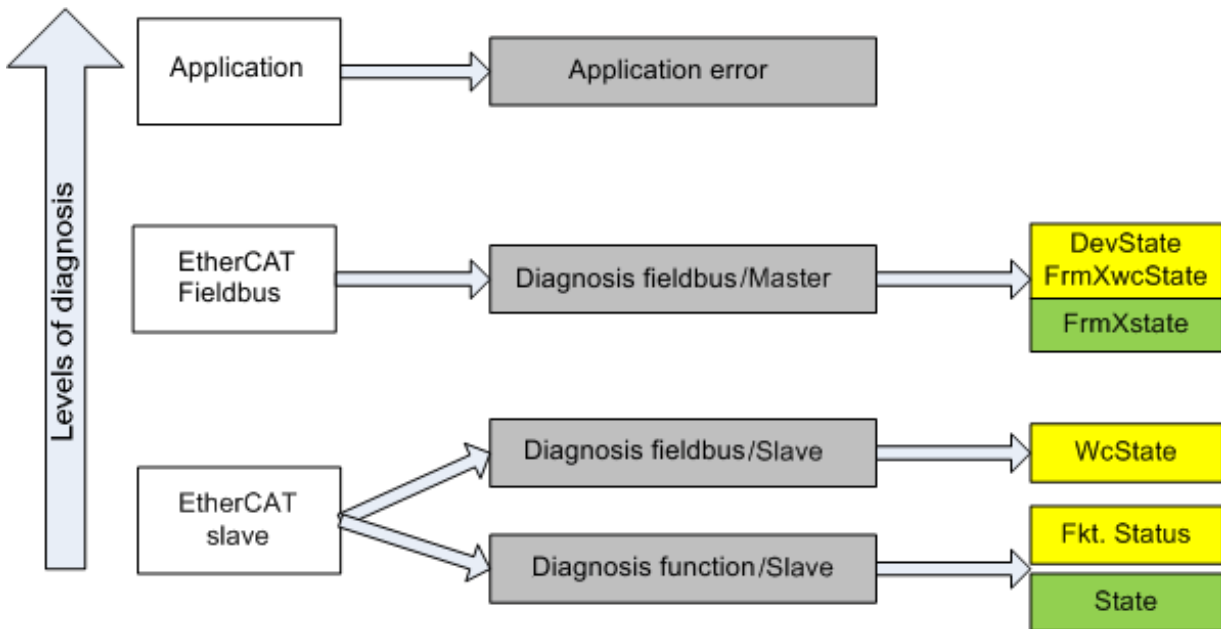
该摘要简单介绍了 TwinCAT 下的 EtherCAT 从站运行的若干方面。关于详细信息，可查看相应章节，例如 EtherCAT 系统文档。

#### 实时诊断：WorkingCounter、EtherCAT State 和 Status

一般来说，EtherCAT 从站提供可供控制任务使用的各种诊断信息。

这种诊断信息与不同的通信层级有关。因此，它有不同的来源，也会在不同的时间进行更新。

任何应用，如果严格要求现场总线的 I/O 数据保持正确和最新，就必须对相应的底层进行诊断性访问。EtherCAT 和 TwinCAT System Manager 全面提供这种诊断要素。下面讨论那些有助于控制任务进行诊断，且在正常运行时（而不仅是在调试阶段），能够在当前周期保持准确刷新的诊断要素。



附图 132: 选择 EtherCAT 从站的诊断信息

一般来说，EtherCAT 从站提供

- 典型的从站通信诊断（成功参与过程数据交换以及正确运行模式的诊断）  
这种诊断对所有从站都一样。

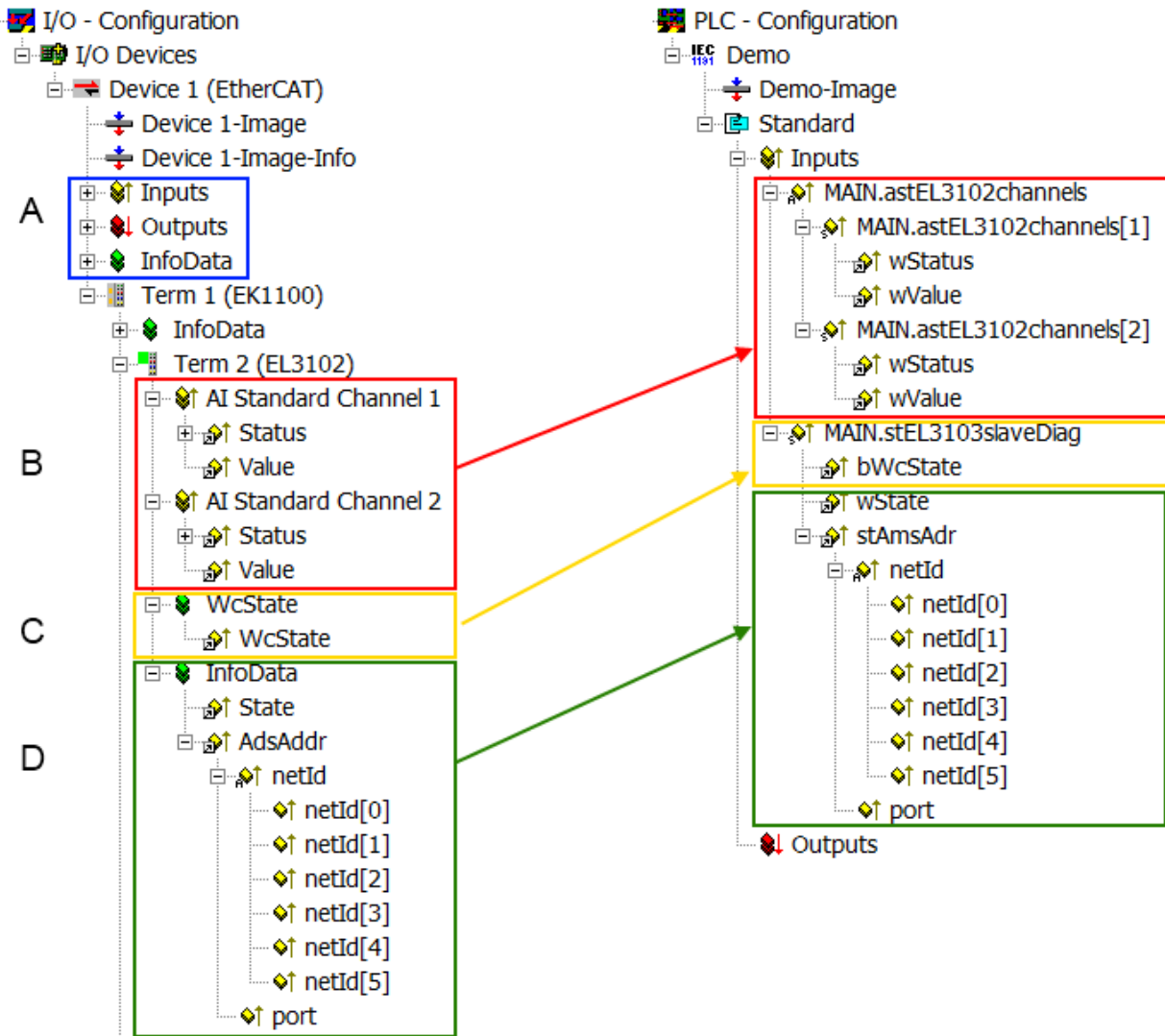
以及

- 典型的通道功能诊断（与设备有关）  
参见相应的设备文件

图选择 EtherCAT 从站的诊断信息中的颜色也与 System Manager（系统管理器）中的变量颜色相对应，参见图 PLC 中的基本 EtherCAT 从站诊断。

颜色	含义
黄色	从站到 EtherCAT 主站的输入变量，在每个周期内更新
红色	EtherCAT 主站到从站的输出变量，在每个周期内更新
绿色	EtherCAT 主站的信息变量，非周期性更新。这意味着，在任意的特定周期，它们有可能并不代表最新的状态。因此，通过 ADS 读取此类变量非常有用。

图 PLC 中的基本 EtherCAT 从站诊断显示了实现基本 EtherCAT 从站诊断的示例。这里使用的是倍福 EL3102（2 通道模拟量输入端子模块），因为它既能提供典型的从站通信诊断，又能提供通道特有的功能诊断。在 PLC 中创建结构体作为输入变量，每个结构体都与过程映像相对应。



附图 133: PLC 中的 EtherCAT 从站基本诊断

这里包括以下几个方面:

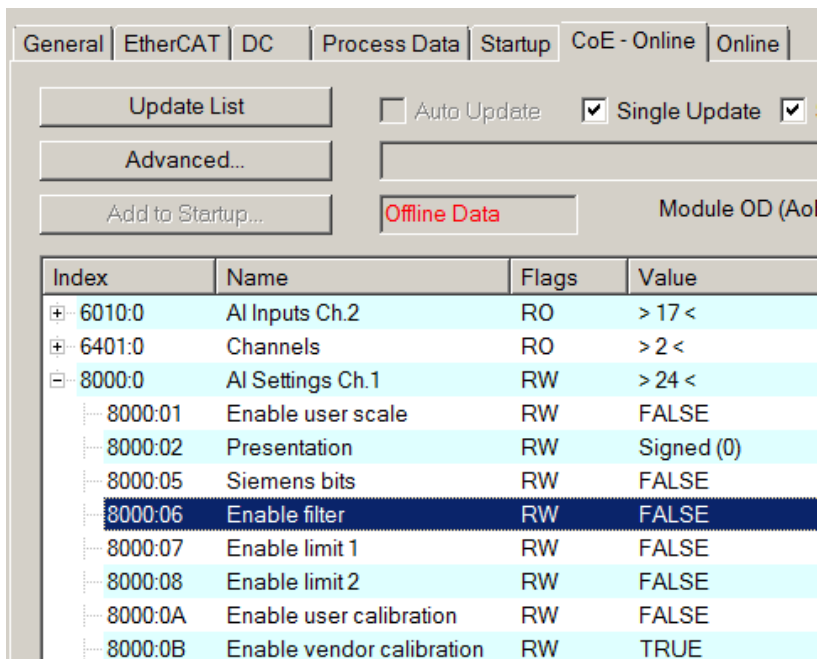
代码	功能	实施	应用/评估
A	EtherCAT 主站的诊断信息 周期性更新（黄色）或非周期性提供（绿色）。		至少要对 PLC 中最近一个周期的 DevState 进行评估。 相对于 EtherCAT 系统文档中所涉及诊断，这里的 EtherCAT 主站诊断信息提供了更多的可能性。几个关键词： <ul style="list-style-type: none"> <li>主站中的 CoE 用于与/通过从站进行通信</li> <li>TcEtherCAT.lib 功能</li> <li>执行在线扫描</li> </ul>
B	在选择的示例中（EL3102），EL3102 包括两个模拟量输入通道，传输最近周期的单一功能状态。	Status <ul style="list-style-type: none"> <li>位符号可参见设备手册</li> <li>其他设备可能提供更多的信息，或者没有典型的从站信息</li> </ul>	为了确保上级 PLC 任务（或相应的控制应用）能够获取正确的数据，必须评估从站功能的状态。因此，此类信息与最近一个周期的过程数据一起提供。
C	对于每个拥有周期性过程数据的 EtherCAT 从站，主站通过 WorkingCounter 显示该从站是否成功、无错误地参与了过程数据的周期性交换。于是在 System Manager 中提供了 EtherCAT 从站在最近周期的这一重要的基本信息，并且与 EtherCAT 主站的综合诊断变量（见 A 点）具有相同的内容，以进行链接。  1. 2.	WcState（工作计数器） 0: 在最近一个周期中有效的实时通信 1: 无效的实时通信 这可能会对位于同一 SyncUnit（同步单元）中其他从站的过程数据产生影响。	为了使上级 PLC 任务（或相应的控制应用）能够依赖正确的数据，必须评估 EtherCAT 从站的通信状态。因此，此类信息与最近一个周期的过程数据一起提供。
D	EtherCAT 主站的诊断信息在从站中表示，以便于链接，但实际上是由主站为相关的从站确定和进行表示。这种信息不能称为实时信息，因为它 <ul style="list-style-type: none"> <li>除了系统启动时，很少/从不 改变。</li> <li>本身是非周期性确定的（例如 EtherCAT 状态）</li> </ul>	State 从站的当前状态（INIT...OP）。正常运行时，从站必须处于 OP (=8) 状态。 AdsAddr ADS 地址用于从 PLC/任务通过 ADS 与 EtherCAT 从站进行通信，例如对 CoE 进行读/写。从站的 AMS-NetID 与 EtherCAT 主站的 AMS-NetID 相对应；通过端口 (= EtherCAT 地址)，可以与各个从站进行通信。	EtherCAT 主站的信息变量，非周期性更新。这意味着，在任意的特定周期，它们有可能并不代表最新的状态。因此，可以通过 ADS 读取此类变量。

**注意**

**诊断信息**  
强烈建议对所提供的诊断信息进行评估，以便应用能够适当的反应。

**CoE 参数目录**

CoE 参数目录（CanOpen-over-EtherCAT）用于管理相关从站的设定值。在某些情况下，当调试一个相对复杂的 EtherCAT 从站时，可能需要在进行这里进行修改。它可以通过 TwinCAT System Manager 访问，参见图 EL3102, CoE 字典:



附图 134: EL3102, CoE 字典

## ● EtherCAT 系统文档

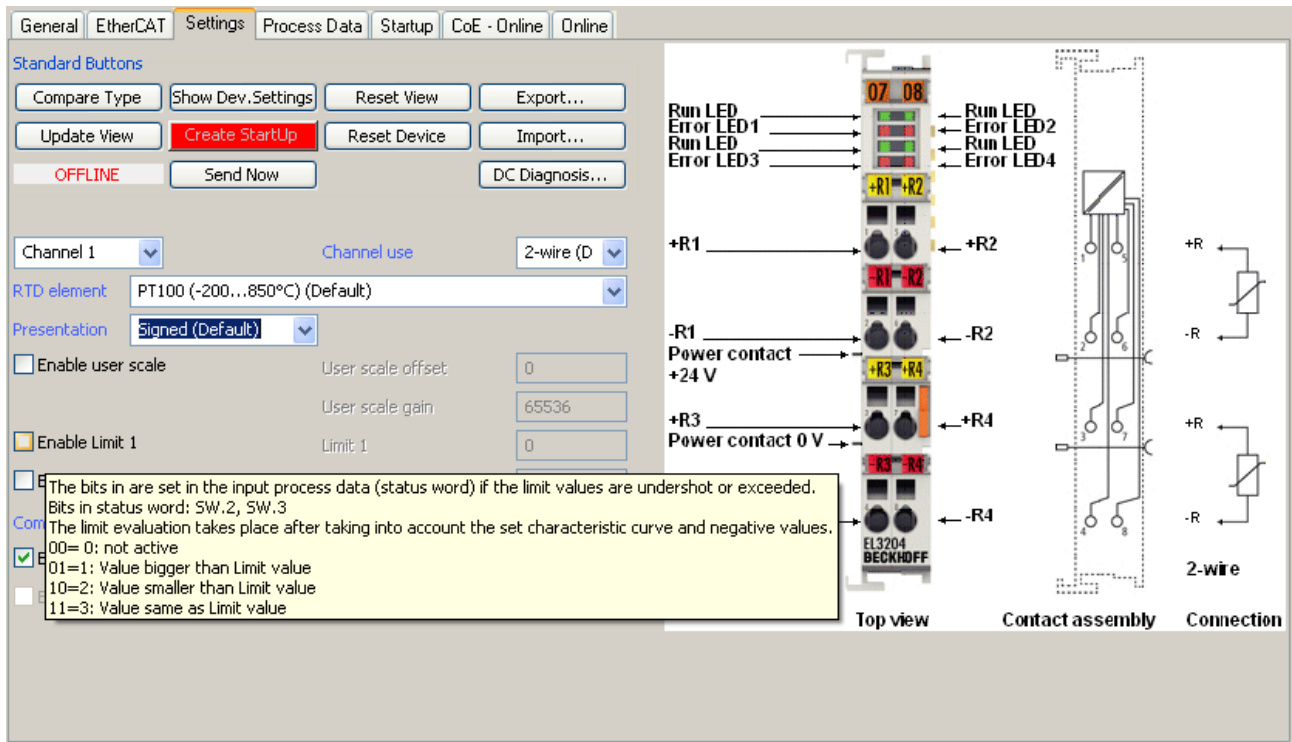
**i** 必须遵守 [EtherCAT 系统文档](#)（EtherCAT 基础知识-->CoE 接口）中的全面描述！

### 简要摘录:

- 在线目录中的更改是否保存在从站本地，这取决于从站设备。EL 端子模块（EL66xx 除外）能够以这种方式保存修改的参数。
- 用户必须管理对“StartUp”列表的更改。

### TwinCAT System Manager 中的调试助手

TwinCAT 中引入了调试界面，这是 EL/EP 等 EtherCAT 设备持续开发过程的一个新增功能。从 TwinCAT 2.11R2 及以上版本开始，都在 TwinCAT System Manager 中提供了调试助手。它们通过适当扩展的 ESI 配置文件集成到 System Manager 中。



附图 135: EL3204 调试助手示例

这个调试过程同时还管理：

- CoE 参数目录
- DC/FreeRun 模式
- 可用的过程数据（PDO）

尽管“Process Data”、“DC”、“Startup”和“CoE-Online”等过去必须的设置选项卡仍然需要显示，但如果使用调试助手，建议不要用它们来改变自动生成的设置。

调试工具并未涵盖 EL/EP 设备所有可能的应用。如果可用的设置选项不够齐全，用户可以像过去一样手动进行 DC、PDO 和 CoE 设置。

### EtherCAT 状态: TwinCAT System Manager 的自动默认行为和手动操作

工作电源接通后，EtherCAT 从站必须经历以下状态

- INIT
- PREOP
- SAFEOP
- OP

以确保稳定运行。EtherCAT 主站根据从站的初始化流程来主导每个从站的状态，该流程是在 ESI/XML 和用户设置（分布时钟（DC）、PDO、CoE）中专为调试设备而定义的。另请参见链接“通信原理，EtherCAT 状态机 [►\_30]”。根据需要完成配置的数量以及整体通信情况，启动过程可能需要几秒钟。

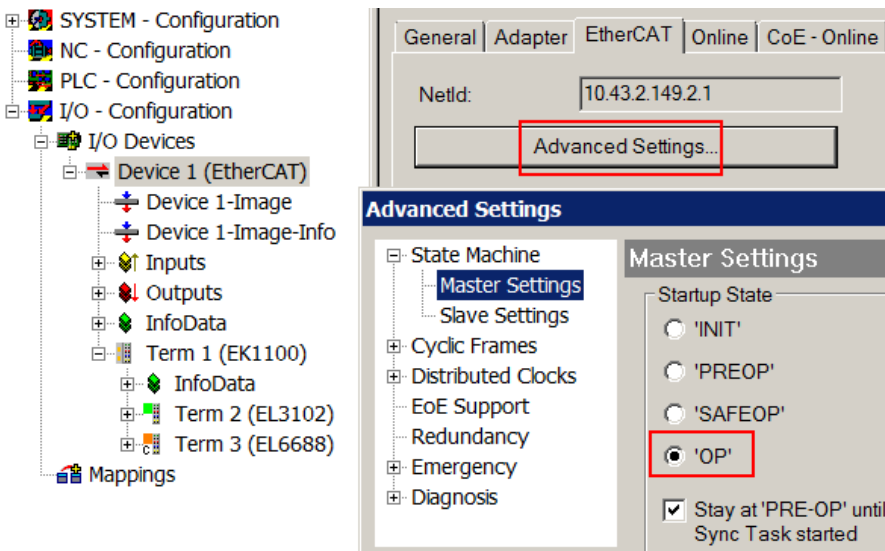
EtherCAT 主站本身在启动时必须经过这些例行的步骤，直到目标状态 OP 。

用户所需的目标状态可在系统管理器中进行设置，TwinCAT 启动时会自动引导状态切换。一旦 TwinCAT 进入 RUN 状态，TwinCAT EtherCAT 主站就会逐步达到目标状态。

### 标准设置

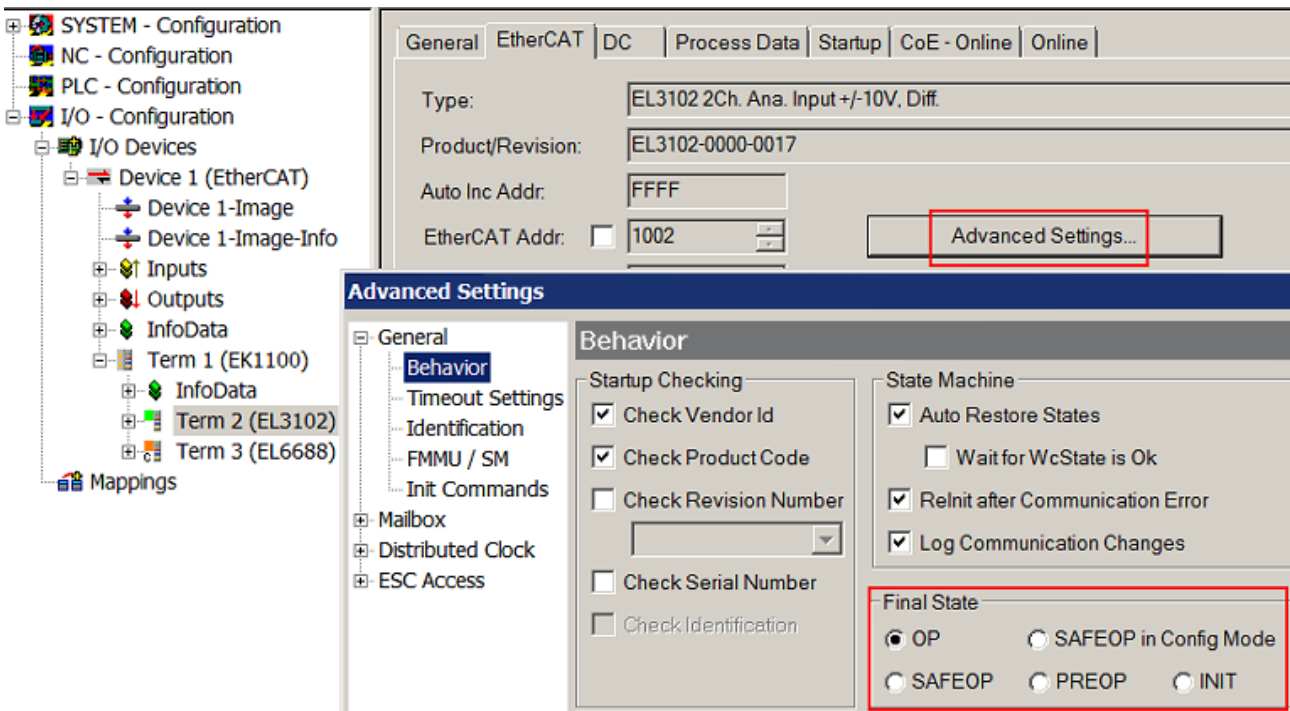
EtherCAT 主站的 advanced settings（高级设置）的标准设置如下：

- EtherCAT 主站: OP
- 从站: OP  
该设置同样适用于所有从站。



附图 136: System Manager 的默认行为

此外，任何特定从站的目标状态均可在“Advanced Settings”对话框中设置；标准设置仍然是 OP。



附图 137: 从站的默认目标状态

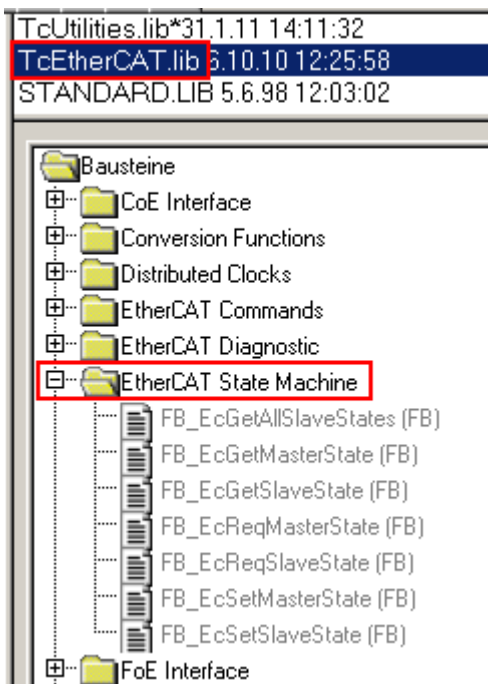
### 手动控制

在某些特殊原因下，可能需要从应用/任务/PLC 中控制EtherCAT状态。例如：

- 出于诊断的原因
- 为了触发一个可控的 EtherCAT 重启过程
- 因为需要改变 EtherCAT 主站的启动时间

此时适合在 PLC 程序中调用来自 *TcEtherCAT.lib* 的 PLC 功能块（包含在TwinCAT 标配功能中）并使用 *FB\_EcSetMasterState* 等以可控的方式推进各种状态的切换。

所以，在 EtherCAT 主站设置中将主站和从站不（的目标状态）都设置到 INIT 状态是很有用的。



附图 138: PLC 功能块

关于 E-Bus 电流的说明

EL/ES 端子模块置于 DIN 导轨上，紧跟在耦合器后面。总线耦合器可以向后续的EL 端子模块供给 5V 的 E-bus 系统电压；耦合器原则上可以负担最多 2 A 的 E-Bus 电流。关于每个 EL 端子模块需要消耗多少 E-bus 电流的信息，可参见倍福公司网站和产品目录。如果后续的端子模块需要的电流超过了耦合器可以提供的电流，则必须在 I/O 站的适当位置插入E-Bus电源端子模块（例如 EL9410）。

预先计算的最大 E-Bus 理论电流在 TwinCAT System Manager 中显示为一列数值。如果预计E-Bus供电不足，剩余电流总额就会是负数，并以感叹号 (!) 标记；在这种位置前面需要插入一个E-Bus电源模块。

General   Adapter   EtherCAT   Online   CoE - Online						
NetId:		10.43.2.149.2.1		Advanced Settings...		
Number	Box Name	Address	Type	In Size	Out S...	E-Bus (..
1	Term 1 (EK1100)	1001	EK1100			
2	Term 2 (EL3102)	1002	EL3102	8.0		1830
3	Term 4 (EL2004)	1003	EL2004		0.4	1730
4	Term 5 (EL2004)	1004	EL2004		0.4	1630
5	Term 6 (EL7031)	1005	EL7031	8.0	8.0	1510
6	Term 7 (EL2808)	1006	EL2808		1.0	1400
7	Term 8 (EL3602)	1007	EL3602	12.0		1210
8	Term 9 (EL3602)	1008	EL3602	12.0		1020
9	Term 10 (EL3602)	1009	EL3602	12.0		830
10	Term 11 (EL3602)	1010	EL3602	12.0		640
11	Term 12 (EL3602)	1011	EL3602	12.0		450
12	Term 13 (EL3602)	1012	EL3602	12.0		260
13	Term 14 (EL3602)	1013	EL3602	12.0		70
14	Term 3 (EL6688)	1014	EL6688	22.0		-240 !

附图 139: E-Bus 电流非法超出电源限值

从 TwinCAT 2.11 及以上版本开始，在该配置激活时，警告信息“E-Bus Power of Terminal...” 将出现在日志窗口：

**Message**

E-Bus Power of Terminal 'Term 3 (EL6688)' may to low (-240 mA) - please check!

附图 140: 超过 E-Bus 电流的警告信息

**注意****注意！可能发生故障！**

一个 I/O 站里面所有 EtherCAT 端子模块的 E-Bus 电源必须使用相同的接地电位！



## 6.4 运行模式和过程数据

### 版本说明

单通道 EL60x1 经过多年开发和改进。EL6001/EL6021 的进一步开发实现了以下功能扩展：

- 在 EL6001 固件 (FW) 05 / 硬件 (HW) 03 及以上版本以及 EL6021 固件 (FW) 04 / 硬件 (HW) 03 及以上版本中，用于状态监控和参数设置的对象也可从 Index 0x6000 (描述文件定义的对象) 中获得，并可根据硬件情况在 TwinCAT System Manager 中进行参数设置。
- EL6021 固件 (FW) 06 / 硬件 (HW) 03 及以上版本支持命令模式 [▶ 142]
- EL6001 固件 08 及以上版本支持可发送/接收 8 位以上数据的 16 位过程数据接口
- EL6001 固件 11 及以上版本可使用 1000...115200 baud 范围内的所有整数波特率。
- 

EL6001/EL6021 的旧版本不支持此功能！

FW/HW 版本及以上	ESI	控制/状态/参数化对象区域
EL6001 01/01 EL6021 01/01	EL6001-0000-0000 及以上 EL6021-0000-0000 及以上	Index 0x300n:01 (Control-Word) Index 0x310n:01 (Status-Word) Index 0x4070 (Data bytes in send buffer) Index 0x4071 (Data bytes in receive buffer) Index 0x4072 (Diagnosis)  Index 0x4073 (Baud rate) Index 0x4074 (Data frame) Index 0x4075 (Feature bits)
EL6001 05/03 EL6021 04/03	EL6001-0000-0016 及以上 EL6021-0000-0016 及以上	除上述对象外还包括： Index 0x6000 (COM inputs) Index 0x7000 (COM outputs) Index 0xA000 (COM Diag data) Index 0x8000 (COM settings)
EL6021 06/03	EL6021-0000-0018 及以上	除上述对象外还包括： Index 0xB000 (Command)
EL6001 08/03	EL6001-0000-0019 及以上	除上述对象外还包括： 16 位 PDO 对象
EL6001 11/11	EL6001-0000-0020 及以上	通过 explicitBaudrate 实现 1000...115200 范围内的所有波特率

修订版本为 -0016 的 EL6002/EL6022 设备从第一版开始就具有 6000、7000 和 8000 系列对象。

#### ● 售后服务期间的产品兼容性



例如：不能用硬件版本较低 (03 以下) 的 EL6001/EL6021 代替硬件 03 及更高版本的 EL6001/EL6021！高版本替代低版本则完全可行！

#### ● 过程数据监视



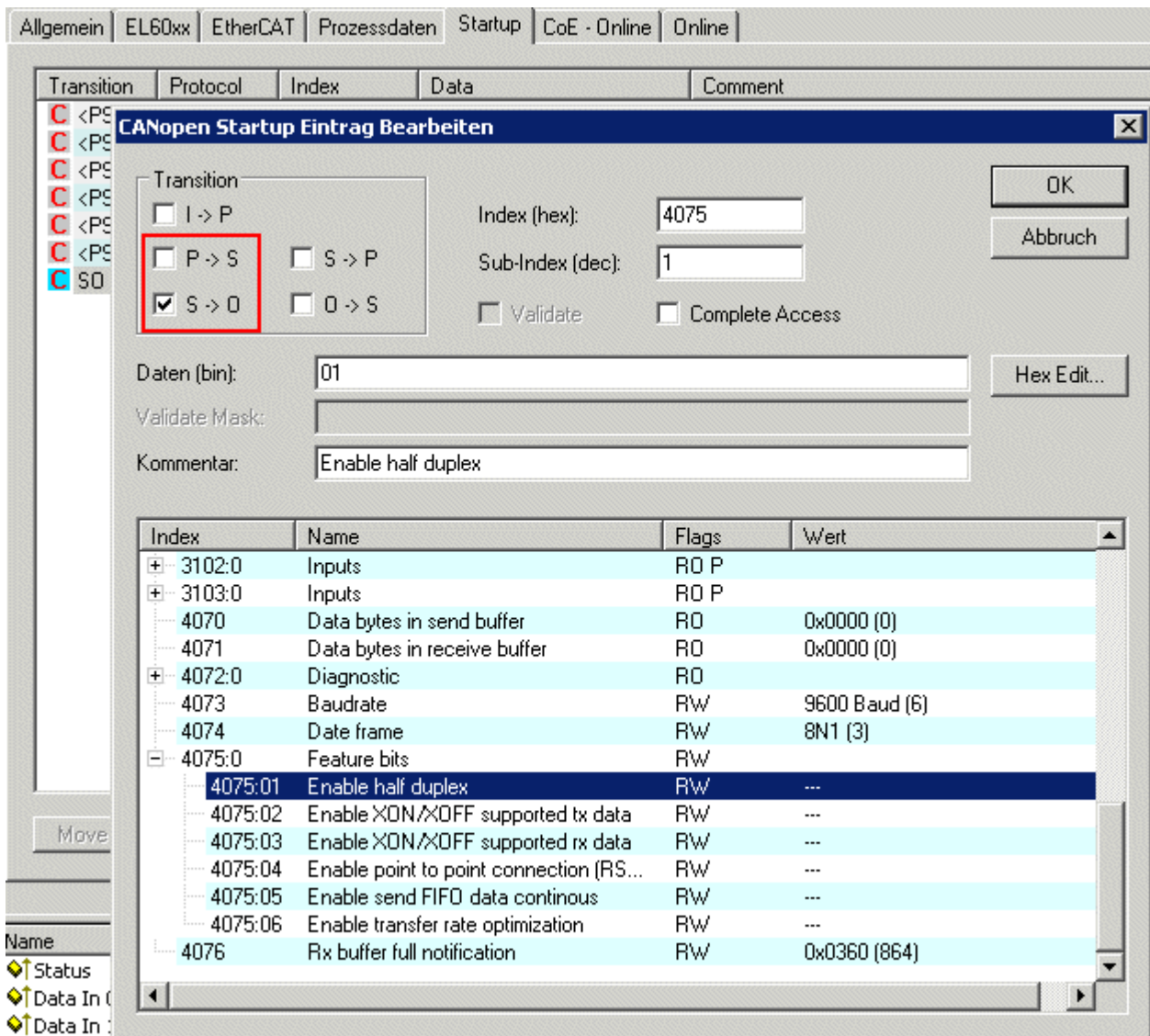
- **WcState:** 如果  $\neq 0$ ，则该 EtherCAT 设备不参与过程数据通信
- **State:** 如果  $\neq 8$ ，则 EtherCAT 设备不处于 OP (运行) 状态
- **TxPDO state, SyncError:** 如果  $\neq 0$ ，则没有有效的过程数据，例如由于断线导致
- **TxPDO Toggle:** 如果该位切换，则表示有新的过程数据集可用

#### StartUp 条目 (硬件版本 < 03)

##### ● StartUp list 条目



对于硬件版本低于 03 的 EL6001/EL6021，只能在从 SafeOP 到 OP (S  $\rightarrow$  O) 的转换过程中设置 StartUp 条目。默认设置为 PreOP  $\rightarrow$  SafeOP (P  $\rightarrow$  S)。创建 StartUp 条目时，确保勾选“S  $\rightarrow$  O”复选框 (见图)！



附图 141: S -> O 转换的 StartUp 条目

### 过程数据

出厂时的产品已传输 22 个字节的用户数据和 1 个控制/状态字。

过程数据由 CoE 对象 0x6000 (Inputs) 和 0x7000 (Outputs) 生成，详见对象描述和参数化 [▶ 184] 章节。

### 扩展的过程映像 50x 16-bit

如果通信数据位大于 8 位，则需要扩大过程数据接口。如果端子模块支持此功能（参见通信功能 [▶ 147]），则可将 50 字接口设置为 22 字节接口的替代（PDO 0x1605 和 0x1A05）。这可用于所有加密规定（7xx、8xx），但只有规定超过 8 位时才有意义，例如 9N1。在每种情况下，数据位必须占用最低有效位，即在 9N1 的情况下，数据字中的 9 个最低有效位。

### EL60x2-0016 及以上版本的过程数据

EL60x2-0016 及以上版本

Sync Manager (SM) - PDO 分配

SM2, PDO 分配 0x1C12				
Index	被排除 PDO 的 Index	大小 (byte.bit)	名称	PDO 内容
0x1600	-	24.0	COM Outputs Channel1	Index 0x7000:01 - Ctrl__Transmit request
				Index 0x7000:02 - Ctrl__Receive accepted
				Index 0x7000:03 - Ctrl__Init request
				Index 0x7000:04 - Ctrl__Send continous
				Index 0x7000:09 - Ctrl__Output length
				Index 0x7000:11 - Data out 0
				---
0x1601	-	24.0	COM Outputs Channel2	Index 0x7010:01
				Index 0x7010:02
				Index 0x7010:03
				Index 0x7010:04
				Index 0x7010:09
				Index 0x7010:11
				---
0x1604 (默认)	-	24.0	COM Outputs Channel1	Index 0x7001:01 - Ctrl
				Index 0x7000:11 - Data out 0
				---
0x1605 (默认)	-	24.0	COM Outputs Channel2	Index 0x7011:01 - Ctrl
				Index 0x7010:11 - Data out 0
				---
				Index 0x7010:26

SM3, PDO 分配 0x1C13				
Index	被排除 PDO 的 Index	大小 (byte.bit)	名称	PDO 内容
0x1A00	-	24.0	COM Inputs Channel1	Index 0x6000:01 - Status_Transmit accepted
				Index 0x6000:02 - Status_Receive request
				Index 0x6000:03 - Status_Init accepted
				Index 0x6000:04 - Status_Buffer full
				Index 0x6000:05 - Status_Input length
				Index 0x6000:06 - Status_Framing error
				Index 0x6000:07 - Status_Overrun error
				Index 0x6000:09 - Status_Input length
				Index 0x6000:11 - Data in 0
				---
Index 0x6000:26				
0x1A01	-	24.0	COM Inputs Channel2	Index 0x6010:01
				Index 0x6010:02
				Index 0x6010:03
				Index 0x6010:04 - Status_Buffer full
				Index 0x6010:05
				Index 0x6010:06
				Index 0x6010:07
				Index 0x6010:09
				Index 0x6010:11
				---
Index 0x6010:26				
0x1A04 (默认)	-	24.0	COM Inputs Channel1	Index 0x6001:01
				Index 0x6000:11
				---
Index 0x6000:26				
0x1A05 (默认)	-	24.0	COM Inputs Channel2	Index 0x6011:01
				Index 0x6010:11
				---
Index 0x6010:26				

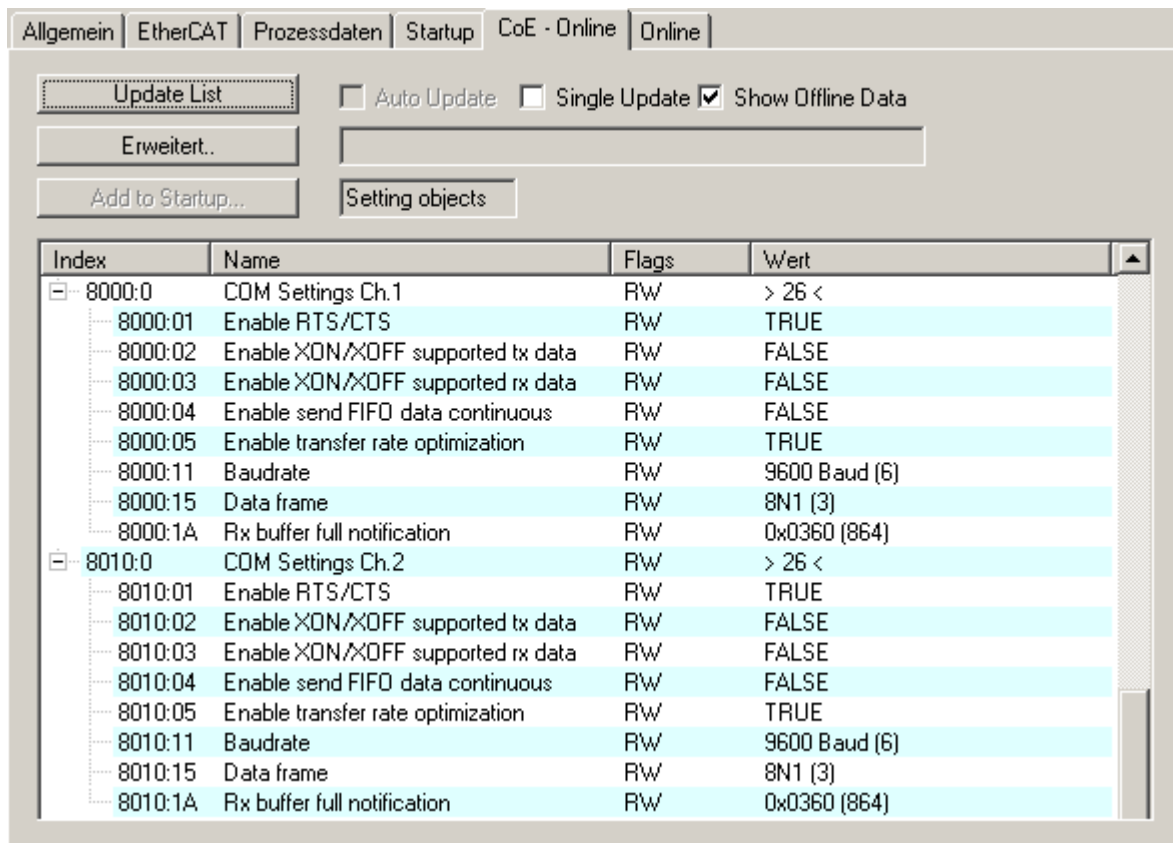
**功能和应用说明**

可选功能通常在 CoE 目录中设置 (CAN over EtherCAT, Index 0x80n0)。

**● 通过 CoE 列表进行参数化 (CAN over EtherCAT)**

**i** 在使用/操作 CoE 参数时, 请注意以下常规 CoE 信息: - 如果组件需要更换, 请保留 Startup List - 注意 Online/Offline 字典的区别, 确保存在当前 XML 描述 - 使用“CoE reload”重置更改。请特别注意, 对于 EL60xx, 如果对通信设置 (波特率、数据帧、功能位) 进行任何更改, 则需要通过控制字发出 InitRequest, 才能使更改生效。

从对象 0x8000 可以进行以下 CoE 设置, 其默认设置如下图所示:



附图 142: “CoE - Online, EL60x2 terminals” 选项卡

## 传输速率

端子模块有一个包含 22 字节用户数据的过程映像。每个周期最多可以发送或接收 22 个字节。数据在第一个周期内从端子模块传输到控制器。在第二个周期中，控制器必须确认其已接受数据。因此，如果周期时间为 10 ms，则每秒可传输 50 次 22 个字节。

如果数据帧设为 8N1，则每个传输字节包括 1 个起始位、8 个数据位和 1 个停止位。这相当于每个用户字节 10 位。

基于上述设置，可实现连续数据传输速率：  
 $50[1/s] * 22[Byte] * 10[Bit] = 11000 \text{ bps}$

。

下一个较低的波特率是 9600 baud。因此，最大波特率为 9600 的连续传输只需 10 ms 的周期时间。

在固件 11 及以上版本中，CoE 0x8000:1B 通过 *explicitBaudrate* 支持 1000...115200 baud 范围内的所有波特率。在选择波特率时，必须考虑最远距离与所设定波特率的兼容性。

如果只是偶尔发送或接收少量数据（如条形码扫描仪），也可将波特率设置得更高，或将周期时间延长。如果控制器来不及从端子模块快速地及时取走数据，数据将被暂时存储在端子模块的内部缓冲区中。接收数据的缓冲区大小为 864 字节。如果该区域被填满，所有后续从串口设备接收到的数据都将丢失。

另一种情况是：控制器向端子模块传输的数据大大超过了端子模块的传输能力。“波特率”设置为 300，“数据帧”设置为 8N1 时，端子模块每秒只能传输 30 个字节。但如果每秒接收的字节数超过 30 字节，这时也会先写入 128 字节的发送缓冲区。一旦填满，所有后续从控制器收到的数据都将丢失。

## 传输速率优化：自动续接所收到的数据

在串行通信中，数据记录通常以连续的字节流形式发送。两个数据记录之间有停顿。端子模块可根据停顿识别数据记录的开始和结束时间。这样，它就能将数据记录的字节组合起来，并以连贯的方式传递给控制器。

运行模式：首先将连续接收到的几个字节收集到接收缓冲区。如果在一个字节后检测到停顿（或缓冲区已满），收集的接收数据将从接收缓冲区传输到过程映像。“Receive Request”位反转，表示有新的接收数据。

## 启用/停用

出厂设置已经通过 CoE 对象 0x80n0:05 “Enable transfer rate optimization” 启用了接收数据自动续接的功能。如果要想让串行端子模块发送的数据尽快传输到控制器，可以停用此功能。如果串行设备连续发送数据不会停顿，也可以停用此功能。如果停用了接收数据自动续接功能，则每个接收到的字节都会立即转送到过程映像中。

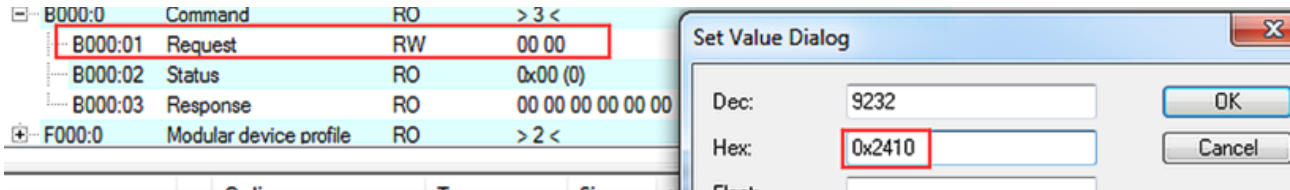
### 需要续接的两个数据串之间的停顿时长可调

注：该功能在以下版本中可用

- EL6001 固件 12 及以上版本
- EL6021 和 EL6021-0021 的固件 10 及以上版本。

如果要更改传输速率优化的标准时间，必须在 CoE 对象 0xB000:01 的低字节输入等待时间 1...254，单位是”位时间“（bit time，即传输 1 个二进制位所需要的时间），高字节固定为 0x24。

例如：0x10 (16<sub>dec</sub>) 表示等待时间为 16 位时间：



附图 143: 将等待时间设置为 16 位时间的示例

在 115.2 kBaud 速率下，该时间相当于 139 μs: (1/115200 [位/秒]) \* 16 [位]。经过这段时间间隔后，数据将自动复制到过程映像中。

## 连续传输数据

通常，EL60xx 端子模块会自动决定何时发送缓冲区中的数据字节。对于许多应用来说，连续的数据流是很有帮助的。为此，倍福 EL60xx 端子模块在“Settings”对象中设置了“启用连续发送 FIFO 数据”（Enable send FIFO data continuous）。如果启用了该设置：

- 来自控制器的数据必须首先填满内部发送缓冲区（128 字节）。为此，数据将像在正常数据传输中那样从控制器发送到端子模块。
- 随着“Send continuous”位上升沿的出现，开始从缓冲区发送数据
- 如果数据已发送，端子模块会通过设置“InitAccepted”位通知控制器。通过“Send continuous”清除“Init accepted”。

即使 EtherCAT 周期时间较长，波特率较高，该设置也能在短时间内传输多达 128 字节时。

端子模块在每个停止位之后无缝跟进下一个起始位，从而尽量缩短报文之间的距离。不过，在双通道模式下，高波特率可能会导致报文之间出现短暂停顿。例如，一个 20 字节的数据块可以分为 5 个加 15 个或 7 个加 13 个报文的两个数据块发送，或者采用其它不同的分配方式。在双通道的串口模块中，通道 1 优先。

## 优先次序

由于接收到的数据通常不会重复，因此其优先级高于要发送的数据。此外，通道序号越靠后，优先级越低。因此，在通道 1 上接收数据的优先级最高。

## 命令模式

EL6021 固件 06/修订版本 -0018 及以上支持命令模式。端子模块的某些功能，可以通过选择命令的组合或顺序，进行使用和控制。目前支持以下功能：

- 多数据帧功能：（固件 06 及以上版本）在发送数据的过程中更改编码方式  
为此，应先将要发送的字节填入端子模块的发送缓冲区（注意缓冲区大小）。通过控制字启动发送过程后，前 n 个字节使用编码 A 发送，缓冲区中的其余字节使用编码 B 发送。然后，缓冲区被重新填满并进行相应发送。举例说明：第一个字节在校验位为 1 时（Mark Parity）发送，其余字节在校验位为 0 时（Space Parity）发送。  
流程：

- 每次启动/重启 EL60xx EtherCAT 从站后，都必须重新进行参数设置。命令功能不是掉电保持的。
- 在请求 0x8000:04 之后写入 0x1，启用“SendContinuous”模式
- 在请求 0xB000:01 后写入 0x2001，启用多数据帧功能  
检查 0xB000:02 是否等于 0
- 在请求 0xB000:01 后写入 0x2100 + [日期帧代码]，指定编码 A  
例如：8E2 = 0x12 --> 写入值 0x2112  
检查 0xB000:02 是否等于 0
- 在请求 0xB000:01 之后写入 0x2200 + [n]，指定编码 A 中要发送的 n 个字节  
检查 0xB000:02 = 0
- 在请求 0xB000:01 之后写入 0x2300 + [日期帧代码]，指定编码 B  
检查 0xB000:02 是否等于 0
- 触发 Ctrl *SendContinuous* 上升沿，开始发送
- 端子模块状态位 *InitAccepted* = 1 完成传输。*InitAccepted* 可以由控制位 *SendContinuous* 重置。

CmdName 的含义 0x2000 + ControlByte, CmdControlByte 位 0: 启用 MultiDataFrame 功能 位 1...7: 不使用 0x2100 + 数值, 第一个数据帧, 从数据帧表格中选择需要的值 0x2200 + NoOfBytes, 字节数, 第一个数据帧传输的字节数 0x2300 + 数值, 第二个数据帧, 从数据帧表格中选择需要的值  
注: 在该模式下, 还支持 8M1 和 8S1 编码, 但不能通过 0x8000:15 选择该编码。

- 可根据要求提供其它功能

## 数据传输示例

### 初始化

初始化在首次传输/接收之前执行。这样，端子模块就可以利用相应 Settings 对象中的数据进行参数化。

注意：x80n0:01 “Enable RTS/CTS” 默认为 “TRUE”，模块固件预设RTS/CTS引脚有接线，可以接收相应的信号。在本示例中，如果这些引脚未接线，则必须将此设置为 FALSE。

程序：

1. 将 “Init request” 设为 1
2. 端子模块通过设置 “Init accepted” 确认初始化成功
3. 重置 “Init request”
4. 端子模块将 “Init accepted” 设置为 0

现在端子模块已准备就绪，可进行数据交换。

### 控制器到端子模块的数据传输（发送 2 个字符）

1. 将 “Output length” 设为 2
2. 在 “Data Out 0” 和 “Data Out 1” 中填入用户数据
3. 更改 “Transmit request” 的状态
4. 端子模块通过改变 “Transmit accepted” 位的状态确认已收到数据。

### 端子模块到控制器的数据传输（接收字符）

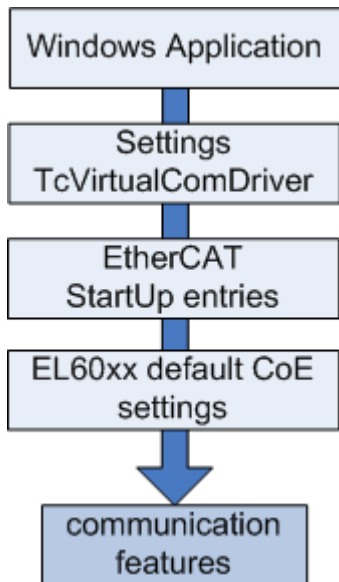
1. 端子模块通过改变 “Receive request” 位的状态来指示过程映像中有新数据。
2. 接收到的字节数写入 “Input length”
3. 控制器通过改变 “Receive request” 的状态确认接受字节



## 6.5 关于 TcVirtualComDriver 的提示信息

如果 EL60xx 要与 TwinCAT Virtual Serial COM Driver 驱动程序配合使用，请留意以下提示信息：

通常，客户特定的上层 Windows 应用程序会根据应用场景设置需要的 COM 接口功能，例如 2400 baud，7N2 编码。因此，之前所做的参数设置通常是无效的，比如在 CoE StartUp 项、“设置”（Settings）对话框、VirtualComPort 或 CoE 中输入的用户自定义参数等等。



附图 144：每个上层应用（如果可用或已启用）都会规定其下一层的通信功能。

**注意**

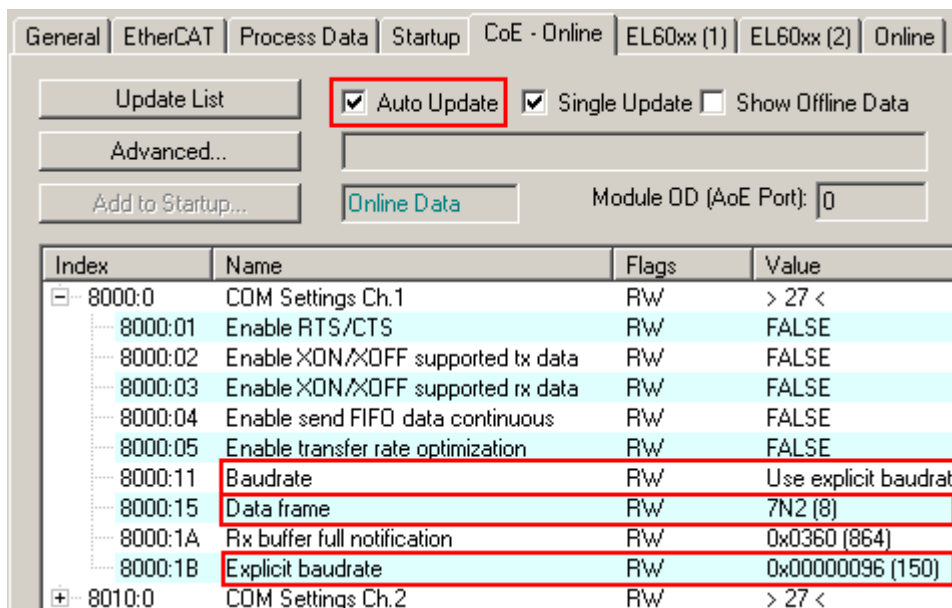
**TcVirtualComDriver 版本**

检查您使用的 TcVirtualComDriver 驱动程序版本所支持的波特率/编码组合，是否这个 EL/EP 串口通讯模块也支持。如果不支持：

- 可能无法打开 COM 端口
- System Manager 的 ADS 记录器中可能会出现 ADS 错误信息
- ADS 记录器中可能会出现 CoE 错误信息

TcVirtualComDriver 1.18 及以上版本支持 ExplicitBaudrate (CoE 0x80n0:1B)。

如果 COM 应用程序在运行期间设置了功能（如波特率），则对照“CoE-Online”选项卡检查该设置是否正确，见图。

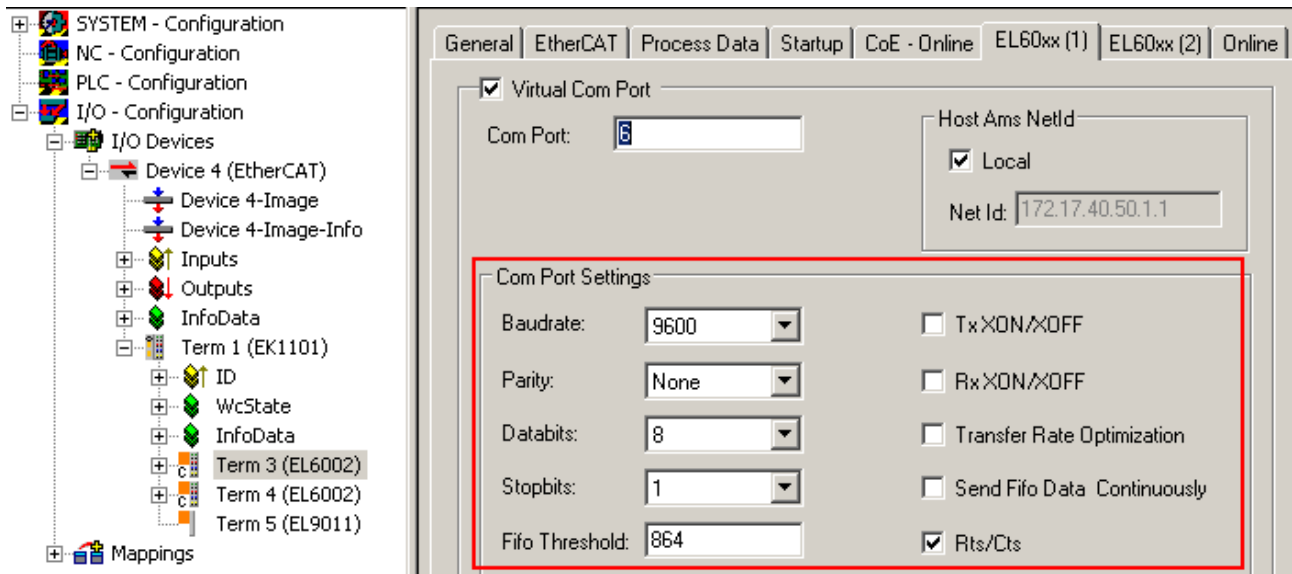


附图 145：在 CoE 中检查 COM 应用程序所需的设置

Transition	Protocol	Index	Data	Comment
C <PS>	CoE	0x1C12 C 0	02 00 00 16 01 16	download pdo 0x1C12 index
C <PS>	CoE	0x1C13 C 0	02 00 00 1A 01 1A	download pdo 0x1C13 index

startup entries not relevant for TcVirtualComDriver

附图 146: EL6002 的默认启动项 (示例) - 只需要默认项 (本例中为 0x1C12 和 0x1C13)



附图 147: VirtualComDriver 的设置 (示例)

仅当应用程序没有设置或者无法设置 COM 口的特性参数时, 这个界面上的设置才有意义。打开COM 端口时使用这些默认设置。

例外情况: *TransferRateOptimization*、*SendFifoDataContinously* 和 *FiFoThreshold* 只能通过 TwinCAT 设置。

## 6.6 通信功能

EL60xx 设备串行通信的主要功能特点是现场侧的波特率和数据报文结构/编码。这些设备提供的功能各不相同。为了查看更加清晰，在此按照固件/硬件版本列出了各个设备的可用参数。

请在倍福网页上查看是否存在本模块的最新文档。

设备概览						
设备	可用波特率			可用编码		
	波特率 [baud]	CoE 值 [0x80n0:11]	FW/HW 版本及 以上	编码	CoE 值 [0x80n0:15]	FW/HW 版本及 以上
EL6001	2400	4		7E1	1	
	4800	5		701	2	
	9600	6		8N1	3	
	12000	14	FW07 及以上版本	8E1	4	
	14400	15	FW07 及以上版本	801	5	
	19200	7		7E2	9	
	38400	8		702	10	
	57600	9		8N2	11	
	115200	10		8E2	12	
	104167	Explicit Baudrate [0x8000:1B]	FW08 及以上版本	802	13	
				9N1	32 Extended data frame [▶ 184] [0x8000:1C]	FW08 及以上版本
1000... 115200*)	Explicit Baudrate [0x8000:1B]	FW11 及以上版本				
EL6002	110	Explicit Baudrate [▶ 145] [0x80n0:1B]	FW03 及以上版本	7E1	1	
	150	Explicit Baudrate [▶ 145] [0x80n0:1B]	FW03 及以上版本	701	2	
	300	1		8N1	3	
	600	2		8E1	4	
	1200	3		801	5	
	2400	4		7N2	8	FW03 及以上版本
	4800	5		7E2	9	
	9600	6		702	10	
	19200	7		8N2	11	
	38400	8		8E2	12	
	57600	9		802	13	
	115200	10				

\*) 任何整数波特率都可以设置为 int32 值。所有波特率的频率误差均小于 0.6%。

设备概览						
设备	可用波特率			可用编码		
	波特率 [baud]	CoE 值 [0x80n0:11]	FW/HW 版本及 以上	编码	CoE 值 [0x80n0:15]	FW/HW 版本及 以上
EL6021	2400	4		7E1	1	
	4800	5		701	2	
	9600	6		8N1	3	
	19200	7		8E1	4	
	38400	8		801	5	
	57600	9		7E2	9	
	115200	10		702	10	
				8N2	11	
				8E2	12	
				802	13	
				8S1	18	FW06 及以上版本具备命令模式
			8M1	19	FW06 及以上版本具备命令模式	
EL6022	300	1		7E1	1	
	600	2		701	2	
	1200	3		8N1	3	
	2400	4		8E1	4	
	4800	5		801	5	
	9600	6		7E2	9	
	19200	7		702	10	
	38400	8		8N2	11	
	57600	9		8E2	12	
	115200	10		802	13	
EP6022	300	1		7E1	1	
	600	2		701	2	
	1200	3		8N1	3	
	2400	4		8E1	4	
	4800	5		801	5	
	9600	6		7E2	9	
	19200	7		702	10	
	38400	8		8N2	11	
	57600	9		8E2	12	
	115200	10		802	13	

## 6.7 EL6001 的 LIN 主站功能

### “LIN 主站支持”的描述 (FW10 及以上版本)

EL6001 实现了 RS232 电平的总线接口。FW10 及以上固件版本的 EL6001 具有发送和接收 LIN 报文的辅助功能。该功能不涉及对 LIN 协议层（“协议栈”）或 LIN 物理层的处理格式。但 EL6001 可以处理 LIN 报文在写入和读取方向上的 Start/End（开始/结束）加长型标记。

例如，端子模块“上层”需要一个在 PLC 中实现的 LIN 栈；端子模块“下层”需要一个 RS232 <-> 5/12 V 电平转换器，即需要使用“RS232-LIN”。

在使用该功能之前，请检查系统的适用性，因为 EL6001 并没有实现全面的 LIN 功能！

一个完整的 LIN 帧由主站查询和随后的从站响应组成。主站发送主站帧并接收从站数据。EL6001 启用 LIN 功能后，会在“受保护标识符”（PID）之前输出“同步中断”和“同步字节”。

事实上，EL6001 端子模块因此能够在 LIN 下以如下方式运行：

- EL6001 作为 LIN 主站运行，向从站发送数据并接收响应。
- EL6001 作为 LIN 从站运行：EL6001 端子模块无法及时响应主站报文！因此：
  - 是否从主站接收数据：是
  - 是否向主站发送数据：否

### 启用

通过对象“Command”可以启用添加“sync break（同步中断）”和“sync field（同步字节）”，LIN 需要它们进行发送。为此，请在 CoE 对象 0xB000:01 [▶ 182] 中输入值 0x3000。可在 CoE 对象 0xB000:03 中读取响应，其值为 0x01 0x00 0x00 0x4C 0x49 0x4E（4C 49 4E = ASCII“LIN”），以确认启用该功能。随后，端子模块准备就绪，可以在 LIN 总线上接收包含 13 位“sync break（同步中断）”和“sync field（同步字节）”（0x55）的帧，在接收数据中只包含以下信息，并在发送时输出这些 ID。

该设置不会永久储存在端子模块中，每次重新启动后都必须重新写入。使用 Sartup [▶ 117] List 可以避免这一问题。

### 用户定义的波特率

可以在对象“Explicit baudrate”（0x8000:1B）中输入任何既有的可用波特率或波特率 10417。

### 其它参数

为确保 LIN 模式下的功能正常运行，对象 0x8000 [▶ 165] 中必须采用以下设置：

Index : SubIndex	名称	数值
0x8000:01	Enable RTS/CTS	FALSE
0x8000:02	Enable XON/XOFF supported tx data	FALSE
0x8000:03	Enable XON/XOFF supported rx data	FALSE
0x8000:04	Enable send FIFO data continuous	FALSE
0x8000:05	Enable transfer rate optimization	TRUE
0x8000:15	Dataframe	3 (8N1)
0x8000:1A	Rx buffer full notification	参见调试对象 [▶ 165]。

### 操作

在 LIN 模式下，控制字和状态字的应用保持不变。

在 LIN 模式下，EL6001 会自动在用户数据之前加上“Break field”和“Sync byte”字段。在接收过程中，这两个字段会被自动删除。

过程映像的使用方法如下：

过程数据	内容
Ctrl.Output length	用户数据字节数 (n) + 2
Data Out 0	受保护的 ID 字段 (PID = 受保护的标识符字段)
Data Out 1	数据字节 1
Data Out n	数据字节 n
Data Out n+1	校验和

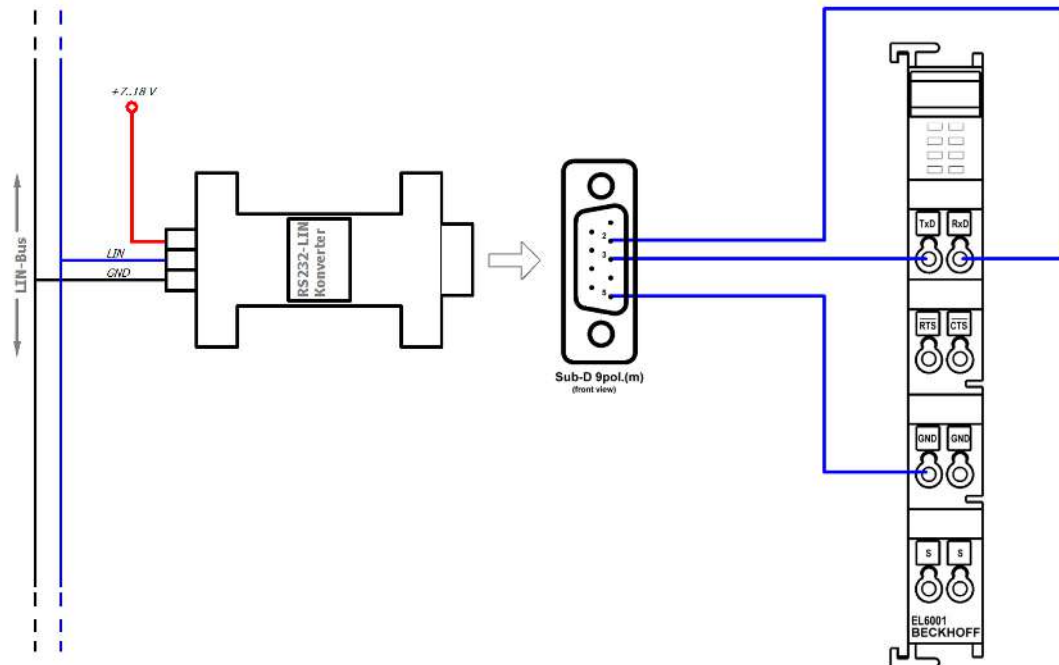
### EL6001 的 LIN 示例

下面的 LIN 通信示例旨在说明通过 0xB000 CoE “Command” 对象启用 LIN 扩展功能，并将 RS-232 物理转换为 LIN 总线，PLC 控制器就可以通过 EL6001 端子模块加入 LIN 总线。

#### 基于 UART-LIN 转换器的硬件连接

物理层的实施是 LIN 通信的基本前提，包括提供相应信号电平为 0V / 12V 的单线总线。最好使用 RS232-LIN 转换器，该转换器与 sub-D 连接器相连，另一端提供三个极，分别用于接地、供电以及与单线 LIN 总线的电气连接。

如图所示，使用 RS232-LIN 转换器需要将 sub-D 9 针连接器连接至 EL6001 端子模块：



附图 148: 连接至 EL6001 端子模块的 RS232-LIN sub-D 连接器

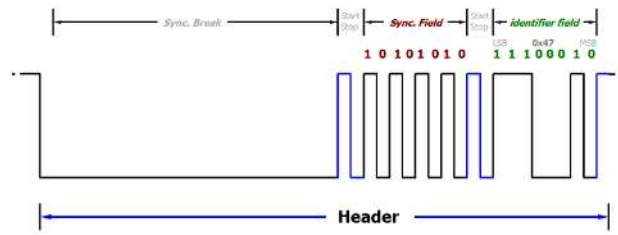
EL6001 端子模块与 sub-D 9 针连接器之间的电气连接：

- RxD → 引脚 2
- TxD → 引脚 3
- GND → 引脚 5

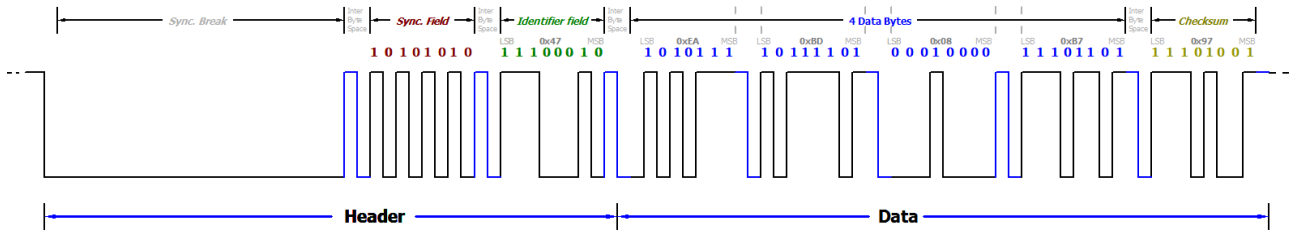
#### ST语言编程

LIN 通信可以用 PLC 的一段“主站程序”来实现，即 PLC “主站程序”就像 LIN 总线一样发送“无条件帧”，且在 LIN 总线上只发出已知从站节点的 PID（协议标识符），象通常那样从目标从站请求数据。调试章节中包含一个相关的编程示例 [► 152]。

在本例中，从站的 ID 是 0x07；因此，主站在总线上发送 ID 0x07 和算得的奇偶校验，生成 PID 字段 0x47。LIN 总线上的信息如下：

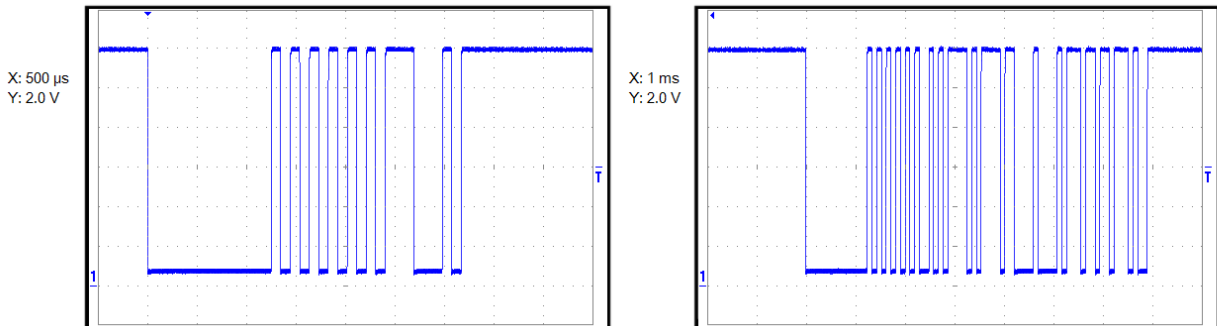


附图 149: LIN 帧示例: 主站向 ID 为 0x07 的节点查询



附图 150: LIN 帧示例: 含数据 0xEA、0xBD、0x08、0xB7 的 ID 0x07 + 校验和 0x97

ID 为 0x07 的 LIN 帧示波器记录:



附图 151: 左图: LIN 总线上的查询, PID 0x47, 右图: 具有相同 PID 和数据 (包括校验和) 的 LIN 帧

## 6.8 示例程序

### ● 使用示例程序

本文件包含我们产品在某些应用领域的应用案例。这里提供的应用说明是基于我们产品的典型特征，仅作为示例。本文件中的说明明确地不涉及具体的应用。因此，客户有责任评估和决定产品是否适用于特定的应用。我们对本文件所包含的源代码的完整性和正确性不承担任何责任。我们保留在任何时候修改本文件内容的权利，对错误和遗漏的信息不承担任何责任。

### 6.8.1 示例程序 1



[https://infosys.beckhoff.com/content/1033/el600x\\_el602x/Resources/1719190795/.zip](https://infosys.beckhoff.com/content/1033/el600x_el602x/Resources/1719190795/.zip)

#### 连接串行条形码扫描仪

在本例中，条形码阅读器与 EL6001 相连。阅读器将读取字符，直至收到 ASCII 字符 0x0D (13<sub>dec</sub>, CR)。

数据：

- 执行串行通信的快速任务：周期时间 1 ms
- 标准 PLC 任务：周期时间 10 ms
- 通道 1 上的条形码扫描仪
- 需要 TwinCAT 2.11
- 需要TwinCAT串口通讯功能插件

有关串行通信库使用方法的详细说明，请访问 Beckhoff Information System。

Beckhoff Information System -> TwinCAT -> TwinCAT PLC -> TwinCAT libraries for PC-based systems  
-> TwinCAT PLC Library: Serial communication

#### 启动示例程序

应用示例已在测试配置下经过测试，并进行了相应的说明。

在设置实际应用程序时可能会出现某些偏差。

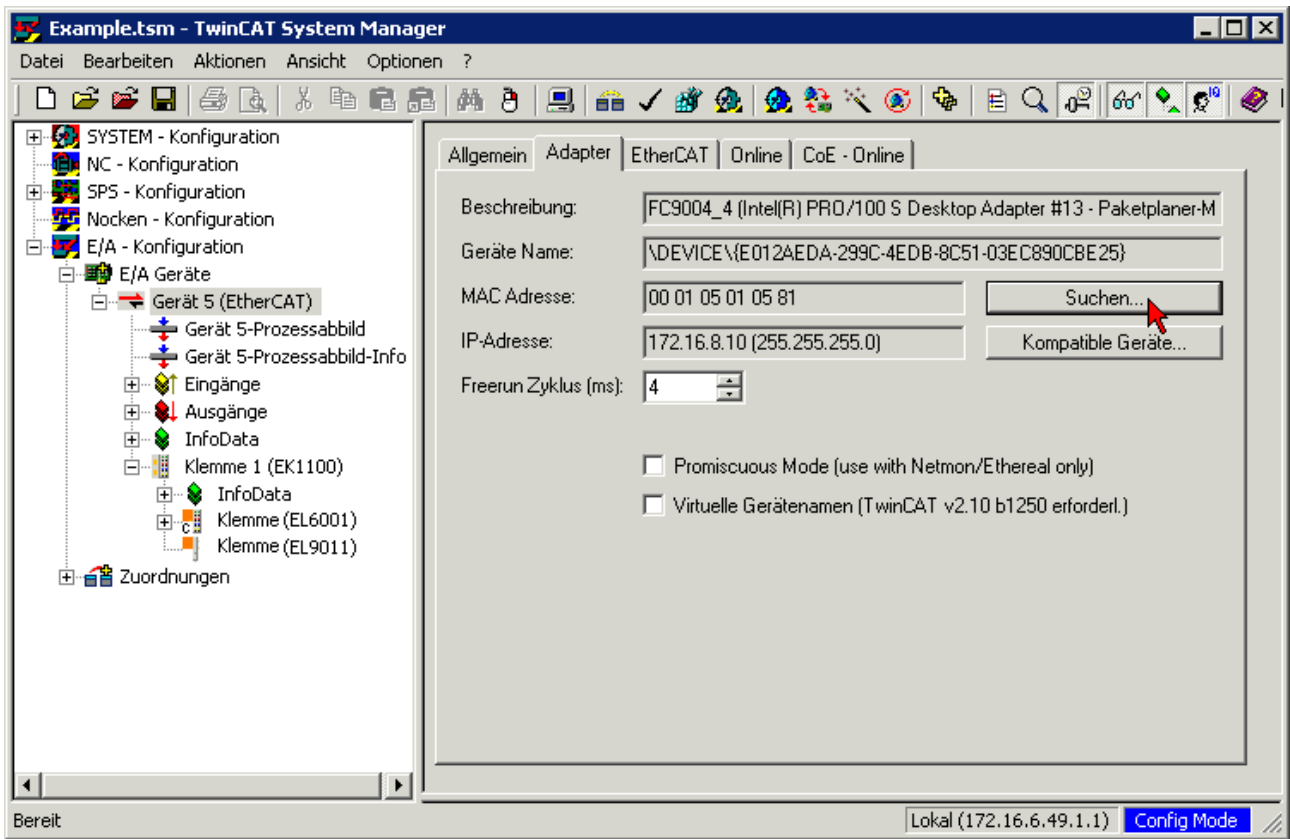
该测试配置使用以下硬件和软件：

- TwinCAT 主站 PC，配备 Windows XP Professional SP 3，TwinCAT 2.11 版本 (Build 1528) 和 INTEL PRO/100 VE 以太网适配器
- 倍福 EK1100 EtherCAT 耦合器，EL6001 端子模块
- 带 sub-D 9 针连接器的串行条形码扫描仪

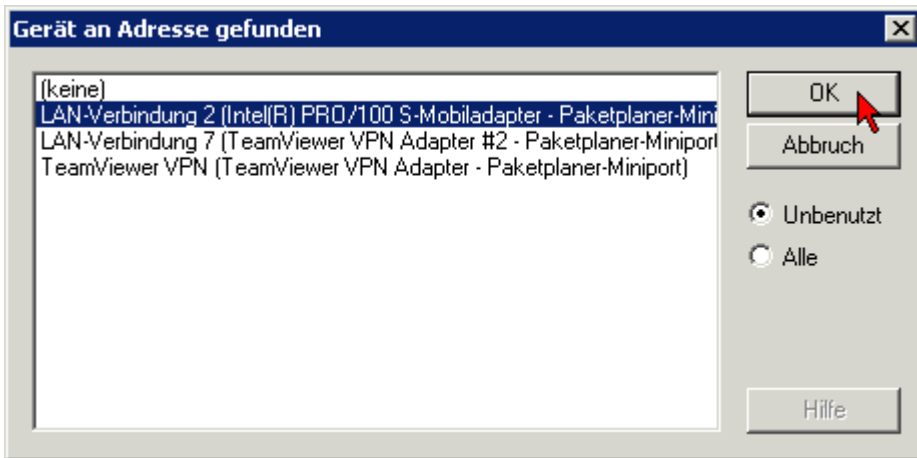
#### 程序启动步骤

- 单击“Download”按钮后，将压缩文件保存在本地硬盘上，并将 \*.TSM (配置) 和 \*.PRO (PLC 程序) 文件解压到临时工作文件夹中
- 运行 \*.TSM 文件和 \*.PRO 文件；将会打开 TwinCAT System Manager 和 TwinCAT PLC
- 按照图 1 连接硬件，并将 PC 的以太网适配器连接至 EtherCAT 耦合器 (更多相关信息请参见相应的耦合器手册)
- 在“System configuration”、“I/O configuration”、“I/O devices”、“Device (EtherCAT)”下选择本地以太网适配器 (带 real-time 实时驱动程序)；在“Adapter”选项卡上选择“Search...”，然后选择合适的适配器并确认 (见图)





附图 152: 搜索以太网适配器



附图 153: 选择并确认以太网适配器

激活配置并确认:

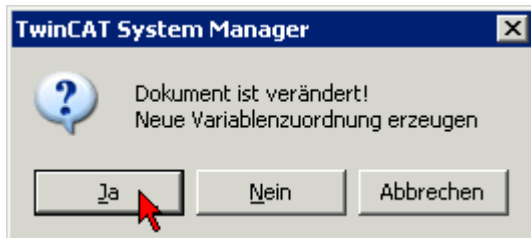


附图 154: 激活配置

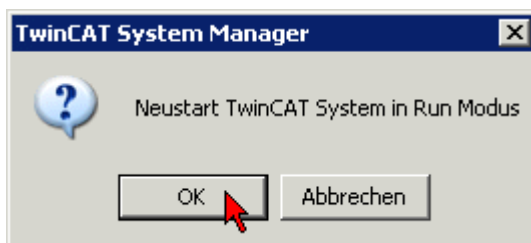


附图 155: 确认已激活配置

- 确认新变量映射后，在 RUN 模式下重启：

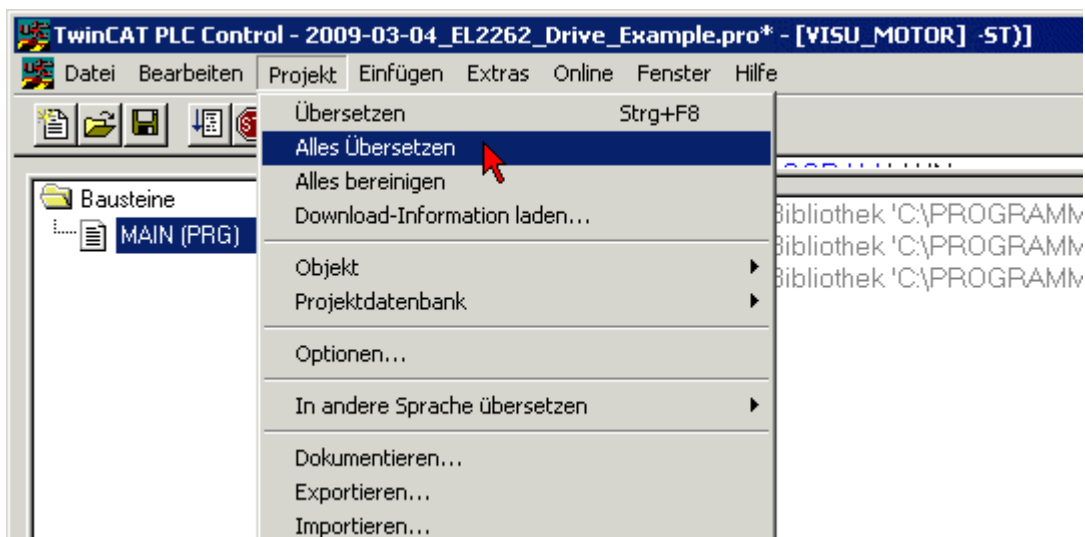


附图 156: 生成变量映射



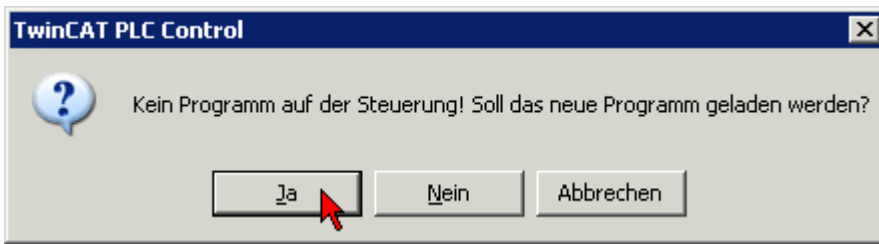
附图 157: 在 RUN 模式下重启 TwinCAT

- 在 TwinCAT PLC 的“Project”菜单下，选择“Rebuild all”以编译项目：



附图 158: 编译项目

- 在 TwinCAT PLC 中：使用“F11”按钮登录，确认加载程序，使用“F5”按钮运行程序：



附图 159: 确认程序启动

- 接收到字符 0x13 后，条形码储存在“LastBarcode”中

<pre> 0001 Receive( 0002     Suffix:= '\$0D',           (*RETURN*) 0003     Timeout:= T#1s, 0004     ReceivedString:= Barcode, 0005     RxBuffer:= RxBuffer, 0006     StringReceived=&gt; BarcodeReceived, 0007     Busy=&gt; ReceiveBusy, 0008     Error=&gt; ReceiveErrorID, 0009     RxTimeout=&gt; ReceiveTimeout); 0010 0011 NewBarcode(CLK := Receive.StringReceived); 0012 IF NewBarcode.Q THEN 0013     LastBarcode := Barcode; 0014 END_IF                 </pre>	<pre> Receive.StringRec... = FALSE   Receive.Busy = FALSE Barcode = '544900017888\$R' BarcodeReceived = FALSE ReceiveBusy = FALSE ReceiveErrorID = COMERRO... ReceiveTimeout = FALSE Receive.StringRec... = FALSE NewBarcode.Q = FALSE LastBarcode = '544900017888...' Barcode = '544900017888\$R'                 </pre>
--	---

附图 160: 接收到的条形码

## 6.8.2 示例程序 2

[https://infosys.beckhoff.com/content/1033/el600x\\_el602x/Resources/1719192971/.zip](https://infosys.beckhoff.com/content/1033/el600x_el602x/Resources/1719192971/.zip)

### 读取和解析时间报文

本示例展示了在 PLC 中处理和解析各种串行时间协议的方法。为此，将介绍 IEC61131-PLC 模块，该模块可以解析到达 PLC 的位流，并在必要时提取时间/地点信息。这些信息可用于同步控制器或记录动作等。

在本示例中，假设数据由倍福 EL/KL60xx 串行通讯端子模块通过 22 字节接口传送。

### 背景信息

在海事应用中通过串行总线传输时间和地点信息：通常一个活动的信息源会通过 RS232/RS485、USB 或 Ethernet 的方式按周期或在事件触发时将时间和地点信息发送给下级监听器。世界上有众多用于发送时间和地点信息报文格式，这些格式也被称为“句子”。这种报文由 n 个字节的数据组成，具有以下特征：

- 起始和结束字符 STX、ETX 用于报文识别，可能有多个结束字符
- 明确定义且固定的报文长度
- 明确定义的结构
- 校验和（非必要）
- 类型标识，必要时也会包含在句子中

各种机构和公司为特定用途研发了开放的或专属的类型格式。因此，本示例将介绍两种实现样例，它们可以很容易进行调整以适用于其它特定协议。示例中的报文格式为 Meinberg 标准格式和 NMEA0183 v2.3 型 RMC 格式。

## 关于时间和位置来源的小知识

全球定位系统（GPS）或无线电控制的时钟网关以串行通信方式发送数据。这些设备接收相应的时间信号（GPS 通过卫星，或无线电通过长波控制时钟），并将其转换为串行有线传输，例如使用 8N1 的 RS232。网关通常包含一个本地时钟源，以便在参考信号（GPS、无线电发射器）发生瞬时故障时，能够在一定时间内继续发布可靠的时间信息。在中欧，可以使用德国 DCF77 发射器。

如有必要，还可以使用来自参考信号的其他信息：

- GPS：位置信息（W/N 和高度），即将发生的时钟变化
- DCF77：天气信息、重大事件

## 下级时间从站的同步

一般来说，应根据时间网关调整下一级从站的时间，即这些从站的时间应进行同步。为此，请注意以下要点：

- Offset（偏移）：从站自身时钟与网关参考时间的绝对偏差 - 例如，如果知道时间信息何时生效，则可以非周期性和串行方式传输该信息。在本示例中，可以通过串行传输路径获得这些偏移信息。
- frequency ratio（频率比）：从网关到从站之间的高精度周期信号可对漂移过程进行补偿，并可标明上述绝对偏移对应的时间。广泛使用的 PPS（每秒脉冲数）就是这种信号的一个例子。

如果网关周期性地将从站报文发送到总线上，该报文第一个 bit 的时间通常被解析为 PPS 信号。本例中的程序并不处理这些信息，而只计算绝对时间信息。

## 时间格式

符合 NMEA0183 标准的时间报文应用广泛。注意：

- NMEA0183 标准中目前（截至 2009 年）有 NMEA0183 1.5 至 4.00 共 8 个版本 - 不同版本之间的报文结构可能有所不同。
- 仅 NMEA0183 v2.30 就定义了 70 种不同的格式；此外，设备制造商还可以自行创建格式。
- 报文又被叫做“句子”。
- 开头的 TalkerID（2 个字符）和 TypeID（3 个字符）定义所使用的句子类型。
- 计算报文的校验和。
- 相关信息请在线访问 [www.nmea.org](http://www.nmea.org) 或其它网站。

此外，还有许多专有格式，如 Meinberg 标准、Siemens SINEC H1 和 SAT 1703，或军用格式，如 IRIG 代码（美国）。

## 使用示例程序

PLC 项目包含 2 个功能块（FB），必须专门与 EL/KL600x 连接。从端子模块接收其收集到的字节，并尽可能解析其内容。功能块涵盖：

- Meinberg 标准格式
- NMEA0183 v2.3 型 RMC 格式

对于其它时间格式，可以基于已知的报文结构创建专门的解释功能块；相关信息请联系设备制造商。

示例中使用的硬件：EK1100 和 EL6001（也包括 EL600x、KL600x）

待测功能块必须将其过程数据连接到 System Manager 中的端子模块（22 字节过程映像和控制/状态字）。

示例中连接的是 NMEA 功能块。端子模块的设置波特率必须匹配于发送数据的设备。

## 参考文献

- [www.beuth.de](http://www.beuth.de), IEC61162: 基于 NMEA2000
- <http://gpsinformation.net> 私域，通过 NMEA，多种格式：<http://www.gpsinformation.org/dale/nmea.htm>  
<http://www.nmea.de/nmea0183datensaetze.html>
- <http://www.meinberg.de/german/info/irig.htm>, IRIG 编码

## 6.8.3 示例程序 3 (LIN)

下载 (TwinCAT 3 示例程序):

 [https://infosys.beckhoff.com/content/1033/el600x\\_el602x/Resources/1805853195/.zip](https://infosys.beckhoff.com/content/1033/el600x_el602x/Resources/1805853195/.zip)

### LIN 主站示例程序的全局变量

```
VAR_GLOBAL
nSetBaudrate : UINT := 10417; // Für Rx-Delay-Berechnung
(* I/O variables for EL6001 terminal acting as Master*)
COMin_EL6001_MASTER AT %I* : EL6inData22B; (* linked to the EL6001 in the TwinCAT System Manager *)
COMout_EL6001_MASTER AT %Q* : EL6outData22B; (* linked to the EL6001 in the TwinCAT System Manager *)
RxBuffer_MASTER : ComBuffer; (* Receive data buffer; used with all receive function blocks *)
TxBuffer_MASTER : ComBuffer; (* Transmit data buffer; used with all receive function blocks *)
END_VAR
```

### LIN 主-从示例程序的数据类型

```
TYPE tDataFrame : ARRAY[0..8] OF BYTE; END_TYPE // Datentyp für ein LIN-Frame
```

用于添加数据长度信息和奇偶校验位的功能:

变量声明部分:

```
FUNCTION F_ADD_LIN_NODE_PARITY : BYTE
VAR_INPUT
nNodeID:BYTE; // Inputvariable: node-Id
nReqLen:BYTE; // Inputvariable: length-identification: 2,4,8 Byte-Frame
END_VAR
VAR
bParity0:BYTE; // Internal intermediate value parity 0
bParity1:BYTE; // Internal intermediate value parity 1
nPrepId:BYTE; // Intermediate value for PID
END_VAR
```

执行代码部分:

```
nPrepId := nNodeID OR SHL(nReqLen,4);

bParity0 :=
(nNodeID AND 2#0001)
XOR (SHR((nNodeID AND 2#0010), 1))
XOR (SHR((nNodeID AND 2#0100), 2))
XOR (SHR((nNodeID AND 2#0001_0000), 4));

bParity1 := 16#01 AND (NOT(
SHR((nNodeID AND 2#0010), 1)
XOR (SHR((nNodeID AND 2#1000),3))
XOR (SHR((nNodeID AND 2#0001_0000), 4))
XOR (SHR((nNodeID AND 2#0010_0000), 5))));

F_ADD_LIN_NODE_PARITY := nPrepId OR SHL(bParity0,6) OR SHL(bParity1,7);
```

用于计算 checksum 校验和的 Function (传统方法)

变量声明部分:

```
FUNCTION F_CALC_LIN_CHKSUM : BYTE
VAR_INPUT
pData:POINTER TO ARRAY[0..10] OF BYTE; // Pointer to source datafield
nLen:BYTE; // Number of bytes used for calculating the checksum
END_VAR
VAR
i: BYTE; // Counter variable
wResult:WORD; // Resulting output value
END_VAR
```

执行代码部分:

```
wResult := BYTE_TO_WORD(pData^[0]);
FOR i := 1 TO (nLen-1) DO
wResult := wResult + BYTE_TO_WORD(pData^[i]);
IF wResult > 255 THEN
wResult := wResult - 255;
END_IF
END_FOR
F_CALC_LIN_CHKSUM := WORD_TO_BYTE(NOT(wResult));
```

主程序应由独立的 Task 任务来调用。在此，每 200 ms 查询一次 LIN 总线中 ID 为 0x07 的节点。

变量声明部分：

```
PROGRAM EL6001_MASTER
VAR
Timer: TON; // Timer for periodical requests by the master
Send: SendData; // Functionblock of TC2_SerialCom
SendBusy: BOOL; // Flag copy of SendData.Busy
SendErrorID: ComError_t; // Flag-Copy of Error-ID
aDataTX:tDataFrame; // Data frame being send
Receive: ReceiveData; // Functionblock of TC2_SerialCom
LastReceivedDataBytes: tDataFrame; // Copy (Latch) of received data
DataReceived: BOOL; // Flag copy of receive confirmation
ReceiveBusy: BOOL; // Flag copy of reception not ready
ReceiveError: BOOL; // Flag copy of receive error
ReceiveErrorID: ComError_t; // Error-ID copy
ReceiveTimeout: BOOL; // Flag copy of receive-timeout
ReceiveCounter: UDINT := 0; // Number of received frames
aDataRX:tDataFrame; // Receiving data frame buffer
nDataLen:BYTE := 4; // Fixed data length
nState:BYTE := 0; // Initial state (start)
bNodeId_SL1: BYTE := 16#07; // ID of slave node
bReqLen_SL1: BYTE := 0; // Optional entry for 4 Data bytes Value 2
nRxChecksum: BYTE; // Storage of received checksum
nCalcChecksum:BYTE; // Storage of calculated checksum
T_ReceiveDelay:TIME; // Storage of calculated delay time
END_VAR
```

执行代码部分：

```
(*=====
Receive data
*)
CASE nState OF
0:
Timer(IN:=TRUE, PT:=T#0.5S); // Call timer for periodical master requests
IF Timer.Q THEN
// Put ID into Tranceive data:
aDataTX[0] := F_ADD_LIN_NODE_PARITY(bNodeId_SL1, bReqLen_SL1);
LastReceivedDataBytes[0] := aDataTX[0];
// Send request to Slave 1 (get Data)
Send(pSendData:= ADR(aDataTX), Length:= 1,
TXbuffer:= TxBuffer_MASTER,
Busy => SendBusy, Error => SendErrorID);
Timer(IN:=FALSE); (* reset timer *)
IF NOT SendBusy THEN // Wait until sending ends
nState := nState + 1;
END_IF
END_IF
1:
// Delaytime by 1/Tbaud * Number of Bytes * (8 Databits + 2 Bit:start-Stop) * 1000ms
T_ReceiveDelay := REAL TO TIME((1/DINT_TO_REAL(nSetBaudrate)) * 33 * 1000); // .. for 1 Byte
Timer(IN:=TRUE, PT:=T_ReceiveDelay);
IF Timer.Q THEN
// Wait until ID is send
nState := nState + 1;
Timer(IN:=FALSE); (* reset timer *)
END_IF
2:
Receive(
pReceiveData:= ADR(aDataRX),
SizeReceiveData:= (nDataLen + 1),
RXbuffer:= RxBuffer_MASTER,
Timeout:= T#1S,
DataReceived=> DataReceived,
busy=> ReceiveBusy,
Error=> ReceiveErrorID,
RxTimeout=> ReceiveTimeout );
IF DataReceived THEN
// DataReceived := FALSE;
ReceiveCounter := ReceiveCounter + 1;
IF NOT ReceiveBusy THEN
// Compare checksum
nRxChecksum := aDataRX[nDataLen];
nCalcChecksum := F_CALC_LIN_CHKSUM(pData := ADR(aDataRX), nLen := nDataLen);
IF(nRxChecksum = nCalcChecksum) THEN
// Response received - clear databuffer:
memset(ADR(LastReceivedDataBytes[1]), 0, (SIZEOF(aDataRX)-1));
// Take-over data when checksum OK:
```

```

memcpy(ADR>LastReceivedDataBytes[1]), ADR(aDataRX), (nDataLen +1));
END_IF
nState := 0;
END_IF
ELSE
Timer(IN:=TRUE, PT:=T#0.1S); // Receive-Timeout 100 ms: no data
IF Timer.Q THEN
nState := 0;
END_IF
END_IF
END_CASE
(*=====
*)

```

### 程序快速任务 / RS232 后台通信

在本例中，应创建第三个Task，其任务周期应尽可能短，专门处理与 EL6001 端子模块（作为主站）之间的后台通信。

变量声明部分：

```

PROGRAM FAST
VAR
(* background communication with the EL6001 as Master device *)
COMportControl_MASTER: SerialLineControl;
COMportControlError_MASTER: BOOL;
COMportControlErrorID_MASTER: ComError_t;
END_VAR

```

执行代码部分：

```

COMportControl_MASTER(
Mode:= SERIALLINEMODE_EL6_22B,
pComIn:= ADR(COMin_EL6001_MASTER), (* I/O data; see global variables *)
pComOut:= ADR(COMout_EL6001_MASTER), (* I/O data; see global variables *)
SizeComIn:= SIZEOF(COMin_EL6001_MASTER), (* I/O data; see global variables *)
TxBuffer:= TxBuffer_MASTER, (* transmit buffer; see global variables *)
RxBuffer:= RxBuffer_MASTER, (* receive buffer; see global variables *)
Error=> COMportControlError_MASTER, (* indicating an error *)
ErrorID=> COMportControlErrorID_MASTER); (* contains the error-ID *)

```

## 7 EL6001、EL6021 的 CoE 对象概览

### 7.1 控制字 Control 和状态字 Status

#### 控制字 Control

控制字 Control Word (CW) 位于输出过程映像中，并从控制器传输到端子模块。

位	CW. 15	CW. 14	CW. 13	CW. 12	CW. 11	CW. 10	CW. 9	CW. 8	CW. 7	CW. 6	CW. 5	CW. 4	CW. 3	CW. 2	CW. 1	CW. 0
名称	OL7	OL6	OL5	OL4	OL3	OL2	OL1	OL0	-	OL2*	OL1*	OL0*	SC	IR	RA	TR

CW. 15 至 CW. 8 位仅在使用大型过程映像时显示。

如果使用小型或中型过程映像，仅 CW. 7 至 CW. 0 位显示为控制字节！从控制器发送到端子模块的等待输出字节数。由 OL2\* ... OL0\* 位显示。

#### 说明表

位	名称	描述
CW. 15 ... CW. 8	OL7 ... OL0 (OutLengt)	$1_{dec} \dots 22_{dec}$ 如果使用大型过程映像：就代表从控制器发送到端子模块的等待输出字节数。 - 如果使用小/中型过程映像：不显示
CW. 7	-	$0_{bin}$ 保留
CW. 6 ... CW. 4	OL2* ... OL0* (OutLengt*)	0 如果使用大型过程映像：保留 1 ... 6 如果使用中型过程映像：就代表从控制器发送到端子模块的等待输出字节数。 1 ... 4 如果使用小型过程映像：就代表从控制器发送到端子模块的等待输出字节数。
CW. 3	SC (SendContinuous)	上升 从 FIFO 连续发送数据。 来自控制器的数据先填满端子模块的发送缓冲区（最多 128 字节）。缓冲区内内容随 SC 位上升沿发送。端子模块通过设置 SW. 2 位确认向控制器发送数据。SW. 2 与 CW. 3 一起取消。
CW. 2	IR (InitRequest)	$1_{bin}$ 控制器请求端子模块进行初始化。发送和接收功能被阻止，FIFO 指针重置，接口使用相关对象的值初始化（波特率 4073 [▶_163]，数据帧 4074 [▶_163]，功能位 4075 [▶_164]）。端子模块通过 SW. 2 [▶_161] 位 (IA) 确认初始化完成。 $0_{bin}$ 控制器再次请求端子模块准备进行串行数据交换。
CW. 1	RA (ReceiveAccepted)	toggle 控制器通过改变该位的状态确认已收到数据。然后，端子模块才会向控制器发送新数据。
CW. 0	TR (TransmitRequest)	toggle 控制器通过改变该位的状态通知端子模块，DataOut 数组包含指示字节数的 OL 位。端子模块通过改变 SW. 0 [▶_161] (TA) 的状态确认收到状态字节中的数据。然后，端子模块才会接收从控制器发来的新数据。



状态字

状态字 (SW) 位于输入过程映像中，并从端子模块传输到控制器。

位	SW. 15	SW. 14	SW. 13	SW. 12	SW. 11	SW. 10	SW. 9	SW. 8	SW. 7	SW. 6	SW. 5	SW. 4	SW. 3	SW. 2	SW. 1	SW. 0
名称 (中小型过程映像)	-	-	-	-	-	-	-	-	-	IL2*	IL1*	ILO*	BUF_F	IA	RR	TA
名称 (大型过程映像)	IL7	IL6	IL5	IL4	IL3	IL2	IL1	ILO	-	OVERRUN ERR	FRAMING ERR	PARITY ERR	BUF_F	IA	RR	TA

如果使用小型或中型过程映像，仅 CW. 7 至 CW. 0 位显示为状态字节！等待从端子模块发送到控制器的输入字节数由 IL2\* ... ILO\* 位显示。

说明表

位	名称	描述
SW. 15 ... SW. 8	IL7 ... ILO (InLenght*)	1 <sub>dec</sub> ... 22 <sub>dec</sub> - 如果使用大型过程映像：就代表等待从端子模块发送到控制器的输入字节数。 如果使用小/中型过程映像：不显示
SW. 7	-	0 <sub>bin</sub> 保留
SW. 6	IL2* (InLenght*)  OVERRUN ERR	0 ... 1 如果使用大型过程映像：发生超限错误。发生错误的数项不会存入端子模块的接收 FIFO，而是直接丢弃。
		1 ... 6 如果使用中型过程映像：就代表等待从端子模块发送到控制器的输入字节数。
		1 ... 4 如果使用小型过程映像：就代表等待从端子模块发送到控制器的输入字节数。
SW. 5	IL1* (InLenght*)  FRAMING ERR	0 ... 1 如果使用大型过程映像：发生帧错误。发生错误的数项不会存入端子模块的接收 FIFO，而是直接丢弃。
		1 ... 6 如果使用中型过程映像：就代表等待从端子模块发送到控制器的输入字节数。
		1 ... 4 如果使用小型过程映像：就代表等待从端子模块发送到控制器的输入字节数。
SW. 4	ILO* (InLenght*)  PARITY ERR	0 ... 1 如果使用大型过程映像：发生奇偶校验错误。发生错误的数项不会存入端子模块的接收 FIFO，而是直接丢弃。
		1 ... 6 如果使用中型过程映像：就代表等待从端子模块发送到控制器的输入字节数。
		1 ... 4 如果使用小型过程映像：就代表等待从端子模块发送到控制器的输入字节数。
SW. 3	BUF_F	1 <sub>bin</sub> 接收 FIFO 已满。所有后续到来的输入数据都将丢失！
SW. 2	IA (InitAccepted- Bit)	1 <sub>bin</sub> 端子模块已完成初始化。
		0 <sub>bin</sub> 端子模块再次准备就绪，可进行串行数据交换。
SW. 1	RR (ReceiveRequest, 接收请求)	0 和 1 交替切换 端子模块通过改变该位的状态通知控制器，DataIn 字节包含通过 IL 位指示的字节数。控制器必须通过改变 CW. 1 [▶_160] (RA) 的状态来确认已收到控制字节中的数据。然后，端子模块才会向控制器发送新数据。
SW. 0	TA (TransmitAccepted)	0 和 1 交替切换 端子模块通过改变该位的状态确认已收到数据。然后，端子模块才会接收从控制器发来的新数据。

## 数据传输示例

这些示例使用的是大型过程映像。

- 初始化:

控制字		状态字		注释
CW. 15 ... CW. 8	CW. 7 ... CW. 0	SW. 15 ... SW. 8	SW. 7 ... SW. 0	
XXXX XXXX <sub>bin</sub>	XXXX XXXX <sub>bin</sub>	XXXX XXXX <sub>bin</sub>	XXXX XXXX <sub>bin</sub>	开始数据传输
0000 0000 <sub>bin</sub>	0000 0100 <sub>bin</sub>	0000 0000 <sub>bin</sub>	0000 0000 <sub>bin</sub>	控制器请求端子模块初始化
0000 0000 <sub>bin</sub>	0000 0100 <sub>bin</sub>	0000 0000 <sub>bin</sub>	0000 0100 <sub>bin</sub>	命令已执行: 端子模块初始化完成
0000 0000 <sub>bin</sub>	0000 0000 <sub>bin</sub>	0000 0000 <sub>bin</sub>	0000 0100 <sub>bin</sub>	控制器请求端子模块准备进行串行数据交换
0000 0000 <sub>bin</sub>	0000 0000 <sub>bin</sub>	0000 0000 <sub>bin</sub>	0000 0000 <sub>bin</sub>	命令已执行: 端子模块已准备就绪, 可进行串行数据交换

- 从控制器向端子模块发送数据:

控制字		状态字		注释
CW. 15 ... CW. 8	CW. 7 ... CW. 0	SW. 15 ... SW. 8	SW. 7 ... SW. 0	
0000 0000 <sub>bin</sub>	0000 0000 <sub>bin</sub>	XXXX XXXX <sub>bin</sub>	0000 x0x0 <sub>bin</sub>	开始数据传输
0000 0010 <sub>bin</sub>	0000 0001 <sub>bin</sub>	XXXX XXXX <sub>bin</sub>	0000 x0x0 <sub>bin</sub>	控制器请求端子模块发送 2 字节
0000 0010 <sub>bin</sub>	0000 0001 <sub>bin</sub>	XXXX XXXX <sub>bin</sub>	0000 x0x1 <sub>bin</sub>	命令已执行: 端子模块已将 2 字节加载至传输 FIFO
0001 0000 <sub>bin</sub>	0000 0000 <sub>bin</sub>	XXXX XXXX <sub>bin</sub>	0000 x0x1 <sub>bin</sub>	控制器请求端子模块发送 16 字节
0001 0000 <sub>bin</sub>	0000 0000 <sub>bin</sub>	XXXX XXXX <sub>bin</sub>	0000 x0x0 <sub>bin</sub>	命令已执行: 端子模块已将 16 个字节加载至传输 FIFO

- 从端子模块向控制器发送数据:

控制字		状态字		注释
CW. 15 ... CW. 8	CW. 7 ... CW. 0	SW. 15 ... SW. 8	SW. 7 ... SW. 0	
XXXX XXXX <sub>bin</sub>	0000 000x <sub>bin</sub>	0000 0000 <sub>bin</sub>	0000 000x <sub>bin</sub>	开始数据传输
XXXX XXXX <sub>bin</sub>	0000 000x <sub>bin</sub>	0000 0011 <sub>bin</sub>	0000 001x <sub>bin</sub>	端子模块请求从控制器传输 3 字节
XXXX XXXX <sub>bin</sub>	0000 001x <sub>bin</sub>	0000 0011 <sub>bin</sub>	0000 001x <sub>bin</sub>	确认: 控制器从接收 FIFO 收到 3 个字节
XXXX XXXX <sub>bin</sub>	0000 001x <sub>bin</sub>	0001 0110 <sub>bin</sub>	0000 000x <sub>bin</sub>	端子模块请求从控制器传输 22 字节
XXXX XXXX <sub>bin</sub>	0000 000x <sub>bin</sub>	0001 0110 <sub>bin</sub>	0000 000x <sub>bin</sub>	确认: 控制器从接收 FIFO 收到 22 个字节

## 7.2 对象描述和参数化

### ● EtherCAT XML 设备描述



此描述与 EtherCAT XML 设备描述中的 CoE 对象相匹配。建议从倍福网站的下载区下载最新 XML 文件，并按照安装说明进行安装。

### ● 参数化



端子模块通过“CoE - Online”选项卡 [▶ 118] 进行参数化（双击相应对象，见下文）。

### 简介

CoE 概述包含针对不同应用的对象：

- 调试过程中配置所需的对象
- 用于指示内部设置的对象（可能是固定的）
- 描述文件定义的对象，表示输入和输出的状态显示（硬件版本 03 [▶ 202] 及以上）

以下章节首先介绍的是正常运行所需的对象，然后完整描述其它对象。

### 7.2.1 用于调试的对象

#### Index 0x1011 Restore default parameters

Index (Hex)	名称	含义	数据类型	标志	默认值
1011:0	恢复默认参数 [▶ 215]	恢复默认参数	UINT8	RO	0x01 (1 <sub>dec</sub> )
1011:01	SubIndex 0x001	如果此对象在设置值对话框中被设置为“0x64616F6C”，所有备份对象都被重置为它们的出厂状态。	UINT32	RW	0x00000000 (0 <sub>dec</sub> )

### ● 新项目使用0x8000进行COM通讯设置



对于新的项目，建议使用0x8000中的以下COM设置列表项，而不是0x407\_设置对象。

#### Index 0x4073 Baud rate

Index (Hex)	名称	含义	数据类型	标志	默认值
4073:0	Baud rate	详细信息请参阅通信功能 [▶ 147] 章节	UINT16	RW	0x0006 (6 <sub>dec</sub> )

#### Index 0x4074 Data frame

Index (Hex)	名称	含义	数据类型	标志	默认值
4074:0	Data frame	详细信息请参阅通信功能 [▶ 147] 章节	UINT16	RW	0x0003 (3 <sub>dec</sub> )

## Index 0x4075 Feature bits

Index (Hex)	名称	含义	数据类型	标志	默认值
4075:0	Feature bits	该对象的长度	UINT8	RO	0x06 (6 <sub>dec</sub> )
4075:01	<b>EL6001:</b> Enable RTS/CTS	0: 未启用 RTS/CTS 1: 已启用 RTS/CTS	BOOLEAN	RW	0x00 (0 <sub>dec</sub> )
	<b>EL6021:</b> Enable half duplex	0: 全双工模式 1: 半双工模式			
4075:02	Enable XON/XOFF supported tx data	0: 不支持 XON/XOFF 发送数据 1: 支持 XON/XOFF 发送数据	BOOLEAN	RW	0x00 (0 <sub>dec</sub> )
4075:03	Enable XON/XOFF supported rx data	0: 不支持 XON/XOFF 接收数据 1: 支持 XON/XOFF 接收数据	BOOLEAN	RW	0x00 (0 <sub>dec</sub> )
4075:04	<b>EL6001:</b> Enable send FIFO data continuous	0: 不从 FIFO 连续发送数据 1: 来自控制器的数据先填满端子模块的发送缓冲区 (最多 128 字节)。缓冲区内容随控制字中 CW.3 [▮_160] 位的上升沿发送。端子模块通过设置状态字中的 SW.2 [▮_161] 位确认向控制器发送数据。SW.2 通过 CW.3 重置。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
	<b>EL6021:</b> Enable point to point connection (RS422)	0: 禁用点对点连接 1: 启用点对点连接			
4075:05	<b>EL6001:</b> Enable transfer rate optimization	0: 关闭传输速率优化 1: 开启传输速率优化: 如果出现以下情况, 输入缓冲区的内容会自动传输到过程映像中: <ul style="list-style-type: none"> <li>在收到数据后的大约 16 位时间内 (即接收 2 个字节所需的时间), 没有再收到其它字节;</li> <li>输入缓冲区已满</li> </ul>	BOOLEAN	RO	0x01 (1 <sub>dec</sub> )
	<b>EL6021:</b> Enable send FIFO data continuous	0: 不从 FIFO 连续发送数据 1: 来自控制器的数据先填满端子模块的发送缓冲区 (最多 128 字节)。缓冲区内容随控制字中 CW.3 [▮_160] 位的上升沿发送。端子模块通过设置状态字中的 SW.2 [▮_161] 位确认向控制器发送数据。SW.2 通过 CW.3 重置。			
4075:06	<b>仅限 EL6021:</b> Enable transfer rate optimization	0: 关闭传输速率优化 1: 开启传输速率优化: 如果出现以下情况, 输入缓冲区的内容会自动传输到过程映像中: <ul style="list-style-type: none"> <li>在收到数据后的大约 16 位时间内 (即接收 2 个字节所需的时间), 没有再收到其它字节;</li> <li>过程映像被填满</li> </ul>	BOOLEAN	RW	0x01 (1 <sub>dec</sub> )

## Index 0x4076 Rx buffer full notification

Index (Hex)	名称	含义	数据类型	标志	默认值
4076:0	Rx buffer full notification	该值决定了状态字节中 SW.3 [▮_161] 位 (BUF_F) 被置位的接收 FIFO 中的数据个数。	UINT16	RW	0x0000 (0 <sub>dec</sub> )

Index 0x8000 COM settings [硬件版本 03 及以上]

Index (Hex)	名称	含义	数据类型	标志	默认值	
8000:0	COM Settings	最大 SubIndex (Hex)	UINT8	RO	0x26 (38 <sub>dec</sub> )	
8000:01**	Enable RTS/CTS	FALSE	未启用 RTS/CTS	BOOLEAN	RW	0x01 (1 <sub>dec</sub> )
		TRUE	已启用 RTS/CTS			
8000:02	Enable XON/XOFF supported tx data	FALSE	不支持 XON/XOFF 发送数据	BOOLEAN	RW	0x00 (0 <sub>dec</sub> )
		TRUE	支持 XON/XOFF 发送数据			
8000:03	Enable XON/XOFF supported rx data	FALSE	不支持 XON/XOFF 接收数据	BOOLEAN	RW	0x00 (0 <sub>dec</sub> )
		TRUE	支持 XON/XOFF 接收数据			
8000:04	Enable send FIFO data continuous	FALSE	不从 FIFO 连续发送数据	BOOLEAN	RW	0x00 (0 <sub>dec</sub> )
		TRUE	启用从 FIFO 连续发送数据： 来自控制器的数据先填满端子模块的发送缓冲区（最多 128 字节）。 填充的缓冲区内容在“SendContinuous [▶_182]”位的上升沿发送。 端子模块通过设置“InitAccepted”位确认向控制器发送数据。 “InitAccepted [▶_181]”通过“SendContinuous”重置。			
8000:05	Enable transfer rate optimization	FALSE	关闭传输速率优化功能	BOOLEAN	RW	0x01 (1 <sub>dec</sub> )
		TRUE	当启用传输速率优化，“Transfer rate optimization”： 如果出现以下情况，输入缓冲区的内容会自动传输到过程映像中： <ul style="list-style-type: none"> <li>在收到数据后的大约 16 位时间内（即接收 2 个字节所需的时间），没有再收到其它字节；</li> <li>过程映像被填满</li> </ul>			
8000:06** *	Enable half duplex	FALSE	全双工模式	BOOLEAN	RW	0x00 (0 <sub>dec</sub> )
		TRUE	半双工模式			
8000:07** *	Enable point to point connection (RS422)	FALSE	点对点连接已禁用	BOOLEAN	RW	0x00 (0 <sub>dec</sub> )
		TRUE	点对点连接已启用			
8000:11	Baud rate	详细信息请参阅通信功能 [▶_147] 章节	BIT4	RW	0x06 (6 <sub>dec</sub> )	
8000:15	Data frame	详细信息请参阅通信功能 [▶_147] 章节	BIT4	RW	0x03 (3 <sub>dec</sub> )	
8000:1A	Rx buffer full notification	该值指定接收 FIFO 中的数据数量，“buffer_full [▶_182]”位由此置位。	UINT16	RW	0x0360 (864 <sub>dec</sub> )	
8000:1B**	Explicit baudrate	在该对象中，可以直接输入所需的波特率数值。仅支 通信功能 [▶_147] 章节中指定的波特率。对象 0x8000:11 和 4073 也采纳了该对象的更改	UINT32	RW	0x00000384 (9600 <sub>dec</sub> )	
8000:1C**	Extended data frame	在该对象中，除常规数据帧（如 9N1）外，还可选择特殊格式。对象 0x8000:15 和 0x4074 也采纳了该对象的更改。	ENUM16	RW	0x0003 (3 <sub>dec</sub> )	

\*\*）EL6001 的 FW11 及以上版本，EL6021 的 FW09 及以上版本

\*\*\*）仅限 EL6021

### 7.2.2 标准对象 (0x1000-0x1FFF)

这些标准对象对所有 EtherCAT 从站具有相同的含义。

#### Index 0x1000 Device type

Index (Hex)	名称	含义	数据类型	标志	默认值
1000:0	Device type	EtherCAT 从站的设备类型：Lo-Word 包含使用的 CoE 设备描述规约（5001）。根据模块化设备配置文件，Hi-Word 包含模块配置文件。	UINT32	RO	0x02581389 (39326601 <sub>dec</sub> )

## Index 0x1008 Device name

Index (Hex)	名称	含义	数据类型	标志	默认值
1008:0	Device name	EtherCAT 从站的设备名称	STRING	RO	EL6001 EL6021

## Index 0x1009 Hardware version

Index (Hex)	名称	含义	数据类型	标志	默认值
1009:0	Hardware version	EtherCAT 从站的硬件版本	STRING	RO	04

## Index 0x100A Software version

Index (Hex)	名称	含义	数据类型	标志	默认值
100A:0	Software version	EtherCAT 从站的固件版本	STRING	RO	05

## Index 0x1018 Identity

Index (Hex)	名称	含义	数据类型	标志	默认值
1018:0	Identity	识别从站的信息	UINT8	RO	0x04 (4 <sub>dec</sub> )
1018:01	Vendor ID	EtherCAT 从站的供应商 ID	UINT32	RO	0x00000002 (2 <sub>dec</sub> )
1018:02	Product code	EtherCAT 从站的产品代码	UINT32	RO	0x17853052 (394604626 <sub>dec</sub> )
1018:03	Revision	EtherCAT 从站的修订号；低位字（位 0-15）表示特殊端子模块编号，高位字（位 16-31）表示设备描述	UINT32	RO	0x00100000 (1048576 <sub>dec</sub> )
1018:04	Serial number	EtherCAT 从站的序列号；低位字的低字节（位 0-7）包含生产年份，低位字的高字节（位 8-15）包含生产周数，高位字（位 16-31）为 0	UINT32	RO	0x00000000 (0 <sub>dec</sub> )

## Index 0x10F0 Backup parameter handling

Index (Hex)	名称	含义	数据类型	标志	默认值
10F0:0	Backup parameter handling	标准化加载和保存备份条目的信息	UINT8	RO	0x01 (1 <sub>dec</sub> )
10F0:01	Checksum	对 EtherCAT 从站的所有备份条目进行校验和	UINT32	RO	0x00000000 (0 <sub>dec</sub> )

## Index 0x1400 RxPDO-Par Outputs

Index (Hex)	名称	含义	数据类型	标志	默认值
1400:0	RxPDO-Par Outputs	PDO 参数 RxPDO 1	UINT8	RO	0x06 (6 <sub>dec</sub> )
1400:06	Exclude RxPDOs	指定不得与 RxPDO 1 一起传输的 RxPDO (RxPDO 映射对象的 Index 0x)	OCTET-STRING [6]	RO	01 16 02 16 04 16

## Index 0x1401 RxPDO-Par Outputs

Index (Hex)	名称	含义	数据类型	标志	默认值
1401:0	RxPDO-Par Outputs	PDO 参数 RxPDO 2	UINT8	RO	0x06 (6 <sub>dec</sub> )
1401:06	Exclude RxPDOs	指定不得与 RxPDO 2 一起传输的 RxPDO (RxPDO 映射对象的 Index 0x)	OCTET-STRING [6]	RO	00 16 02 16 04 16

## Index 0x1402 RxPDO-Par Outputs

Index (Hex)	名称	含义	数据类型	标志	默认值
1402:0	RxPDO-Par Outputs	PDO 参数 RxPDO 3	UINT8	RO	0x06 (6 <sub>dec</sub> )
1402:06	Exclude RxPDOs	指定不得与 RxPDO 3 一起传输的 RxPDO (RxPDO 映射对象的 Index 0x)	OCTET-STRING [6]	RO	00 16 01 16 04 16

**Index 0x1600 RxPDO-Map Outputs**

Index (Hex)	名称	含义	数据类型	标志	默认值
1600:0	RxPDO-Map Outputs	PDO 映射 RxPDO 1	UINT8	R0	0x04 (4 <sub>dec</sub> )
1600:01	SubIndex 0x001	1. PDO映射条目 (对象 0x3001 (Outputs), 条目 0x01 (Ctrl))	UINT32	R0	0x3001:01, 8
1600:02	SubIndex 0x002	2. PDO映射条目 (对象 0x3001 (Outputs), 条目 0x02 (Data Out 0))	UINT32	R0	0x3001:02, 8
1600:03	SubIndex 0x003	3. PDO映射条目 (对象 0x3001 (Outputs), 条目 0x03 (Data Out 1))	UINT32	R0	0x3001:03, 8
1600:04	SubIndex 0x004	4. PDO映射条目 (对象 0x3001 (Outputs), 条目 0x04 (Data Out 2))	UINT32	R0	0x3001:04, 8

**Index 0x1601 RxPDO-Map Outputs**

Index (Hex)	名称	含义	数据类型	标志	默认值
1601:0	RxPDO-Map Outputs	PDO 映射 RxPDO 2	UINT8	R0	0x06 (6 <sub>dec</sub> )
1601:01	SubIndex 0x001	1. PDO映射条目 (对象 0x3002 (Outputs), 条目 0x01 (Ctrl))	UINT32	R0	0x3002:01, 8
1601:02	SubIndex 0x002	2. PDO映射条目 (对象 0x3002 (Outputs), 条目 0x02 (Data Out 0))	UINT32	R0	0x3002:02, 8
1601:03	SubIndex 0x003	3. PDO映射条目 (对象 0x3002 (Outputs), 条目 0x03 (Data Out 1))	UINT32	R0	0x3002:03, 8
1601:04	SubIndex 0x004	4. PDO映射条目 (对象 0x3002 (Outputs), 条目 0x04 (Data Out 2))	UINT32	R0	0x3002:04, 8
1601:05	SubIndex 0x005	5. PDO映射条目 (对象 0x3002 (Outputs), 条目 0x05 (Data Out 3))	UINT32	R0	0x3002:05, 8
1601:06	SubIndex 0x006	6. PDO映射条目 (对象 0x3002 (Outputs), 条目 0x06 (Data Out 4))	UINT32	R0	0x3002:06, 8

## Index 0x1602 RxPDO-Map Outputs

Index (Hex)	名称	含义	数据类型	标志	默认值
1602:0	RxPDO-Map Outputs	PDO 映射 RxPDO 3	UINT8	RO	0x17 (23 <sub>dec</sub> )
1602:01	SubIndex 0x001	1. PDO映射条目 (对象 0x3003 (Outputs), 条目 0x01 (Ctrl))	UINT32	RO	0x3003:01, 16
1602:02	SubIndex 0x002	2. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x02 (Data Out 0))	UINT32	RO	0x3003:02, 8
1602:03	SubIndex 0x003	3. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x03 (Data Out 1))	UINT32	RO	0x3003:03, 8
1602:04	SubIndex 0x004	4. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x04 (Data Out 2))	UINT32	RO	0x3003:04, 8
1602:05	SubIndex 0x005	5. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x05 (Data Out 3))	UINT32	RO	0x3003:05, 8
1602:06	SubIndex 0x006	6. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x06 (Data Out 4))	UINT32	RO	0x3003:06, 8
1602:07	SubIndex 0x007	7. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x07 (Data Out 5))	UINT32	RO	0x3003:07, 8
1602:08	SubIndex 0x008	8. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x08 (Data Out 6))	UINT32	RO	0x3003:08, 8
1602:09	SubIndex 0x009	9. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x09 (Data Out 7))	UINT32	RO	0x3003:09, 8
1602:0A	SubIndex 0x010	10. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x0A (Data Out 8))	UINT32	RO	0x3003:0A, 8
1602:0B	SubIndex 0x011	11. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x0B (Data Out 9))	UINT32	RO	0x3003:0B, 8
1602:0C	SubIndex 0x012	12. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x0C (Data Out 10))	UINT32	RO	0x3003:0C, 8
1602:0D	SubIndex 0x013	13. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x0D (Data Out 11))	UINT32	RO	0x3003:0D, 8
1602:0E	SubIndex 0x014	14. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x0E (Data Out 12))	UINT32	RO	0x3003:0E, 8
1602:0F	SubIndex 0x015	15. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x0F (Data Out 13))	UINT32	RO	0x3003:0F, 8
1602:10	SubIndex 0x016	16. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x10 (Data Out 14))	UINT32	RO	0x3003:10, 8
1602:11	SubIndex 0x017	17. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x11 (Data Out 15))	UINT32	RO	0x3003:11, 8
1602:12	SubIndex 0x018	18. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x12 (Data Out 16))	UINT32	RO	0x3003:12, 8
1602:13	SubIndex 0x019	19. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x13 (Data Out 17))	UINT32	RO	0x3003:13, 8
1602:14	SubIndex 0x020	20. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x14 (Data Out 18))	UINT32	RO	0x3003:14, 8
1602:15	SubIndex 0x021	21. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x15 (Data Out 19))	UINT32	RO	0x3003:15, 8
1602:16	SubIndex 0x022	22. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x16 (Data Out 20))	UINT32	RO	0x3003:16, 8
1602:17	SubIndex 0x023	23. PDO 映射条目 (对象 0x3003 (Outputs), 条目 0x17 (Data Out 21))	UINT32	RO	0x3003:17, 8



Index 0x1604 COM RxPDO-Map Outputs [硬件版本 03 及以上]

Index (Hex)	名称	含义	数据类型	标志	默认值
1604:0	COM RxPDO-Map Outputs	PDO 映射 RxPDO 5	UINT8	RO	0x1C (28 <sub>dec</sub> )
1604:01	SubIndex 0x001	1. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x01 (Transmit request))	UINT32	RO	0x7000:01, 1
1604:02	SubIndex 0x002	2. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x02 (Receive accepted))	UINT32	RO	0x7000:02, 1
1604:03	SubIndex 0x003	3. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x03 (Init request))	UINT32	RO	0x7000:03, 1
1604:04	SubIndex 0x004	4. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x04 (Send continues))	UINT32	RO	0x7000:04, 1
1604:05	SubIndex 0x005	5. PDO 映射条目 (4 位对齐)	UINT32	RO	0x0000:00, 4
1604:06	SubIndex 0x006	6. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x09 (Output length))	UINT32	RO	0x7000:09, 8
1604:07	SubIndex 0x007	7. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x11 (Data Out 0))	UINT32	RO	0x7000:11, 8
1604:08	SubIndex 0x008	8. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x12 (Data Out 1))	UINT32	RO	0x7000:12, 8
1604:09	SubIndex 0x009	9. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x13 (Data Out 2))	UINT32	RO	0x7000:13, 8
1604:0A	SubIndex 0x010	10. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x14 (Data Out 3))	UINT32	RO	0x7000:14, 8
1604:0B	SubIndex 0x011	11. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x15 (Data Out 4))	UINT32	RO	0x7000:15, 8
1604:0C	SubIndex 0x012	12. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x16 (Data Out 5))	UINT32	RO	0x7000:16, 8
1604:0D	SubIndex 0x013	13. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x17 (Data Out 6))	UINT32	RO	0x7000:17, 8
1604:0E	SubIndex 0x014	14. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x18 (Data Out 7))	UINT32	RO	0x7000:18, 8
1604:0F	SubIndex 0x015	15. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x19 (Data Out 8))	UINT32	RO	0x7000:19, 8
1604:10	SubIndex 0x016	16. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x1A (Data Out 9))	UINT32	RO	0x7000:1A, 8
1604:11	SubIndex 0x017	17. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x1B (Data Out 10))	UINT32	RO	0x7000:1B, 8
1604:12	SubIndex 0x018	18. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x1C (Data Out 11))	UINT32	RO	0x7000:1C, 8
1604:13	SubIndex 0x019	19. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x1D (Data Out 12))	UINT32	RO	0x7000:1D, 8
1604:14	SubIndex 0x020	20. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x1E (Data Out 13))	UINT32	RO	0x7000:1E, 8
1604:15	SubIndex 0x021	21. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x1F (Data Out 14))	UINT32	RO	0x7000:1F, 8
1604:16	SubIndex 0x022	22. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x20 (Data Out 15))	UINT32	RO	0x7000:20, 8
1604:17	SubIndex 0x023	23. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x21 (Data Out 16))	UINT32	RO	0x7000:21, 8
1604:18	SubIndex 0x024	24. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x22 (Data Out 17))	UINT32	RO	0x7000:22, 8
1604:19	SubIndex 0x025	25. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x23 (Data Out 18))	UINT32	RO	0x7000:23, 8
1604:1A	SubIndex 0x026	26. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x24 (Data Out 19))	UINT32	RO	0x7000:24, 8
1604:1B	SubIndex 0x027	27. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x25 (Data Out 20))	UINT32	RO	0x7000:25, 8
1604:1C	SubIndex 0x028	28. PDO 映射条目 (对象 0x7000 (COM Outputs), 条目 0x26 (Data Out 21))	UINT32	RO	0x7000:26, 8

## Index 0x1605 COM ext. outputs

Index (Hex)	名称	含义	数据类型	标志	默认值
1605:0	COM ext. outputs	PDO 映射 RxPDO 6	UINT8	RO	0x38 (56 <sub>dec</sub> )
1605:01	SubIndex 0x001	1. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x01 (Transmit request))	UINT32	RO	0x7001:01, 1
1605:02	SubIndex 0x002	2. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x02 (Receive accepted))	UINT32	RO	0x7001:02, 1
1605:03	SubIndex 0x003	3. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x03 (Init request))	UINT32	RO	0x7001:03, 1
1605:04	SubIndex 0x004	4. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x04 (Send continues))	UINT32	RO	0x7001:04, 1
1605:05	SubIndex 0x005	5. PDO 映射条目 (4 位对齐)	UINT32	RO	0x0000:00, 4
1605:06	SubIndex 0x006	6. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x09 (Output length))	UINT32	RO	0x7001:09, 8
1605:07	SubIndex 0x007	7. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x11 (Data Out 0))	UINT32	RO	0x7001:11, 16
1605:08	SubIndex 0x008	8. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x12 (Data Out 1))	UINT32	RO	0x7001:12, 16
1605:09	SubIndex 0x009	9. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x13 (Data Out 2))	UINT32	RO	0x7001:13, 16
1605:0A	SubIndex 0x010	10. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x14 (Data Out 3))	UINT32	RO	0x7001:14, 16
1605:0B	SubIndex 0x011	11. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x15 (Data Out 4))	UINT32	RO	0x7001:15, 16
1605:0C	SubIndex 0x012	12. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x16 (Data Out 5))	UINT32	RO	0x7001:16, 16
1605:0D	SubIndex 0x013	13. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x17 (Data Out 6))	UINT32	RO	0x7001:17, 16
1605:0E	SubIndex 0x014	14. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x18 (Data Out 7))	UINT32	RO	0x7001:18, 16
1605:0F	SubIndex 0x015	15. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x19 (Data Out 8))	UINT32	RO	0x7001:19, 16
1605:10	SubIndex 0x016	16. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x1A (Data Out 9))	UINT32	RO	0x7001:1A, 16
1605:11	SubIndex 0x017	17. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x1B (Data Out 10))	UINT32	RO	0x7001:1B, 16
1605:12	SubIndex 0x018	18. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x1C (Data Out 11))	UINT32	RO	0x7001:1C, 16
1605:13	SubIndex 0x019	19. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x1D (Data Out 12))	UINT32	RO	0x7001:1D, 16
1605:14	SubIndex 0x020	20. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x1E (Data Out 13))	UINT32	RO	0x7001:1E, 16
1605:15	SubIndex 0x021	21. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x1F (Data Out 14))	UINT32	RO	0x7001:1F, 16
1605:16	SubIndex 0x022	22. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x20 (Data Out 15))	UINT32	RO	0x7001:20, 16
1605:17	SubIndex 0x023	23. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x21 (Data Out 16))	UINT32	RO	0x7001:21, 16
1605:18	SubIndex 0x024	24. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x22 (Data Out 17))	UINT32	RO	0x7001:22, 16
1605:19	SubIndex 0x025	25. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x23 (Data Out 18))	UINT32	RO	0x7001:23, 16
1605:1A	SubIndex 0x026	26. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x24 (Data Out 19))	UINT32	RO	0x7001:24, 16
1605:1B	SubIndex 0x027	27. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x25 (Data Out 20))	UINT32	RO	0x7001:25, 16
1605:1C	SubIndex 0x028	28. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x26 (Data Out 21))	UINT32	RO	0x7001:26, 16
1605:1D	SubIndex 0x029	29. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x27 (Data Out 22))	UINT32	RO	0x7001:27, 16
1605:1E	SubIndex 0x030	30. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x28 (Data Out 23))	UINT32	RO	0x7001:28, 16
1605:1F	SubIndex 0x031	31. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x29 (Data Out 24))	UINT32	RO	0x7001:29, 16

Index (Hex)	名称	含义	数据类型	标志	默认值
1605:20	SubIndex 0x032	32. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x2A (Data Out 25))	UINT32	RO	0x7001:2A, 16
1605:21	SubIndex 0x033	33. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x2B (Data Out 26))	UINT32	RO	0x7001:2B, 16
1605:22	SubIndex 0x034	34. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x2C (Data Out 27))	UINT32	RO	0x7001:2C, 16
1605:23	SubIndex 0x035	35. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x2D (Data Out 28))	UINT32	RO	0x7001:2D, 16
1605:24	SubIndex 0x036	36. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x2E (Data Out 29))	UINT32	RO	0x7001:2E, 16
1605:25	SubIndex 0x037	37. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x2F (Data Out 30))	UINT32	RO	0x7001:2F, 16
1605:26	SubIndex 0x038	38. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x30 (Data Out 31))	UINT32	RO	0x7001:30, 16
1605:27	SubIndex 0x039	39. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x31 (Data Out 32))	UINT32	RO	0x7001:31, 16
1605:28	SubIndex 0x040	40. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x32 (Data Out 33))	UINT32	RO	0x7001:32, 16
1605:29	SubIndex 0x041	41. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x33 (Data Out 34))	UINT32	RO	0x7001:33, 16
1605:2A	SubIndex 0x042	42. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x34 (Data Out 35))	UINT32	RO	0x7001:34, 16
1605:2B	SubIndex 0x043	43. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x35 (Data Out 36))	UINT32	RO	0x7001:35, 16
1605:2C	SubIndex 0x044	44. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x36 (Data Out 37))	UINT32	RO	0x7001:36, 16
1605:2D	SubIndex 0x045	45. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x37 (Data Out 38))	UINT32	RO	0x7001:37, 16
1605:2E	SubIndex 0x046	46. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x38 (Data Out 39))	UINT32	RO	0x7001:38, 16
1605:2F	SubIndex 0x047	47. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x39 (Data Out 40))	UINT32	RO	0x7001:39, 16
1605:30	SubIndex 0x048	48. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x3A (Data Out 41))	UINT32	RO	0x7001:3A, 16
1605:31	SubIndex 0x049	49. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x3B (Data Out 42))	UINT32	RO	0x7001:3B, 16
1605:32	SubIndex 0x050	50. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x3C (Data Out 43))	UINT32	RO	0x7001:3C, 16
1605:33	SubIndex 0x051	51. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x3D (Data Out 44))	UINT32	RO	0x7001:3D, 16
1605:34	SubIndex 0x052	52. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x3E (Data Out 45))	UINT32	RO	0x7001:3E, 16
1605:35	SubIndex 0x053	53. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x3F (Data Out 46))	UINT32	RO	0x7001:3F, 16
1605:36	SubIndex 0x054	54. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x40 (Data Out 47))	UINT32	RO	0x7001:40, 16
1605:37	SubIndex 0x055	55. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x41 (Data Out 48))	UINT32	RO	0x7001:41, 16
1605:38	SubIndex 0x056	56. PDO 映射条目 (对象 0x7001 (COM ext. outputs), 条目 0x42 (Data Out 49))	UINT32	RO	0x7001:42, 16

Index 0x1800 TxPDO-Par Inputs

Index (Hex)	名称	含义	数据类型	标志	默认值
1800:0	TxPDO-Par Inputs	PDO 参数 TxPDO 1	UINT8	RO	0x06 (6 <sub>dec</sub> )
1800:06	Exclude TxPDOs	指定不得与 TxPDO 1 一起传输的 TxPDO (TxPDO 映射对象的Index 0x)	OCTET-STRING [6]	RO	01 1A 02 1A 04 1A

Index 0x1801 TxPDO-Par Inputs

Index (Hex)	名称	含义	数据类型	标志	默认值
1801:0	TxPDO-Par Inputs	PDO 参数 TxPDO 2	UINT8	RO	0x06 (6 <sub>dec</sub> )
1801:06	Exclude TxPDOs	指定不得与 TxPDO 2 一起传输的 TxPDO (TxPDO 映射对象的Index 0x)	OCTET-STRING [6]	RO	00 1A 02 1A 04 1A

## Index 0x1802 TxPDO-Par Inputs

Index (Hex)	名称	含义	数据类型	标志	默认值
1802:0	TxPDO-Par Inputs	PDO 参数 TxPDO 3	UINT8	RO	0x06 (6 <sub>dec</sub> )
1802:06	Exclude TxPDOs	指定不得与 TxPDO 3 一起传输的 TxPDO (TxPDO 映射对象的Index 0x)	OCTET-STRING [6]	RO	00 1A 01 1A 04 1A

## Index 0x1804 COM TxPDO-Par Inputs

Index (Hex)	名称	含义	数据类型	标志	默认值
1804:0	COM TxPDO-Par Inputs	PDO 参数 TxPDO 5	UINT8	RO	0x06 (6 <sub>dec</sub> )
1804:06	Exclude TxPDOs	指定不得与 TxPDO 5 一起传输的 TxPDO (TxPDO 映射对象的Index 0x)	OCTET-STRING [8]	RO	00 1A 01 1A 02 1A 05 1A

## Index 0x1805 COM ext. inputs

Index (Hex)	名称	含义	数据类型	标志	默认值
1805:0	COM ext. inputs	PDO 参数 TxPDO 6	UINT8	RO	0x06 (6 <sub>dec</sub> )
1805:06	Exclude TxPDOs	指定不得与 TxPDO 6 一起传输的 TxPDO (TxPDO 映射对象的Index 0x)	OCTET-STRING [8]	RO	00 1A 01 1A 02 1A 04 1A

## Index 0x1A00 TxPDO-Map Inputs

Index (Hex)	名称	含义	数据类型	标志	默认值
1A00:0	TxPDO-Map Inputs	PDO 映射 TxPDO 1	UINT8	RO	0x04 (4 <sub>dec</sub> )
1A00:01	SubIndex 0x001	1. PDO 映射条目 (对象 0x3101 (Inputs), 条目 0x01 (Status))	UINT32	RO	0x3101:01, 8
1A00:02	SubIndex 0x002	2. PDO 映射条目 (对象 0x3101 (Inputs), 条目 0x02 (Data In 0))	UINT32	RO	0x3101:02, 8
1A00:03	SubIndex 0x003	3. PDO 映射条目 (对象 0x3101 (Inputs), 条目 0x03 (Data In 1))	UINT32	RO	0x3101:03, 8
1A00:04	SubIndex 0x004	4. PDO 映射条目 (对象 0x3101 (Inputs), 条目 0x04 (Data In 2))	UINT32	RO	0x3101:04, 8

## Index 0x1A01 TxPDO-Map Inputs

Index (Hex)	名称	含义	数据类型	标志	默认值
1A01:0	TxPDO-Map Inputs	PDO 映射 TxPDO 2	UINT8	RO	0x06 (6 <sub>dec</sub> )
1A01:01	SubIndex 0x001	1. PDO 映射条目 (对象 0x3102 (Inputs), 条目 0x01 (Status))	UINT32	RO	0x3102:01, 8
1A01:02	SubIndex 0x002	2. PDO 映射条目 (对象 0x3102 (Inputs), 条目 0x02 (Data In 0))	UINT32	RO	0x3102:02, 8
1A01:03	SubIndex 0x003	3. PDO 映射条目 (对象 0x3102 (Inputs), 条目 0x03 (Data In 1))	UINT32	RO	0x3102:03, 8
1A01:04	SubIndex 0x004	4. PDO 映射条目 (对象 0x3102 (Inputs), 条目 0x04 (Data In 2))	UINT32	RO	0x3102:04, 8
1A01:05	SubIndex 0x005	5. PDO 映射条目 (对象 0x3102 (Inputs), 条目 0x05 (Data In 3))	UINT32	RO	0x3102:05, 8
1A01:06	SubIndex 0x006	6. PDO 映射条目 (对象 0x3102 (Inputs), 条目 0x06 (Data In 4))	UINT32	RO	0x3102:06, 8

Index 0x1A02 TxPDO-Map Inputs

Index (Hex)	名称	含义	数据类型	标志	默认值
1A02:0	TxPDO-Map Inputs	PDO 映射 TxPDO 3	UINT8	RO	0x17 (23 <sub>dec</sub> )
1A02:01	SubIndex 0x001	1. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x01 (Status))	UINT32	RO	0x3103:01, 16
1A02:02	SubIndex 0x002	2. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x02 (Data In 0))	UINT32	RO	0x3103:02, 8
1A02:03	SubIndex 0x003	3. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x03 (Data In 1))	UINT32	RO	0x3103:03, 8
1A02:04	SubIndex 0x004	4. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x04 (Data In 2))	UINT32	RO	0x3103:04, 8
1A02:05	SubIndex 0x005	5. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x05 (Data In 3))	UINT32	RO	0x3103:05, 8
1A02:06	SubIndex 0x006	6. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x06 (Data In 4))	UINT32	RO	0x3103:06, 8
1A02:07	SubIndex 0x007	7. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x07 (Data In 5))	UINT32	RO	0x3103:07, 8
1A02:08	SubIndex 0x008	8. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x08 (Data In 6))	UINT32	RO	0x3103:08, 8
1A02:09	SubIndex 0x009	9. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x09 (Data In 7))	UINT32	RO	0x3103:09, 8
1A02:0A	SubIndex 0x010	10. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x0A (Data In 8))	UINT32	RO	0x3103:0A, 8
1A02:0B	SubIndex 0x011	11. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x0B (Data In 9))	UINT32	RO	0x3103:0B, 8
1A02:0C	SubIndex 0x012	12. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x0C (Data In 10))	UINT32	RO	0x3103:0C, 8
1A02:0D	SubIndex 0x013	13. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x0D (Data In 11))	UINT32	RO	0x3103:0D, 8
1A02:0E	SubIndex 0x014	14. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x0E (Data In 12))	UINT32	RO	0x3103:0E, 8
1A02:0F	SubIndex 0x015	15. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x0F (Data In 13))	UINT32	RO	0x3103:0F, 8
1A02:10	SubIndex 0x016	16. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x10 (Data In 14))	UINT32	RO	0x3103:10, 8
1A02:11	SubIndex 0x017	17. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x11 (Data In 15))	UINT32	RO	0x3103:11, 8
1A02:12	SubIndex 0x018	18. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x12 (Data In 16))	UINT32	RO	0x3103:12, 8
1A02:13	SubIndex 0x019	19. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x13 (Data In 17))	UINT32	RO	0x3103:13, 8
1A02:14	SubIndex 0x020	20. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x14 (Data In 18))	UINT32	RO	0x3103:14, 8
1A02:15	SubIndex 0x021	21. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x15 (Data In 19))	UINT32	RO	0x3103:15, 8
1A02:16	SubIndex 0x022	22. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x16 (Data In 20))	UINT32	RO	0x3103:16, 8
1A02:17	SubIndex 0x023	23. PDO 映射条目 (对象 0x3103 (Inputs), 条目 0x17 (Data In 21))	UINT32	RO	0x3103:17, 8

## Index 0x1A04 COM TxPDO-Map Inputs [硬件版本 03 及以上]

Index (Hex)	名称	含义	数据类型	标志	默认值
1A04:0	COM TxPDO-Map Inputs	PDO 映射 TxPDO 5	UINT8	RO	0x1F (31 <sub>dec</sub> )
1A04:01	SubIndex 0x001	1. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x01 (Transmit accepted))	UINT32	RO	0x6000:01, 1
1A04:02	SubIndex 0x002	2. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x02 (Receive request))	UINT32	RO	0x6000:02, 1
1A04:03	SubIndex 0x003	3. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x03 (Init accepted))	UINT32	RO	0x6000:03, 1
1A04:04	SubIndex 0x004	4. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x04 (Buffer full))	UINT32	RO	0x6000:04, 1
1A04:05	SubIndex 0x005	5. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x05 (Parity error))	UINT32	RO	0x6000:05, 1
1A04:06	SubIndex 0x006	6. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x06 (Framing error))	UINT32	RO	0x6000:06, 1
1A04:07	SubIndex 0x007	7. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x07 (Overrun error))	UINT32	RO	0x6000:07, 1
1A04:08	SubIndex 0x008	8. PDO 映射条目 (1 位对齐)	UINT32	RO	0x0000:00, 1
1A04:09	SubIndex 0x009	9. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x09 (Input length))	UINT32	RO	0x6000:09, 8
1A04:0A	SubIndex 0x010	10. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x11 (Data In 0))	UINT32	RO	0x6000:11, 8
1A04:0B	SubIndex 0x011	11. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x12 (Data In 1))	UINT32	RO	0x6000:12, 8
1A04:0C	SubIndex 0x012	12. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x13 (Data In 2))	UINT32	RO	0x6000:13, 8
1A04:0D	SubIndex 0x013	13. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x14 (Data In 3))	UINT32	RO	0x6000:14, 8
1A04:0E	SubIndex 0x014	14. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x15 (Data In 4))	UINT32	RO	0x6000:15, 8
1A04:0F	SubIndex 0x015	15. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x16 (Data In 5))	UINT32	RO	0x6000:16, 8
1A04:10	SubIndex 0x016	16. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x17 (Data In 6))	UINT32	RO	0x6000:17, 8
1A04:11	SubIndex 0x017	17. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x18 (Data In 7))	UINT32	RO	0x6000:18, 8
1A04:12	SubIndex 0x018	18. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x19 (Data In 8))	UINT32	RO	0x6000:19, 8
1A04:13	SubIndex 0x019	19. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x1A (Data In 9))	UINT32	RO	0x6000:1A, 8
1A04:14	SubIndex 0x020	20. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x1B (Data In 10))	UINT32	RO	0x6000:1B, 8
1A04:15	SubIndex 0x021	21. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x1C (Data In 11))	UINT32	RO	0x6000:1C, 8
1A04:16	SubIndex 0x022	22. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x1D (Data In 12))	UINT32	RO	0x6000:1D, 8
1A04:17	SubIndex 0x023	23. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x1E (Data In 13))	UINT32	RO	0x6000:1E, 8
1A04:18	SubIndex 0x024	24. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x1F (Data In 14))	UINT32	RO	0x6000:1F, 8
1A04:19	SubIndex 0x025	25. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x20 (Data In 15))	UINT32	RO	0x6000:20, 8
1A04:1A	SubIndex 0x026	26. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x21 (Data In 16))	UINT32	RO	0x6000:21, 8
1A04:1B	SubIndex 0x027	27. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x22 (Data In 17))	UINT32	RO	0x6000:22, 8
1A04:1C	SubIndex 0x028	28. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x23 (Data In 18))	UINT32	RO	0x6000:23, 8
1A04:1D	SubIndex 0x029	29. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x24 (Data In 19))	UINT32	RO	0x6000:24, 8
1A04:1E	SubIndex 0x030	30. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x25 (Data In 20))	UINT32	RO	0x6000:25, 8
1A04:1F	SubIndex 0x031	31. PDO 映射条目 (对象 0x6000 (COM Inputs), 条目 0x26 (Data In 21))	UINT32	RO	0x6000:26, 8

Index 0x1A05 COM ext. inputs

Index (Hex)	名称	含义	数据类型	标志	默认值
1A05:0	COM ext. inputs	PDO 映射 TxPDO 6	UINT8	RO	0x3B (59 <sub>dec</sub> )
1A05:01	SubIndex 0x001	1. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x01 (Transmit accepted))	UINT32	RO	0x6001:01, 1
1A05:02	SubIndex 0x002	2. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x02 (Receive request))	UINT32	RO	0x6001:02, 1
1A05:03	SubIndex 0x003	3. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x03 (Init accepted))	UINT32	RO	0x6001:03, 1
1A05:04	SubIndex 0x004	4. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x04 (Buffer full))	UINT32	RO	0x6001:04, 1
1A05:05	SubIndex 0x005	5. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x05 (Parity error))	UINT32	RO	0x6001:05, 1
1A05:06	SubIndex 0x006	6. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x06 (Framing error))	UINT32	RO	0x6001:06, 1
1A05:07	SubIndex 0x007	7. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x07 (Overrun error))	UINT32	RO	0x6001:07, 1
1A05:08	SubIndex 0x008	8. PDO 映射条目 (1 位对齐)	UINT32	RO	0x0000:00, 1
1A05:09	SubIndex 0x009	9. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x09 (Input length))	UINT32	RO	0x6001:09, 8
1A05:0A	SubIndex 0x010	10. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x11 (Data In 0))	UINT32	RO	0x6001:11, 16
1A05:0B	SubIndex 0x011	11. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x12 (Data In 1))	UINT32	RO	0x6001:12, 16
1A05:0C	SubIndex 0x012	12. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x13 (Data In 2))	UINT32	RO	0x6001:13, 16
1A05:0D	SubIndex 0x013	13. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x14 (Data In 3))	UINT32	RO	0x6001:14, 16
1A05:0E	SubIndex 0x014	14. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x15 (Data In 4))	UINT32	RO	0x6001:15, 16
1A05:0F	SubIndex 0x015	15. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x16 (Data In 5))	UINT32	RO	0x6001:16, 16
1A05:10	SubIndex 0x016	16. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x17 (Data In 6))	UINT32	RO	0x6001:17, 16
1A05:11	SubIndex 0x017	17. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x18 (Data In 7))	UINT32	RO	0x6001:18, 16
1A05:12	SubIndex 0x018	18. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x19 (Data In 8))	UINT32	RO	0x6001:19, 16
1A05:13	SubIndex 0x019	19. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x1A (Data In 9))	UINT32	RO	0x6001:1A, 16
1A05:14	SubIndex 0x020	20. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x1B (Data In 10))	UINT32	RO	0x6001:1B, 16
1A05:15	SubIndex 0x021	21. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x1C (Data In 11))	UINT32	RO	0x6001:1C, 16
1A05:16	SubIndex 0x022	22. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x1D (Data In 12))	UINT32	RO	0x6001:1D, 16
1A05:17	SubIndex 0x023	23. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x1E (Data In 13))	UINT32	RO	0x6001:1E, 16
1A05:18	SubIndex 0x024	24. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x1F (Data In 14))	UINT32	RO	0x6001:1F, 16
1A05:19	SubIndex 0x025	25. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x20 (Data In 15))	UINT32	RO	0x6001:20, 16
1A05:1A	SubIndex 0x026	26. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x21 (Data In 16))	UINT32	RO	0x6001:21, 16
1A05:1B	SubIndex 0x027	27. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x22 (Data In 17))	UINT32	RO	0x6001:22, 16
1A05:1C	SubIndex 0x028	28. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x23 (Data In 18))	UINT32	RO	0x6001:23, 16
1A05:1D	SubIndex 0x029	29. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x24 (Data In 19))	UINT32	RO	0x6001:24, 16
1A05:1E	SubIndex 0x030	30. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x25 (Data In 20))	UINT32	RO	0x6001:25, 16
1A05:1F	SubIndex 0x031	31. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x26 (Data In 21))	UINT32	RO	0x6001:26, 16

Index (Hex)	名称	含义	数据类型	标志	默认值
1A05:20	SubIndex 0x032	32. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x27 (Data In 22))	UINT32	RO	0x6001:27, 16
1A05:21	SubIndex 0x033	33. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x28 (Data In 23))	UINT32	RO	0x6001:28, 16
1A05:22	SubIndex 0x034	34. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x29 (Data In 24))	UINT32	RO	0x6001:29, 16
1A05:23	SubIndex 0x035	35. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x2A (Data In 25))	UINT32	RO	0x6001:2A, 16
1A05:24	SubIndex 0x036	36. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x2B (Data In 26))	UINT32	RO	0x6001:2B, 16
1A05:25	SubIndex 0x037	37. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x2C (Data In 27))	UINT32	RO	0x6001:2C, 16
1A05:26	SubIndex 0x038	38. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x2D (Data In 28))	UINT32	RO	0x6001:2D, 16
1A05:27	SubIndex 0x039	39. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x2E (Data In 29))	UINT32	RO	0x6001:2E, 16
1A05:28	SubIndex 0x040	40. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x2F (Data In 30))	UINT32	RO	0x6001:2F, 16
1A05:29	SubIndex 0x041	41. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x30 (Data In 31))	UINT32	RO	0x6001:30, 16
1A05:2A	SubIndex 0x042	42. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x31 (Data In 32))	UINT32	RO	0x6001:31, 16
1A05:2B	SubIndex 0x043	43. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x32 (Data In 33))	UINT32	RO	0x6001:32, 16
1A05:2C	SubIndex 0x044	44. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x33 (Data In 34))	UINT32	RO	0x6001:33, 16
1A05:2D	SubIndex 0x045	45. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x34 (Data In 35))	UINT32	RO	0x6001:34, 16
1A05:2E	SubIndex 0x046	46. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x35 (Data In 36))	UINT32	RO	0x6001:35, 16
1A05:2F	SubIndex 0x047	47. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x36 (Data In 37))	UINT32	RO	0x6001:36, 16
1A05:30	SubIndex 0x048	48. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x37 (Data In 38))	UINT32	RO	0x6001:37, 16
1A05:31	SubIndex 0x049	49. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x38 (Data In 39))	UINT32	RO	0x6001:38, 16
1A05:32	SubIndex 0x050	50. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x39 (Data In 40))	UINT32	RO	0x6001:39, 16
1A05:33	SubIndex 0x051	51. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x3A (Data In 41))	UINT32	RO	0x6001:3A, 16
1A05:34	SubIndex 0x052	52. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x3B (Data In 42))	UINT32	RO	0x6001:3B, 16
1A05:35	SubIndex 0x053	53. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x3C (Data In 43))	UINT32	RO	0x6001:3C, 16
1A05:36	SubIndex 0x054	54. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x3D (Data In 44))	UINT32	RO	0x6001:3D, 16
1A05:37	SubIndex 0x055	55. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x3E (Data In 45))	UINT32	RO	0x6001:3E, 16
1A05:38	SubIndex 0x056	56. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x3F (Data In 46))	UINT32	RO	0x6001:3F, 16
1A05:39	SubIndex 0x057	57. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x40 (Data In 47))	UINT32	RO	0x6001:40, 16
1A05:3A	SubIndex 0x058	58. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x41 (Data In 48))	UINT32	RO	0x6001:41, 16
1A05:3B	SubIndex 0x059	59. PDO 映射条目 (对象 0x6001 (COM ext. inputs), 条目 0x42 (Data In 49))	UINT32	RO	0x6001:42, 16



Index 0x1C00 Sync manager type

Index (Hex)	名称	含义	数据类型	标志	默认值
1C00:0	Sync manager type	使用 Sync Manager	UINT8	RO	0x04 (4 <sub>dec</sub> )
1C00:01	SubIndex 0x001	Sync Manager 类型通道 1: 邮箱写入	UINT8	RO	0x01 (1 <sub>dec</sub> )
1C00:02	SubIndex 0x002	Sync Manager 类型通道 2: 邮箱读取	UINT8	RO	0x02 (2 <sub>dec</sub> )
1C00:03	SubIndex 0x003	Sync Manager 类型通道 3: 过程数据写入 (输出)	UINT8	RO	0x03 (3 <sub>dec</sub> )
1C00:04	SubIndex 0x004	Sync Manager 类型通道 4: 过程数据读取 (输入)	UINT8	RO	0x04 (4 <sub>dec</sub> )

Index 0x1C12 RxPDO assign

Index (Hex)	名称	含义	数据类型	标志	默认值
1C12:0	RxPDO assign	PDO 分配输出	UINT8	RW	0x01 (1 <sub>dec</sub> )
1C12:01	SubIndex 0x001	第 1 个分配的 RxPDO (包含相关 RxPDO 映射对象的 Index 0x)	UINT16	RW	0x1602 (5634 <sub>dec</sub> )
1C12:02	SubIndex 0x002	第 2 个分配的 RxPDO (包含相关 RxPDO 映射对象的 Index 0x)	UINT16	RW	0x0000 (0 <sub>dec</sub> )

Index 0x1C13 TxPDO assign

Index (Hex)	名称	含义	数据类型	标志	默认值
1C13:0	TxPDO assign	PDO 分配输入	UINT8	RW	0x01 (1 <sub>dec</sub> )
1C13:01	SubIndex 0x001	第 1 个分配的 TxPDO (包含相关 TxPDO 映射对象的 Index 0x)	UINT16	RW	0x1A02 (6658 <sub>dec</sub> )
1C13:02	SubIndex 0x002	第 2 个分配的 TxPDO (包含相关 TxPDO 映射对象的 Index 0x)	UINT16	RW	0x0000 (0 <sub>dec</sub> )

Index 0x1C32 SM output parameter [硬件版本 03 及以上]

Index (Hex)	名称	含义	数据类型	标志	默认值
1C32:0	SM output parameter	输出的SM同步参数	UINT8	RO	0x20 (32 <sub>dec</sub> )
1C32:01	Sync mode	当前的同步模式: <ul style="list-style-type: none"> <li>0: Free Run</li> <li>1: 与 SM 2 事件同步</li> <li>2: DC 模式 - 与 SYNC0 事件同步</li> <li>3: DC 模式 - 与 SYNC1 事件同步</li> </ul>	UINT16	RW	0x0001 (1 <sub>dec</sub> )
1C32:02	Cycle time	周期时间 (单位: ns): <ul style="list-style-type: none"> <li>Free Run: 本地定时器的周期时间</li> <li>与 SM 2 事件同步: 主站周期时间</li> <li>DC 模式: SYNC0/SYNC1 周期时间</li> </ul>	UINT32	RW	0x0007A120 (500000 <sub>dec</sub> )
1C32:03	Shift time	从 SYNC0 事件到输出的时间 (单位: ns, 仅 DC 模式)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C32:04	Sync modes supported	支持的同步模式: <ul style="list-style-type: none"> <li>位 0 = 1: 支持 Free Run</li> <li>位 1 = 1: 支持与 SM 2 事件同步</li> <li>位 2-3 = 01: 支持 DC 模式</li> <li>位 4-5 = 10: SYNC1 事件下的输出偏移 (仅 DC 模式)</li> <li>位 14 = 1: 动态周期 (在写入 1C32:08 [▶ 177] 时开始测量)</li> </ul>	UINT16	RO	0xC007 (49159 <sub>dec</sub> )
1C32:05	Minimum cycle time	最小周期时间 (单位: ns)	UINT32	RO	0x00004E20 (20000 <sub>dec</sub> )

Index (Hex)	名称	含义	数据类型	标志	默认值
1C32:06	Calc and copy time	SYNC0 和 SYNC1 事件之间的最小时间（单位：ns，仅 DC 模式）	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C32:08	Command	<ul style="list-style-type: none"> <li>0: 本地周期时间的测量停止</li> <li>1: 本地周期时间的测量开始</li> </ul> 条目 1C32:03、1C32:05、1C32:06、1C32:09 [► 177]、1C33:03、1C33:06、1C33:09 [► 178] 更新为最大测量值。 对于后续的测量，测量值被重置	UINT16	RW	0x0000 (0 <sub>dec</sub> )
1C32:09	Delay time	从 SYNC1 事件到输出的时间（单位：ns，仅 DC 模式）	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C32:0B	SM event missed counter	OPERATIONAL 期间缺失的 SM 事件数量（仅 DC 模式）	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C32:0C	Cycle exceeded counter	OPERATIONAL 期间超过周期时间的次数（周期没有及时完成或下一个周期开始得太早）	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C32:0D	Shift too short counter	SYNC0 和 SYNC1 事件之间的间隔次数太小（仅 DC 模式）	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C32:20	Sync error	在上一周期同步出错（输出太晚；仅 DC 模式）	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )

## Index 0x1C33 SM input parameter [硬件版本 03 及以上]

Index (Hex)	名称	含义	数据类型	标志	默认值
1C33:0	SM input parameter	输入的SM同步参数	UINT8	RO	0x20 (32 <sub>dec</sub> )
1C33:01	Sync mode	当前的同步模式： <ul style="list-style-type: none"> <li>0: Free Run</li> <li>1: 与 SM 3 事件同步（无输出可用）</li> <li>2: DC - 与 SYNC0 事件同步</li> <li>3: DC - 与 SYNC1 事件同步</li> <li>34: 与 SM 2 事件同步（输出可用）</li> </ul>	UINT16	RW	0x0022 (34 <sub>dec</sub> )
1C33:02	Cycle time	如 1C32:02 [► 177]	UINT32	RW	0x0007A120 (500000 <sub>dec</sub> )
1C33:03	Shift time	从 SYNC0 事件到读取输入的时间（单位：ns，仅 DC 模式）	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C33:04	Sync modes supported	支持的同步模式： <ul style="list-style-type: none"> <li>位 0: 支持 Free Run</li> <li>位 1: 支持与 SM 2 事件同步（输出可用）</li> <li>位 1: 支持与 SM 3 事件同步（无输出可用）</li> <li>位 2-3 = 01: 支持 DC 模式</li> <li>位 4-5=01: 基于本地事件的输入偏移（输出可用）</li> <li>位 4-5 = 10: 基于 SYNC1 事件的输入偏移（无输出可用）</li> <li>位 14 = 1: 动态周期（从写入 1C32:08 [► 177] 或 1C33:08 [► 178] 开始测量）</li> </ul>	UINT16	RO	0xC007 (49159 <sub>dec</sub> )
1C33:05	Minimum cycle time	如 1C32:05 [► 177]	UINT32	RO	0x00004E20 (20000 <sub>dec</sub> )
1C33:06	Calc and copy time	从读取输入到主站可以使用输入值之间的时间（单位：ns，仅 DC 模式）	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C33:08	Command	如 1C32:08 [► 177]	UINT16	RW	0x0000 (0 <sub>dec</sub> )

Index (Hex)	名称	含义	数据类型	标志	默认值
1C33:09	Delay time	从 SYNC1 事件到读取输入的时间（单位：ns，仅 DC 模式）	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C33:0B	SM event missed counter	如 1C32:11 [► 177]	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C33:0C	Cycle exceeded counter	如 1C32:12 [► 177]	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C33:0D	Shift too short counter	如 1C32:13 [► 177]	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C33:20	Sync error	如 1C32:32 [► 177]	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )

## Index 0x3001 Outputs

Index (Hex)	名称	含义	数据类型	标志	默认值
3001:0	Outputs	该对象的长度	UINT8	RO	0x04 (4 <sub>dec</sub> )
3001:01	Ctrl	控制字节 [▶ 160]	UINT8	RO	0x00 (0 <sub>dec</sub> )
3001:02	Data Out 0	输出字节 0	UINT8	RO	0x00 (0 <sub>dec</sub> )
3001:03	Data Out 1	输出字节 1	UINT8	RO	0x00 (0 <sub>dec</sub> )
3001:04	Data Out 2	输出字节 2	UINT8	RO	0x00 (0 <sub>dec</sub> )

## Index 0x3002 Outputs

Index (Hex)	名称	含义	数据类型	标志	默认值
3002:0	Outputs	该对象的长度	UINT8	RO	0x06 (6 <sub>dec</sub> )
3002:01	Ctrl	控制字节 [▶ 160]	UINT8	RO	0x00 (0 <sub>dec</sub> )
3002:02	Data Out 0	输出字节 0	UINT8	RO	0x00 (0 <sub>dec</sub> )
3002:03	Data Out 1	输出字节 1	UINT8	RO	0x00 (0 <sub>dec</sub> )
3002:04	Data Out 2	输出字节 2	UINT8	RO	0x00 (0 <sub>dec</sub> )
3002:05	Data Out 3	输出字节 3	UINT8	RO	0x00 (0 <sub>dec</sub> )
3002:06	Data Out 4	输出字节 4	UINT8	RO	0x00 (0 <sub>dec</sub> )

## Index 0x3003 Outputs

Index (Hex)	名称	含义	数据类型	标志	默认值
3003:0	Outputs	该对象的长度	UINT8	RO	0x17 (23 <sub>dec</sub> )
3003:01	Ctrl	控制字 [▶ 160]	UINT16	RO	0x0000 (0 <sub>dec</sub> )
3003:02	Data Out 0	输出字节 0	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:03	Data Out 1	输出字节 1	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:04	Data Out 2	输出字节 2	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:05	Data Out 3	输出字节 3	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:06	Data Out 4	输出字节 4	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:07	Data Out 5	输出字节 5	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:08	Data Out 6	输出字节 6	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:09	Data Out 7	输出字节 7	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:0A	Data Out 8	输出字节 8	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:0B	Data Out 9	输出字节 9	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:0C	Data Out 10	输出字节 10	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:0D	Data Out 11	输出字节 11	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:0E	Data Out 12	输出字节 12	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:0F	Data Out 13	输出字节 13	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:10	Data Out 14	输出字节 14	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:11	Data Out 15	输出字节 15	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:12	Data Out 16	输出字节 16	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:13	Data Out 17	输出字节 17	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:14	Data Out 18	输出字节 18	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:15	Data Out 19	输出字节 19	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:16	Data Out 20	输出字节 20	UINT8	RO	0x00 (0 <sub>dec</sub> )
3003:17	Data Out 21	输出字节 21	UINT8	RO	0x00 (0 <sub>dec</sub> )

## Index 0x3101 Inputs

Index (Hex)	名称	含义	数据类型	标志	默认值
3101:0	Inputs	该对象的长度	UINT8	RO	0x04 (4 <sub>dec</sub> )
3101:01	Status	状态字节 [▶ 160]	UINT8	RO	0x00 (0 <sub>dec</sub> )
3101:02	Data In 0	输入字节 0	UINT8	RO	0x00 (0 <sub>dec</sub> )
3101:03	Data In 1	输入字节 1	UINT8	RO	0x00 (0 <sub>dec</sub> )
3101:04	Data In 2	输入字节 2	UINT8	RO	0x00 (0 <sub>dec</sub> )

## Index 0x3102 Inputs

Index (Hex)	名称	含义	数据类型	标志	默认值
3102:0	Inputs	该对象的长度	UINT8	RO	0x06 (6 <sub>dec</sub> )
3102:01	Status	状态字节 [►_160]	UINT8	RO	0x00 (0 <sub>dec</sub> )
3102:02	Data In 0	输入字节 0	UINT8	RO	0x00 (0 <sub>dec</sub> )
3102:03	Data In 1	输入字节 1	UINT8	RO	0x00 (0 <sub>dec</sub> )
3102:04	Data In 2	输入字节 2	UINT8	RO	0x00 (0 <sub>dec</sub> )
3102:05	Data In 3	输入字节 3	UINT8	RO	0x00 (0 <sub>dec</sub> )
3102:06	Data In 4	输入字节 4	UINT8	RO	0x00 (0 <sub>dec</sub> )

## Index 0x3103 Inputs

Index (Hex)	名称	含义	数据类型	标志	默认值
3103:0	Inputs	该对象的长度	UINT8	RO	0x17 (23 <sub>dec</sub> )
3103:01	Status	状态字 [►_160]	UINT16	RO	0x0000 (0 <sub>dec</sub> )
3103:02	Data In 0	输入字节 0	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:03	Data In 1	输入字节 1	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:04	Data In 2	输入字节 2	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:05	Data In 3	输入字节 3	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:06	Data In 4	输入字节 4	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:07	Data In 5	输入字节 5	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:08	Data In 6	输入字节 6	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:09	Data In 7	输入字节 7	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:0A	Data In 8	输入字节 8	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:0B	Data In 9	输入字节 9	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:0C	Data In 10	输入字节 10	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:0D	Data In 11	输入字节 11	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:0E	Data In 12	输入字节 12	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:0F	Data In 13	输入字节 13	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:10	Data In 14	输入字节 14	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:11	Data In 15	输入字节 15	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:12	Data In 16	输入字节 16	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:13	Data In 17	输入字节 17	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:14	Data In 18	输入字节 18	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:15	Data In 19	输入字节 19	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:16	Data In 20	输入字节 20	UINT8	RO	0x00 (0 <sub>dec</sub> )
3103:17	Data In 21	输入字节 21	UINT8	RO	0x00 (0 <sub>dec</sub> )

## Index 0x4070 Data bytes in send buffer

Index (Hex)	名称	含义	数据类型	标志	默认值
4070:0	Data bytes in send buffer	发送 FIFO 中的数据字节数	UINT16	RO	0x0000 (0 <sub>dec</sub> )

## Index 0x4071 Data bytes in receive buffer

Index (Hex)	名称	含义	数据类型	标志	默认值
4071:0	Data bytes in receive buffer	接收 FIFO 中的数据字节数	UINT16	RO	0x0000 (0 <sub>dec</sub> )

Index 0x4072 Diagnostic

Index (Hex)	名称	含义	数据类型	标志	默认值
4072:0	Diagnostic	该对象的长度	UINT8	RO	0x05 (5 <sub>dec</sub> )
4072:01	Buffer overflow	缓冲区溢出。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
4072:02	Parity error	发生奇偶校验错误。发生错误的数项不会存入端子模块的接收 FIFO，而是直接丢弃。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
4072:03	Framing error	发生帧错误。发生错误的数项不会存入端子模块的接收 FIFO，而是直接丢弃。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
4072:04	Overrun error	发生超限错误。发生错误的数项不会存入端子模块的接收 FIFO，而是直接丢弃。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
4072:05	Buffer full	接收 FIFO 已满。所有后续到来的输入数据都将丢失！	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )

7.2.3 描述文件定义的对象 (0x6000-0xFFFF) [硬件版本 03 及以上]

对于所有支持 5001 设备描述规约的 EtherCAT 从站，描述文件定义的对象具有相同的含义。

Index 0x6000 COM Inputs

Index (Hex)	名称	含义	数据类型	标志	默认值
6000:0	COM Inputs	最大 SubIndex (Hex)	UINT8	RO	0x26 (38 <sub>dec</sub> )
6000:01	Transmit accepted	端子模块通过改变该位的状态确认已收到数据。然后，端子模块才会接收从控制器发来的新数据。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
6000:02	Receive request	端子模块通过改变这个状态位来通知控制器，DataIn 数组包含了“Input length [▶ 181]”中显示的字节数。控制器必须通过改变 ReceiveAccepted [▶ 182] 状态位来确认已收到数据。然后，端子模块才会向控制器发送新数据。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
6000:03	Init accepted	0 端子模块再次准备就绪，可进行串行数据交换。 1 端子模块已完成初始化。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
6000:04	Buffer full	接收 FIFO 已满。所有后续到来的输入数据都将丢失！	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
6000:05	Parity error	发生奇偶校验错误。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
6000:06	Framing error	发生帧错误。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
6000:07	Overrun error	发生超限错误。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
6000:09	Input length	等待从端子模块发送到控制器的输入字节数。	UINT8	RO	0x00 (0 <sub>dec</sub> )
6000:11	Data In 0	输入字节 0	UINT8	RO	0x00 (0 <sub>dec</sub> )
...	...	....	...	...	...
6000:26	Data In 21	输入字节 21	UINT8	RO	0x00 (0 <sub>dec</sub> )

Index 0x6001 COM ext. inputs

Index (Hex)	名称	含义	数据类型	标志	默认值
6001:0	COM ext. inputs		UINT8	RO	0x00 (0 <sub>dec</sub> )
6001:01	Transmit accepted	与 Index 0x6000 相同	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
6001:02	Receive request		BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
6001:03	Init accepted		BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
6001:04	Buffer full		BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
6001:05	Parity error		BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
6001:06	Framing error		BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
6001:07	Overrun error		BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
6001:09	Input length		UINT8	RO	0x00 (0 <sub>dec</sub> )
6001:11	Data In 0		UINT16	RO	0x0000 (0 <sub>dec</sub> )
	....				
6001:42	Data In 49		UINT16	RO	0x0000 (0 <sub>dec</sub> )

## Index 0x7000 COM Outputs

Index (Hex)	名称	含义	数据类型	标志	默认值
7000:0	COM Outputs	最大 SubIndex (Hex)	UINT8	RO	0x26 (38 <sub>dec</sub> )
7000:01	Transmit request	控制器通过改变该位的状态通知端子模块，DataOut 字节包含“Output length [▸ 182]”中显示的字节数。端子模块通过改变“TransmitAccepted [▸ 181]”位的状态确认已收到数据。然后，端子模块才会接收从控制器发来的新数据。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
7000:02	Receive accepted	控制器通过改变该位的状态确认已收到数据。然后，端子模块才会向控制器发送新数据。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
7000:03	Init request	0 控制器再次请求端子模块准备进行串行数据交换。 1 控制器请求端子模块进行初始化。发送和接收功能将被阻止，FIFO 指针将重置，串口将使用相关 Settings 对象的值进行初始化。端子模块将通过“Init accepted [▸ 181]”位确认初始化的执行。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
7000:04	Send continuous	从 FIFO 连续发送数据。来自控制器的数据先填满端子模块的发送缓冲区（最多 128 字节）。填充的缓冲区内容将在该位的上升沿发送。如果数据已发送，端子模块会通过设置“Init accepted [▸ 181]”位通知控制器。“Init accepted [▸ 181]”通过“Send continuous [▸ 182]”清除。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
7000:09	Output length	从控制器发送到端子模块的等待输出字节数。	UINT8	RO	0x00 (0 <sub>dec</sub> )
7000:11	Data Out 0	输出字节 0	UINT8	RO	0x00 (0 <sub>dec</sub> )
...	...	...	...	...	...
7000:26	Data Out 21	输出字节 21	UINT8	RO	0x00 (0 <sub>dec</sub> )

## Index 0x7001 COM ext. outputs

Index (Hex)	名称	含义	数据类型	标志	默认值
7001:0	COM ext. outputs		UINT8	RO	0x00 (0 <sub>dec</sub> )
7001:01	Transmit request	与 Index 0x7000 相同	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
7001:02	Receive accepted		BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
7001:03	Init request		BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
7001:04	Send continuous		BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
7001:09	Output length		UINT8	RO	0x00 (0 <sub>dec</sub> )
7001:11	Data Out 0		UINT16	RO	0x0000 (0 <sub>dec</sub> )
	...		<	<	<
7001:42	Data Out 49		UINT16	RO	0x0000 (0 <sub>dec</sub> )

## Index 0xA000 COM Diag data

Index (Hex)	名称	含义	数据类型	标志	默认值
A000:0	COM Diag data	最大 SubIndex (Hex)	UINT8	RO	0x21 (33 <sub>dec</sub> )
A000:01	Buffer overflow	缓冲区溢出。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
A000:02	Parity error	发生奇偶校验错误。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
A000:03	Framing error	发生帧错误	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
A000:04	Overrun error	发生超限错误。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
A000:05	Buffer full	接收 FIFO 已满。所有后续到来的输入数据都将丢失！	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
A000:11	Data bytes in send buffer	发送 FIFO 中的数据字节数	UINT16	RO	0x0000 (0 <sub>dec</sub> )
A000:21	Data bytes in receive buffer	接收 FIFO 中的数据字节数	UINT16	RO	0x0000 (0 <sub>dec</sub> )

## Index 0xB000 Command

仅限 EL6021 的 FW06 及以上版本，以及 EL6001 的 FW08 及以上版本。

Index (Hex)	名称	含义	数据类型	标志	默认值
B000:0	Command	最大 SubIndex (Hex)	UINT8	RO	0x03 (3 <sub>dec</sub> )
B000:01	Request	可以通过请求对象向端子模块发送命令 参见命令模式 [▶_140]	OCTET-STRING [2]	RW	{0}
B000:02	Status	当前正在执行的命令的状态 0: 命令执行无误。 255: 正在执行命令	UINT8	RO	0x00 (0 <sub>dec</sub> )
B000:03	Response	命令的可选响应值 字节 0: 见 B000:02 字节 1: 未使用 2-n: 服务响应数据	OCTET-STRING [6]	RO	{0}

Index 0xF000 Modular device profile

Index (Hex)	名称	含义	数据类型	标志	默认值
F000:0	Modular device profile	模块化设备描述文件的常规信息	UINT8	RO	0x02 (2 <sub>dec</sub> )
F000:01	Module Index Oxdistance	通道之间对象 Index (Hex) 的间隔	UINT16	RO	0x0010 (16 <sub>dec</sub> )
F000:02	Maximum number of modules	通道数量	UINT16	RO	0x0002 (2 <sub>dec</sub> )

Index 0xF008 Code word

Index (Hex)	名称	含义	数据类型	标志	默认值
F008:0	Code word	保留	UINT32	RW	0x00000000 (0 <sub>dec</sub> )

Index 0xF010 Module list

Index (Hex)	名称	含义	数据类型	标志	默认值
F010:0	Module list	最大 SubIndex (Hex)	UINT8	RW	0x02 (2 <sub>dec</sub> )
F010:01	SubIndex 0x001	-	UINT32	RW	0x00000000 (0 <sub>dec</sub> )
F010:02	SubIndex 0x002	-	UINT32	RW	0x00000258 (600 <sub>dec</sub> )

## 8 EL6002、EL6022 的 CoE 对象概览

### 8.1 对象描述和参数化

#### ● EtherCAT XML 设备描述

**i** 此描述与 EtherCAT XML 设备描述中的 CoE 对象相匹配。建议从[倍福网站](#)的下载区下载最新 XML 文件，并按照安装说明进行安装。

#### ● 参数化

**i** 端子模块通过“CoE - Online”选项卡 [▶ 118] 进行参数化（双击相应对象，见下文）。

#### 简介

CoE 概述包含针对不同应用的对象：

- 调试过程中配置所需的对象 [▶ 184]
- 用于指示内部设置的对象 [▶ 185]（可能是固定的）
- 描述文件定义的对象 [▶ 196]，表示输入和输出的状态显示。

以下章节首先介绍的是正常运行所需的对象，然后完整描述其它对象。

#### 8.1.1 调试对象

##### Index 0x1011 Restore default parameters

Index (Hex)	名称	含义	数据类型	标志	默认值
1011:0	恢复默认参数 [▶ 215]	恢复默认参数	UINT8	RO	0x01 (1 <sub>dec</sub> )
1011:01	SubIndex 0x001	如果此对象在设置值对话框中被设置为“0x64616F6C”，所有对象都被重置为它们的出厂状态。	UINT32	RW	0x00000000 (0 <sub>dec</sub> )



Index 0x80n0 COM Settings Ch. 1 (n = 0) , Ch. 2 (n = 1)

Index (Hex)	名称	含义	数据类型	标志	默认值
80n0:0	COM Settings Ch. 1 + Ch. 2	最大 SubIndex (Hex)	UINT8	RO	0x1A (26 <sub>dec</sub> )
80n0:01**	Enable RTS/CTS	FALSE	BOOLEAN	RW	0x01 (1 <sub>dec</sub> )
		TRUE			
80n0:02	Enable XON/XOFF supported tx data	FALSE	BOOLEAN	RW	0x00 (0 <sub>dec</sub> )
		TRUE			
80n0:03	Enable XON/XOFF supported rx data	FALSE	BOOLEAN	RW	0x00 (0 <sub>dec</sub> )
		TRUE			
80n0:04	Enable send FIFO data continuous	FALSE	BOOLEAN	RW	0x00 (0 <sub>dec</sub> )
		TRUE			
80n0:05	Enable transfer rate optimization	FALSE	BOOLEAN	RW	0x01 (1 <sub>dec</sub> )
		TRUE			
80n0:06** *	Enable half duplex	FALSE	BOOLEAN	RW	0x00 (0 <sub>dec</sub> )
		TRUE			
80n0:07** *	Enable point to point connection (RS422)	FALSE	BOOLEAN	RW	0x00 (0 <sub>dec</sub> )
		TRUE			
80n0:11	Baud rate	详细信息请参阅通信功能 [▶ 147] 章节	BIT4	RW	0x06 (6 <sub>dec</sub> )
80n0:15	Data frame	详细信息请参阅通信功能 [▶ 147] 章节	BIT4	RW	0x03 (3 <sub>dec</sub> )
80n0:1A	Rx buffer full notification	该值指定接收 FIFO 中的数据数量，“buffer full”位由此置位。	UINT16	RW	0x0360 (864 <sub>dec</sub> )
80n0:1B** **	Explicit baudrate	更多详细信息请参阅 TcVirtualComDriver [▶ 145] 和通信功能 [▶ 147] 章节	UINT32	RW	0x00002580 (9600 <sub>dec</sub> )

\*\*）仅限 EL6002

\*\*\*）仅限 EL6022

\*\*\*\*）仅限 EL6002 的固件 03 [▶ 202] 及以上版本

### 8.1.2 标准对象 (0x1000-0x1FFF)

这些标准对象对所有 EtherCAT 从站具有相同的含义。

#### Index 0x1000 Device type

Index (Hex)	名称	含义	数据类型	标志	默认值
1000:0	Device type	EtherCAT 从站的设备类型：Lo-Word 包含使用的 CoE 设备描述规约 (5001)。根据模块化设备配置文件，Hi-Word 包含模块配置文件。	UINT32	RO	0x02581389 (39326601 <sub>dec</sub> )

#### Index 0x1008 Device name

Index (Hex)	名称	含义	数据类型	标志	默认值
1008:0	Device name	EtherCAT 从站的设备名称	STRING	RO	EL60xx

## Index 0x1009 Hardware version

Index (Hex)	名称	含义	数据类型	标志	默认值
1009:0	Hardware version	EtherCAT 从站的硬件版本	STRING	RO	00

## Index 0x100A Software version

Index (Hex)	名称	含义	数据类型	标志	默认值
100A:0	Software version	EtherCAT 从站的固件版本	STRING	RO	01

## Index 0x1018 Identity

Index (Hex)	名称	含义	数据类型	标志	默认值
1018:0	Identity	识别从站的信息	UINT8	RO	0x04 (4 <sub>dec</sub> )
1018:01	Vendor ID	EtherCAT 从站的供应商 ID	UINT32	RO	0x00000002 (2 <sub>dec</sub> )
1018:02	Product code	EtherCAT 从站的产品代码	UINT32	RO	EL6002: 0x17723052 (393359442 <sub>dec</sub> ) EL6022: 0x17863052 (394670162 <sub>dec</sub> )
1018:03	Revision	EtherCAT 从站的修订号；低位字（位 0-15）表示特殊端子模块编号，高位字（位 16-31）表示设备描述	UINT32	RO	0x00100000 (1048576 <sub>dec</sub> )
1018:04	Serial number	EtherCAT 从站的序列号；低位字的低字节（位 0-7）包含生产年份，低位字的高字节（位 8-15）包含生产周数，高位字（位 16-31）为 0	UINT32	RO	0x00000000 (0 <sub>dec</sub> )

## Index 0x10F0 Backup parameter handling

Index (Hex)	名称	含义	数据类型	标志	默认值
10F0:0	Backup parameter handling	标准化加载和保存备份条目的信息	UINT8	RO	0x01 (1 <sub>dec</sub> )
10F0:01	Checksum	对 EtherCAT 从站的所有备份条目进行校验和	UINT32	RO	0x00000000 (0 <sub>dec</sub> )

**Index 0x1600 COM RxPDO-Map Outputs Ch. 1**

Index (Hex)	名称	含义	数据类型	标志	默认值
1600:0	COM RxPDO-Map Outputs Ch. 1	PDO 映射 RxPDO 1	UINT8	RO	0x1C (28 <sub>dec</sub> )
1600:01	SubIndex 0x001	1. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x01 (Transmit request))	BOOLEAN	RO	0x7000:01, 1
1600:02	SubIndex 0x002	2. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x02 (Receive accepted))	BOOLEAN	RO	0x7000:02, 1
1600:03	SubIndex 0x003	3. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x03 (Init request))	BOOLEAN	RO	0x7000:03, 1
1600:04	SubIndex 0x004	4. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x04 (Send continuous))	BOOLEAN	RO	0x7000:04, 1
1600:05	SubIndex 0x005	5. PDO 映射条目 (4 位对齐)	Align4	RO	0x0000:00, 4
1600:06	SubIndex 0x006	6. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x09 (Output length))	UINT8	RO	0x7000:09, 8
1600:07	SubIndex 0x007	7. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x11 (Data Out 0))	UINT8	RO	0x7000:11, 8
1600:08	SubIndex 0x008	8. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x12 (Data Out 1))	UINT8	RO	0x7000:12, 8
1600:09	SubIndex 0x009	9. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x13 (Data Out 2))	UINT8	RO	0x7000:13, 8
1600:0A	SubIndex 0x010	10. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x14 (Data Out 3))	UINT8	RO	0x7000:14, 8
1600:0B	SubIndex 0x011	11. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x15 (Data Out 4))	UINT8	RO	0x7000:15, 8
1600:0C	SubIndex 0x012	12. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x16 (Data Out 5))	UINT8	RO	0x7000:16, 8
1600:0D	SubIndex 0x013	13. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x17 (Data Out 6))	UINT8	RO	0x7000:17, 8
1600:0E	SubIndex 0x014	14. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x18 (Data Out 7))	UINT8	RO	0x7000:18, 8
1600:0F	SubIndex 0x015	15. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x19 (Data Out 8))	UINT8	RO	0x7000:19, 8
1600:10	SubIndex 0x016	16. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x1A (Data Out 9))	UINT8	RO	0x7000:1A, 8
1600:11	SubIndex 0x017	17. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x1B (Data Out 10))	UINT8	RO	0x7000:1B, 8
1600:12	SubIndex 0x018	18. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x1C (Data Out 11))	UINT8	RO	0x7000:1C, 8
1600:13	SubIndex 0x019	19. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x1D (Data Out 12))	UINT8	RO	0x7000:1D, 8
1600:14	SubIndex 0x020	20. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x1E (Data Out 13))	UINT8	RO	0x7000:1E, 8
1600:15	SubIndex 0x021	21. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x1F (Data Out 14))	UINT8	RO	0x7000:1F, 8
1600:16	SubIndex 0x022	22. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x20 (Data Out 15))	UINT8	RO	0x7000:20, 8
1600:17	SubIndex 0x023	23. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x21 (Data Out 16))	UINT8	RO	0x7000:21, 8
1600:18	SubIndex 0x024	24. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x22 (Data Out 17))	UINT8	RO	0x7000:22, 8
1600:19	SubIndex 0x025	25. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x23 (Data Out 18))	UINT8	RO	0x7000:23, 8
1600:1A	SubIndex 0x026	26. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x24 (Data Out 19))	UINT8	RO	0x7000:24, 8
1600:1B	SubIndex 0x027	27. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x25 (Data Out 20))	UINT8	RO	0x7000:25, 8
1600:1C	SubIndex 0x028	28. PDO 映射条目 (对象 0x7000 (COM Outputs Ch. 1), 条目 0x26 (Data Out 21))	UINT8	RO	0x7000:26, 8

## Index 0x1601 COM RxPDO-Map Outputs Ch. 2

Index (Hex)	名称	含义	数据类型	标志	默认值
1601:0	COM RxPDO-Map Outputs Ch. 2	PDO 映射 RxPDO 2	UINT8	RO	0x1C (28 <sub>dec</sub> )
1601:01	SubIndex 0x001	1. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x01 (Transmit request))	BOOLEAN	RO	0x7010:01, 1
1601:02	SubIndex 0x002	2. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x02 (Receive accepted))	BOOLEAN	RO	0x7010:02, 1
1601:03	SubIndex 0x003	3. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x03 (Init request))	BOOLEAN	RO	0x7010:03, 1
1601:04	SubIndex 0x004	4. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x04 (Send continuous))	BOOLEAN	RO	0x7010:04, 1
1601:05	SubIndex 0x005	5. PDO 映射条目 (4 位对齐)	Align4	RO	0x0000:00, 4
1601:06	SubIndex 0x006	6. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x09 (Output length))	UINT8	RO	0x7010:09, 8
1601:07	SubIndex 0x007	7. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x11 (Data Out 0))	UINT8	RO	0x7010:11, 8
1601:08	SubIndex 0x008	8. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x12 (Data Out 1))	UINT8	RO	0x7010:12, 8
1601:09	SubIndex 0x009	9. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x13 (Data Out 2))	UINT8	RO	0x7010:13, 8
1601:0A	SubIndex 0x010	10. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x14 (Data Out 3))	UINT8	RO	0x7010:14, 8
1601:0B	SubIndex 0x011	11. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x15 (Data Out 4))	UINT8	RO	0x7010:15, 8
1601:0C	SubIndex 0x012	12. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x16 (Data Out 5))	UINT8	RO	0x7010:16, 8
1601:0D	SubIndex 0x013	13. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x17 (Data Out 6))	UINT8	RO	0x7010:17, 8
1601:0E	SubIndex 0x014	14. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x18 (Data Out 7))	UINT8	RO	0x7010:18, 8
1601:0F	SubIndex 0x015	15. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x19 (Data Out 8))	UINT8	RO	0x7010:19, 8
1601:10	SubIndex 0x016	16. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x1A (Data Out 9))	UINT8	RO	0x7010:1A, 8
1601:11	SubIndex 0x017	17. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x1B (Data Out 10))	UINT8	RO	0x7010:1B, 8
1601:12	SubIndex 0x018	18. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x1C (Data Out 11))	UINT8	RO	0x7010:1C, 8
1601:13	SubIndex 0x019	19. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x1D (Data Out 12))	UINT8	RO	0x7010:1D, 8
1601:14	SubIndex 0x020	20. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x1E (Data Out 13))	UINT8	RO	0x7010:1E, 8
1601:15	SubIndex 0x021	21. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x1F (Data Out 14))	UINT8	RO	0x7010:1F, 8
1601:16	SubIndex 0x022	22. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x20 (Data Out 15))	UINT8	RO	0x7010:20, 8
1601:17	SubIndex 0x023	23. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x21 (Data Out 16))	UINT8	RO	0x7010:21, 8
1601:18	SubIndex 0x024	24. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x22 (Data Out 17))	UINT8	RO	0x7010:22, 8
1601:19	SubIndex 0x025	25. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x23 (Data Out 18))	UINT8	RO	0x7010:23, 8
1601:1A	SubIndex 0x026	26. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x24 (Data Out 19))	UINT8	RO	0x7010:24, 8
1601:1B	SubIndex 0x027	27. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x25 (Data Out 20))	UINT8	RO	0x7010:25, 8
1601:1C	SubIndex 0x028	28. PDO 映射条目 (对象 0x7010 (COM Outputs Ch. 2), 条目 0x26 (Data Out 21))	UINT8	RO	0x7010:26, 8

Index 0x1604 COM RxPDO-Map Outputs Ch. 1

Index (Hex)	名称	含义	数据类型	标志	默认值
1604:0	COM RxPDO-Map Outputs Ch. 1	PDO 映射 RxPDO 1	UINT8	RO	0x17 (23 <sub>dec</sub> )
1604:01	SubIndex 0x001	1. PDO 映射条目 (对象 0x7001 (Ctrl Ch.1), 条目 0x01 (Ctrl))	UINT16	RO	0x7001:01, 16
1604:02	SubIndex 0x002	2. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x11 (Data Out 0))	UINT8	RO	0x7000:11, 8
1604:03	SubIndex 0x003	3. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x12 (Data Out 1))	UINT8	RO	0x7000:12, 8
1604:04	SubIndex 0x004	4. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x13 (Data Out 2))	UINT8	RO	0x7000:13, 8
1604:05	SubIndex 0x005	5. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x14 (Data Out 3))	UINT8	RO	0x7000:14, 8
1604:06	SubIndex 0x006	6. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x15 (Data Out 4))	UINT8	RO	0x7000:15, 8
1604:07	SubIndex 0x007	7. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x16 (Data Out 5))	UINT8	RO	0x7000:16, 8
1604:08	SubIndex 0x008	8. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x17 (Data Out 6))	UINT8	RO	0x7000:17, 8
1604:09	SubIndex 0x009	9. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x18 (Data Out 7))	UINT8	RO	0x7000:18, 8
1604:0A	SubIndex 0x010	10. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x19 (Data Out 8))	UINT8	RO	0x7000:19, 8
1604:0B	SubIndex 0x011	11. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x1A (Data Out 9))	UINT8	RO	0x7000:1A, 8
1604:0C	SubIndex 0x012	12. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x1B (Data Out 10))	UINT8	RO	0x7000:1B, 8
1604:0D	SubIndex 0x013	13. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x1C (Data Out 11))	UINT8	RO	0x7000:1C, 8
1604:0E	SubIndex 0x014	14. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x1D (Data Out 12))	UINT8	RO	0x7000:1D, 8
1604:0F	SubIndex 0x015	15. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x1E (Data Out 13))	UINT8	RO	0x7000:1E, 8
1604:10	SubIndex 0x016	16. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x1F (Data Out 14))	UINT8	RO	0x7000:1F, 8
1604:11	SubIndex 0x017	17. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x20 (Data Out 15))	UINT8	RO	0x7000:20, 8
1604:12	SubIndex 0x018	18. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x21 (Data Out 16))	UINT8	RO	0x7000:21, 8
1604:13	SubIndex 0x019	19. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x22 (Data Out 17))	UINT8	RO	0x7000:22, 8
1604:14	SubIndex 0x020	20. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x23 (Data Out 18))	UINT8	RO	0x7000:23, 8
1604:15	SubIndex 0x021	21. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x24 (Data Out 19))	UINT8	RO	0x7000:24, 8
1604:16	SubIndex 0x022	22. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x25 (Data Out 20))	UINT8	RO	0x7000:25, 8
1604:17	SubIndex 0x023	23. PDO 映射条目 (对象 0x7000 (COM Outputs Ch.1), 条目 0x26 (Data Out 21))	UINT8	RO	0x7000:26, 8

## Index 0x1605 COM RxPDO-Map Outputs Ch. 2

Index (Hex)	名称	含义	数据类型	标志	默认值
1605:0	COM RxPDO-Map Outputs Ch. 1	PDO 映射 RxPDO 1	UINT8	RO	0x17 (23 <sub>dec</sub> )
1605:01	SubIndex 0x001	1. PDO 映射条目 (对象 0x7011 (Ctrl Ch.2), 条目 0x01 (Ctrl))	UINT16	RO	0x7011:01, 16
1605:02	SubIndex 0x002	2. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x11 (Data Out 0))	UINT8	RO	0x7010:11, 8
1605:03	SubIndex 0x003	3. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x12 (Data Out 1))	UINT8	RO	0x7010:12, 8
1605:04	SubIndex 0x004	4. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x13 (Data Out 2))	UINT8	RO	0x7010:13, 8
1605:05	SubIndex 0x005	5. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x14 (Data Out 3))	UINT8	RO	0x7010:14, 8
1605:06	SubIndex 0x006	6. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x15 (Data Out 4))	UINT8	RO	0x7010:15, 8
1605:07	SubIndex 0x007	7. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x16 (Data Out 5))	UINT8	RO	0x7010:16, 8
1605:08	SubIndex 0x008	8. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x17 (Data Out 6))	UINT8	RO	0x7010:17, 8
1605:09	SubIndex 0x009	9. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x18 (Data Out 7))	UINT8	RO	0x7010:18, 8
1605:0A	SubIndex 0x010	10. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x19 (Data Out 8))	UINT8	RO	0x7010:19, 8
1605:0B	SubIndex 0x011	11. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x1A (Data Out 9))	UINT8	RO	0x7010:1A, 8
1605:0C	SubIndex 0x012	12. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x1B (Data Out 10))	UINT8	RO	0x7010:1B, 8
1605:0D	SubIndex 0x013	13. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x1C (Data Out 11))	UINT8	RO	0x7010:1C, 8
1605:0E	SubIndex 0x014	14. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x1D (Data Out 12))	UINT8	RO	0x7010:1D, 8
1605:0F	SubIndex 0x015	15. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x1E (Data Out 13))	UINT8	RO	0x7010:1E, 8
1605:10	SubIndex 0x016	16. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x1F (Data Out 14))	UINT8	RO	0x7010:1F, 8
1605:11	SubIndex 0x017	17. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x20 (Data Out 15))	UINT8	RO	0x7010:20, 8
1605:12	SubIndex 0x018	18. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x21 (Data Out 16))	UINT8	RO	0x7010:21, 8
1605:13	SubIndex 0x019	19. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x22 (Data Out 17))	UINT8	RO	0x7010:22, 8
1605:14	SubIndex 0x020	20. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x23 (Data Out 18))	UINT8	RO	0x7010:23, 8
1605:15	SubIndex 0x021	21. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x24 (Data Out 19))	UINT8	RO	0x7010:24, 8
1605:16	SubIndex 0x022	22. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x25 (Data Out 20))	UINT8	RO	0x7010:25, 8
1605:17	SubIndex 0x023	23. PDO 映射条目 (对象 0x7010 (COM Outputs Ch.2), 条目 0x26 (Data Out 21))	UINT8	RO	0x7010:26, 8

Index 0x1A00 COM TxPDO-Map Inputs Ch. 1

Index (Hex)	名称	含义	数据类型	标志	默认值
1A00:0	COM TxPDO-Map Inputs Ch. 1	PDO 映射 TxPDO 1	UINT8	RO	0x1F (31 <sub>dec</sub> )
1A00:01	SubIndex 0x001	1. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x01 (Transmit accepted))	BOOLEAN	RO	0x6000:01, 1
1A00:02	SubIndex 0x002	2. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x02 (Receive request))	BOOLEAN	RO	0x6000:02, 1
1A00:03	SubIndex 0x003	3. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x03 (Init accepted))	BOOLEAN	RO	0x6000:03, 1
1A00:04	SubIndex 0x004	4. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x04 (Buffer full))	BOOLEAN	RO	0x6000:04, 1
1A00:05	SubIndex 0x005	5. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x05 (Parity error))	BOOLEAN	RO	0x6000:05, 1
1A00:06	SubIndex 0x006	6. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x06 (Framing error))	BOOLEAN	RO	0x6000:06, 1
1A00:07	SubIndex 0x007	7. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x07 (Overrun error))	BOOLEAN	RO	0x6000:07, 1
1A00:08	SubIndex 0x008	8. PDO 映射条目 (1 位对齐)	Align1	RO	0x0000:00, 1
1A00:09	SubIndex 0x009	9. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x09 (Input length))	UINT8	RO	0x6000:09, 8
1A00:0A	SubIndex 0x010	10. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x10 (Data In 0))	UINT8	RO	0x6000:11, 8
1A00:0B	SubIndex 0x011	11. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x11 (Data In 1))	UINT8	RO	0x6000:12, 8
1A00:0C	SubIndex 0x012	12. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x12 (Data In 2))	UINT8	RO	0x6000:13, 8
1A00:0D	SubIndex 0x013	13. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x13 (Data In 3))	UINT8	RO	0x6000:14, 8
1A00:0E	SubIndex 0x014	14. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x14 (Data In 4))	UINT8	RO	0x6000:15, 8
1A00:0F	SubIndex 0x015	15. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x15 (Data In 5))	UINT8	RO	0x6000:16, 8
1A00:10	SubIndex 0x016	16. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x16 (Data In 6))	UINT8	RO	0x6000:17, 8
1A00:11	SubIndex 0x017	17. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x17 (Data In 7))	UINT8	RO	0x6000:18, 8
1A00:12	SubIndex 0x018	18. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x18 (Data In 8))	UINT8	RO	0x6000:19, 8
1A00:13	SubIndex 0x019	19. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x19 (Data In 9))	UINT8	RO	0x6000:1A, 8
1A00:14	SubIndex 0x020	20. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x1A (Data In 10))	UINT8	RO	0x6000:1B, 8
1A00:15	SubIndex 0x021	21. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x1B (Data In 11))	UINT8	RO	0x6000:1C, 8
1A00:16	SubIndex 0x022	22. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x1C (Data In 12))	UINT8	RO	0x6000:1D, 8
1A00:17	SubIndex 0x023	23. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x1D (Data In 13))	UINT8	RO	0x6000:1E, 8
1A00:18	SubIndex 0x024	24. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x1E (Data In 14))	UINT8	RO	0x6000:1F, 8
1A00:19	SubIndex 0x025	25. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x1F (Data In 15))	UINT8	RO	0x6000:20, 8
1A00:1A	SubIndex 0x026	26. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x20 (Data In 16))	UINT8	RO	0x6000:21, 8
1A00:1B	SubIndex 0x027	27. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x21 (Data In 17))	UINT8	RO	0x6000:22, 8
1A00:1C	SubIndex 0x028	28. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x22 (Data In 18))	UINT8	RO	0x6000:23, 8
1A00:1D	SubIndex 0x029	29. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x23 (Data In 19))	UINT8	RO	0x6000:24, 8
1A00:1E	SubIndex 0x030	30. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x24 (Data In 20))	UINT8	RO	0x6000:25, 8
1A00:1F	SubIndex 0x031	31. PDO 映射条目 (对象 0x6000 (COM Inputs Ch. 1), 条目 0x25 (Data In 21))	UINT8	RO	0x6000:26, 8

## Index 0x1A01 COM TxPDO-Map Inputs Ch. 2

Index (Hex)	名称	含义	数据类型	标志	默认值
1A01:0	COM TxPDO-Map Inputs Ch. 2	PDO 映射 TxPDO 2	UINT8	RO	0x1F (31 <sub>dec</sub> )
1A01:01	SubIndex 0x001	1. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x01 (Transmit accepted))	BOOLEAN	RO	0x6010:01, 1
1A01:02	SubIndex 0x002	2. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x02 (Receive request))	BOOLEAN	RO	0x6010:02, 1
1A01:03	SubIndex 0x003	3. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x03 (Init accepted))	BOOLEAN	RO	0x6010:03, 1
1A01:04	SubIndex 0x004	4. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x04 (Buffer full))	BOOLEAN	RO	0x6010:04, 1
1A01:05	SubIndex 0x005	5. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x05 (Parity error))	BOOLEAN	RO	0x6010:05, 1
1A01:06	SubIndex 0x006	6. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x06 (Framing error))	BOOLEAN	RO	0x6010:06, 1
1A01:07	SubIndex 0x007	7. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x07 (Overrun error))	BOOLEAN	RO	0x6010:07, 1
1A01:08	SubIndex 0x008	8. PDO 映射条目 (1 位对齐)	Align1	RO	0x0000:00, 1
1A01:09	SubIndex 0x009	9. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x09 (Input length))	UINT8	RO	0x6010:09, 8
1A01:0A	SubIndex 0x010	10. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x10 (Data In 0))	UINT8	RO	0x6010:11, 8
1A01:0B	SubIndex 0x011	11. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x11 (Data In 1))	UINT8	RO	0x6010:12, 8
1A01:0C	SubIndex 0x012	12. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x12 (Data In 2))	UINT8	RO	0x6010:13, 8
1A01:0D	SubIndex 0x013	13. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x13 (Data In 3))	UINT8	RO	0x6010:14, 8
1A01:0E	SubIndex 0x014	14. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x14 (Data In 4))	UINT8	RO	0x6010:15, 8
1A01:0F	SubIndex 0x015	15. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x15 (Data In 5))	UINT8	RO	0x6010:16, 8
1A01:10	SubIndex 0x016	16. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x16 (Data In 6))	UINT8	RO	0x6010:17, 8
1A01:11	SubIndex 0x017	17. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x17 (Data In 7))	UINT8	RO	0x6010:18, 8
1A01:12	SubIndex 0x018	18. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x18 (Data In 8))	UINT8	RO	0x6010:19, 8
1A01:13	SubIndex 0x019	19. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x19 (Data In 9))	UINT8	RO	0x6010:1A, 8
1A01:14	SubIndex 0x020	20. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x1A (Data In 10))	UINT8	RO	0x6010:1B, 8
1A01:15	SubIndex 0x021	21. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x1B (Data In 11))	UINT8	RO	0x6010:1C, 8
1A01:16	SubIndex 0x022	22. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x1C (Data In 12))	UINT8	RO	0x6010:1D, 8
1A01:17	SubIndex 0x023	23. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x1D (Data In 13))	UINT8	RO	0x6010:1E, 8
1A01:18	SubIndex 0x024	24. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x1E (Data In 14))	UINT8	RO	0x6010:1F, 8
1A01:19	SubIndex 0x025	25. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x1F (Data In 15))	UINT8	RO	0x6010:20, 8
1A01:1A	SubIndex 0x026	26. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x20 (Data In 16))	UINT8	RO	0x6010:21, 8
1A01:1B	SubIndex 0x027	27. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x21 (Data In 17))	UINT8	RO	0x6010:22, 8
1A01:1C	SubIndex 0x028	28. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x22 (Data In 18))	UINT8	RO	0x6010:23, 8
1A01:1D	SubIndex 0x029	29. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x23 (Data In 19))	UINT8	RO	0x6010:24, 8
1A01:1E	SubIndex 0x030	30. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x24 (Data In 20))	UINT8	RO	0x6010:25, 8
1A01:1F	SubIndex 0x031	31. PDO 映射条目 (对象 0x6010 (COM Inputs Ch. 2), 条目 0x25 (Data In 21))	UINT8	RO	0x6010:26, 8



Index 0x1A04 COM TxPDO-Map Inputs Ch. 1

Index (Hex)	名称	含义	数据类型	标志	默认值
1A04:0	COM TxPDO-Map Inputs Ch. 1	PDO 映射 TxPDO 1	UINT8	RO	0x17 (23 <sub>dec</sub> )
1A04:01	SubIndex 0x001	1. PDO 映射条目 (对象 0x6001 (Status Ch.1), 条目 0x01 (Status))	UINT16	RO	0x6001:01, 16
1A04:02	SubIndex 0x002	2. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x11 (Data In 0))	UINT8	RO	0x6000:11, 8
1A04:03	SubIndex 0x003	3. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x12 (Data In 1))	UINT8	RO	0x6000:12, 8
1A04:04	SubIndex 0x004	4. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x13 (Data In 2))	UINT8	RO	0x6000:13, 8
1A04:05	SubIndex 0x005	5. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x14 (Data In 3))	UINT8	RO	0x6000:14, 8
1A04:06	SubIndex 0x006	6. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x15 (Data In 4))	UINT8	RO	0x6000:15, 8
1A04:07	SubIndex 0x007	7. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x16 (Data In 5))	UINT8	RO	0x6000:16, 8
1A04:08	SubIndex 0x008	8. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x17 (Data In 6))	UINT8	RO	0x6000:17, 8
1A04:09	SubIndex 0x009	9. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x18 (Data In 7))	UINT8	RO	0x6000:18, 8
1A04:0A	SubIndex 0x010	10. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x19 (Data In 8))	UINT8	RO	0x6000:19, 8
1A04:0B	SubIndex 0x011	11. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x1A (Data In 9))	UINT8	RO	0x6000:1A, 8
1A04:0C	SubIndex 0x012	12. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x1B (Data In 10))	UINT8	RO	0x6000:1B, 8
1A04:0D	SubIndex 0x013	13. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x1C (Data In 11))	UINT8	RO	0x6000:1C, 8
1A04:0E	SubIndex 0x014	14. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x1D (Data In 12))	UINT8	RO	0x6000:1D, 8
1A04:0F	SubIndex 0x015	15. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x1E (Data In 13))	UINT8	RO	0x6000:1E, 8
1A04:10	SubIndex 0x016	16. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x1F (Data In 14))	UINT8	RO	0x6000:1F, 8
1A04:11	SubIndex 0x017	17. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x20 (Data In 15))	UINT8	RO	0x6000:20, 8
1A04:12	SubIndex 0x018	17. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x21 (Data In 16))	UINT8	RO	0x6000:21, 8
1A04:13	SubIndex 0x019	19. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x22 (Data In 17))	UINT8	RO	0x6000:22, 8
1A04:14	SubIndex 0x020	20. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x23 (Data In 18))	UINT8	RO	0x6000:23, 8
1A04:15	SubIndex 0x021	21. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x24 (Data In 19))	UINT8	RO	0x6000:24, 8
1A04:16	SubIndex 0x022	22. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x25 (Data In 20))	UINT8	RO	0x6000:25, 8
1A04:17	SubIndex 0x023	23. PDO 映射条目 (对象 0x6000 (COM Inputs Ch.1), 条目 0x26 (Data In 21))	UINT8	RO	0x6000:26, 8

## Index 0x1A05 COM TxPDO-Map Inputs Ch. 2

Index (Hex)	名称	含义	数据类型	标志	默认值
1A05:0	COM TxPDO-Map Inputs Ch. 2	PDO 映射 TxPDO 1	UINT8	RO	0x17 (23 <sub>dec</sub> )
1A05:01	SubIndex 0x001	1. PDO 映射条目 (对象 0x6011 (Status Ch.2), 条目 0x01 (Status))	UINT16	RO	0x6011:01, 16
1A05:02	SubIndex 0x002	2. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x11 (Data In 0))	UINT8	RO	0x6010:11, 8
1A05:03	SubIndex 0x003	3. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x12 (Data In 1))	UINT8	RO	0x6010:12, 8
1A05:04	SubIndex 0x004	4. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x13 (Data In 2))	UINT8	RO	0x6010:13, 8
1A05:05	SubIndex 0x005	5. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x14 (Data In 3))	UINT8	RO	0x6010:14, 8
1A05:06	SubIndex 0x006	6. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x15 (Data In 4))	UINT8	RO	0x6010:15, 8
1A05:07	SubIndex 0x007	7. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x16 (Data In 5))	UINT8	RO	0x6010:16, 8
1A05:08	SubIndex 0x008	8. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x17 (Data In 6))	UINT8	RO	0x6010:17, 8
1A05:09	SubIndex 0x009	9. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x18 (Data In 7))	UINT8	RO	0x6010:18, 8
1A05:0A	SubIndex 0x010	10. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x19 (Data In 8))	UINT8	RO	0x6010:19, 8
1A05:0B	SubIndex 0x011	11. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x1A (Data In 9))	UINT8	RO	0x6010:1A, 8
1A05:0C	SubIndex 0x012	12. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x1B (Data In 10))	UINT8	RO	0x6010:1B, 8
1A05:0D	SubIndex 0x013	13. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x1C (Data In 11))	UINT8	RO	0x6010:1C, 8
1A05:0E	SubIndex 0x014	14. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x1D (Data In 12))	UINT8	RO	0x6010:1D, 8
1A05:0F	SubIndex 0x015	15. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x1E (Data In 13))	UINT8	RO	0x6010:1E, 8
1A05:10	SubIndex 0x016	16. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x1F (Data In 14))	UINT8	RO	0x6010:1F, 8
1A05:11	SubIndex 0x017	17. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x20 (Data In 15))	UINT8	RO	0x6010:20, 8
1A05:12	SubIndex 0x018	17. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x21 (Data In 16))	UINT8	RO	0x6010:21, 8
1A05:13	SubIndex 0x019	19. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x22 (Data In 17))	UINT8	RO	0x6010:22, 8
1A05:14	SubIndex 0x020	20. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x23 (Data In 18))	UINT8	RO	0x6010:23, 8
1A05:15	SubIndex 0x021	21. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x24 (Data In 19))	UINT8	RO	0x6010:24, 8
1A05:16	SubIndex 0x022	22. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x25 (Data In 20))	UINT8	RO	0x6010:25, 8
1A05:17	SubIndex 0x023	23. PDO 映射条目 (对象 0x6010 (COM Inputs Ch.2), 条目 0x26 (Data In 21))	UINT8	RO	0x6010:26, 8

## Index 0x1C00 Sync manager type

Index (Hex)	名称	含义	数据类型	标志	默认值
1C00:0	Sync manager type	使用 Sync Manager	UINT8	RO	0x04 (4 <sub>dec</sub> )
1C00:01	SubIndex 0x001	Sync Manager 类型通道 1: 邮箱写入	UINT8	RO	0x01 (1 <sub>dec</sub> )
1C00:02	SubIndex 0x002	Sync Manager 类型通道 2: 邮箱读取	UINT8	RO	0x02 (2 <sub>dec</sub> )
1C00:03	SubIndex 0x003	Sync Manager 类型通道 3: 过程数据写入 (输出)	UINT8	RO	0x03 (3 <sub>dec</sub> )
1C00:04	SubIndex 0x004	Sync Manager 类型通道 4: 过程数据读取 (输入)	UINT8	RO	0x04 (4 <sub>dec</sub> )

Index 0x1C12 RxPDO assign

Index (Hex)	名称	含义	数据类型	标志	默认值
1C12:0	RxPDO assign	PDO 分配输出	UINT8	RO	0x04 (4 <sub>dec</sub> )
1C12:01	SubIndex 0x001	第 1 个分配的 RxPDO (包含相关 RxPDO 映射对象的 Index 0x)	UINT16	RO	0x1600 (5632 <sub>dec</sub> )
1C12:02	SubIndex 0x002	第 2 个分配的 RxPDO (包含相关 RxPDO 映射对象的 Index 0x)	UINT16	RO	0x1601 (5633 <sub>dec</sub> )

Index 0x1C13 TxPDO assign

Index (Hex)	名称	含义	数据类型	标志	默认值
1C13:0	TxPDO assign	PDO 分配输入	UINT8	RO	0x04 (4 <sub>dec</sub> )
1C13:01	SubIndex 0x001	第 1 个分配的 TxPDO (包含相关 TxPDO 映射对象的 Index 0x)	UINT16	RO	0x1A00 (6656 <sub>dec</sub> )
1C13:02	SubIndex 0x002	第 2 个分配的 TxPDO (包含相关 TxPDO 映射对象的 Index 0x)	UINT16	RO	0x1A01 (6657 <sub>dec</sub> )

Index 0x1C32 SM output parameter

Index (Hex)	名称	含义	数据类型	标志	默认值
1C32:0	SM output parameter	输出的 SM 同步参数	UINT8	RO	0x20 (32 <sub>dec</sub> )
1C32:01	Sync mode	当前的同步模式： <ul style="list-style-type: none"> <li>0: Free Run</li> <li>1: 与 SM 2 事件同步</li> <li>2: DC 模式 - 与 SYNC0 事件同步</li> <li>3: DC 模式 - 与 SYNC1 事件同步</li> </ul>	UINT16	RW	0x0000 (0 <sub>dec</sub> )
1C32:02	Cycle time	周期时间 (单位: ns)： <ul style="list-style-type: none"> <li>Free Run: 本地定时器的周期时间</li> <li>与 SM 2 事件同步: 主站周期时间</li> <li>DC 模式: SYNC0/SYNC1 周期时间</li> </ul>	UINT32	RW	0x0003D090 (250000 <sub>dec</sub> )
1C32:03	Shift time	从 SYNC0 事件到输出的时间 (单位: ns, 仅 DC 模式)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C32:04	Sync modes supported	支持的同步模式： <ul style="list-style-type: none"> <li>位 0 = 1: 支持 Free Run</li> <li>位 1 = 1: 支持与 SM 2 事件同步</li> <li>位 2-3 = 01: 支持 DC 模式</li> <li>位 4-5 = 10: SYNC1 事件下的输出偏移 (仅 DC 模式)</li> <li>位 14 = 1: 动态周期 (在写入 1C32:08 时开始测量)</li> </ul>	UINT16	RO	0xC007 (49159 <sub>dec</sub> )
1C32:05	Minimum cycle time	最小周期时间 (单位: ns)	UINT32	RO	0x00004E20 (20000 <sub>dec</sub> )
1C32:06	Calc and copy time	SYNC0 和 SYNC1 事件之间的最小时间 (单位: ns, 仅 DC 模式)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C32:08	Command	<ul style="list-style-type: none"> <li>0: 本地周期时间的测量停止</li> <li>1: 本地周期时间的测量开始</li> </ul> 条目 1C32:03、1C32:05、1C32:06、1C32:09、1C33:03、1C33:06、1C33:09 更新为最大测量值。对于后续测量, 测量值被重置	UINT16	RW	0x0000 (0 <sub>dec</sub> )
1C32:09	Delay time	从 SYNC1 事件到输出的时间 (单位: ns, 仅 DC 模式)	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C32:0B	SM event missed counter	OPERATIONAL 期间缺失的 SM 事件数量 (仅 DC 模式)	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C32:0C	Cycle exceeded counter	OPERATIONAL 期间超过周期时间的次数 (周期没有及时完成或下一个周期开始得太早)	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C32:0D	Shift too short counter	SYNC0 和 SYNC1 事件之间的间隔次数太小 (仅 DC 模式)	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C32:20	Sync error	在上一周期同步出错 (输出太晚; 仅 DC 模式)	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )

## Index 0x1C33 SM input parameter

Index (Hex)	名称	含义	数据类型	标志	默认值
1C33:0	SM input parameter	输入的SM同步参数	UINT8	RO	0x20 (32 <sub>dec</sub> )
1C33:01	Sync mode	当前的同步模式： <ul style="list-style-type: none"> <li>0: Free Run</li> <li>1: 与 SM 3 事件同步（无输出可用）</li> <li>2: DC - 与 SYNC0 事件同步</li> <li>3: DC - 与 SYNC1 事件同步</li> <li>34: 与 SM 2 事件同步（输出可用）</li> </ul>	UINT16	RW	0x0000 (0 <sub>dec</sub> )
1C33:02	Cycle time	如 1C32:02	UINT32	RW	0x0003D090 (250000 <sub>dec</sub> )
1C33:03	Shift time	从 SYNC0 事件到读取输入的时间（单位：ns，仅 DC 模式）	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C33:04	Sync modes supported	支持的同步模式： <ul style="list-style-type: none"> <li>位 0: 支持 Free Run</li> <li>位 1: 支持与 SM 2 事件同步（输出可用）</li> <li>位 1: 支持与 SM 3 事件同步（无输出可用）</li> <li>位 2-3 = 01: 支持 DC 模式</li> <li>位 4-5=01: 基于本地事件的输入偏移（输出可用）</li> <li>位 4-5 = 10: 基于 SYNC1 事件的输入偏移（无输出可用）</li> <li>位 14 = 1: 动态周期（从写入 1C32:08 或 1C33:08 开始测量）</li> </ul>	UINT16	RO	0xC007 (49159 <sub>dec</sub> )
1C33:05	Minimum cycle time	如 1C32:05	UINT32	RO	0x00004E20 (20000 <sub>dec</sub> )
1C33:06	Calc and copy time	从读取输入到主站可以使用输入值之间的时间（单位：ns，仅 DC 模式）	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C33:08	Command	如 1C32:08	UINT16	RW	0x0000 (0 <sub>dec</sub> )
1C33:09	Delay time	从 SYNC1 事件到读取输入的时间（单位：ns，仅 DC 模式）	UINT32	RO	0x00000000 (0 <sub>dec</sub> )
1C33:0B	SM event missed counter	如 1C32:11	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C33:0C	Cycle exceeded counter	如 1C32:12	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C33:0D	Shift too short counter	如 1C32:13	UINT16	RO	0x0000 (0 <sub>dec</sub> )
1C33:20	Sync error	如 1C32:32	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )

### 8.1.3 描述文件定义的对象 (0x6000-0xFFFF) [硬件版本 03 及以上]

对于所有支持 5001 设备描述规约的 EtherCAT 从站，描述文件定义的对象具有相同的含义。

Index 0x60n0 COM Inputs Ch. 1 (n = 0) , Ch. 2 (n = 1)

Index (Hex)	名称	含义	数据类型	标志	默认值
60n0:0	COM Inputs Ch. 1 + Ch. 2	最大 SubIndex (Hex)	UINT8	RO	0x26 (38 <sub>dec</sub> )
60n0:01	Transmit accepted	端子模块通过改变该位的状态确认已收到数据。然后，端子模块才会接收从控制器发来的新数据。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
60n0:02	Receive request	端子模块通过改变这个状态位来通知控制器，DataIn 数组包含了“Input length [▮_197]”中显示的字节数。控制器必须通过改变 ReceiveAccepted [▮_197] 状态位来确认已收到数据。然后，端子模块才会向控制器发送新数据。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
60n0:03	Init accepted	0 端子模块再次准备就绪，可进行串行数据交换。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
		1 端子模块已完成初始化。			
60n0:04	Buffer full	接收 FIFO 已满。所有后续到来的输入数据都将丢失！	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
60n0:05	Parity error	发生奇偶校验错误。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
60n0:06	Framing error	发生帧错误。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
60n0:07	Overrun error	发生超限错误。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
60n0:09	Input length	等待从端子模块发送到控制器的输入字节数。	UINT8	RO	0x00 (0 <sub>dec</sub> )
60n0:11	Data In 0	输入字节 0	UINT8	RO	0x00 (0 <sub>dec</sub> )
...	...	....	...	...	...
60n0:26	Data In 21	输入字节 21	UINT8	RO	0x00 (0 <sub>dec</sub> )

Index 0x60n1 Status Ch. 1 (n = 0) , Ch. 2 (n = 1)

Index (Hex)	名称	含义	数据类型	标志	默认值
60n1:0	Status Ch. 1 + Ch. 2	最大 SubIndex (Hex)	UINT8	RO	0x01 (01 <sub>dec</sub> )
60n1:01	Status	状态字	UINT16	RO	0x00 (0 <sub>dec</sub> )

Index 0x70n0 COM Outputs Ch. 1 (n = 0) , Ch. 2 (n = 1)

Index (Hex)	名称	含义	数据类型	标志	默认值
70n0:0	COM Outputs Ch. 1 + Ch. 2	最大 SubIndex (Hex)	UINT8	RO	0x26 (38 <sub>dec</sub> )
70n0:01	Transmit request	控制器通过改变该位的状态通知端子模块，DataOut 字节包含“Output length [▮_197]”中显示的字节数。端子模块通过改变“TransmitAccepted [▮_197]”位的状态确认已收到数据。然后，端子模块才会接收从控制器发来的新数据。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
70n0:02	Receive accepted	控制器通过改变该位的状态确认已收到数据。然后，端子模块才会向控制器发送新数据。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
70n0:03	Init request	0 控制器再次请求端子模块准备进行串行数据交换。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
		1 控制器请求端子模块进行初始化。发送和接收功能将被阻止，FIFO 指针将重置，串口将使用相关 Settings 对象的值进行初始化。端子模块将通过“Init accepted [▮_197]”位确认初始化的执行。			
70n0:04	Send continuous	从 FIFO 连续发送数据。来自控制器的数据先填满端子模块的发送缓冲区（最多 128 字节）。填充的缓冲区内容将在该位的上升沿发送。如果数据已发送，端子模块会通过设置“Init accepted [▮_197]”位通知控制器。“Init accepted [▮_197]”通过“Send continuous [▮_197]”清除。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
70n0:09	Output length	从控制器发送到端子模块的等待输出字节数。	UINT8	RO	0x00 (0 <sub>dec</sub> )
70n0:11	Data Out 0	输出字节 0	UINT8	RO	0x00 (0 <sub>dec</sub> )
...	...	....	...	...	...
70n0:26	Data Out 21	输出字节 21	UINT8	RO	0x00 (0 <sub>dec</sub> )

## Index 0x70n1 Ctrl Ch. 1 (n = 0) , Ch. 2 (n = 1)

Index (Hex)	名称	含义	数据类型	标志	默认值
70n1:0	Ctrl Ch. 1 + Ch. 2	最大 SubIndex (Hex)	UINT8	RO	0x01 (0 <sub>dec</sub> )
70n1:01	Status	控制字	UINT16	RO	0x00 (0 <sub>dec</sub> )

## Index 0xA0n0 COM Diag data Ch. 1 (n = 0) , Ch. 2 (n = 1)

Index (Hex)	名称	含义	数据类型	标志	默认值
A0n0:0	COM Diag data Ch. 1 + Ch. 2	最大 SubIndex (Hex)	UINT8	RO	0x12 (18 <sub>dec</sub> )
A0n0:01	Buffer overflow	缓冲区溢出。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
A0n0:02	Parity error	发生奇偶校验错误。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
A0n0:03	Framing error	发生帧错误	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
A0n0:04	Overrun error	发生超限错误。	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
A0n0:05	Buffer full	接收 FIFO 已满。所有后续到来的输入数据都将丢失!	BOOLEAN	RO	0x00 (0 <sub>dec</sub> )
A0n0:11	Data bytes in send buffer	发送 FIFO 中的数据字节数	UINT16	RO	0x0000 (0 <sub>dec</sub> )
A0n0:12	Data bytes in receive buffer	接收 FIFO 中的数据字节数	UINT16	RO	0x0000 (0 <sub>dec</sub> )

## Index 0xF000 Modular device profile

Index (Hex)	名称	含义	数据类型	标志	默认值
F000:0	Modular device profile	模块化设备描述文件的常规信息	UINT8	RO	0x02 (2 <sub>dec</sub> )
F000:01	Module Index Oxdistance	通道之间对象 Index (Hex) 的间隔	UINT16	RO	0x0010 (16 <sub>dec</sub> )
F000:02	Maximum number of modules	通道数量	UINT16	RO	0x0004 (4 <sub>dec</sub> )

## Index 0xF008 Code word

Index (Hex)	名称	含义	数据类型	标志	默认值
F008:0	Code word	保留	UINT32	RW	0x00000000 (0 <sub>dec</sub> )

## Index 0xF010 Module list

Index (Hex)	名称	含义	数据类型	标志	默认值
F010:0	Module list	最大 SubIndex (Hex)	UINT8	RW	0x04 (4 <sub>dec</sub> )
F010:01	SubIndex 0x001	-	UINT32	RW	0x00000258 (600 <sub>dec</sub> )
F010:02	SubIndex 0x002	-	UINT32	RW	0x00000258 (600 <sub>dec</sub> )

## 8.2 控制字和状态字

控制字和状态字位于输入和输出过程映像的前 16 位。端子模块与控制器之间的通信借助这两个 word 进行控制。

### 状态字

数位	名称	含义	数据类型
0	Transmit accepted	端子模块通过改变该位的状态确认已收到数据。然后，端子模块才会接收从控制器发来的新数据。	BOOLEAN
1	Receive request	端子模块通过改变该位的状态通知控制器，DataIn 字节包含“Input length”中显示的字节数。控制器必须通过改变 ReceiveAccepted 位的状态确认已收到数据。然后，端子模块才会向控制器发送新数据。	BOOLEAN
2	Init accepted	0 端子模块再次准备就绪，可进行串行数据交换。	BOOLEAN
		1 端子模块已完成初始化。	
3	Buffer full	接收 FIFO 已满。所有后续到来的输入数据都将丢失！	BOOLEAN
4	Parity error	发生奇偶校验错误。	BOOLEAN
5	Framing error	发生帧错误。	BOOLEAN
6	Overrun error	发生超限错误。	BOOLEAN
7	-		
8...15	Input length	等待从端子模块发送到控制器的输入字节数。	UINT8

### 控制数据

数位	名称	含义	数据类型
0	Transmit request	控制器通过改变该位的状态来通知端子模块，DataOut 字节包含“Output length”中显示的字节数。端子模块通过改变“TransmitAccepted”位的状态确认已收到数据。然后，端子模块才会接收从控制器发来的新数据。	BOOLEAN
1	Receive accepted	控制器通过改变该位的状态确认已收到数据。然后，端子模块才会向控制器发送新数据。	BOOLEAN
2	Init request	0 控制器再次请求端子模块准备进行串行数据交换。	BOOLEAN
		1 控制器请求端子模块进行初始化。发送和接收功能将被阻止，FIFO 指针将重置，串口将使用相关 Settings 对象的值进行初始化。端子模块将通过“Init accepted”位确认初始化的执行。	
3	Send continuous	从 FIFO 连续发送数据。 来自控制器的数据先填满端子模块的发送缓冲区（最多 128 字节）。填充的缓冲区内容将在该位的上升沿发送。如果数据已发送，端子模块会通过设置“Init accepted”位通知控制器。“Init accepted”通过“Send continuous”清除。	BOOLEAN
4...7	-		
8...15	Output length	从控制器发送到端子模块的等待输出字节数。	UINT8

### PDO 分配

端子模块为每个通道提供两套 Inputs/Outputs 过程映像。它们的区别仅在于 control/status 数据的表示方式不同。

在按位表示的情况下，数据如上述各表所示。在按字表示的情况下，过程数据的前 16 位合并为一个字。

标准的过程数据对象以 control/status 开始。这种 PDO 分配是使用“TwinCAT PLC Serial Communication”库要求的。

状态输入 Status	控制输出 Control	表示法
0x1A00	0x1600	按位表示, 通道 1
0x1A01	0x1601	按位表示, 通道 2
0x1A02	0x1602	按位表示, 通道 3
0x1A03	0x1603	按位表示, 通道 4
0x1A04	0x1604	按字表示, 通道 1
0x1A05	0x1605	按字表示, 通道 2
0x1A06	0x1606	按字表示, 通道 3
0x1A07	0x1607	按字表示, 通道 4



## 9 附录

### 9.1 EtherCAT AL 状态代码

详细信息请参见 [EtherCAT系统描述](#)。

## 9.2 固件兼容性

倍福 EtherCAT 设备在交付时都装有最新的固件版本。固件和硬件必须相互兼容；但不是每种组合都能确保兼容性。下面的概述显示了可以运行固件的硬件版本。

### 注意

- 建议为相应的硬件使用可用的最新固件
- 对于已交付的产品，倍福没有任何义务为客户提供免费固件更新。

### 注意

#### 设备损坏风险！

请注意单独页面 [▶ 204] 上的固件更新说明。

如果在固件更新时设备处于 BOOTSTRAP 模式，则在下载新固件时不会检查新固件的适用性。这可能导致设备损坏！因此，请务必确保固件适用于硬件版本！

EL6001			
硬件 (HW)	固件 (FW)	修订版本号	发布日期
00 - 02	01		05/2006
	02		12/2006
	03		04/2008
	04		04/2008
03 - 17*	05	EL6001-0000-0016	12/2009
	06		04/2010
	07	EL6001-0000-0017	03/2011
		EL6001-0000-0018	10/2012
	08	EL6001-0000-0019	05/2014
	09	EL6001-0000-0020	08/2014
	10		05/2015
	11		06/2017
	12	EL6001-0000-0021	01/2023
	13		05/2023
	14		07/2023
	15	EL6001-0000-0022	01/2024
16*		02/2024	

EL6002			
硬件 (HW)	固件 (FW)	修订版本号	发布日期
00 - 18*	01	EL6002-0000-0016	12/2009
	02		06/2010
	03	EL6002-0000-0017	11/2012
		EL6002-0000-0018	08/2013
	04*	EL6002-0000-0019	05/2014

EL6021			
硬件 (HW)	固件 (FW)	修订版本号	发布日期
00 - 02	01		05/2005
	02		12/2006
	03		04/2008
03 - 17*	04	EL6021-0000-0016	11/2009
	05		04/2010
		EL6021-0000-0017	10/2012
	06	EL6021-0000-0018	08/2013
	07	EL6021-0000-0019	06/2014
	08	EL6021-0000-0020	10/2014
	09	EL6021-0000-0021	01/2020
	10*	EL6021-0000-0022	01/2023

EL6021-0021			
硬件 (HW)	固件 (FW)	修订版本号	发布日期
00 - 03*	06	EL6021-0021-0018	09/2013
	07	EL6021-0021-0019	06/2014
	08	EL6021-0021-0020	10/2014
	09	EL6021-0021-0021	01/2020
	10	EL6021-0021-0022	01/2023

EL6022				
硬件 (HW)	固件 (FW)	修订版本号	发布日期	
00 - 19*	01	EL6022-0000-0016	01/2010	
	02		06/2010	
	03*			09/2011
		EL6022-0000-0017	08/2012	
		EL6022-0000-0018	08/2013	
		EL6022-0000-0019	03/2015	

\*) 这是在编写本文件的时兼容的固件/硬件版本。请在倍福网页上查看是否有更多最新文档。

## 9.3 固件更新 EL/ES/EM/ELM/EPxxxx

本节介绍了倍福 EL/ES、ELM、EM、EK 和 EP 系列 EtherCAT 从站设备的更新情况。只有在与倍福支持部门协商后才能进行固件更新。

### 注意

#### 仅使用 TwinCAT 3 软件!

必须在安装了 TwinCAT 3 之后才能进行倍福 IO 设备的固件更新。建议尽可能使用最新的固件，可在倍福公司网站上免费下载 <https://www.beckhoff.com/en-us/>。

为了更新固件，TwinCAT 可以在 FreeRun 模式下运行，不需要付费许可。

待更新的设备通常可以保留在安装位置，但 TwinCAT 必须在 FreeRun 模式下运行。请确保 EtherCAT 通讯良好（没有丢失帧等）。

不应使用其他 EtherCAT 主站软件，例如 EtherCAT Configurator，因为它们可能不支持复杂的更新固件、EEPROM 和其他设备组件。

### 储存地点

一个 EtherCAT 从站最多可以在三个位置上存储运行数据：

- 每个 EtherCAT 从站都有一个设备描述文件，包括标识（名称、产品代码）、时序定义、通信设置等。该设备描述文件（ESI: EtherCAT Slave Information）可以从 Beckhoff 网站下载区的 [zip 文件](#) 中下载，并在 EtherCAT 主站中用于离线组态，例如在 TwinCAT 中。最重要的是，每个 EtherCAT 从站都将其可供电子读取的设备描述文件（ESI）存放在其本地存储芯片，即 **ESI EEPROM** 中。从站上电以后，该描述文件将加载到从站本地，并告知其通信配置；另一方面，EtherCAT 主站可以通过这种方式识别从站，并相应地设置 EtherCAT 通信。

### 注意

#### 用项目定义的 ESI-EEPROM 写入

ESI 文件是设备制造商根据 ETG 标准为相应产品开发 and 发布的。

- ESI 文件的含义：禁止从使用侧（比如用户）进行修改。
- ESI EEPROM 的含义：即使技术上允许写入，EEPROM 中的 ESI 部分和可能存在的空闲存储区域也不得在正常更新过程之外进行更改。特别是对于周期性的内存写入（运行时间计数器等），必须使用专门的存储器产品，例如 EL6080 或 IPC 自己的 NOVRAM（掉电保持存储器选项）。

- 根据功能和性能的不同，EtherCAT 从站有一个或几个本地处理器来处理 I/O 数据。相应的程序就称作 Firmware **固件**，文件格式为 \*.efw。
- 在一些 EtherCAT 从站中，EtherCAT 通讯也可能集成在这些本地处理器中。此时，本地处理器通常是一个 **FPGA** 芯片，带有 \*.rbf 固件。

客户可以通过 EtherCAT 现场总线及其通讯机制来访问 Firmware（固件）。Firmware 的更新或读取是通过非周期性邮箱通信（mailbox）或对 ESC 的寄存器访问实现的。

如果要更新从站的固件，TwinCAT System Manager 提供使用新固件刷新上述三处运行数据的机制。从站通常不会检查新的固件是否合适，也就是说，如果下载了错误的固件，从站可能就无法再运行。

### 通过 bundle firmware（捆绑固件）简化更新

使用所谓的 **bundle firmware（捆绑固件）** 进行更新更为方便：此时从站处理器的固件和 ESI 描述组合在一个 \*.efw 文件中；固件更新期间，在端子模块中的 Firmware 和 ESI 都会改变。要实现这种功能，要求以下几点：

- 固件为打包格式：可通过文件名识别，其中还包含修订版本号，例如 ELxxxx-xxxx\_REV0016\_SW01.efw
- 在下载对话框中输入密码=1 时，使用捆绑固件更新。如果密码=0（默认设置），则只进行固件更新，不进行 ESI 更新。
- 只用于支持此功能的设备。打包文件的内容通常不能再修改；这个功能是自 2016 年以来诸多新开发功能的一部分。

更新之后，应确认是否成功

- ESI/Revision: 例如，通过 TwinCAT ConfigMode/FreeRun 中的在线扫描，这是确定固件修订版本的简便方法

- Firmware: 例如, 通过查看设备的 CoE Online 数据

**注意**

**设备损坏风险!**

- ✓ 下载新设备文件时注意以下几点
  - a) EtherCAT 设备的固件下载不能中断
  - b) 必须确保通畅的 EtherCAT 通讯。必须避免 CRC 错误或丢帧。
  - c) 供电必须稳定。信号电平必须符合规范。
- ⇒ 如果在更新过程中出现故障, EtherCAT 设备可能无法使用, 只能返回制造商重新调试。

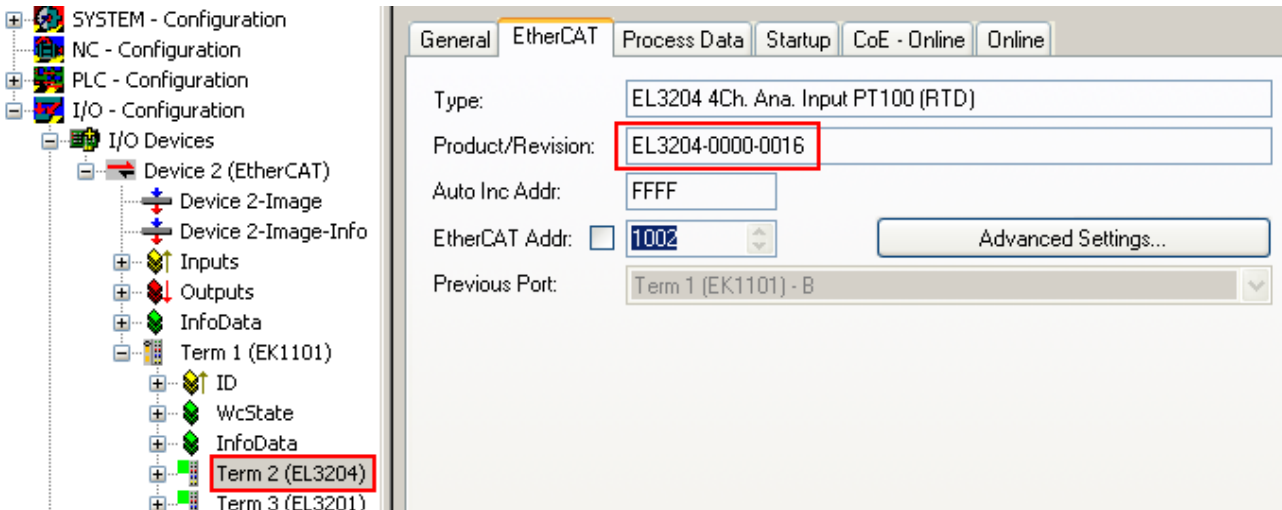
### 9.3.1 设备描述 ESI 文件/XML

**注意**

**关于更新 ESI 描述文件/EEPROM 的注意事项**

一些从站在 EEPROM 中存储了用于生产的校准和配置数据。在更新过程中, 这些信息会被覆盖, 无法恢复。

ESI 设备描述存储在从站上, 并在启动时加载。每个设备描述都有一个唯一标识符, 包括从站名称 (9 个字符/9 位数) 和修订版本号 (4 位数)。在 System Manager 中配置的每个从站都在 EtherCAT 选项卡中显示其标识符:



附图 161: 由名称 EL3204-0000 和修订版本号 0016 组成的设备标识符

配置的标识符必须与作为硬件使用的实际设备描述兼容, 即从站在启动时加载的描述 (本例中为 EL3204)。通常情况下, 配置的版本必须与端子模块网络中实际存在的版本相同或更低。

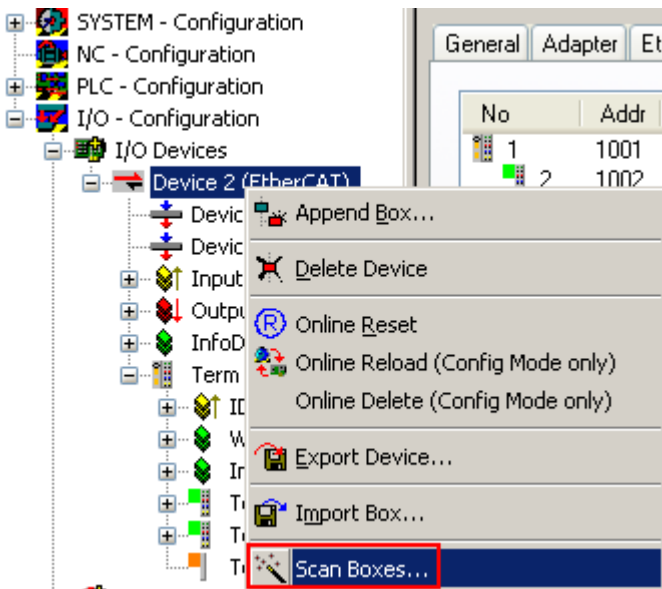
有关这方面的进一步信息, 请参考 [EtherCAT 系统文件](#)。

**● XML/ESI 描述的更新**

**I** 设备的修订版本与所使用的 Firmware (固件) 和 Hardware (硬件) 密切相关。不兼容的组合会导致故障, 甚至使设备最终关闭。只有在与倍福支持 (售后) 部门协商后才能进行相应的更新。

**ESI 从站标识符的显示**

确定所配置的设备描述和实际设备是否相符的最简单方法是在 TwinCAT Config Mode/FreeRun 模式下扫描 EtherCAT 从站:



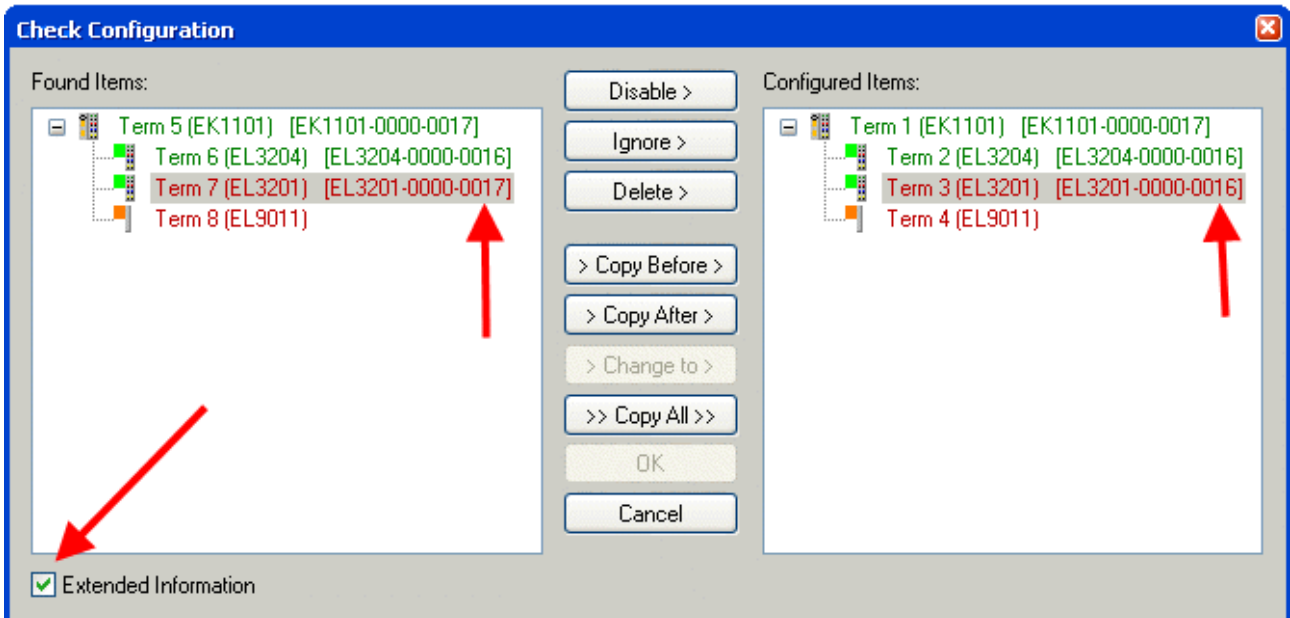
附图 162: 右键单击 EtherCAT Device 扫描下级从站

如果找到的内容与配置的内容相符，则显示



附图 163: 配置是相同的

否则就会出现一个更改对话框，用于选择实际配置。



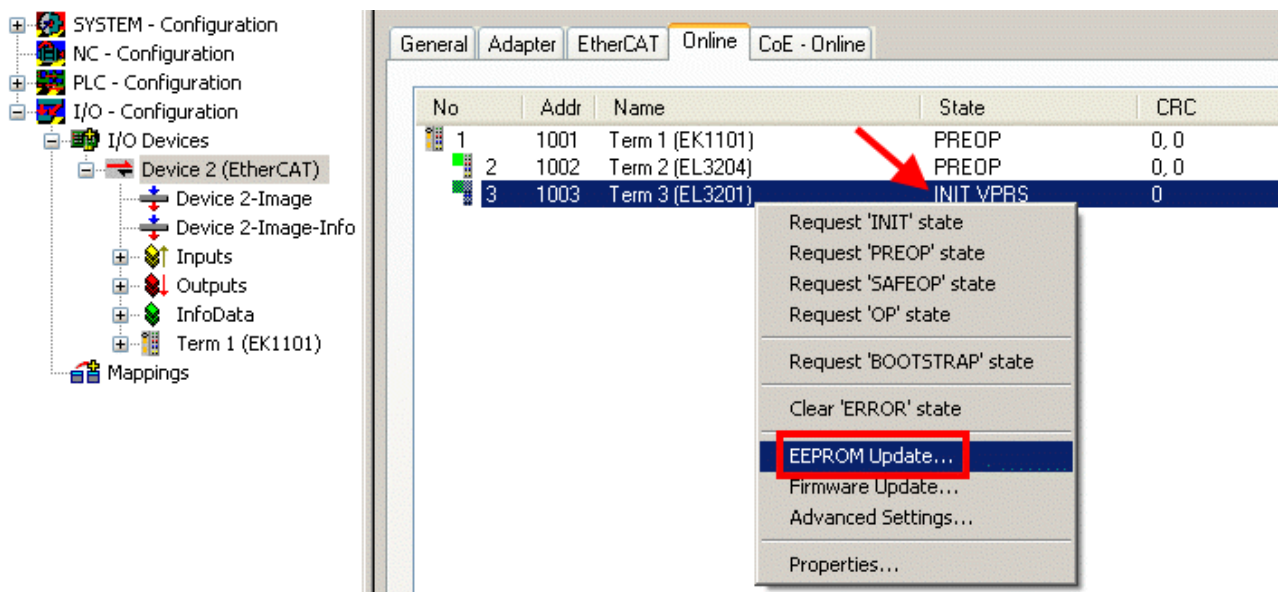
附图 164: 更改对话框

在图更改对话框的示例中，发现了一个 EL3201-0000-0017，而原配置中是 EL3201-0000-0016。此时可以通过 Copy Before 按钮来调整配置。必须选中 Extended Information 复选框，以显示修订版本。

### 更改 ESI 从站标识符

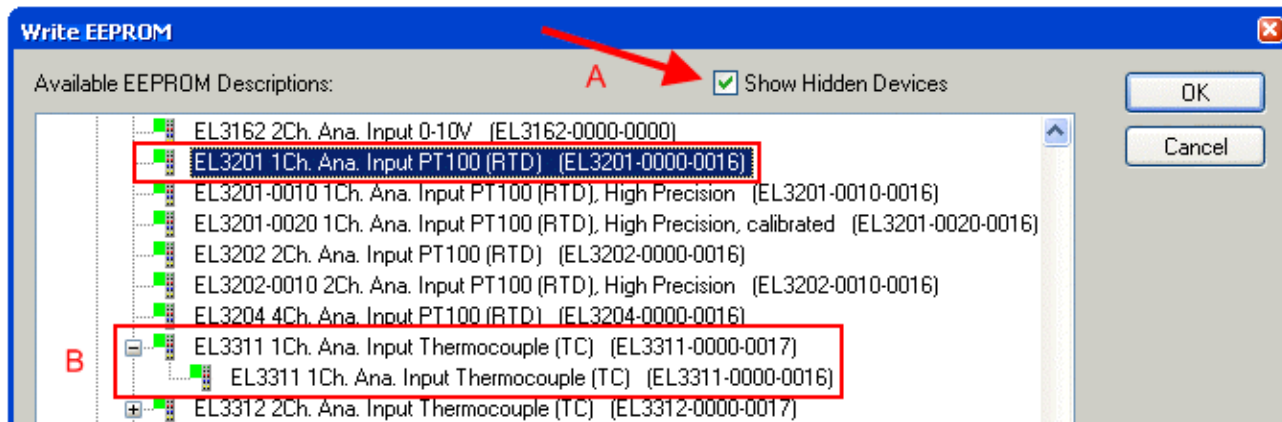
ESI/EEPROM 标识符可以在 TwinCAT 下按如下方式更新：

- 必须与从站建立正确无误的 EtherCAT 通讯。
- 从站的状态无关紧要。
- 右键单击 Online 中显示的从站，打开 *EEPROM Update* 对话框，参见图 *EEPROM 更新*



附图 165: EEPROM 更新

在以下对话框中选择新的 ESI 描述，参见图 *选择新的 ESI*。通过复选框 *Show Hidden Devices* 还能显示旧的、通常隐藏的从站版本。



附图 166: 选择新的 ESI

System Manager 弹出一个进度条，显示 EEPROM 写入的进度。首先写入数据，然后进行验证。

**●** 只有在设备重新启动后，以上更改才会生效。

**i** 大多数 EtherCAT 设备会立即或从 INIT 启动后读取修改后的 ESI 描述。一些通信设置（例如：分布时钟）只在开机时读取。因此，EtherCAT 从站必须短暂地关机，以使更改生效。

### 9.3.2 Firmware (固件) 说明

#### 确定固件版本

##### 通过 TwinCAT System Manager 确定版本

如果主站可以在线访问从站，TwinCAT System Manager 会显示从站处理器的固件版本。点击需要检查其处理器固件的 E-Bus 端子模块（在此例中为端子模块 2 (EL3204)），并选择选项卡 *CoE Online* (CAN over EtherCAT)。

#### ● CoE Online 和 Offline CoE (在线 CoE 和离线 CoE)

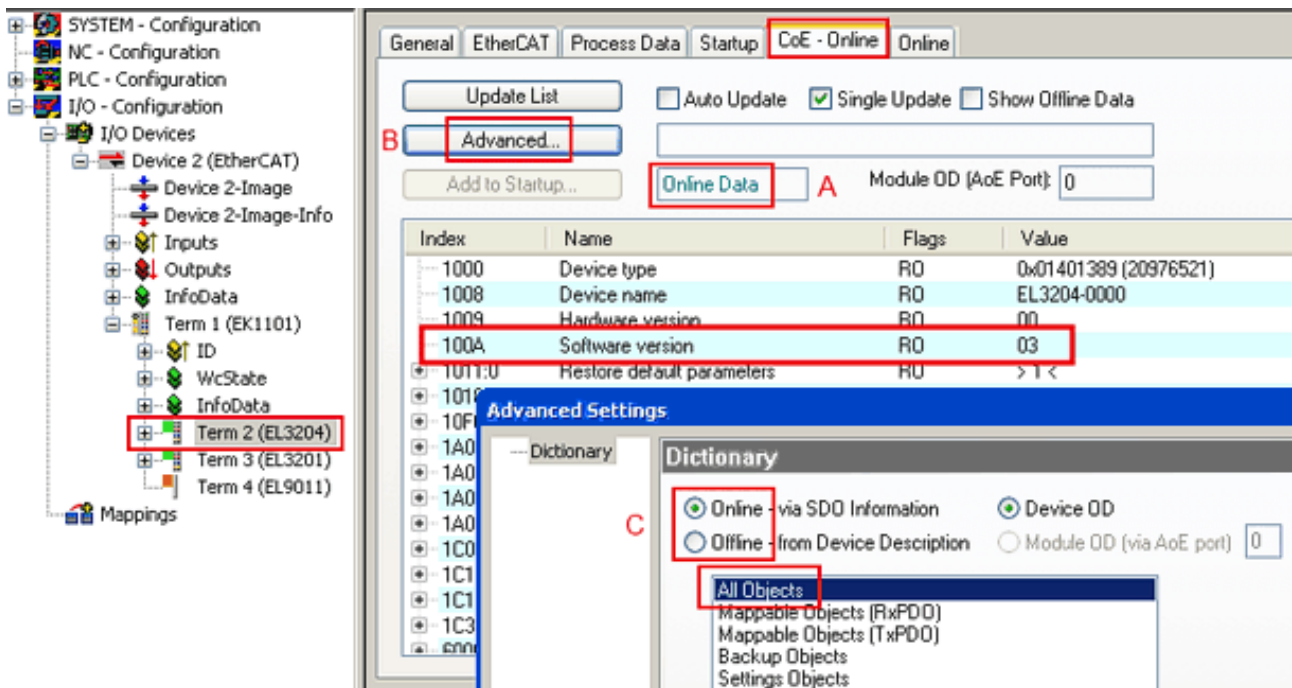
**i** 可用的 CoE 目录有两套：

**online:** 如果 EtherCAT 从站支持，从站处理器会提供该功能。该 CoE 目录只有在从站连接并运行时才能显示。

**Offline:** EtherCAT 从站信息文件 ESI/XML 包含的 CoE 默认内容。只有在 ESI 中包含了 CoE 目录才能显示（例如“倍福 EL5xxx.xml”）。

要在两个视图之间切换，必须使用按钮“Advanced”。

在图 EL3204 固件版本的显示中，所选 EL3204 的固件版本在 CoE 条目 0x100A 中显示为 03。



附图 167: EL3204 固件版本的显示

在 (A) 处，TwinCAT 2.11 表明当前显示的是“Online CoE”目录。如果不是，可以通过 Advanced Settings 中 (B) 处的 *Online* 选项和双击 *All Objects* 来加载 Online 目录。

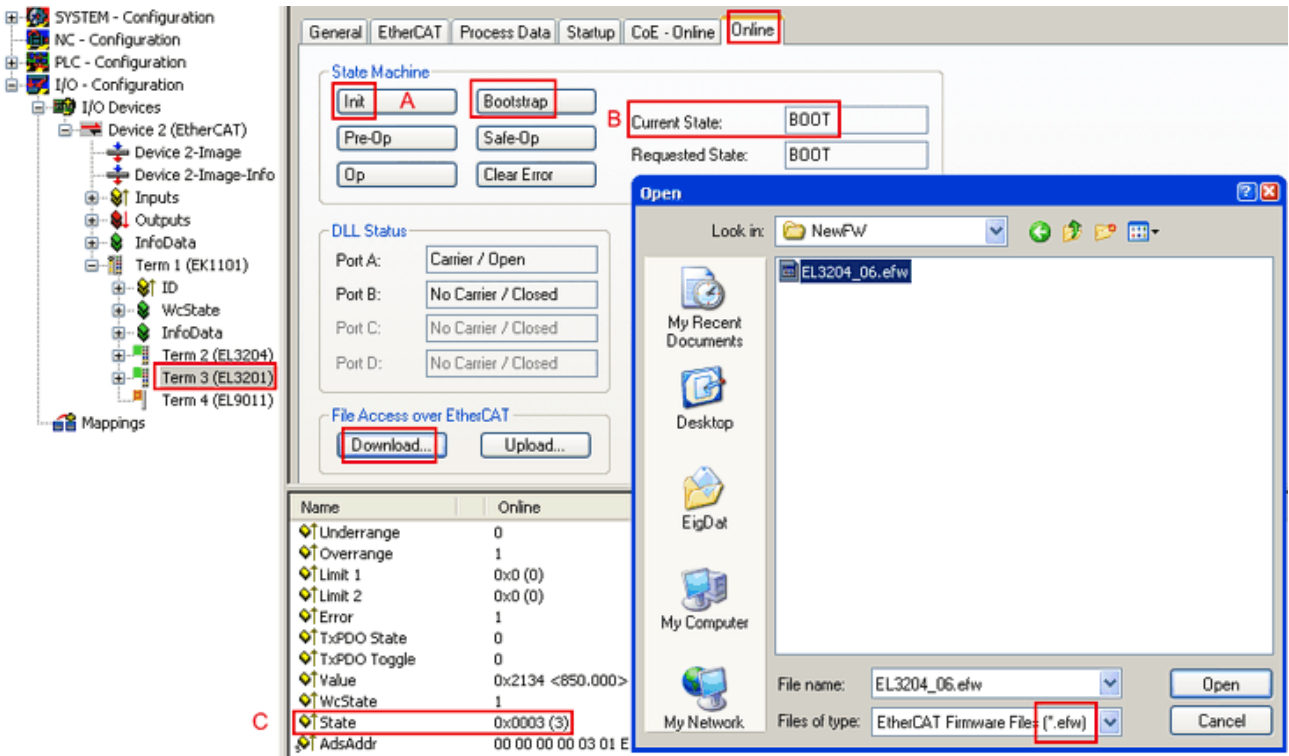
### 9.3.3 更新从站处理器的固件 \*.efw

#### ● CoE 目录

**i** Online CoE 目录由从站处理器管理，并存储在专用的 EEPROM 中，在固件更新期间一般不会改变。

切换到 *Online* 选项卡，更新从站处理器的固件，参见图 *固件更新*。

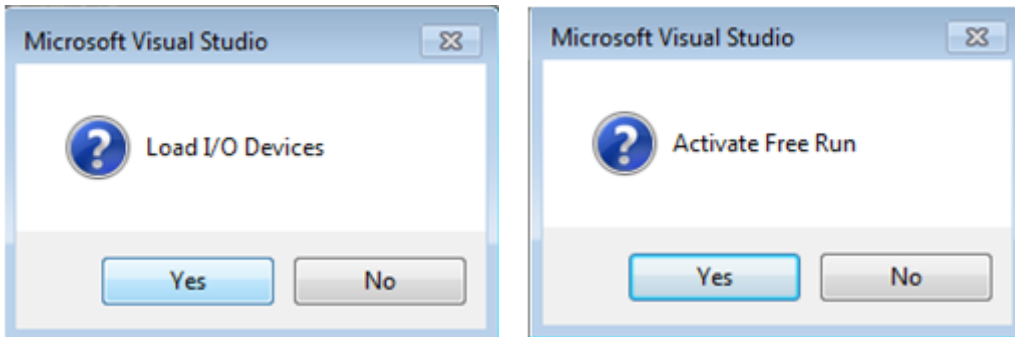




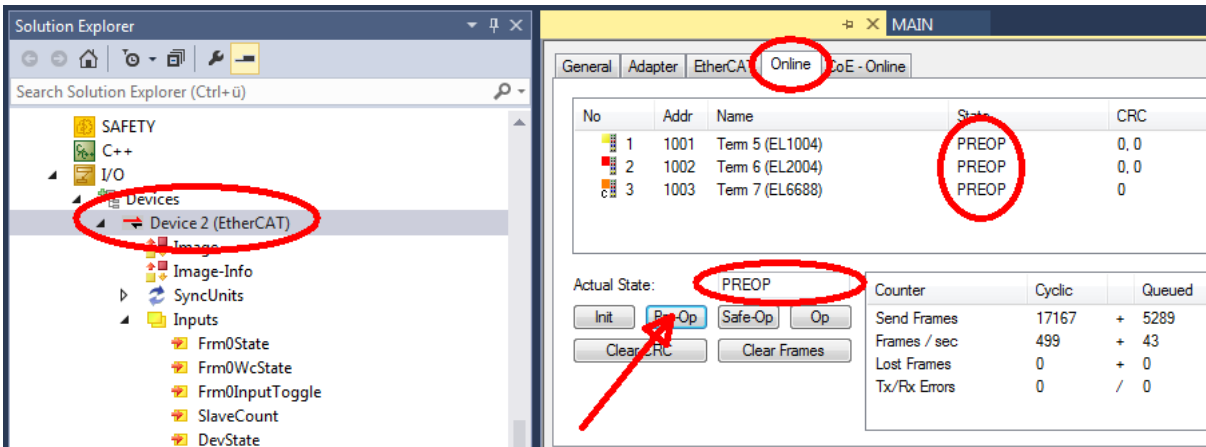
附图 168: 固件更新

除非倍福的支持（售后）部门另有说明，否则请按以下步骤进行。适用于 TwinCAT 2 和 TwinCAT 3 作为 EtherCAT 主站的情况。

- 将 TwinCAT 系统切换到 Config Mode/FreeRun，周期时间  $\geq 1$  ms（配置模式下默认为 4 ms）。不建议在实时核运行时（Running 模式）进行固件更新。

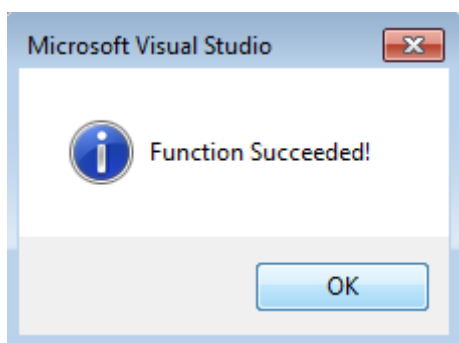


- 将 EtherCAT 主站切换到 PreOP



- 将从站切换到 INIT (A)
- 将从站切换到 BOOTSTRAP

- 检查当前状态（B、C）
- 下载新的 \*.efw文件（一直等待，直到下载结束）。通常不需要密码。



- 下载完成后，切换到 INIT，再到 PreOP
- 短时切断从站电源（不要拉低电压！）
- 在 CoE 0x100A 内检查固件状态（FW版本）是否被正确替换。

### 9.3.4 FPGA 固件 \*.rbf

如果是用 FPGA 芯片处理 EtherCAT 通信，固件更新则通过 \*.rbf 文件完成。

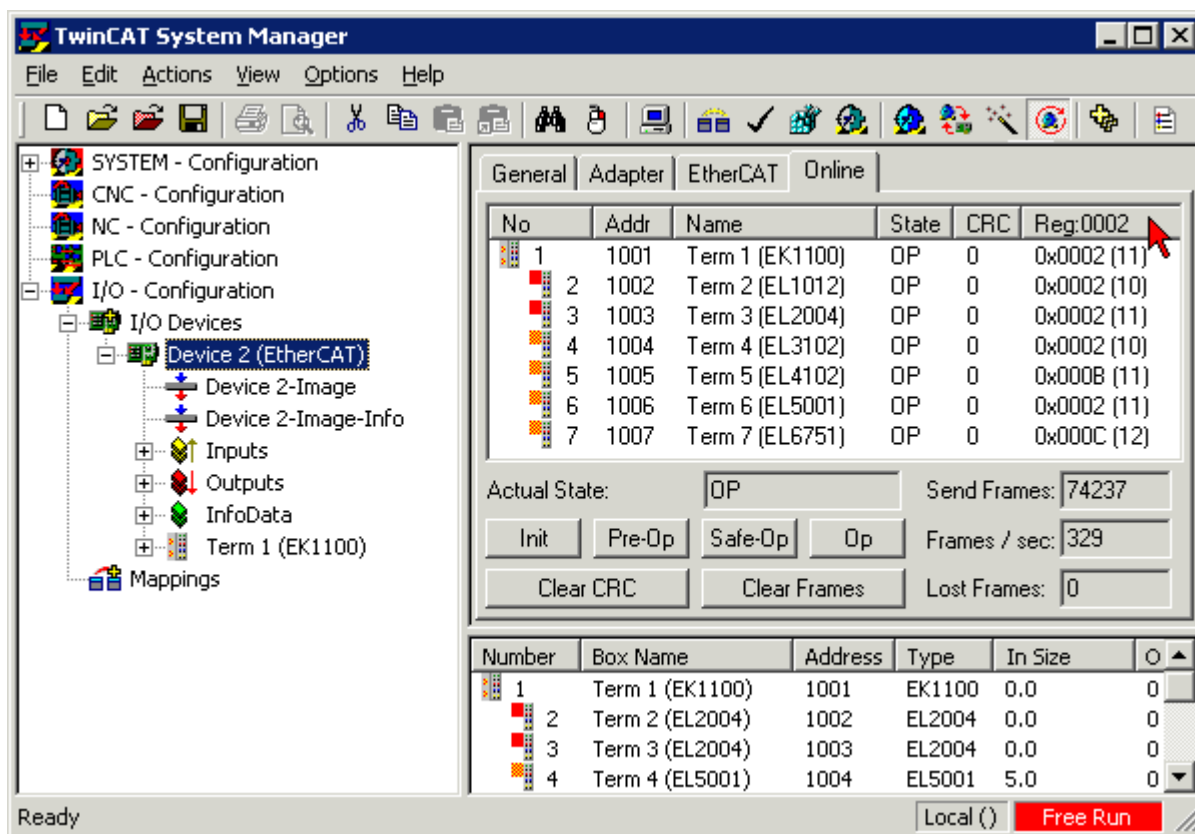
- 用于处理 I/O 信号的从站处理器固件
- 用于 EtherCAT 通讯的 FPGA 固件（仅适用于带 FPGA 的端子模块）

端子模块序列号中包含的固件版本号包含这两个固件成分。如果修改了其中任何一个，固件版本号都会更新。

#### 通过 TwinCAT System Manager 确定版本

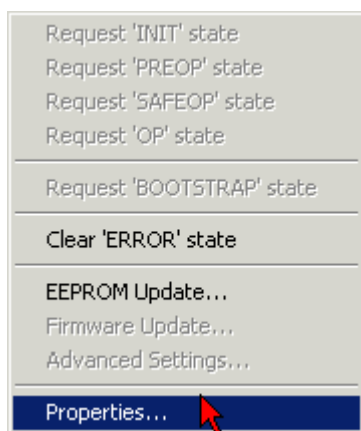
TwinCAT System Manager 显示 FPGA 固件版本。点击 EtherCAT 总线的以太网卡（例中的 Device 2），选择 *Online* 选项卡。

*Reg:0002* 栏表示各个 EtherCAT 设备的固件版本，以十六进制和十进制表示。



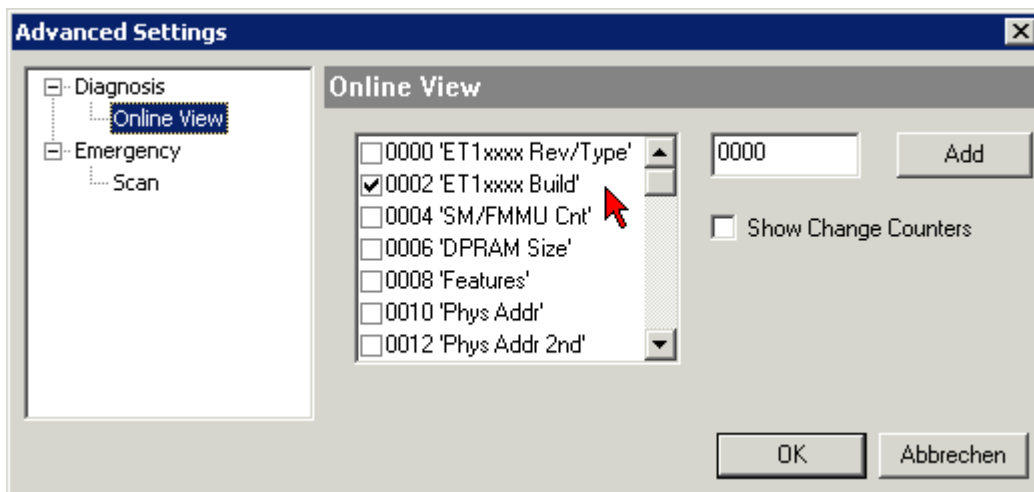
附图 169: FPGA 固件版本定义

如果没有显示 *Reg:0002* 列, 请右击表头, 在右键菜单中选择 *Properties* 。



附图 170: 右键菜单 *Properties*

出现 *Advanced Settings* 对话框, 可以选择要显示的列。在 *Diagnosis/Online View* 下, 选择 '*0002 ETxxxx Build*' 复选框, 以便激活 FPGA 固件版本显示。

附图 171: 对话框 *Advanced Settings*

## 更新

要更新以下 FPGA 固件

- EtherCAT 耦合器的 FPGA 固件：耦合器必须具有 FPGA 固件版本 11 或更高版本；
- E-Bus 端子模块的 FPGA 固件：端子模块必须有 FPGA 固件版本 10 或更高版本。

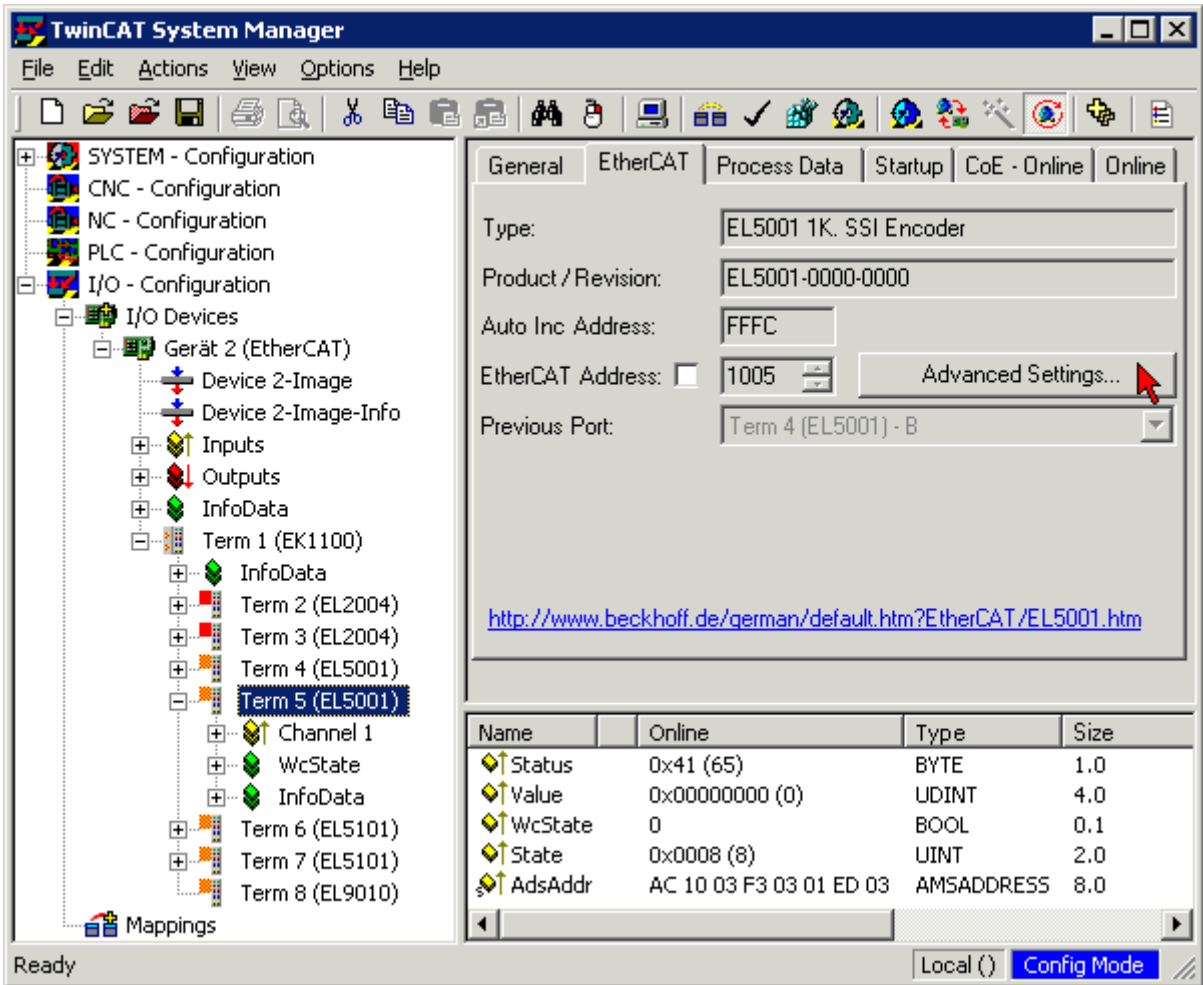
旧的固件版本只能由制造商进行更新！

## 更新一个 EtherCAT 设备

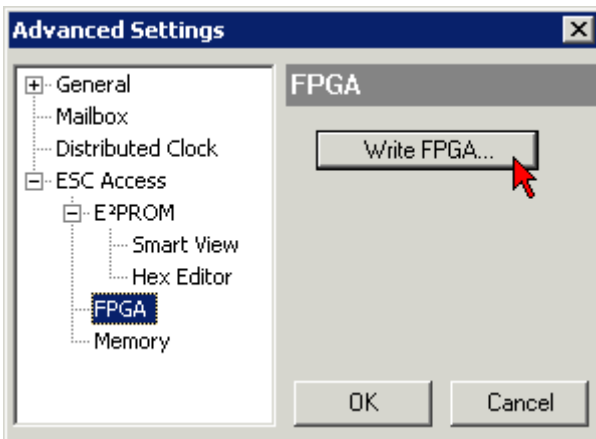
如果没有给出其他规定（例如来自倍福支持部门），则必须满足以下顺序：

- 将 TwinCAT 系统切换到 Config Mode/FreeRun，周期时间  $\geq 1$  ms（配置模式下默认为 4 ms）。不建议在实时核运行时（Running 模式）进行固件更新。

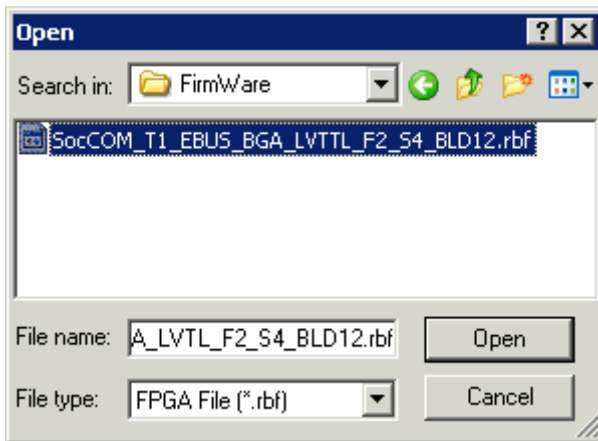
在 TwinCAT System Manager 中，选择需要更新 FPGA 固件的端子模块（例如：端子模块 5：EL5001），并在 *EtherCAT* 选项卡中点击 *Advanced Settings* 按钮：



出现 *Advanced Settings* 对话框。在 *ESC Access/E<sup>2</sup>PROM/FPGA* 下，点击 *Write FPGA...* 按钮：



- 选择带有新 FPGA 固件的文件 (\*.rbf)，并将其传输到 EtherCAT 设备上：



- 一直等待，直到下载结束
- 短时切断从站电源（不要拉低电压！）。为了激活新的 FPGA 固件，需要重新启动 EtherCAT 设备（断电重启）。
- 检查新的 FPGA 状态

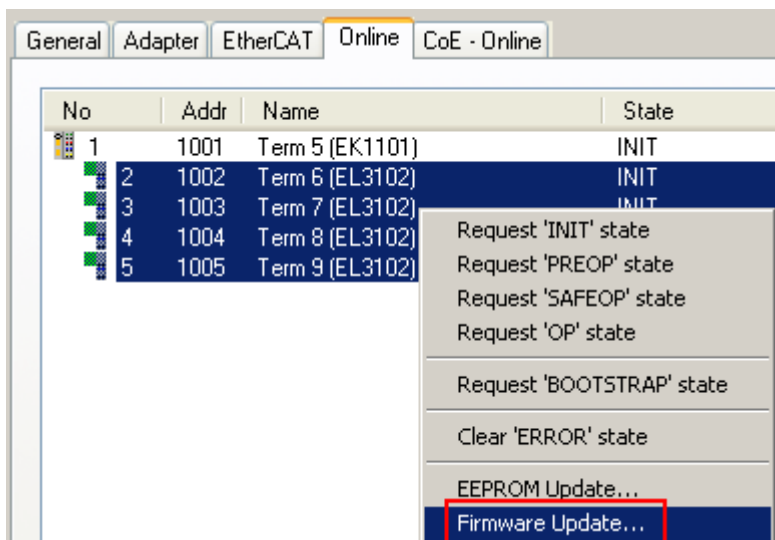
### 注意

#### 设备损坏风险！

在任何情况下，都不能中断 EtherCAT 设备下载固件的过程！如果下载固件的过程中发生了断电或者断网，EtherCAT 设备只能返回制造商重新调试！

## 9.3.5 同时更新多个 EtherCAT 设备

如果几个设备有相同的固件文件/ESI，这些设备的固件和 ESI 描述可以同时更新。

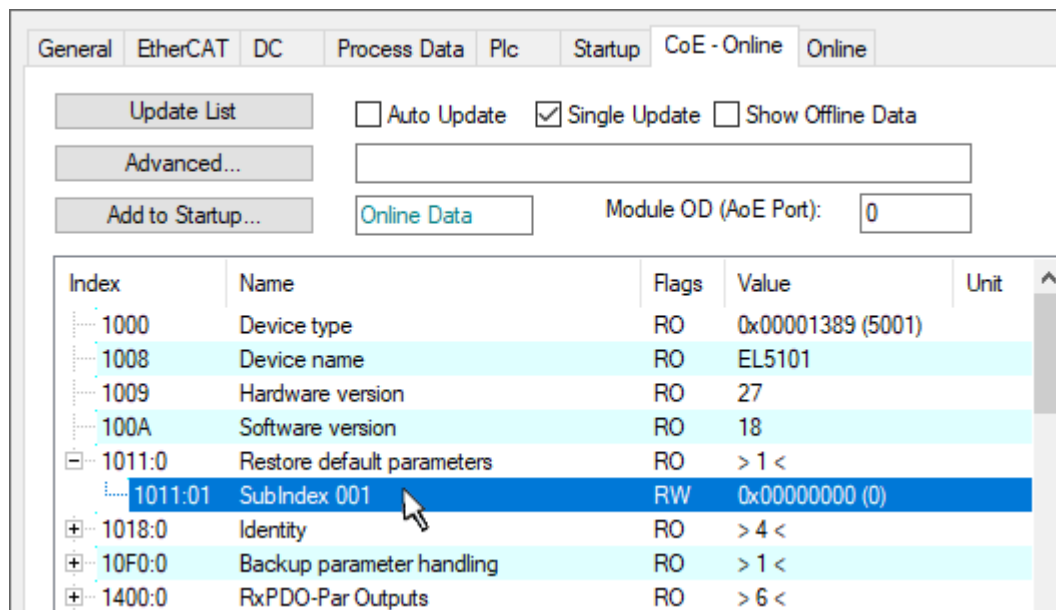


附图 172：多重选择和固件更新

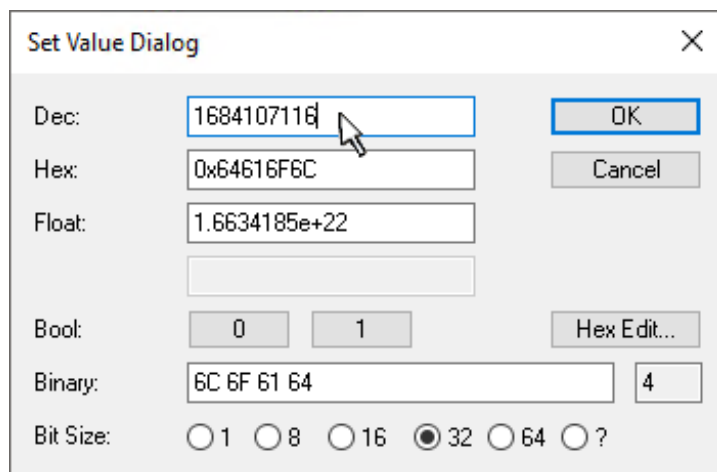
选择所需的从站，如上所述在 BOOTSTRAP 模式下进行固件更新。

## 9.4 恢复出厂状态

要恢复 EtherCAT 设备（“从站”）CoE 对象的交付状态（出厂设置），可通过 EtherCAT 主站（例如 TwinCAT）使用 CoE 对象 Index 1011 *Restore default parameters*（参见图选择 *Restore default parameters*）。



附图 173: 选择 *Restore default parameters*



附图 174: 在 Set Value dialog 中输入一个恢复值

双击 *SubIndex 001*，进入设置值对话框。将重置值 **1684107116** 输入字段 *Dec* 中，或将数值 **0x64616F6C** 输入字段 *Hex* 中，并按 *OK* 确认（图：在 *Set Value dialog* 中输入恢复值）。

- 从站中所有可写的条目都将重置为默认值。
- 只有直接对从站的 Online CoE 进行重置，才能成功恢复出厂值。在离线 CoE 中不能更改任何值。
- 为此，TwinCAT 必须处于 RUN 或 CONFIG/Freerun 状态，即保持 EtherCAT 数据交换。确保 EtherCAT 传输正确无误。
- 由于进行了重置，因此不会进行单独确认。如要进行验证，可以事先向某个可写对象写入一个值。
- 该重置过程也可以作为从站 Startup List 的第一个条目，例如在状态转换 PREOP→SAFEOP 中，或者如图 *CoE reset* 作为 *Startup* 条目所示的 SAFEOP→OP 中。

CoE 中的所有备份对象重置为交付状态。

---

**i** 替代的恢复值

在一些较旧的端子模块（FW 创建时间约在 2007 年之前）中，备份对象可以用另一套替代的恢复值进行切换：十进制值：1819238756，十六进制值：0x6C6F6164。

恢复值输入错误不会产生任何影响。

---



## 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。

### 倍福分公司和代表处

有关倍福产品本地支持和服务方面的信息，请联系倍福分公司或代表处！

世界各地倍福分公司和代表处的地址可参见以下网页：<http://www.beckhoff.com>

该网页还提供更多倍福产品组件的文档。

### 支持

倍福支持部门提供全面的技术援助，不仅帮助使用各种倍福产品，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](mailto:support@beckhoff.com)  
网址： [www.beckhoff.com/support](http://www.beckhoff.com/support)

### 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](mailto:service@beckhoff.com)  
网址： [www.beckhoff.com/service](http://www.beckhoff.com/service)

### 德国总部

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20  
33415 Verl  
Germany

电话： +49 5246 963 0  
电子邮箱： [info@beckhoff.com](mailto:info@beckhoff.com)  
网址： [www.beckhoff.com](http://www.beckhoff.com)



更多信息:

[www.beckhoff.com/EL6xxx](http://www.beckhoff.com/EL6xxx)

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Germany  
电话号码: +49 5246 9630  
[info@beckhoff.com](mailto:info@beckhoff.com)  
[www.beckhoff.com](http://www.beckhoff.com)

