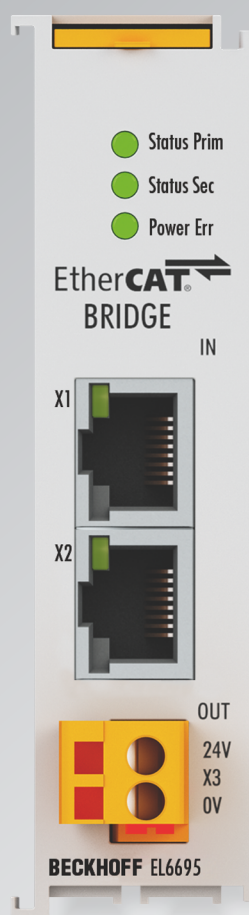


文件资料 | ZH

EL6695

EtherCAT 桥接端子模块



目录

1 前言	7
1.1 文档说明	7
1.2 安全说明	8
1.3 文档发行状态	9
1.4 文档指南	11
1.5 EtherCAT 设备的版本标识	12
1.5.1 关于标识的一般说明	12
1.5.2 EL 端子模块的版本标识	13
1.5.3 倍福识别码 (BIC)	14
1.5.4 BIC 电子读取 (eBIC)	15
2 产品描述	18
2.1 简介	18
2.2 技术数据	19
2.2.1 与倍福 EtherCAT 数据交换设备比较	20
2.3 安装的工况条件	21
3 基本通讯	22
3.1 EtherCAT 基础知识	22
3.2 EtherCAT 布线 - 线缆连接	22
3.3 设置看门狗的一般注意事项	24
3.4 EtherCAT 状态机	26
3.5 CoE 接口	28
3.6 分布时钟 (Distributed Clock)	34
4 安装和接线	35
4.1 静电防护的说明	35
4.2 UL 声明	36
4.3 注意事项 - 电源	37
4.4 安装和拆卸 - 正面带拆卸手柄的端子模块	38
4.5 安装位置	40
4.6 无通讯模块的安装位置	43
4.7 屏蔽说明	44
4.8 LED 和接线	45
4.9 诊断	46
4.10 处理	47
5 调试	48
5.1 TwinCAT 快速入门	48
5.1.1 TwinCAT 2	51
5.1.2 TwinCAT 3	61
5.2 TwinCAT 开发环境	75
5.2.1 TwinCAT real-time 实时驱动程序的安装	75
5.2.2 关于 ESI 设备描述文件的说明	81
5.2.3 创建 OFFLINE 配置	84

5.2.4	创建ONLINE配置.....	89
5.2.5	EtherCAT 设备的配置.....	96
5.3	EtherCAT 从站的一般调试说明	106
5.4	CoE 对象概览.....	113
5.4.1	标准 CoE 对象	113
5.4.2	端子模块特定 CoE 对象	113
5.4.3	配置文件特定 CoE 对象	120
6	功能和运行模式.....	123
6.1	基本功能原理.....	124
6.1.1	通过 TwinCAT 访问 EL6695	125
6.2	与 EL6692 的兼容性	126
6.3	状态机 EL6695	127
6.4	周期性过程数据 PDO	128
6.4.1	流程控制.....	128
6.4.2	对称 PDO 映射.....	132
6.4.3	选择性 PDO 映射	144
6.4.4	FSoE 传输.....	154
6.5	本地邮箱协议.....	155
6.5.1	CoE - Can over EtherCAT	155
6.6	具备传输功能的邮箱协议.....	156
6.6.1	CoE 访问过程中的 AoE 应用.....	156
6.6.2	EoE - Ethernet over EtherCAT.....	158
6.6.3	AoE - ADS over EtherCAT	161
6.6.4	FoE - Filetransfer over EtherCAT	163
6.6.5	VoE - vendor-specific protocol over EtherCAT	173
6.6.6	SoE - Servo Drive Profile over EtherCAT	173
6.7	分布式时钟	174
6.8	在线扫描.....	177
6.9	EL6692 兼容模式.....	178
6.9.1	带 PDO 映射与分配的模块操作.....	178
6.10	EL6695 性能模式.....	179
6.10.1	PDO 映射的基本原理.....	179
6.10.2	无 Init 命令	179
6.10.3	对象描述下载.....	179
6.10.4	设备仿真.....	179
6.11	应用特定变量定义.....	181
7	附录.....	182
7.1	EtherCAT AL 状态代码	182
7.2	固件兼容性	183
7.3	固件更新.....	184
7.3.1	设备描述 ESI 文件/XML	185
7.3.2	Firmware (固件) 说明.....	188
7.3.3	更新从站处理器的固件 *.efw.....	188

7.3.4 同时更新多个 EtherCAT 设备 190

7.4 固件（FW）/FPGA 更新的补充说明 191

7.5 恢复出厂状态..... 194

7.5.1 常规设备重置..... 194

7.5.2 恢复出厂状态..... 194

7.6 技术支持和服务 198

1 前言

1.1 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®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® 和 XTS® 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。



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。

第三方品牌

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

1.2 安全说明

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。

1.3 文档发行状态

版本	注释
1.9.0	• 更新“恢复出厂状态”章节 • 更新“功能和运行模式”章节 • 结构更新
1.8.0	• 更新“安装和布线”章节 • 更新“LED 和连接”章节 • 结构更新
1.7.0	• 更新“EL/ES/ELM/EM/EPxxxx 固件更新”章节 • 更新“固件 (FW) /FPGA 更新的补充说明”章节 • 更新“固件兼容性”章节 • 更新“恢复出厂状态”章节 • 新增“文档指南”章节 (前言) • 删除“推荐的安装导轨”章节
1.6.0	• 更新“连接”章节 • 结构更新
1.5.0	• 更新“技术数据”章节 • 更新“安装和布线”章节 • 更新“诊断”章节 • 结构更新 • 更新修订状态
1.4.6	• 更新“附录”中的“CoE 对象概览”
1.4.5	• “对称 PDO 映射”章节的新增内容 • 结构更新
1.4.4	• 更新“对称 PDO 映射”章节 • 结构更新
1.4.3	• 更新“分布式时钟”章节 • 结构更新
1.4.2	• 更新“对称 PDO 映射”章节 • 结构更新
1.4.1	• 结构更新
1.4.0	• 新增“FoE 数据吞吐率”的示例 • 各项增补/更正 • 结构更新 • 更新修订状态
1.3.0	• 更新“功能和运行模式”章节 (新增 FoE 运行模式)
1.2.0	• 更新“简介”章节 • 更新“功能和运行模式”章节 • 新增“连接”章节 • 更新“诊断 LED”章节
1.1.0	• 更新“功能和运行模式”章节 • 更新“技术数据”章节 • 结构更新 • 更新修订状态

版本	注释
1.0.0	• 首次发布
0.9.8	• “选择性 PDO 映射”部分补充了“转换为全局数据类型” • （临时版本 – 可能变更）
0.9.7	• 更正“用于 CoE 访问的 AoE 应用”部分
0.9.6	• 扩展“功能和运行模式”部分
0.9.5	• 在“支持传输的邮箱协议”部分中插入“用于 CoE 访问的 AoE 应用”部分
0.9	• 首批系列的临时文件
0.1	• 初版

1.4 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)	• 系统概览 • EtherCAT 基础知识 • 电缆冗余 • 热连接 • EtherCAT 设备配置
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) 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](#)。

如果您对我们的文档有任何建议或意见，请发送电子邮件至documentation@beckhoff.com，并注明文档标题和版本号。

1.5 EtherCAT 设备的版本标识

1.5.1 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“EL2872，修订版本号 0022 和序列号 01200815”。
- 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。

1.5.2 EL 端子模块的版本标识

倍福 IO 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

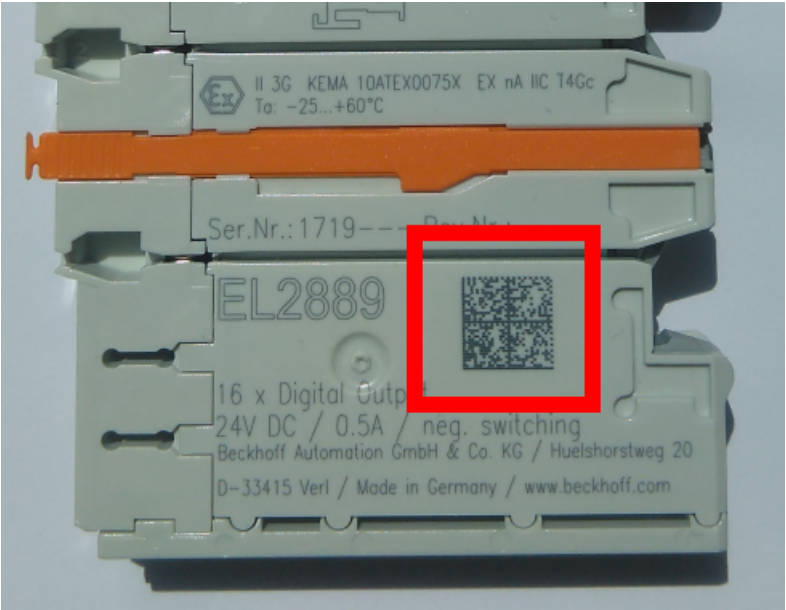
02 - 硬件版本为 02



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

1.5.3 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	12	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**1KEL**1809 **Q1 51S**678294

对应的DMC如下：



附图 3: 示例 DMC **1P072222SBTN**k4p562d7**1KEL**1809 **Q1 51S**678294

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。

1.5.4 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)

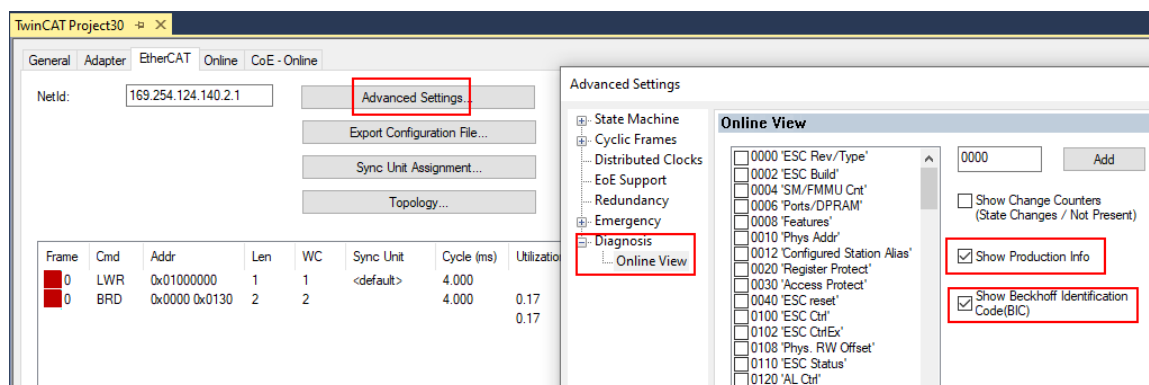
倍福的所有 EtherCAT 设备都有一个 ESI-EEPROM，其中包含 EtherCAT 标识和修订版本号。EtherCAT 从站信息，一般也被称为 EtherCAT 主站的 ESI/XML 配置文件，储存在其中。具体关系请参见 EtherCAT 系统手册中的相应章节（[链接](#)）。

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	1P158442SBTN0008jekp1KELM3704 Q1 2P482001000016
10F0:0	Backup parameter handling	RO	> 1 <
10F3:0	Diagnosis History	RO	> 21 <
10F8	Actual Time Stamp	RO	0x170bf277e

- 对象 0x10E2 将在批量产品的必要固件修订过程中优先引入。
- 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_Uilities 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 string (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。

2 产品描述

2.1 简介



附图 4: EL6695

EtherCAT 桥接端子模块

EtherCAT 桥接端子模块 EL6695 可以实现属于不同主站的 EtherCAT 网段之间的实时数据交换。还支持通过各种协议进行异步通信。分布式时钟可以双向同步。EL6695 和 EL6692（该产品将继续销售）不同的是，它具有灵活的 CoE 配置和一个设备仿真选项，并且大大提高了数据吞吐率。与 EL6692 类似，TwinCAT System Manager 提供一个“Extension”接口方便配置。副边（RJ45）通过外部连接供电，原边通过 E-bus 供电。得益于灵活的 CoE 配置，桥接端子模块也可用于将子 IPC 系统作为 EtherCAT 从站进行集成。在这种情况下，用户可以在 CoE 中定义一个参数集，并使该子系统对外部呈现为用户定义的 EtherCAT 从站。

2.2 技术数据

技术数据	EL6695
Ports (端口)	原边: E-bus (端子模块网段), 副边: 2 个 100 Mbit/s 以太网, RJ45, 输入/输出
Function	EtherCAT 分布式时钟同步, 同步/异步数据交换
电缆长度	副边端口: 最大 100 m 100BASE-TX
硬件诊断	状态 LED
电源 电流消耗	原边: 通过 E-bus, 典型值 400 mA 副边: 24 V _{DC} (-15%/+20%), 典型值 80 mA, 可插拔 仅需其中一个电压即可运行。如果两个电压均存在, 则内部电源模块将优先使用 24 V 电源
分布式时钟	是, 需要 TC3.1 b4018.4 及以上版本的 TwinCAT
电气隔离	500 V (E-bus/副边)
周期性过程数据	每个方向最大 3 kbyte 注意: 支持多少个周期性 PDO 主要取决于所使用的 EcMaster 使用 TC2.11b2248 或 TC3.1 b4018 时: 255 个变量, 最大总大小 MTU (~1400 字节)
支持的异步协议	CoE、EoE、AoE、FoE、(VoE、SoE)
PDO 传输速率	取决于运行模式和数据量 (通常为 10 至 100 μs)
EtherCAT 最短循环时间	50 μs
功能特点	• 可在 TwinCAT 中用作外部参考时钟 • 同步数据交换 • 灵活的 CoE 定义/CoE 设备仿真 • 可用于直接的双侧 DC 同步 • PDO 映射 (对称或选择性) • ADS 路由 (处理来自两个主站的 ADS 查询) • 每个 FoE 的用户数据传输速率最高可达 5 kbytes
运行时允许的环境温度	0 至 +55 °C
储存时允许的环境温度	-25 至 +85 °C
允许的相对湿度	95%, 无冷凝
抗振 / 抗冲击性能	符合 EN 60068-2-6 / EN 60068-2-27 标准
EMC 抗干扰 / 辐射	符合 EN 61000-6-2 / EN 61000-6-4 标准
防护等级	IP20
安装位置	可变
外形尺寸	宽度: 24 mm (并排安装) 高度: 100 mm 深度: 68 mm
认证 / 标识*	CE、cULus [▶ 36]、UKCA、EAC

*) 真正适用的认证/标志见侧面的型号牌 (产品标志)。

2.2.1 与倍福 EtherCAT 数据交换设备比较

	EL6695	EL6692	FC1100	FC1121	CXnnnn-B110
设计	24mm 端子模块		PCIe® 插卡		集成在嵌入式控制器内部
PDO 周期性过程数据	每个方向最大 3 kB ⁽¹⁾ 执行 TC2.11b2248 或 TC3.1b4018: 255 个变量, 最大整体大小 MTU (约 1400 字节)	每个方向最大 480 字节	每个方向最大 1024 字节		每个方向最大 480 字节
PDO 传输速度 (设备内部, 不含总线周期)	例如: 200 字节输入/200 字节输出: 典型值 15 μs 例如: 1400 字节输入/1400 字节输出: 典型值 50 μs	典型值 1..4 ms 例如: 200 字节输入/200 字节输出: 典型值 1 ms	例如: 480 字节输入/480 字节输出: 约 300 μs ⁽²⁾		例如: 480 字节输入/480 字节输出: 在 CX5020 上约为 250 μs ⁽²⁾
支持的异步协议	CoE、EoE、AoE、FoE、(VoE、SoE)	CoE、AoE、EoE			
Mailbox (邮箱)	128-1498 字节	128-1024 字节	64-1024 字节		
邮箱默认设置	1024 字节	256 字节	1024 字节		512 字节
最短允许的 EtherCAT 周期时间 ⁽³⁾	50 μs (使用 SyncMan 中断)	无限制 (不使用 SyncMan 中断)			
分布式时钟同步	是				
可在 TwinCAT 中用作外部参考时钟	是				
具体特性	• 同步数据交换 • 灵活的 CoE 定义/CoE 设备仿真 • 适用于直接的双向 DC 同步 • PDO 映射 (对称或选择性) • ADS 路由 (由两个主站处理 ADS 请求) • 通过 FoE 传输多达 5 kB 用户数据 • 原边/副边独立供电	• 适用于通过 EC 主站间接实现的 DC 同步 • ADS 路由			

¹ 支持的周期性 PDO 的数量取决于 EtherCAT 主站

² 传输值在很大程度上取决于 IPC 环境和控制

³ 此限值表示设备的一个运行限值。实际上, 两侧各自合理的最低周期主要取决于待传输的数据量 (以及 PDO 传输时间) – 应选择一个合理的周期, 以便端子模块能够在每个循环中获取/设置数据。

2.3 安装的工况条件

请确保仅在规定的条件下运输、储存和操作 EL6695 EtherCAT 端子模块（参见技术数据）！

⚠ 警告

严禁在非正常工况下运行！

在下列工况下不得使用 EL6695：

- 在电离辐射的影响下
- 在腐蚀性环境中
- 在导致总线端子模块受到严重污染的环境下

安装的安全指引

在安装和调试 EtherCAT 端子模块之前，请先阅读本文档前言中的安全说明。

运输/存储

在运输或存储 EtherCAT 端子模块时，务必使用其交付时的原始包装。

电气安装

请注意，如果 E-bus 和 24 V 电源同时可用，EL6695 将优先使用 24 V 电源。为确保 EL6695 通过 E-bus 供电，建议在断开 24 V 电源的情况下进行调试。否则，如果 E-bus 电源未预先正确计算容量，则在 24 V 电源故障时，端子模块网段中的 E-bus 电源可能会发生过载。

Number	Box Name	Address	Type	In Size	Out Size	E-Bus (mA)
 1	Box 1 (EL6695)	1001	EL6695	2.0		
 2	Term 2 (EK1100)	1002	EK1100			
 3	Term 3 (EL6695)	1003	EL6695	2.0		1600
 4	Term 4 (EL1008)	1004	EL1008	1.0		1510
 5	Term 5 (EL1008)	1005	EL1008	1.0		1420

3 基本通讯

3.1 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](#)。

3.2 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

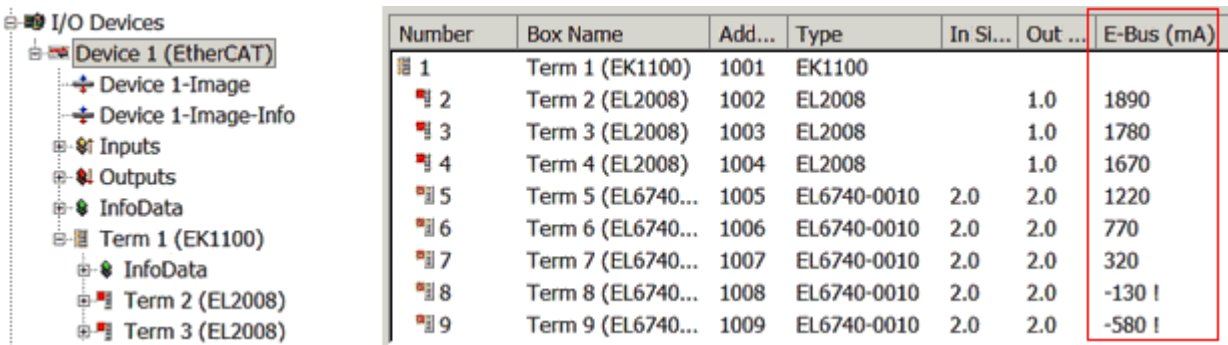
用于连接 EtherCAT 设备的合适电缆可参见[倍福公司网站](#)！

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。



附图 5: 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!

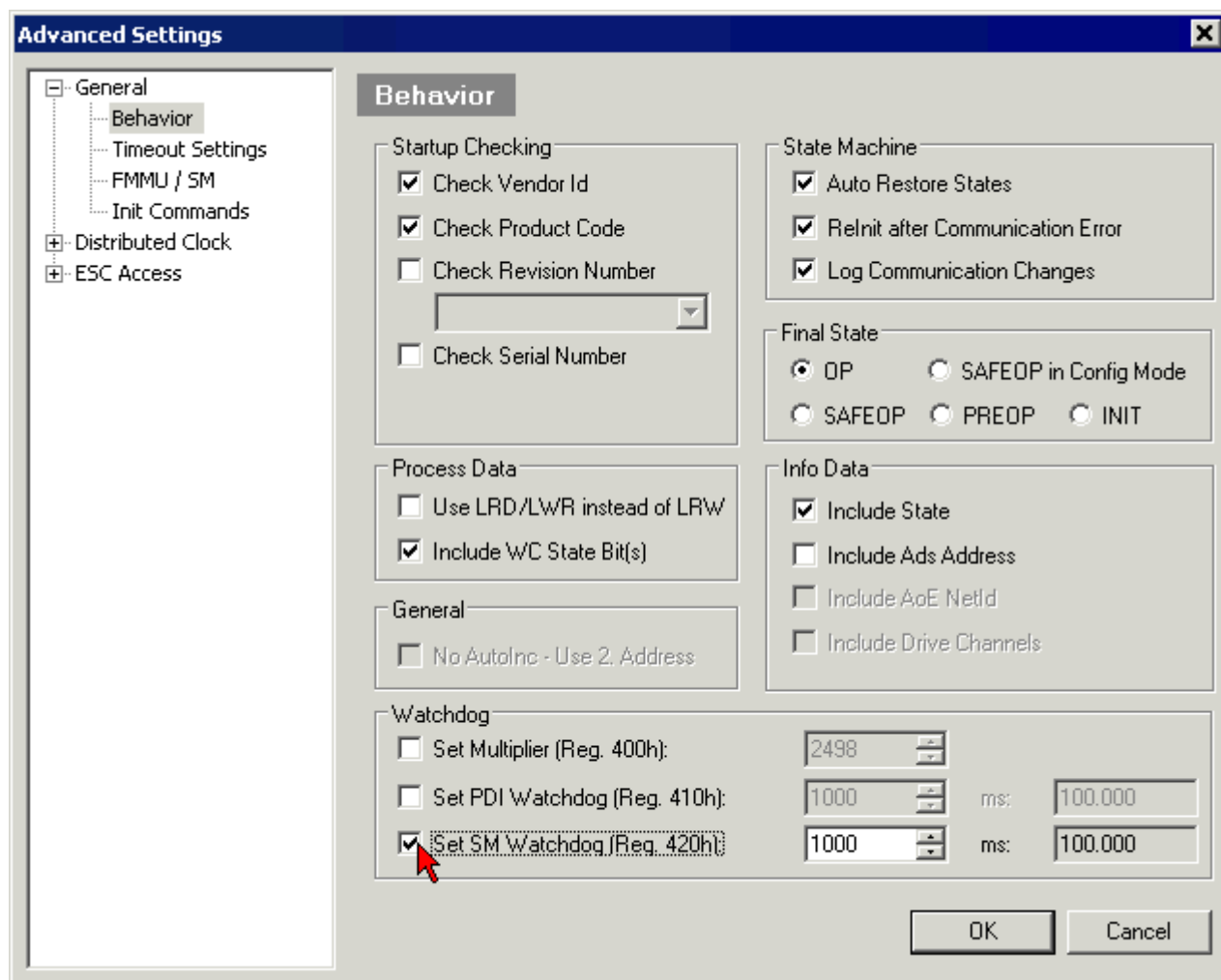
3.3 设置看门狗的一般注意事项

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 6: 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 (Sync Manager 看门狗)

Sync Manager 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，Sync Manager 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 μ 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 -> 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。

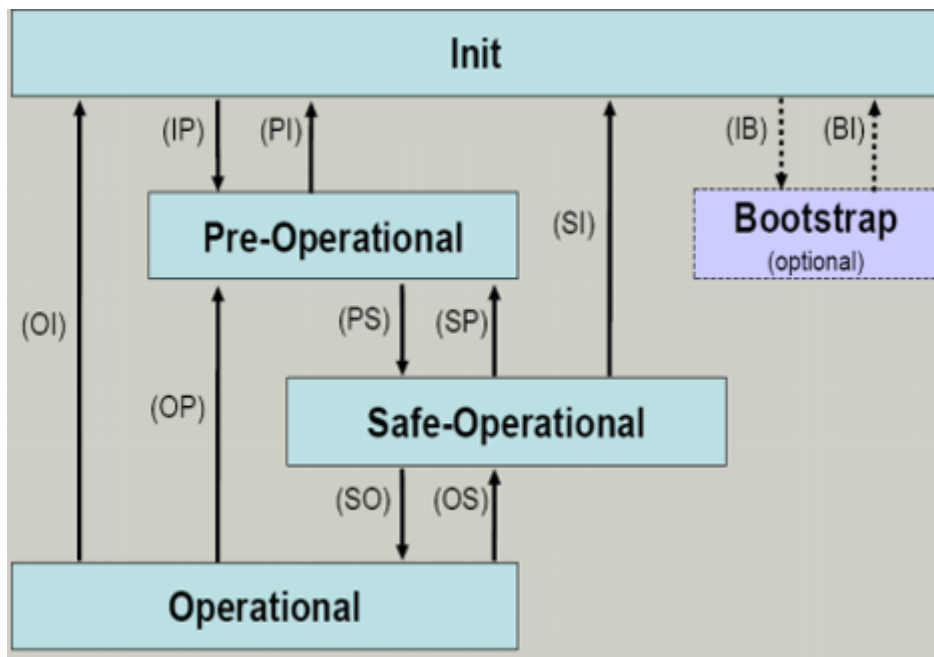
3.4 EtherCAT 状态机

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 7: 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

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。

3.5 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_{dec})
- 子索引 SubIndex：0x00...0xFF (0...255_{dec})

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：

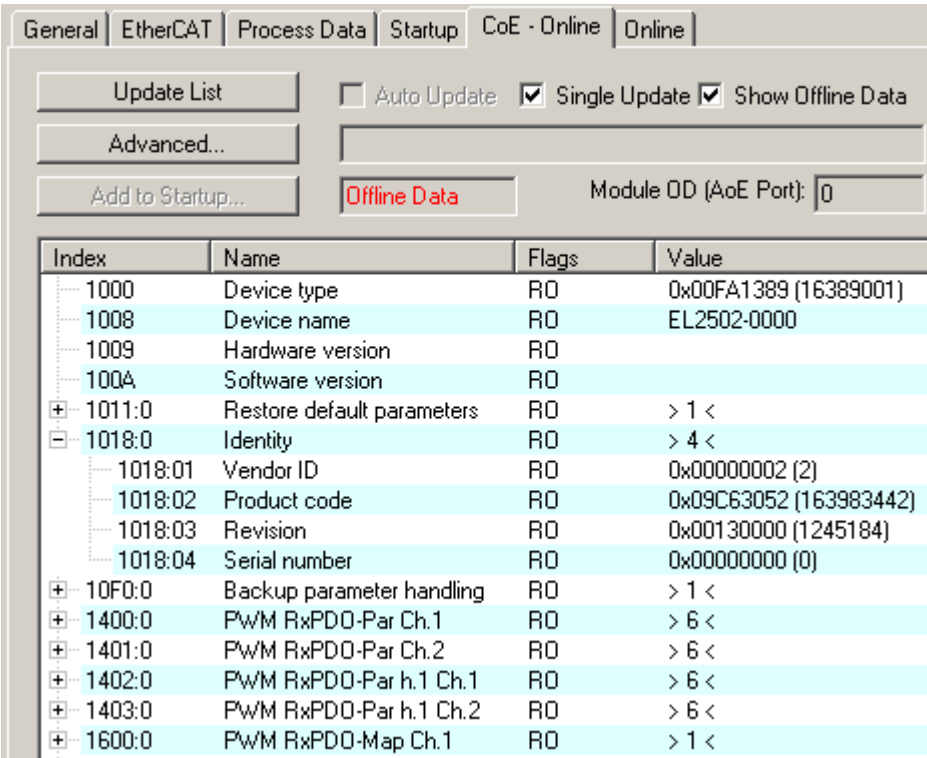
- 0x4000：有些 EtherCAT 设备在此存储通道参数。过去，这是在引入 0x8000 区域之前的第一个参数区。以前用 0x4000 配置参数的 EtherCAT 设备改用 0x8000 后，出于兼容性的考虑，这两个 CoE 索引范围都支持，并在内部进行映射。
- 0x6000：input PDO（“input”，是指从 EtherCAT 主站的角度看是输入）
- 0x7000：output PDO（“output”，是指从 EtherCAT 主站的角度看是输出）



适用性

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 8: “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 [TwinCAT 3 | 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。

i

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。

i

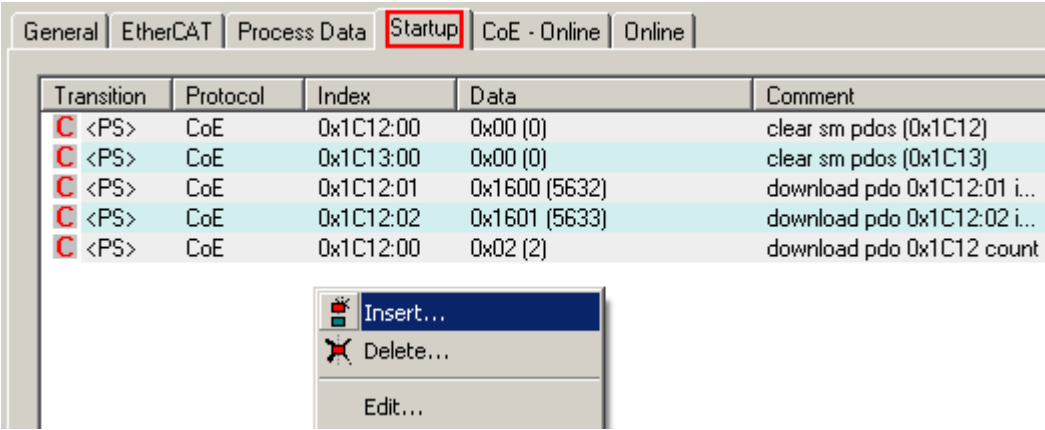
Startup List

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 9: 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。

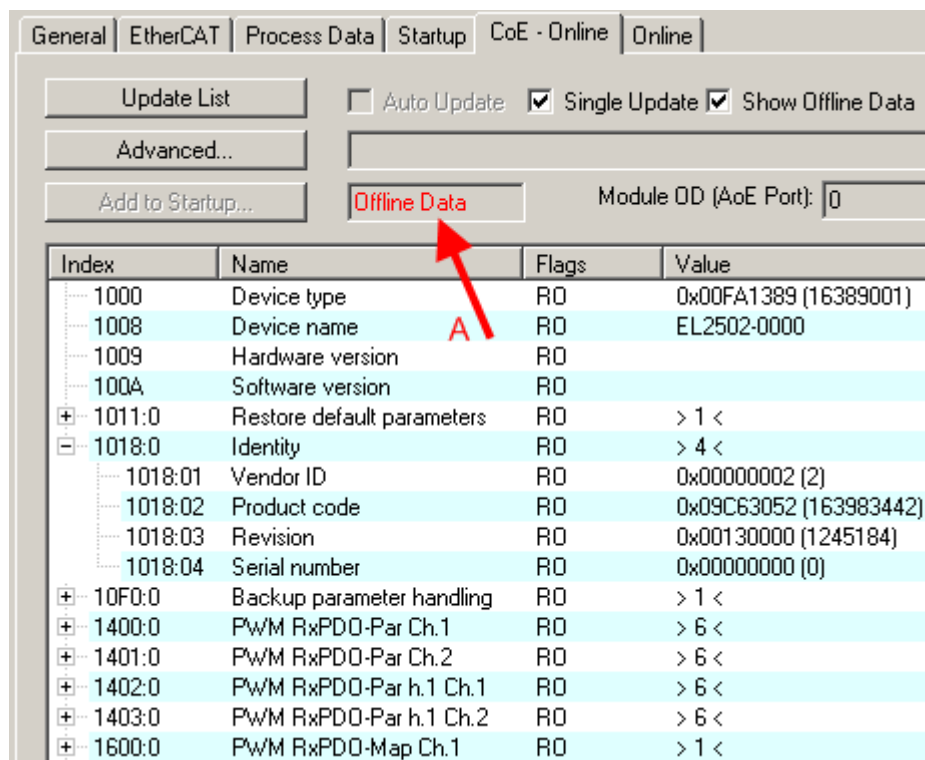
online/offline列表

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。

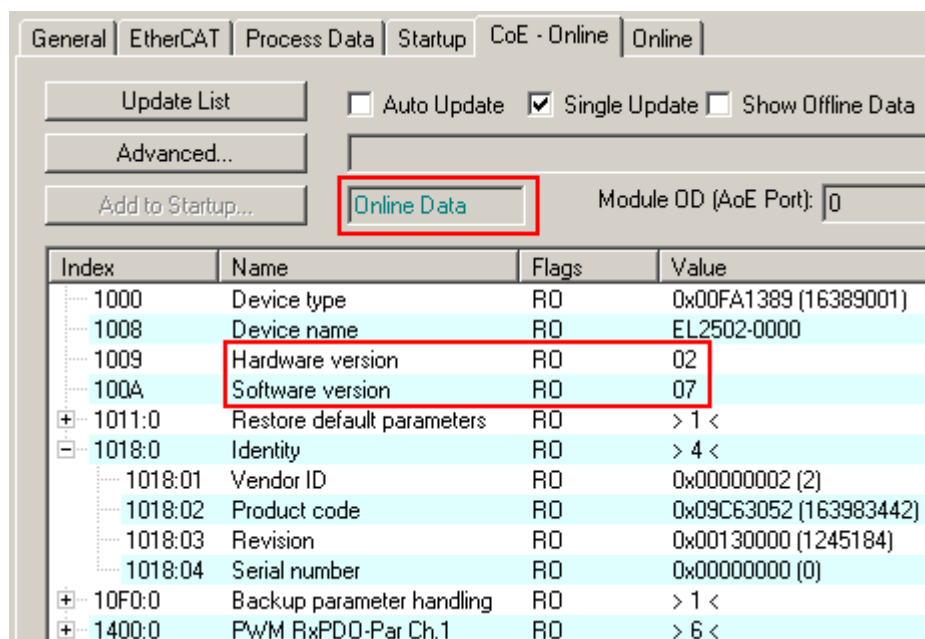
- 如果从站处于离线状态
 - 显示 ESI 文件中的离线列表。此时修改 CoE 参数是没有意义的，也无法进行。

- 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 10: 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 11: 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](#)。

备份对象和校验和 0x10F0:01

为了保留信息，在其它对象之外，还在CoE 参数/对象目录中定义了以下对象类型，但不必每个设备中都同时出现这些类型：

- 供应商对象
 - 在设备中永久存储（故障安全）。
 - 技术上属性为 ReadWrite（RW）。
 - 只有在知道相应供应商密码的情况下才能修改/删除。
 - 用于特定供应商的调整或身份数据。
- 备份对象
 - 这些对象在设备中永久存储，即使在更改后也是如此。
 - 技术上属性为 ReadWrite（RW）。
 - 它们可通过 EtherCAT 主站的 CoE 访问权限随时修改/删除
 - 特别是，通过选择 "Restore Default Parameters"（参见 "Restore Default Parameters" 一章），可将它们重置为初始状态，该状态永久存储在固件中。由于重置为以前的值类似于恢复备份，因此这些对象被称为 "备份对象"。
 - 它们用于常规功能参数，决定设备行为。
- 具有写保护选项的备份对象。
 - 备份对象也是如此。
 - 此外，用户还可以使用 xF009 中的密码激活写保护，以防止意外更改。详情可查阅包含这些对象的设备的设备文档。
- 易失对象
 - 这些对象不会持久地存储在设备中。
 - 它们用于显示内部信息（过程数据、状态、温度.....），可作为只读参数 ReadOnly（RO）或功能参数（ReadWrite）使用。但是，如果后者要使用非默认值，则必须通过 EtherCAT 主站在每次上电时写入。

设备用 32 位对象 0x10F0:01 CheckSum（SubIndex 01）中的 16 位来显示所谓备份对象当前状态的 CRC 校验和：

10F0:0	Backup parameter handling	RO	> 1 <
10F0:01	Checksum	RO	0x00003C62 (15458)

附图 12: CoE index 10F0

如果备份对象被改变了，固件会相应计算新的校验和。这可用于检测备份对象的变化。

注意：校验和的初始值可能因固件版本而异，因为功能扩展可能向 CRC 的校验区间中增加其他对象。

3.6 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 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 ns。

详细信息请参见 [EtherCAT 系统描述](#)。

4 安装和接线

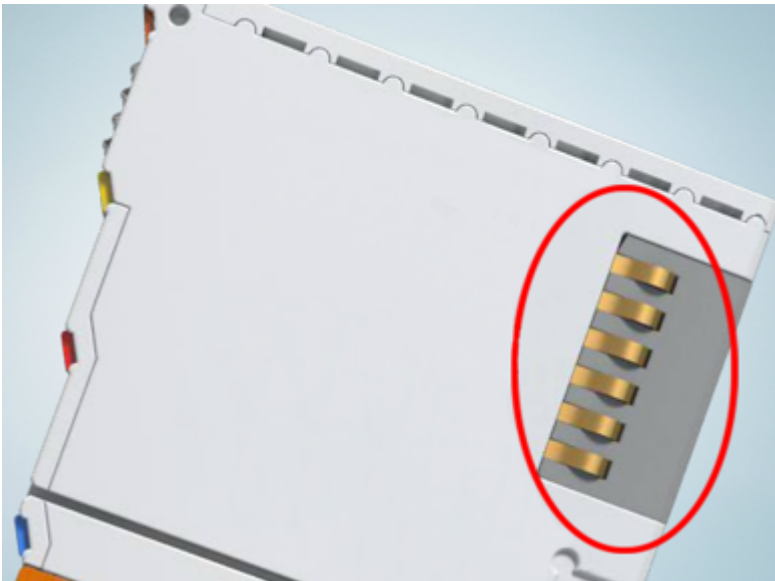
4.1 静电防护的说明

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 13: 倍福 I/O 组件的弹簧触点

4.2 UL 声明

⚠ 谨慎	
	应用 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。
⚠ 谨慎	
	检查 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）。
⚠ 谨慎	
	带有以太网连接器的设备 不可用于连接通信电路（telecommunication circuits）。

基本原则

符合 UL508 的 UL 认证。有这种认证的设备带有此标志：



4.3 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。

4.4 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

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）。

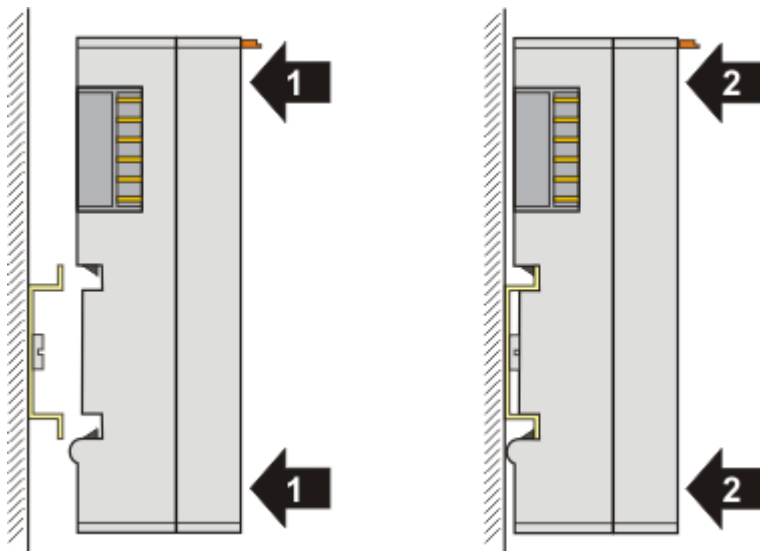
⚠ 警告

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。

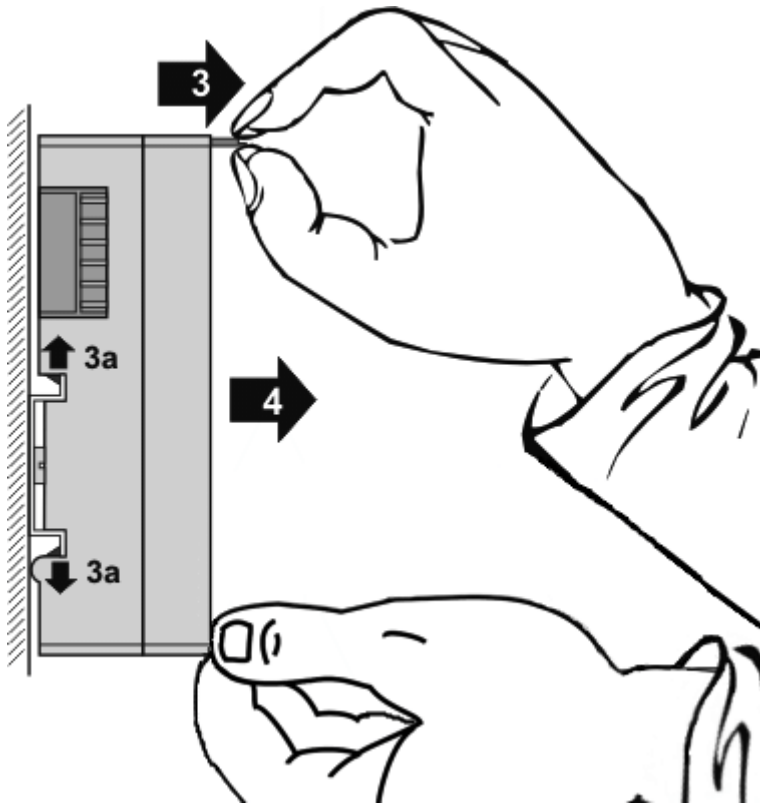


然后将接线端子模块向安装导轨上按压（1），直至模块在安装导轨上锁紧（2）。

- 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。

4.5 安装位置

注意

关于安装位置和工作温度范围的限制

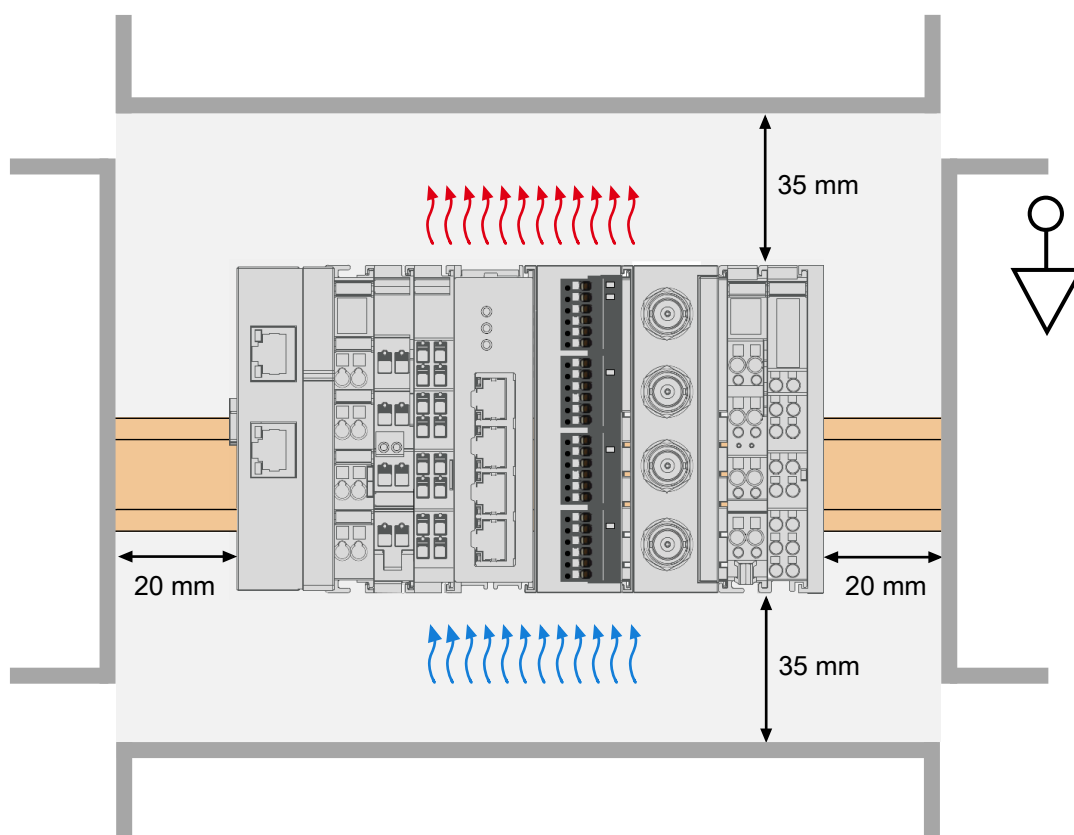
- 请参考设备的技术数据，以确定是否规定了关于安装位置和/或工作温度范围的任何限制。
- 在安装散热增强型设备时，应确保在运行期间设备上方和下方留有足够空间，以保证充分的通风！

安装导轨上各设备的安装位置及其名称如下所示。下图中的设备图示为示例。
以下内容适用于所有安装位置：参考方向“下”（见箭头）为重力加速度方向。

水平安装（标准安装）

安装导轨水平安装在垂直安装板上。
设备的接线端面朝前方。

设备从下面通风，通过对流实现电子元件的最佳散热。因此，这也是推荐的安装位置。



附图 14: 标准安装位置的建议最小距离

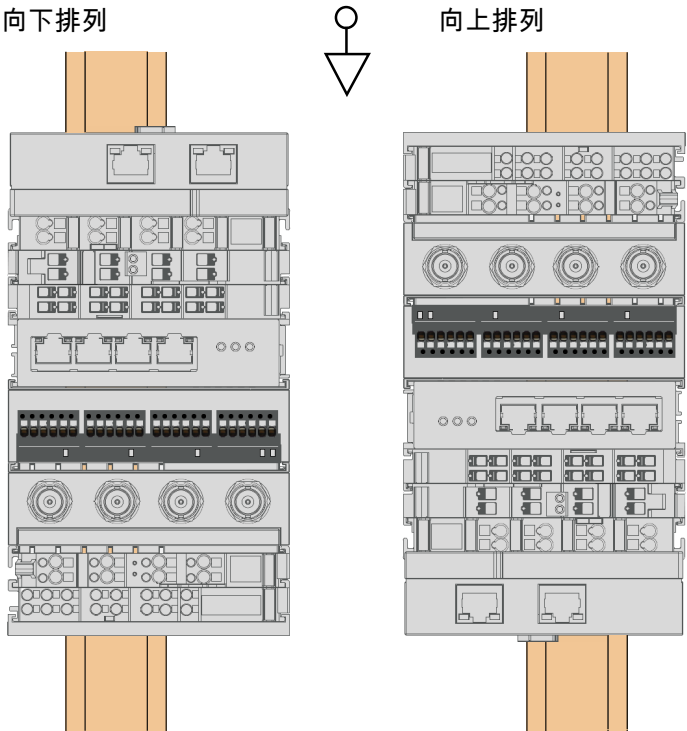
注意

遵守最小距离要求

强烈建议在所有安装位置都遵守“标准安装位置的建议最小距离”图中所示的最小距离。

垂直安装

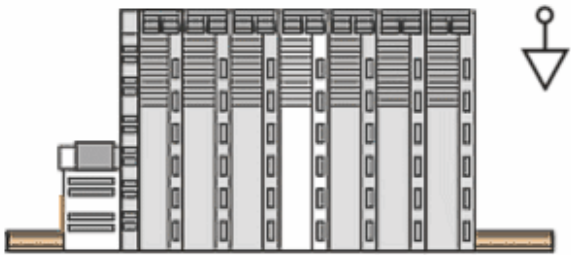
安装导轨垂直安装在垂直安装板上。
设备的接线端面朝前方。
设备可按如下方式排列：



附图 15: 垂直安装位置，向下排列（左）/向上排列（右）

水平安装

在水平安装位置，安装导轨铺设在水平安装板上。
设备的接线端面朝上。



附图 16: 水平安装位置

注意

滑脱安装导轨的危险

特别在“垂直”安装位置以及其他承受相应机械负载的安装位置，整节端子模块可能会在安装导轨上发生滑动。这些情况都可能导致意外故障。

- 如果存在这种风险，则应使用适当的锁定装置固定整节端子模块，例如将其夹紧在安装导轨上。

注意

遵守最小距离要求

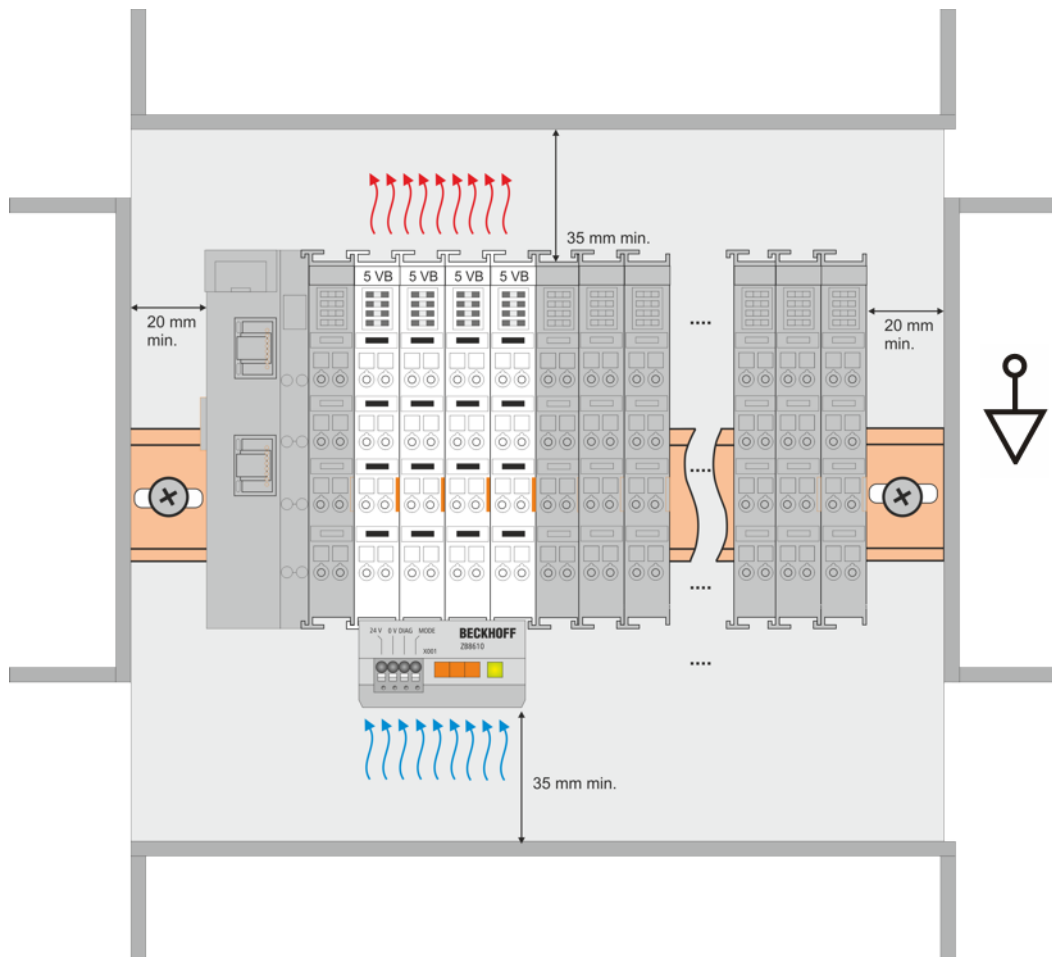
强烈建议在所有安装位置都遵守“标准安装位置的建议最小距离”图中所示的最小距离。

带 ZB8610 风扇盒的安装位置

如果特定应用需要或必须增强冷却效果，可在设备底部安装 ZB8610 风扇盒。在水平安装位置，设备通过风扇盒实现自下而上的通风。对流通风可进一步优化冷却效果（见下图）。

任何安装位置都可以使用风扇盒。

有关带风扇及无风扇运行的更多信息，请参见设备的技术数据（例如降额、安装位置信息等）。



附图 17: 以水平安装位置为例，带风扇运行的建议最小距离

注意

遵守最小距离要求

强烈建议遵守“带风扇运行的建议最小距离”图中所示的最小距离。

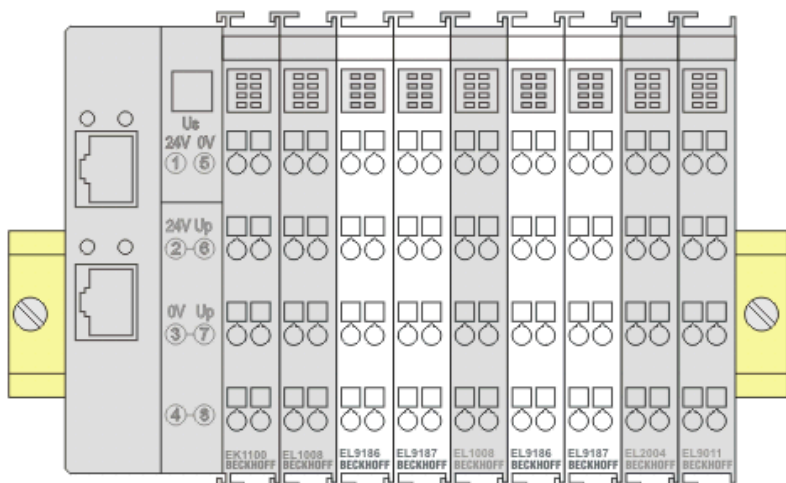
4.6 无通讯模块的安装位置

● 关于总线端子 I/O 站中无通讯模块安装位置的提示

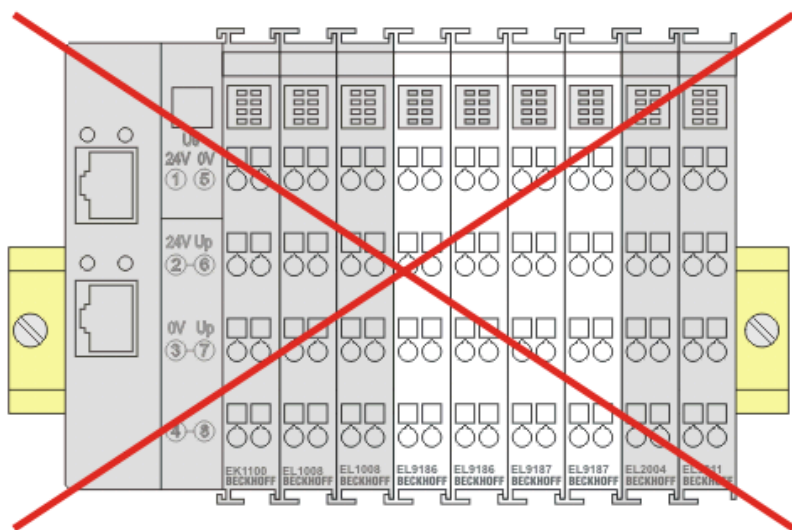
i 那些在总线端子 I/O 站中不参与数据传输的 EtherCAT 端子模块，即所谓的无通讯模块。无通讯模块不消耗 E-Bus 的电流。

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 18: 正确的安装位置



附图 19: 错误的安装位置

4.7 屏蔽说明

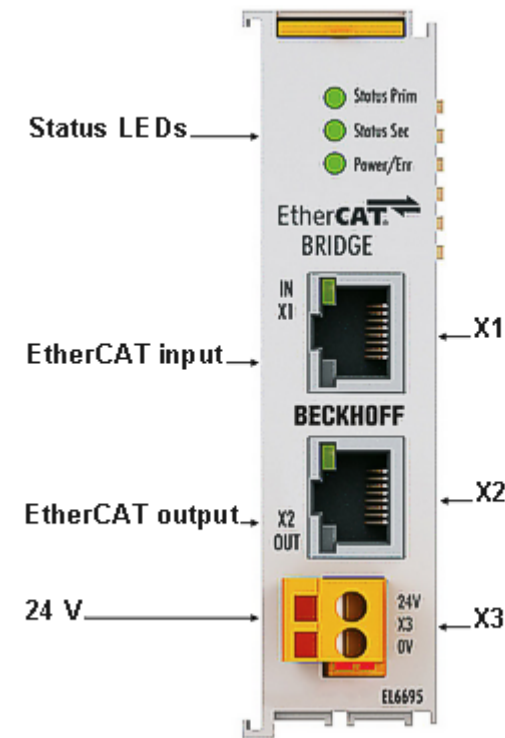
注意

通过外部屏蔽连接实现低阻屏蔽

EL66xx 的屏蔽层通过电容连接方式连接至安装导轨。

如果希望或需要低阻屏蔽，则必须将屏蔽层连接到外部屏蔽支架上（如通过用于屏蔽连接的固定支架 ZB5800）。

4.8 LED 和接线



附图 20: EL6695 - 连接

连接

端子模块接点	描述
X1	EtherCAT 输入（RJ45，支持 10BASE-T/100BASE-TX 以太网）
X2	EtherCAT 输出（RJ45，支持 10BASE-T/100BASE-TX 以太网）
X3	2 针插座端子连接（24 V _{DC} ），副边电源

LED

LED	颜色	含义
X1/X2	绿色	熄灭
		未连接到 EtherCAT 网络
		常亮
		已建立与 EtherCAT 网络的连接（LINK）
		闪烁
		与 EtherCAT 网络中的设备进行数据交换（ACT）

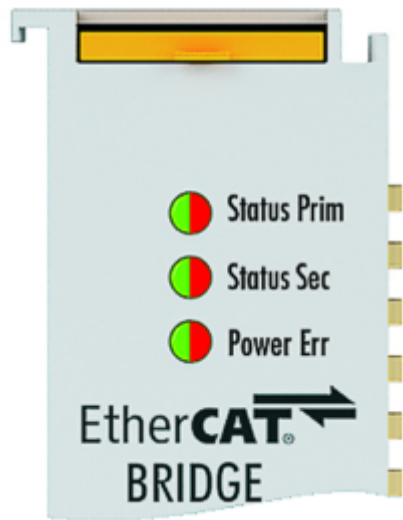


诊断 LED 指示灯

有关 LED 的说明，请参阅[诊断 LED \[► 46\]](#) 章节

4.9 诊断

诊断 LED 指示灯



附图 21: 通过诊断 LED 显示运行状态

LED	显示屏	含义
Status Prim Status Sec	熄灭	EtherCAT 状态机的状态: INIT = 端子模块的初始化或 BOOTSTRAP = 用于端子模块固件更新的功能
EtherCAT 状态 LED (红色/绿色), 相应 从站侧的运行状态	绿色闪烁	EtherCAT 状态机的状态: PREOP = 邮箱通信的功能和不同的标准 设置
	绿色单次闪烁	EtherCAT 状态机的状态: SAFEOP = 验证 <u>Sync Manager</u> (同步 管理器) 通道和分布式时钟。 输出保持安全状态
	绿色常亮	EtherCAT 状态机的状态: OP = 正常工作状态; 可以进行邮箱和 过程数据通信
	红色闪烁	内部错误
	红色单次闪烁	AL 状态寄存器错误
	红色双闪	过程数据 Watchdog (看门狗) 超时
	红色常亮	PDI Watchdog (看门狗) 超时, 或自检错误
	熄灭	无供电
Power/Error (电源/ 错误) (供电状态)	橙色	仅从一侧供电, 原边或副边
	闪烁红色	FW 更新失败, 将端子模块送至维修部门 [► 198]
	绿色	两侧均有供电, 使用 24 V 输入

4.10 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。必须遵守国家对废弃电气和电子设备的处理规定。

5 调试

5.1 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。

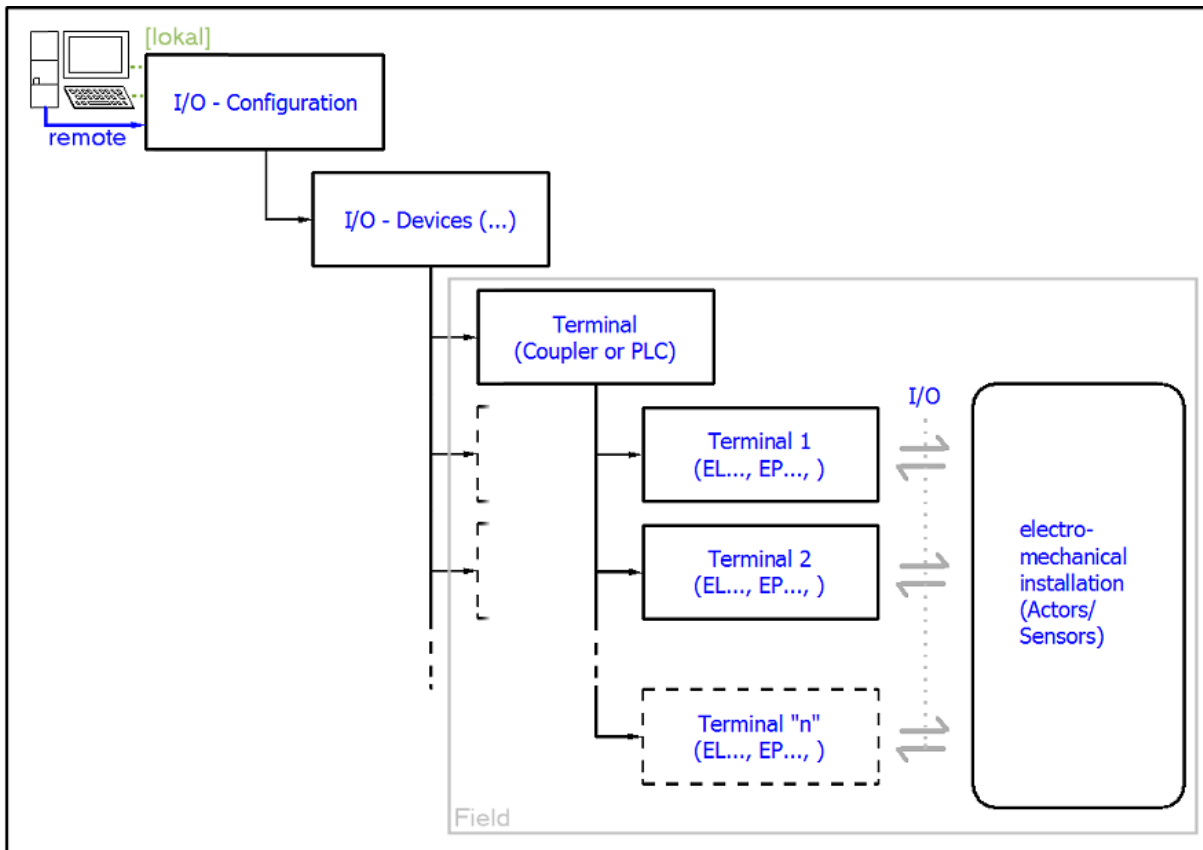
更多信息请参考 <http://infosys.beckhoff.com>：

- **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：



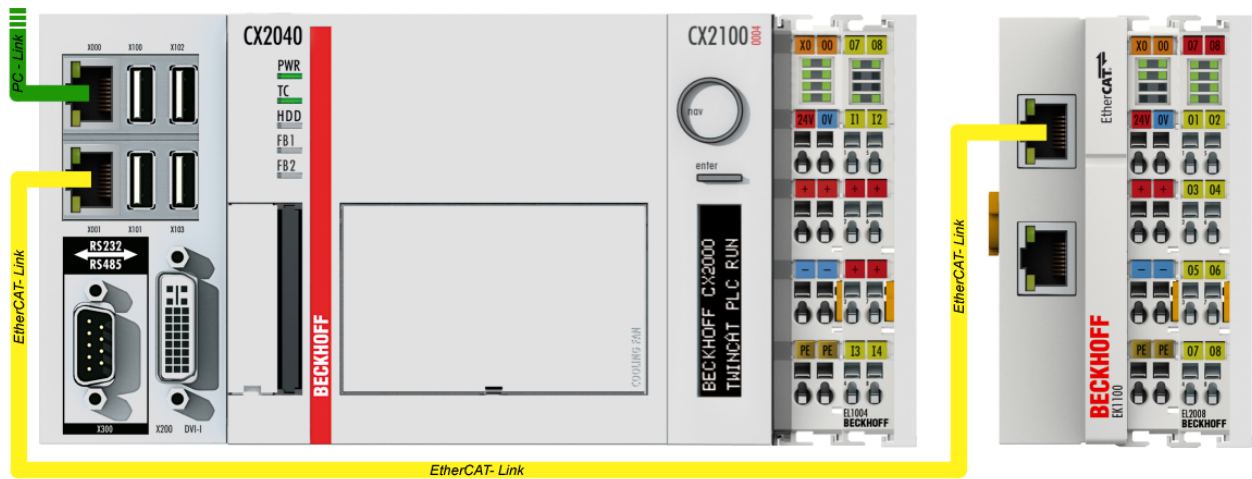
附图 22: 用户侧（调试）和实际安装组件之间的关系

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 DC）
- 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 DC；0.5 A）
- （X000 可选连接外部 PC，提供用户接口）



附图 23: 系统配置，含嵌入式控制器、输入（EL1004）和输出（EL2008）

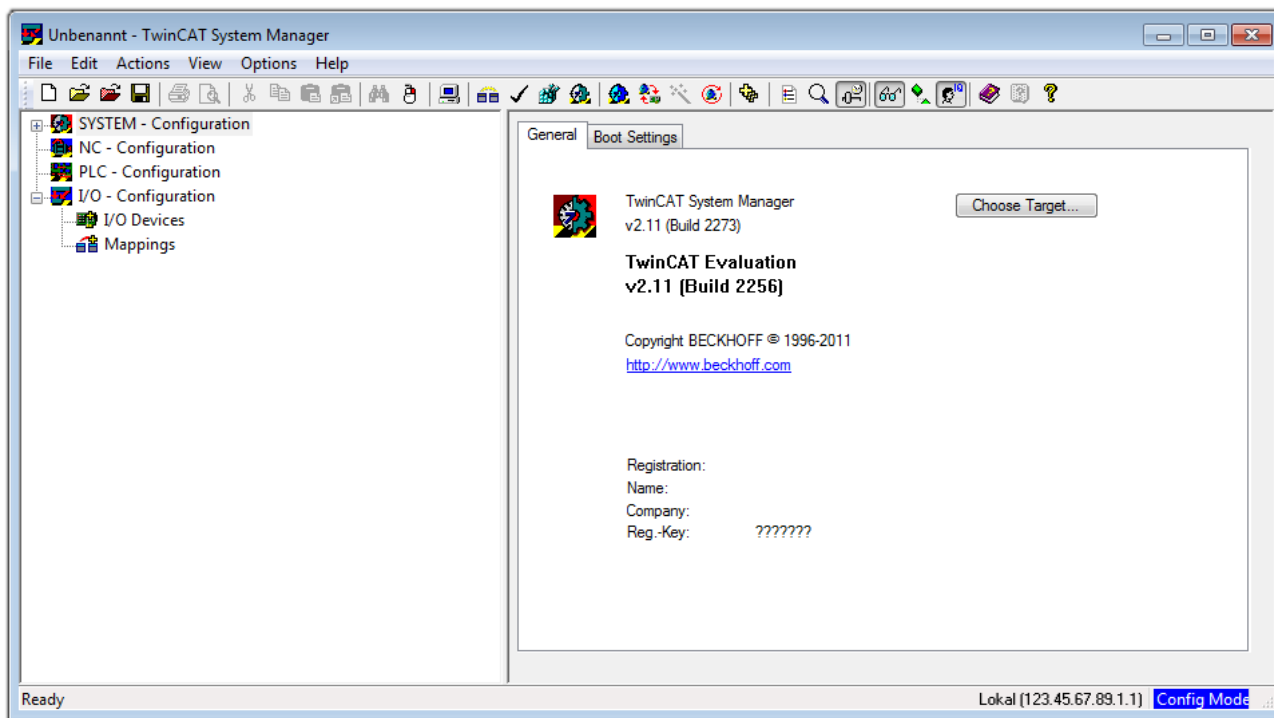
请注意，一个配置可能有多个任意组合；例如，EL1004 端子模块可以接在耦合器之后，或者 EL2008 端子模块可以附加到 CX2040 的右侧而无须 EK1100 耦合器。

5.1.1 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：



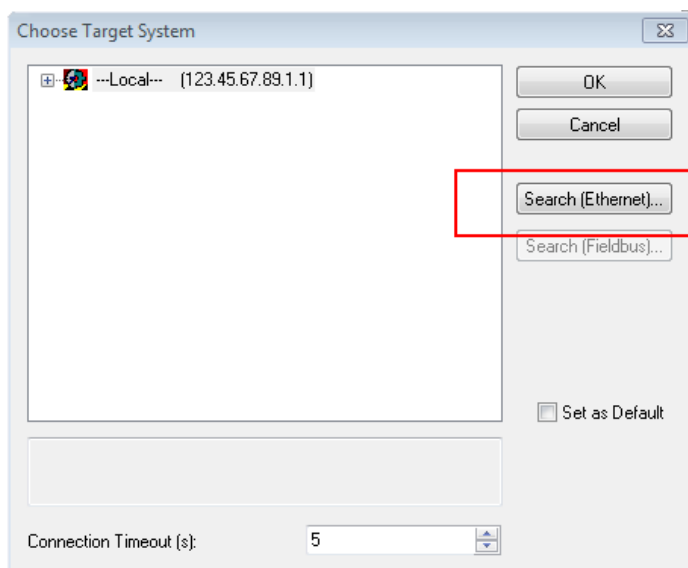
附图 24: TwinCAT 2 的初始用户界面

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） [► 53]”。

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



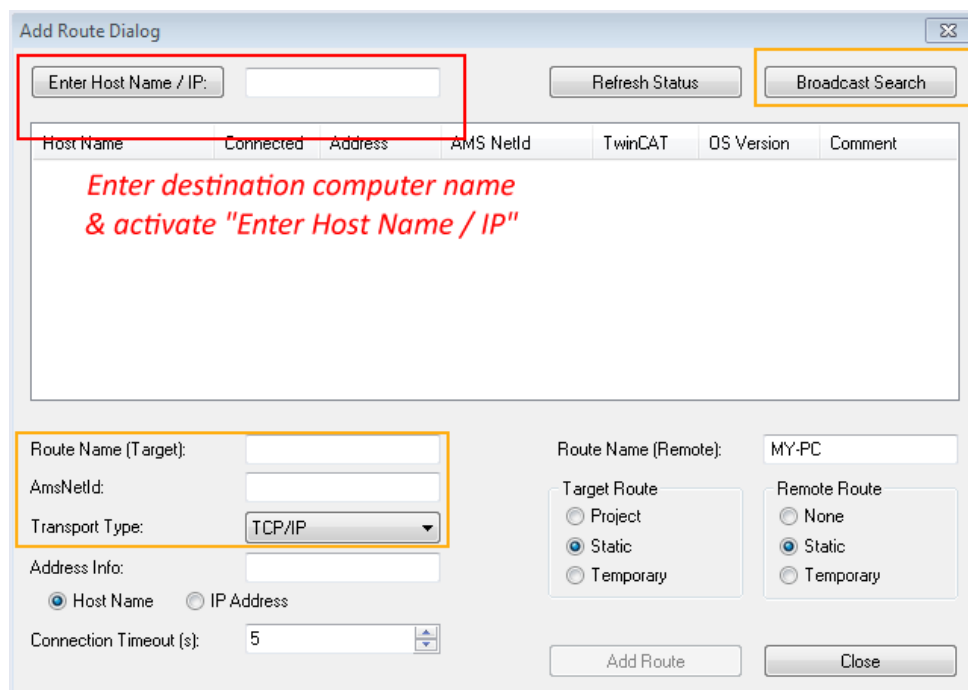
“ ” 或“F8”键，打开以下窗口：



附图 25: 选择目标系统

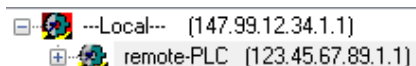
使用“Search (Ethernet)...”进入目标系统。弹出下一个对话框，在此可以选择：

- 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 26: 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）：



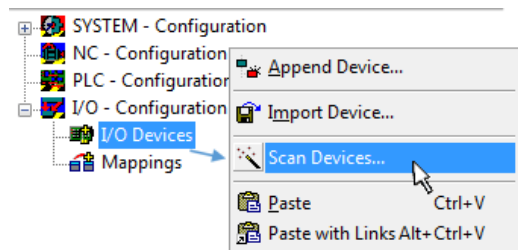
按“OK”确认，就可以通过 TwinCAT System Manager 访问目标系统。

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

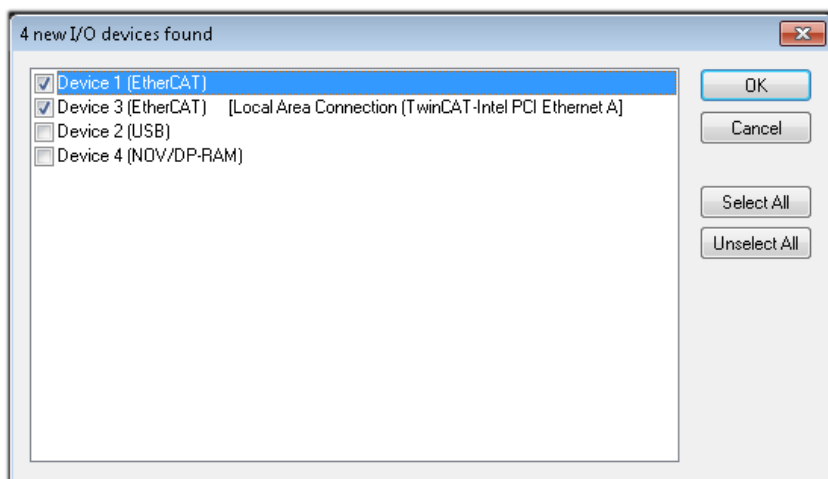
菜单，选择 “Scan Devices...”，或者通过  在菜单栏中启动该操作。首先，可能需要点击  或通过菜单

“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 27: 选择 “Scan Devices...”

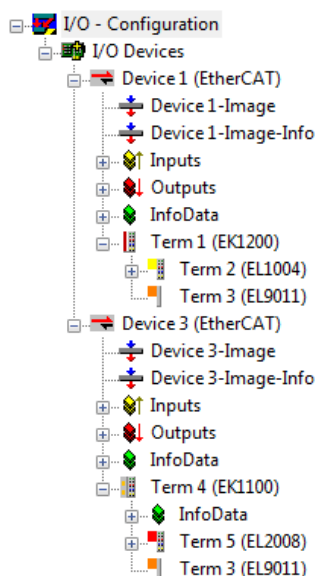
确认接下来的警告信息，并在对话框中选择 “EtherCAT” 设备：



附图 28: 自动检测 I/O 设备：选择要集成的设备

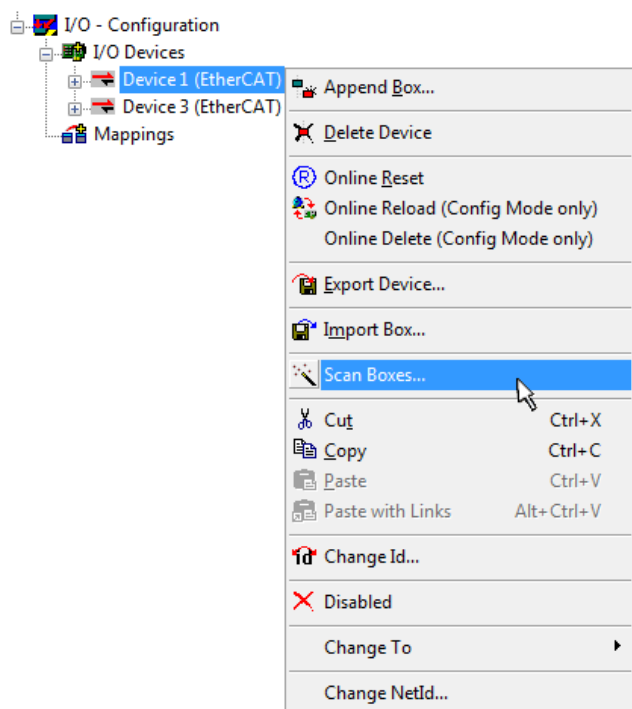
确认 “Find new boxes” 信息，以确定连接到设备的端子模块。“Free Run” 自由运行功能允许在 “Config Mode” 配置模式下操作输入和输出值，这个功能也需要确认才能启用。

根据本节开头描述的示例配置 [\[► 49\]](#)，结果如下：



附图 29: 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 30: 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：

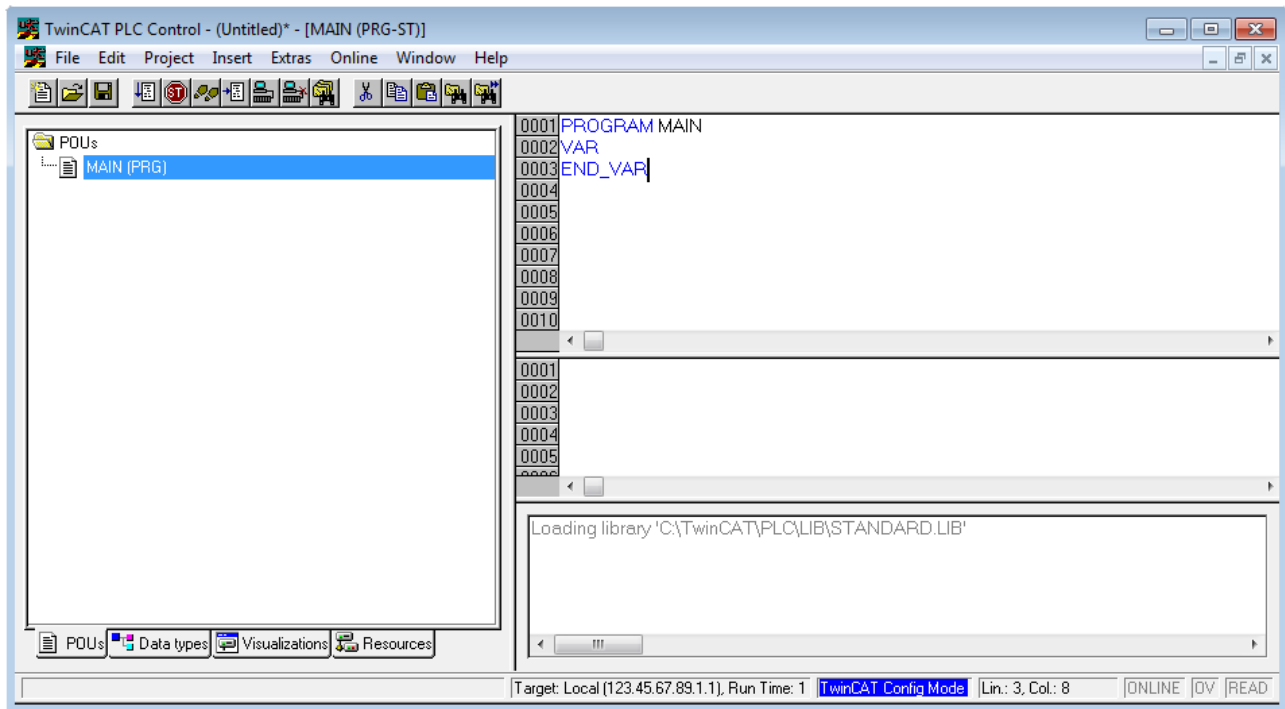
- 基于文本的语言
 - 指令表 (IL)
 - 结构化文本 (ST)

- 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)

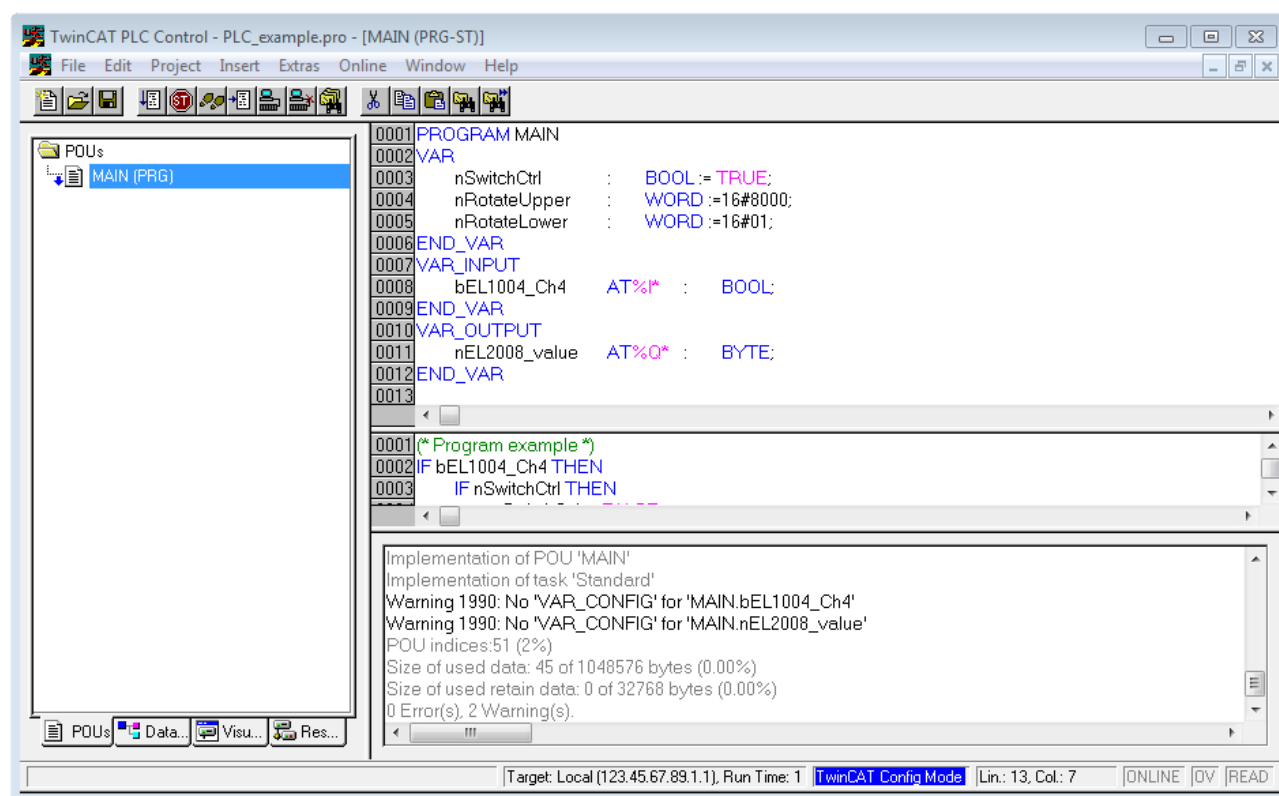
下面的内容只用到结构化文本 (ST)。

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



附图 31: 刚启动的 TwinCAT PLC Control

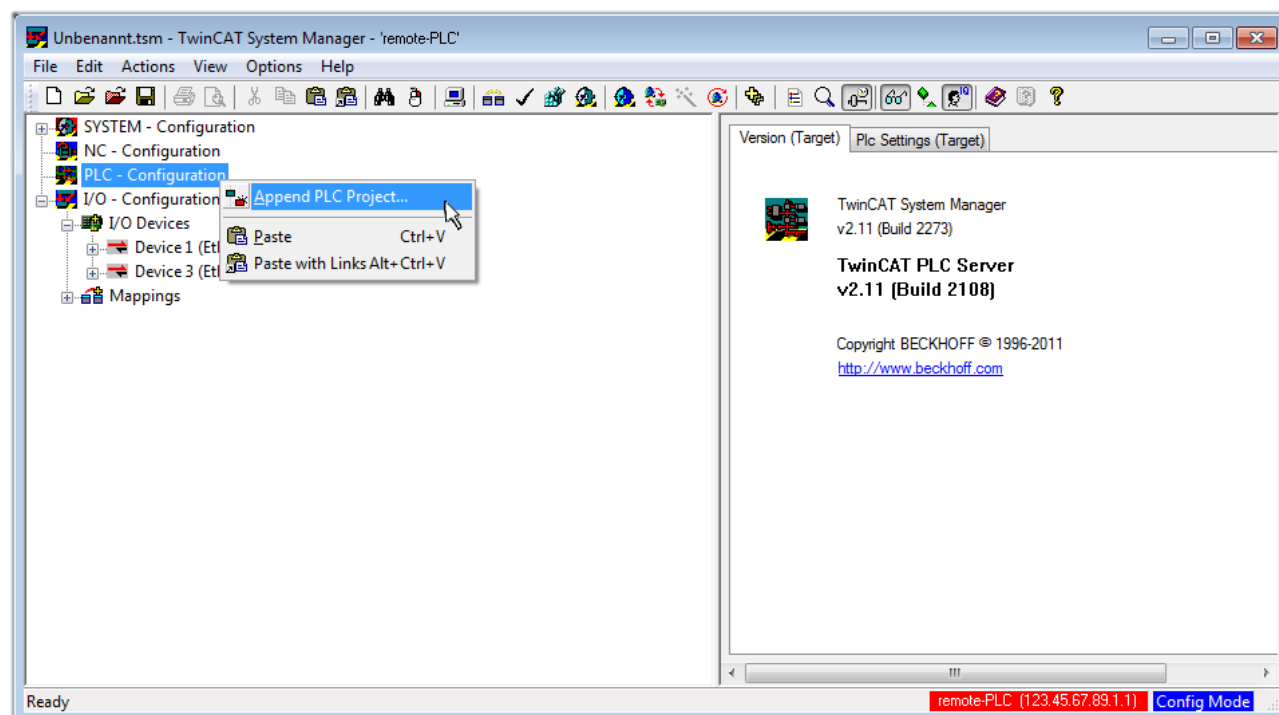
在示例项目中创建变量和程序，并保存为名称 “PLC_example.pro”：



附图 32: 示例程序编译后，包含未分配地址的变量

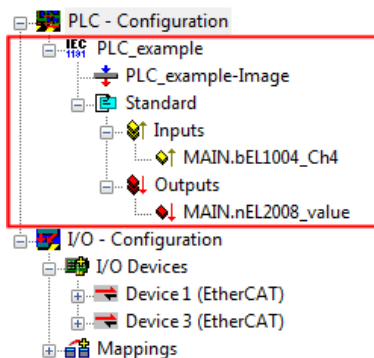
编译过程后的警告 Warning 1990 (missing “VAR_CONFIG”)，表示定义的外部变量（含标志符 “AT%I*” 或 “AT%Q*” ）还没有分配地址。编译成功后，TwinCAT PLC Control 会在项目路径下创建一个 “*.tpy” 文件，该文件包含了指定的 IO 变量，但 System Manager 还没有为其分配地址，因此出现了警告。只要在 System Manager 中引入该 .tpy 文件并保存，再次编译时警告就不会再出现了。

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 33: 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：

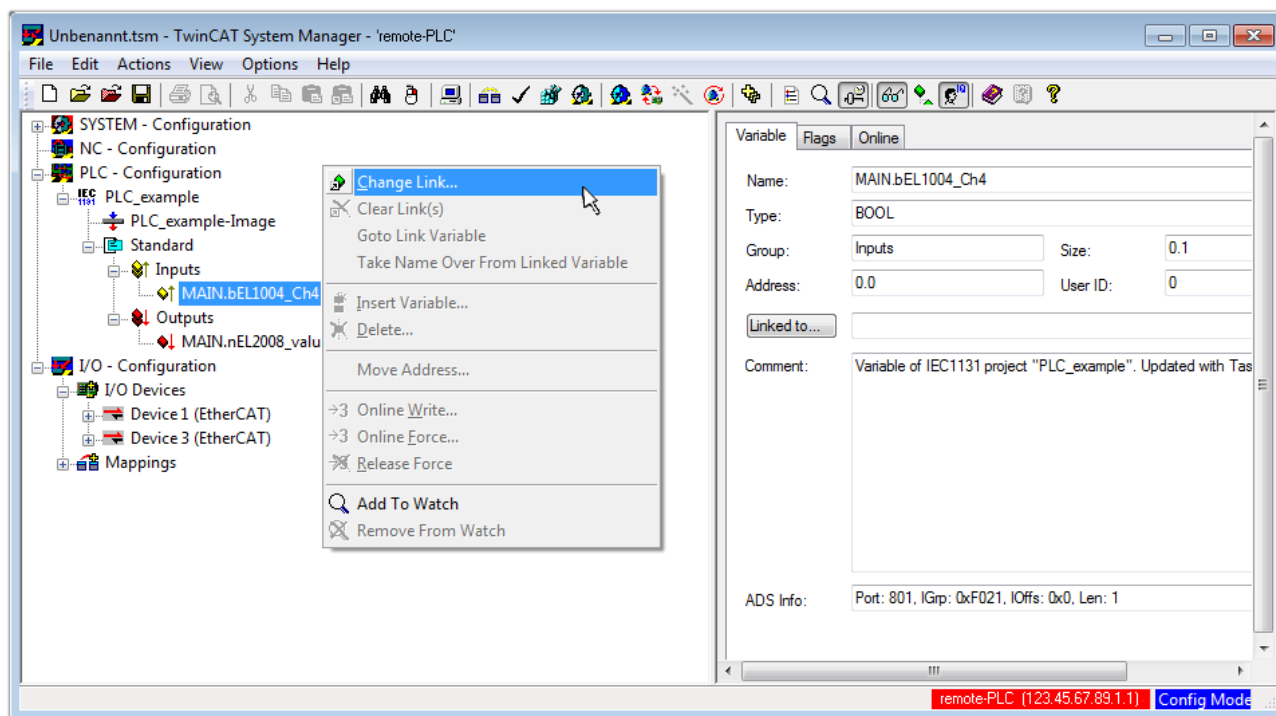


附图 34: 在 System Manager 的 PLC Configuration 下导入 PLC 项目

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。

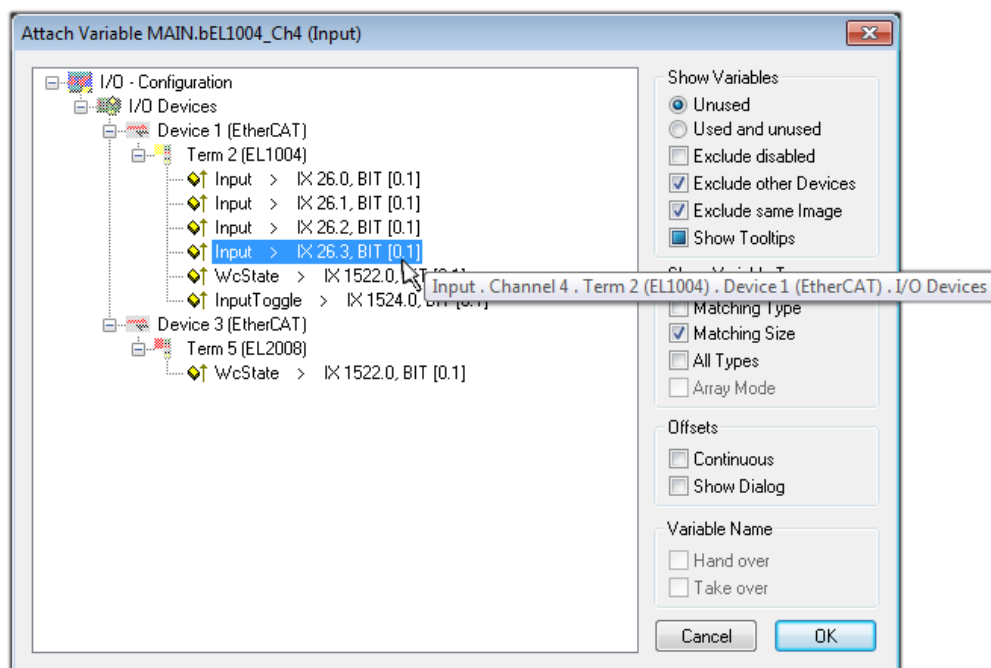
变量分配

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”：



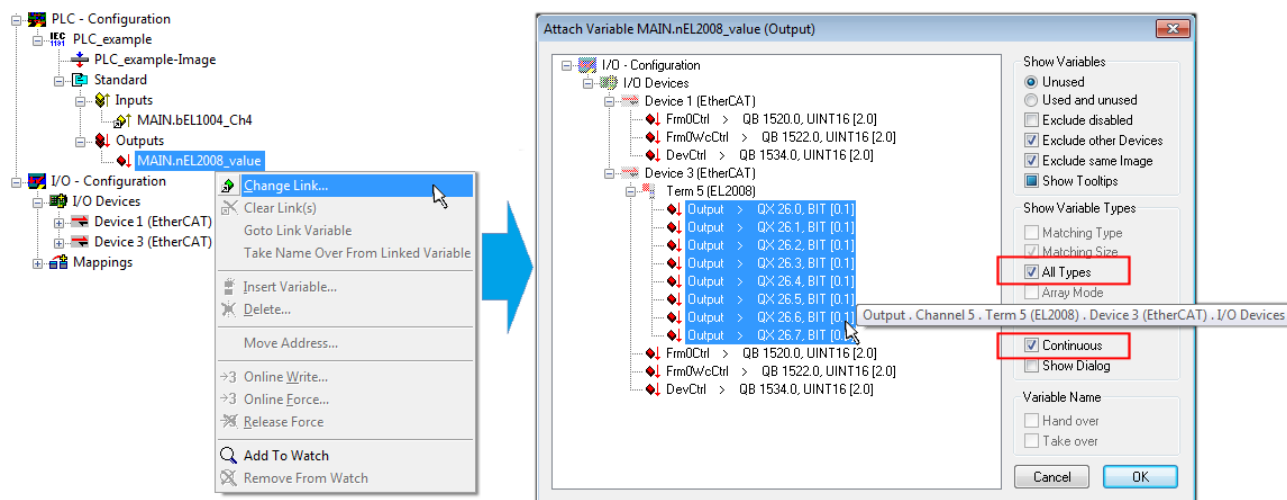
附图 35: 在 PLC 变量和过程对象之间建立链接

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：

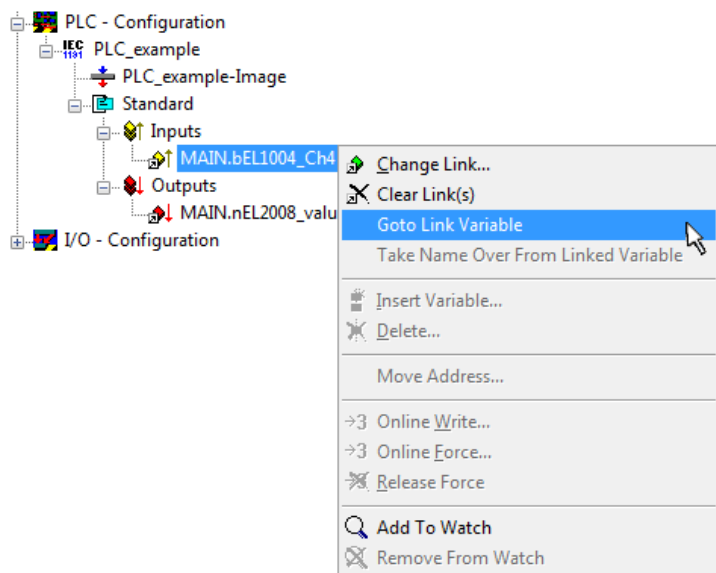


附图 36: 选择 BOOL 类型的 PDO

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 37: 同时选择几个 PDO：勾选“Continuous”和“**All types**”

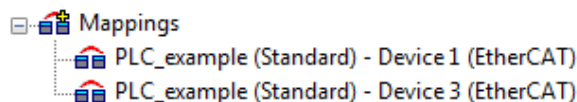
请注意，“Continuous”复选框也要勾选。这种设计旨在将变量“nEL2008_value”的字节中包含的位按顺序分配给 EL2008 端子模块的所有八个选定的输出位。这样就可以在 PLC 程序中用一个字节对应端子模块的所有 8 个输出，字节的第 0 到 7 位分别对应模块的第 1 到 8 通道。在变量的黄色或红色对象处有一个特殊符号（），表示变量已链接。也可以通过从变量的右键菜单中选择“Goto Link Variable（转到链接变量）”来检查链接。此时，链接的对方（在这种情况下是 PDO）被自动选中：



附图 38: “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

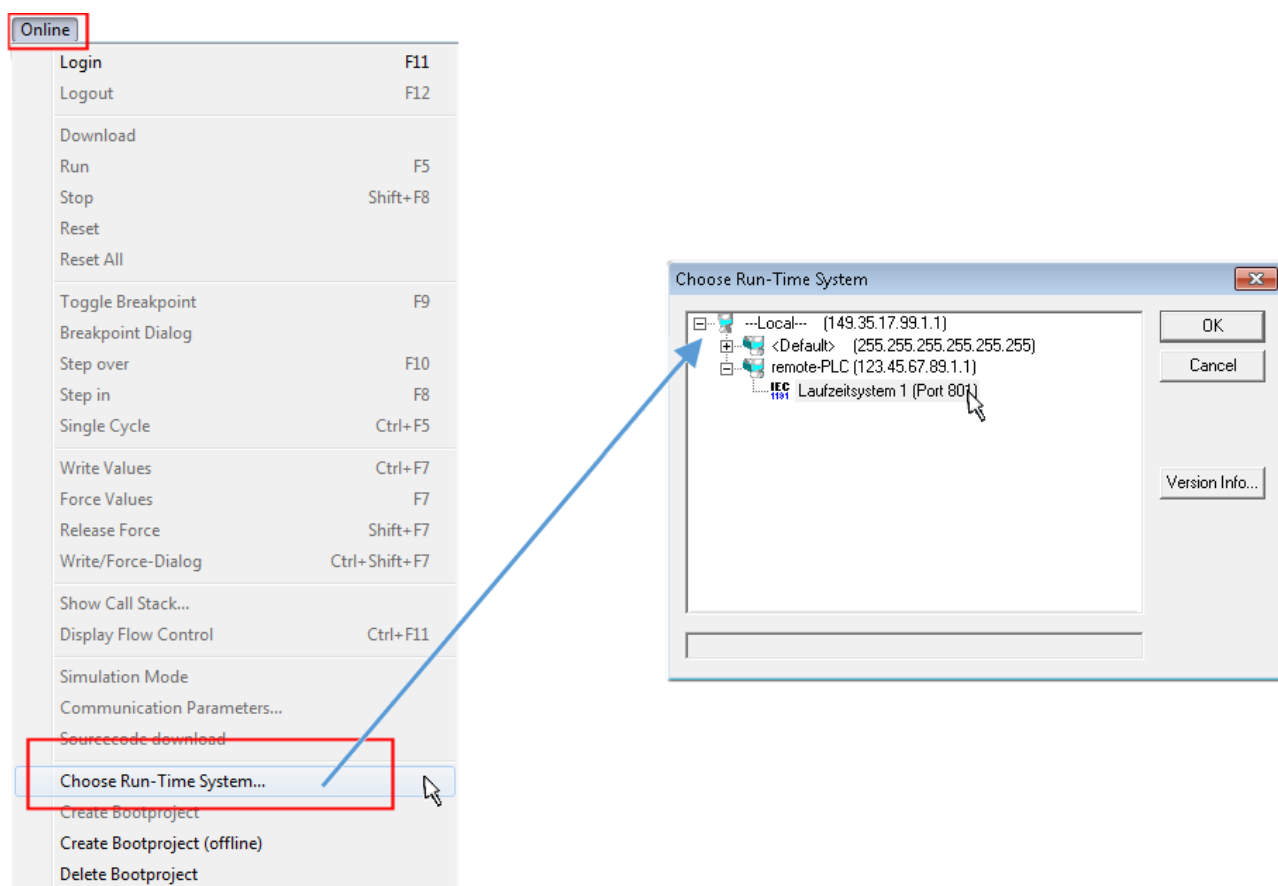
PDO 到 PLC 变量的分配过程建立了从控制器到端子模块的输入和输出的连接。下面激活该配置。首先，可以

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 **RTime 0%** 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

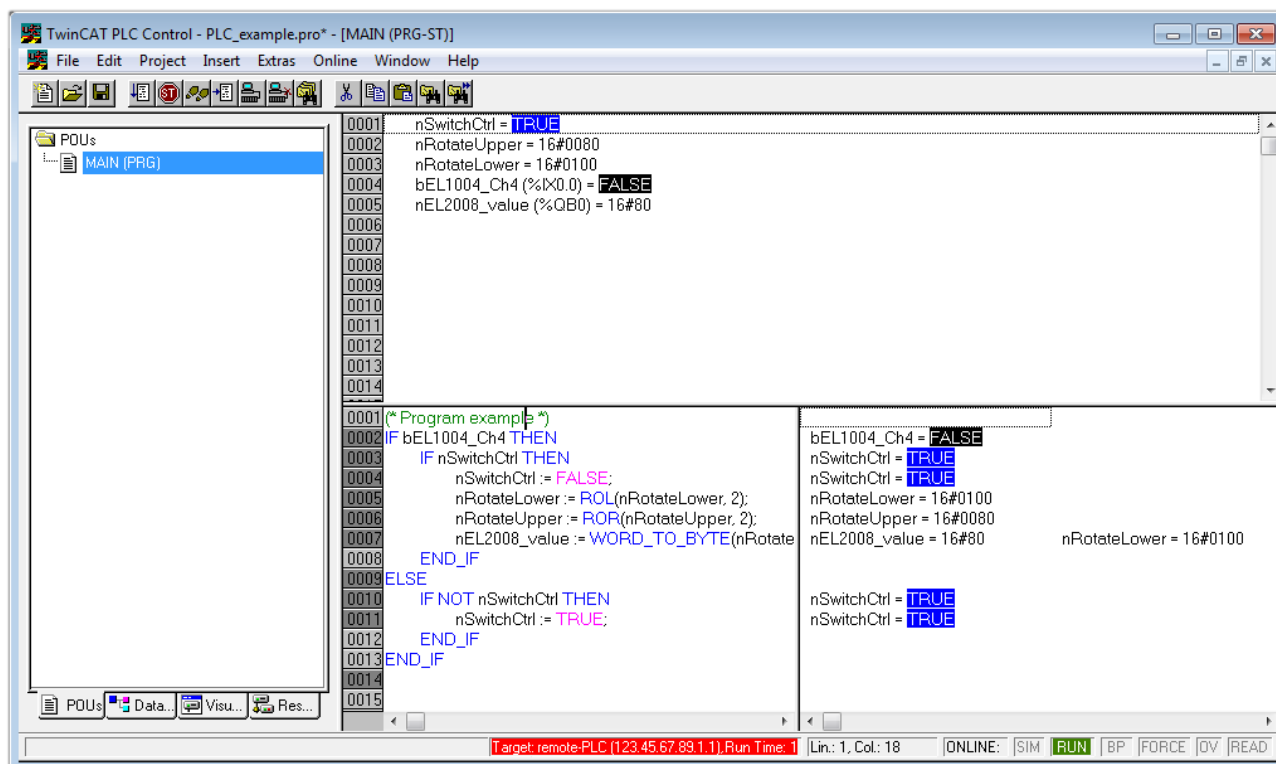
从远程系统操作控制器，必须先在 PLC Control 中通过菜单 “Online” → “Choose Runtime System...”，以便连接到 IPC/EPC：



附图 39: 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 40: PLC Control 登录，做好程序启动准备

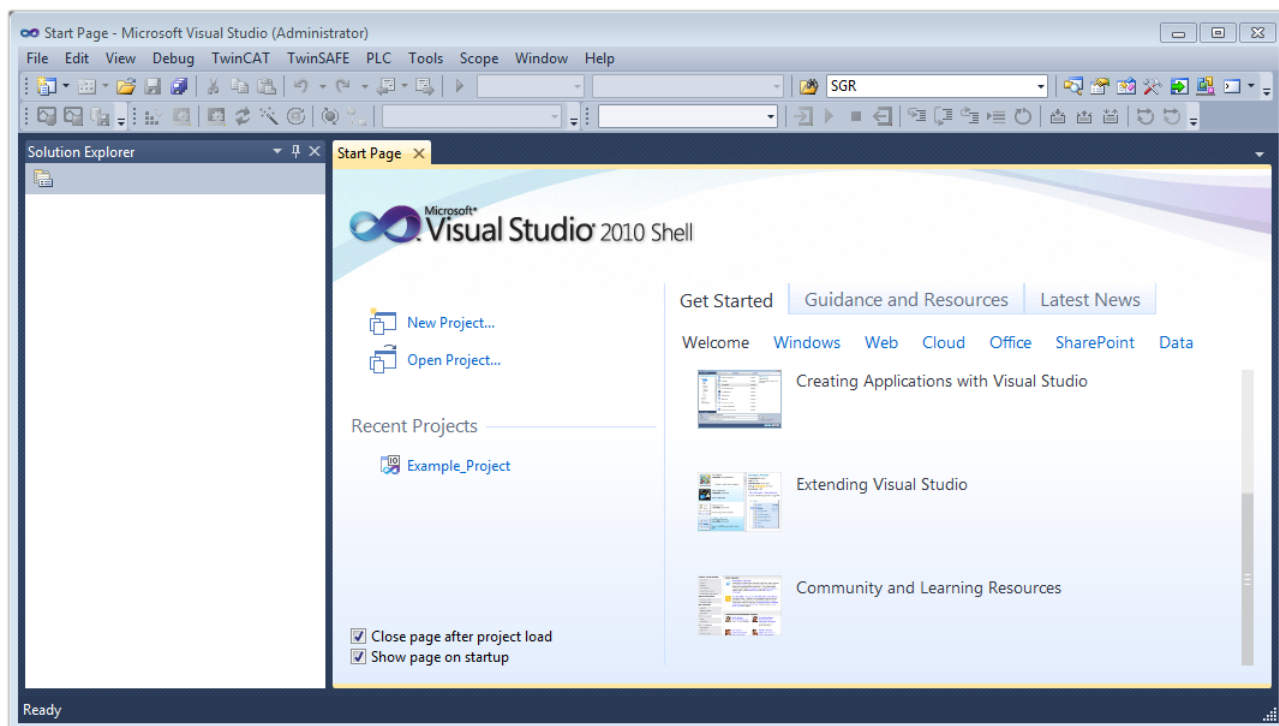
然后就可以通过“Online” → “Run”，F5 键或  启动 PLC。

5.1.2 TwinCAT 3

启动

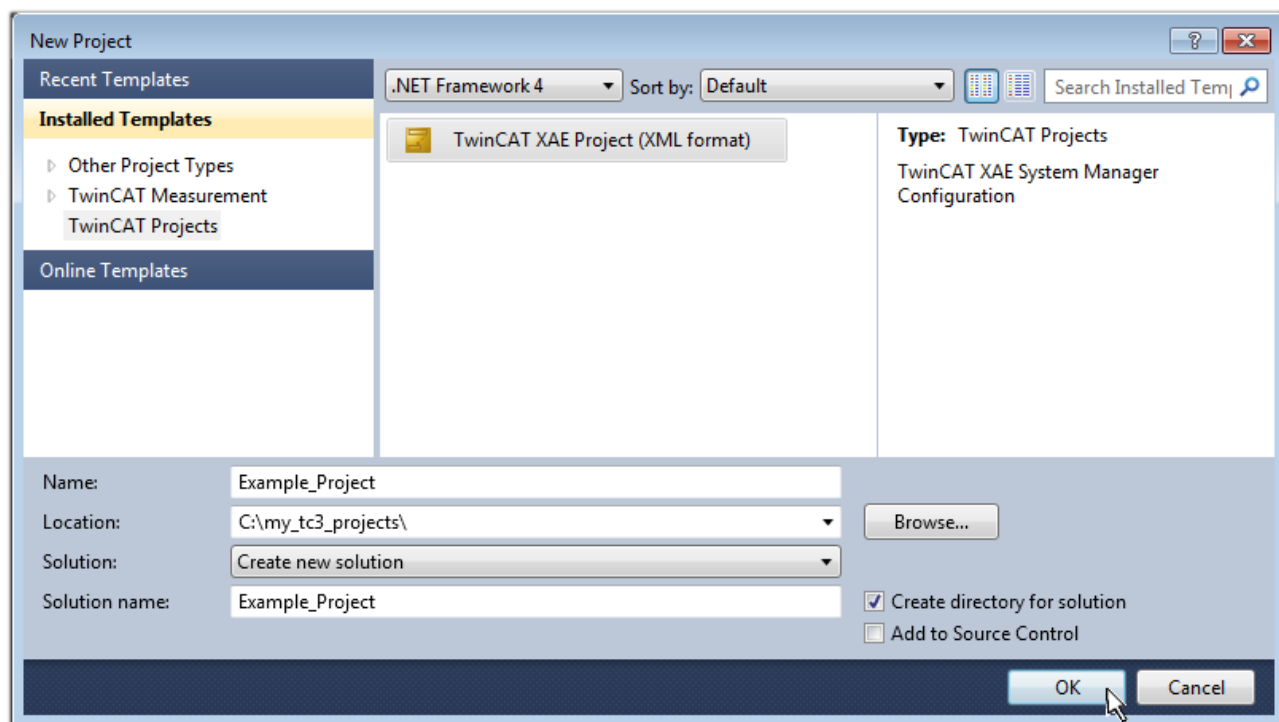
TwinCAT 3 集成于 Microsoft Visual Studio，所有功能都包含在一个开发环境中：启动后，项目文件浏览器显示在通用窗口区域的左侧（参见 TwinCAT 2 的“TwinCAT System Manager”），用于与电气组件进行通信。

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：



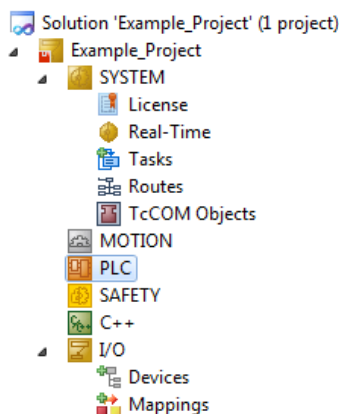
附图 41: TwinCAT 3 的初始用户界面

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 42: 新建 TwinCAT 3 项目

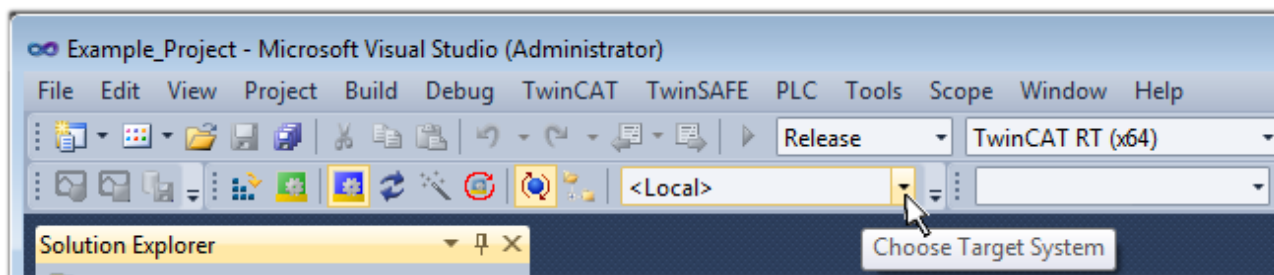
然后，在项目文件浏览器中就会显示新建的项目：



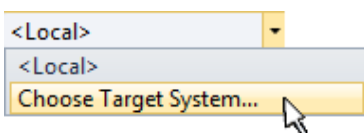
附图 43: 项目文件浏览器中的 TwinCAT 3 新建项目

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 \[► 64\]](#)”。

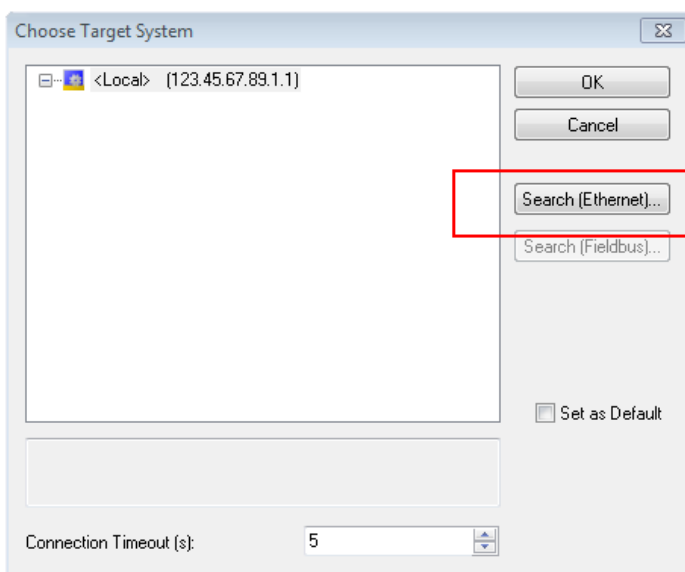
如果要从 TwinCAT 开发环境连接到另一个安装在远程 PLC 上的 TwinCAT Runtime 运行环境, 就得先识别到 Target System(目标系统)。通过菜单栏中的符号:



展开下拉菜单:



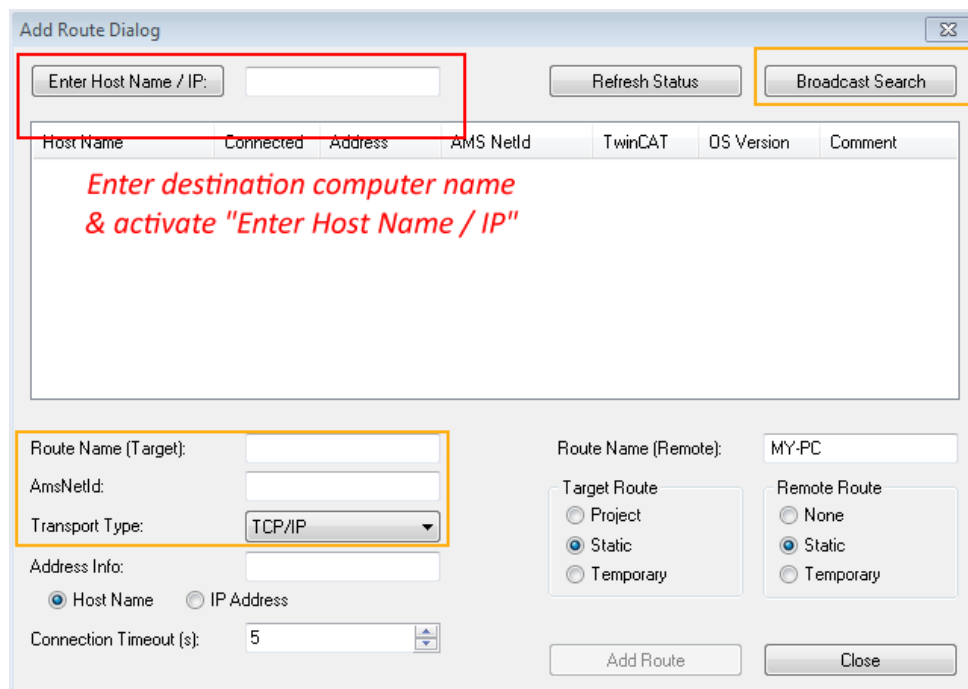
并打开以下窗口:



附图 44: 选择对话框: Choose the target system

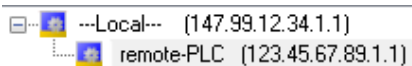
使用“Search (Ethernet)...”进入目标系统。弹出下一个对话框，在此可以选择：

- 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 45: 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）：



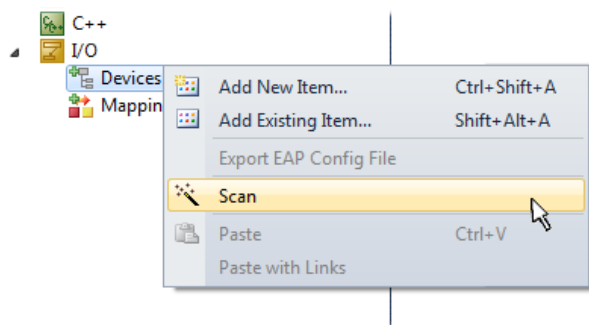
点击“OK”按钮确认，就可以通过 Visual Studio shell 访问目标系统。

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

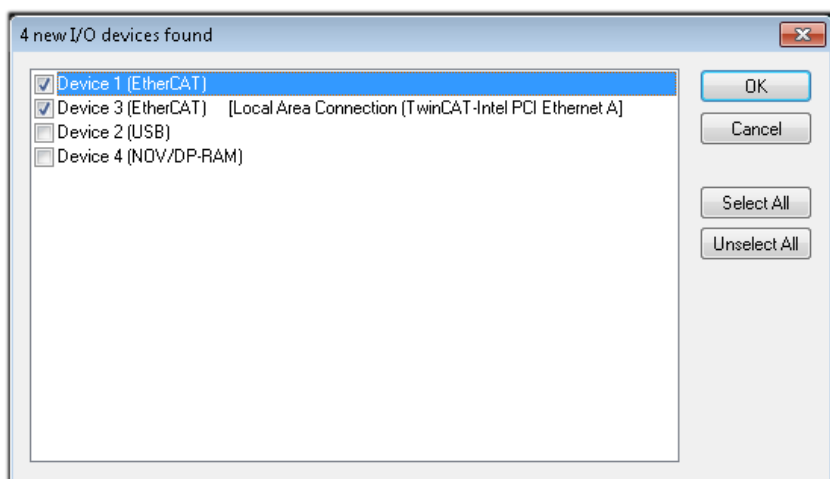
打开右键菜单，选择“Scan”或通过菜单栏中的  开始操作。首先，TwinCAT System Manager 可能需

要通过  或通过菜单“TwinCAT”→“Restart TwinCAT (Config Mode)”设置成“Config Mode”。



附图 46: 选择“Scan(扫描)”

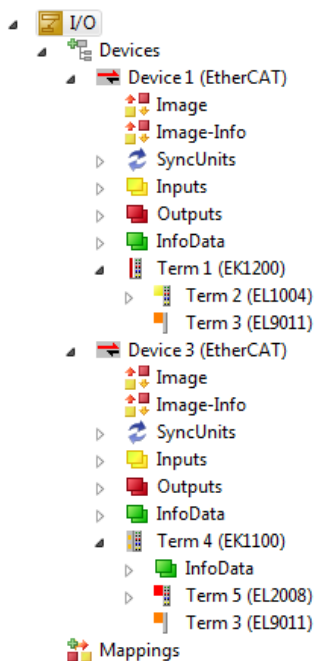
确认接下来的警告信息，并在对话框中选择“EtherCAT”设备：



附图 47: 自动检测 I/O 设备：选择要集成的设备

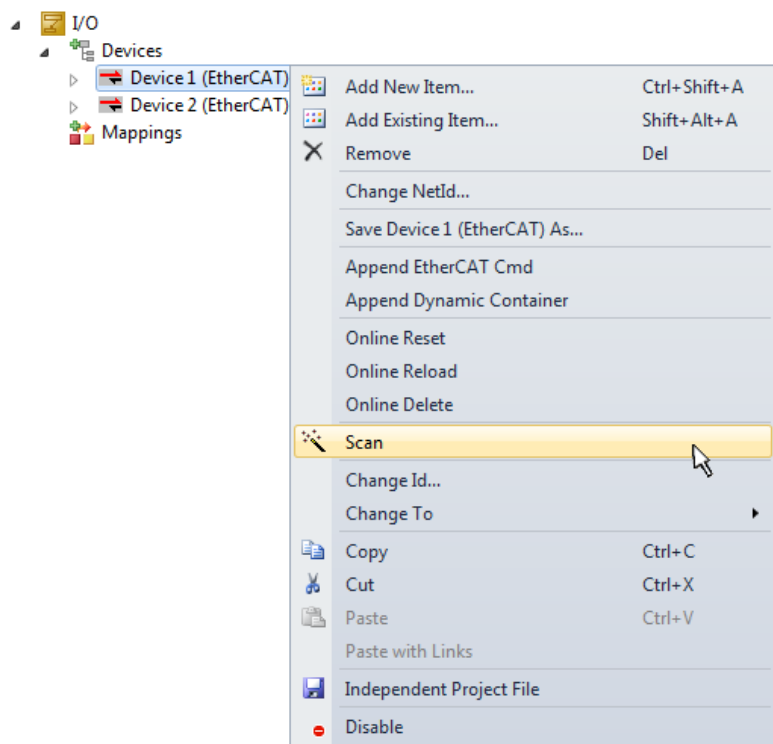
确认“Find new boxes”信息，以确定连接到设备的端子模块。“Free Run”自由运行功能允许在“Config Mode”配置模式下操作输入和输出值，这个功能也需要确认才能启用。

根据本节开头描述的示例配置 [► 49]，结果如下：



附图 48: 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 49: 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。

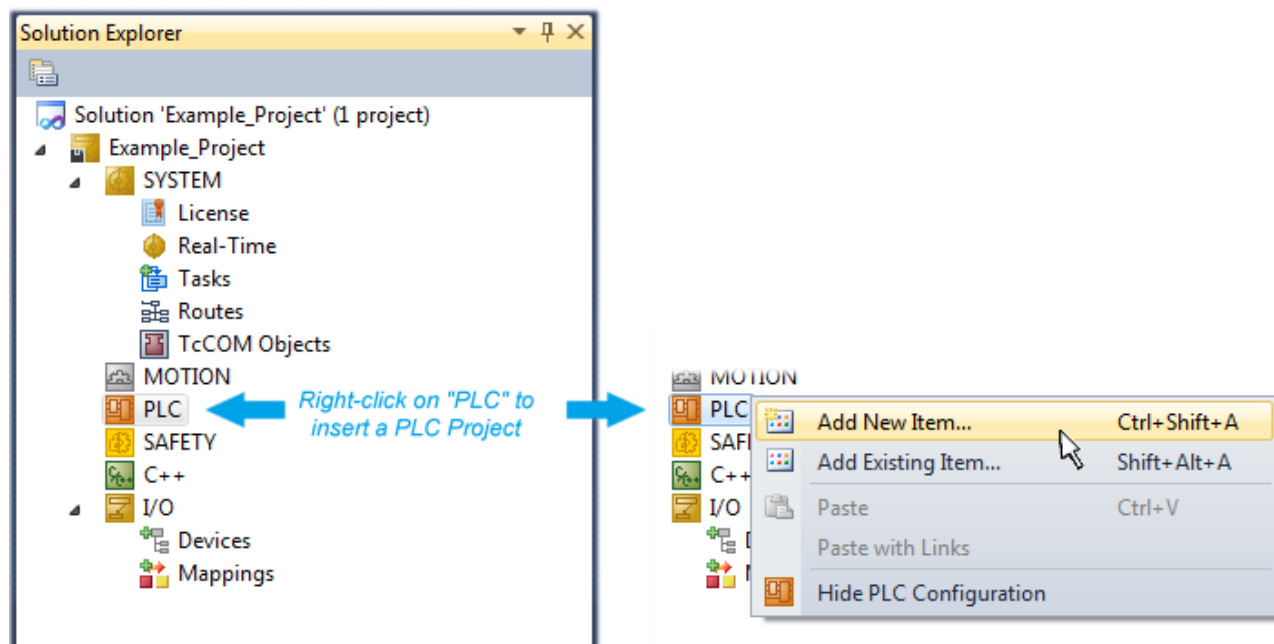
对 PLC 进行编程

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)

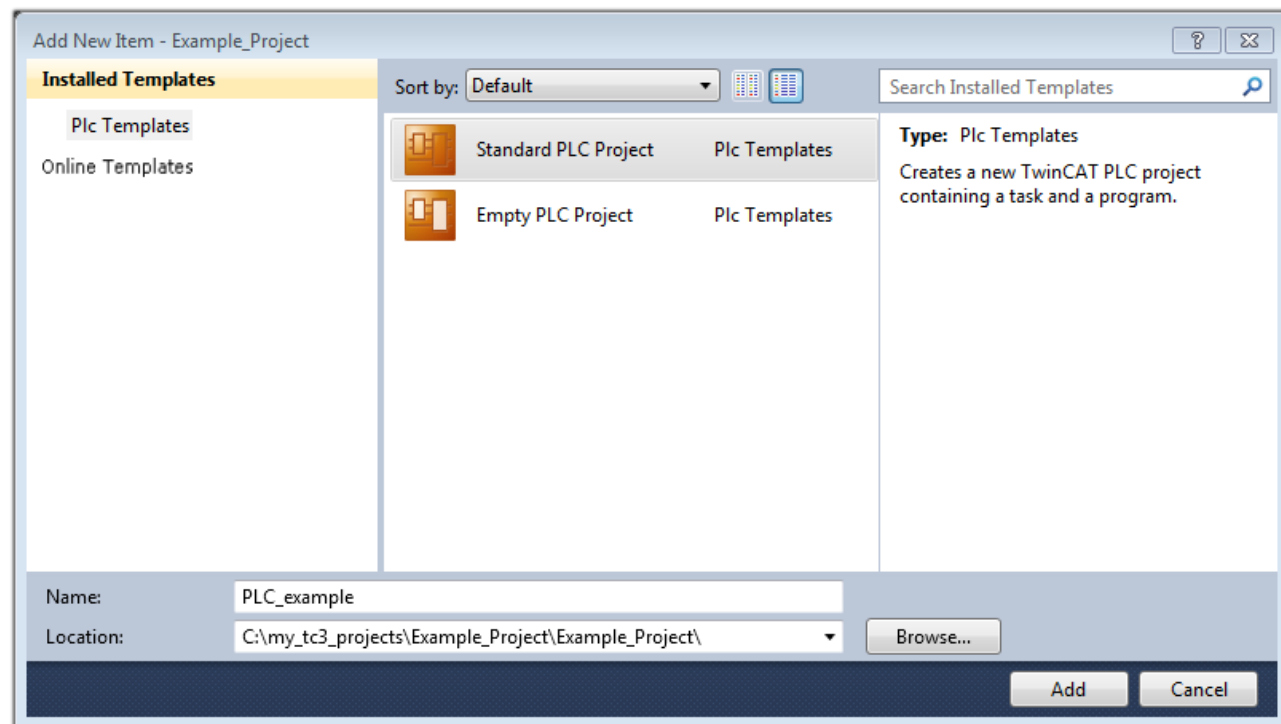
下面的内容只用到结构化文本 (ST)。

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：



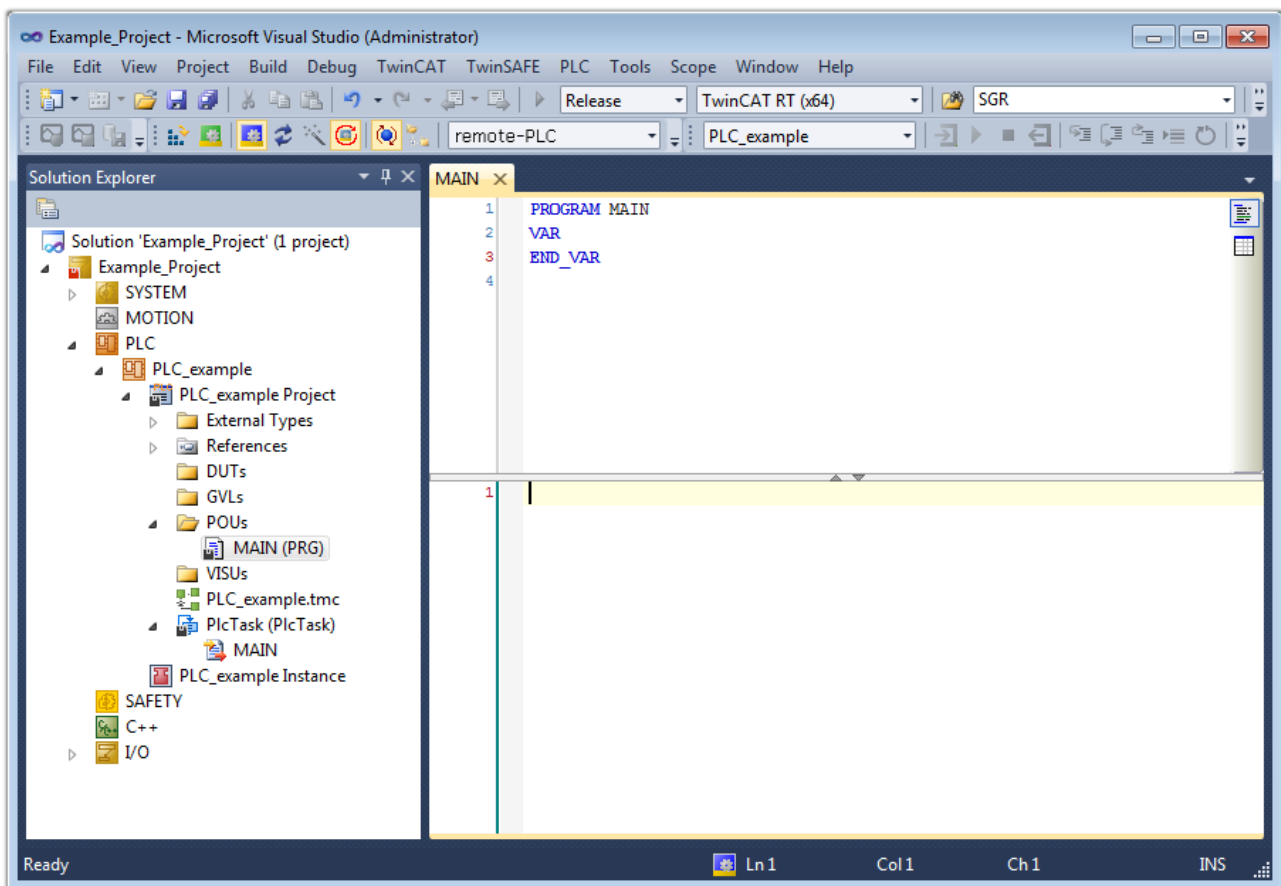
附图 50: 在节点“PLC”下添加编程环境

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：



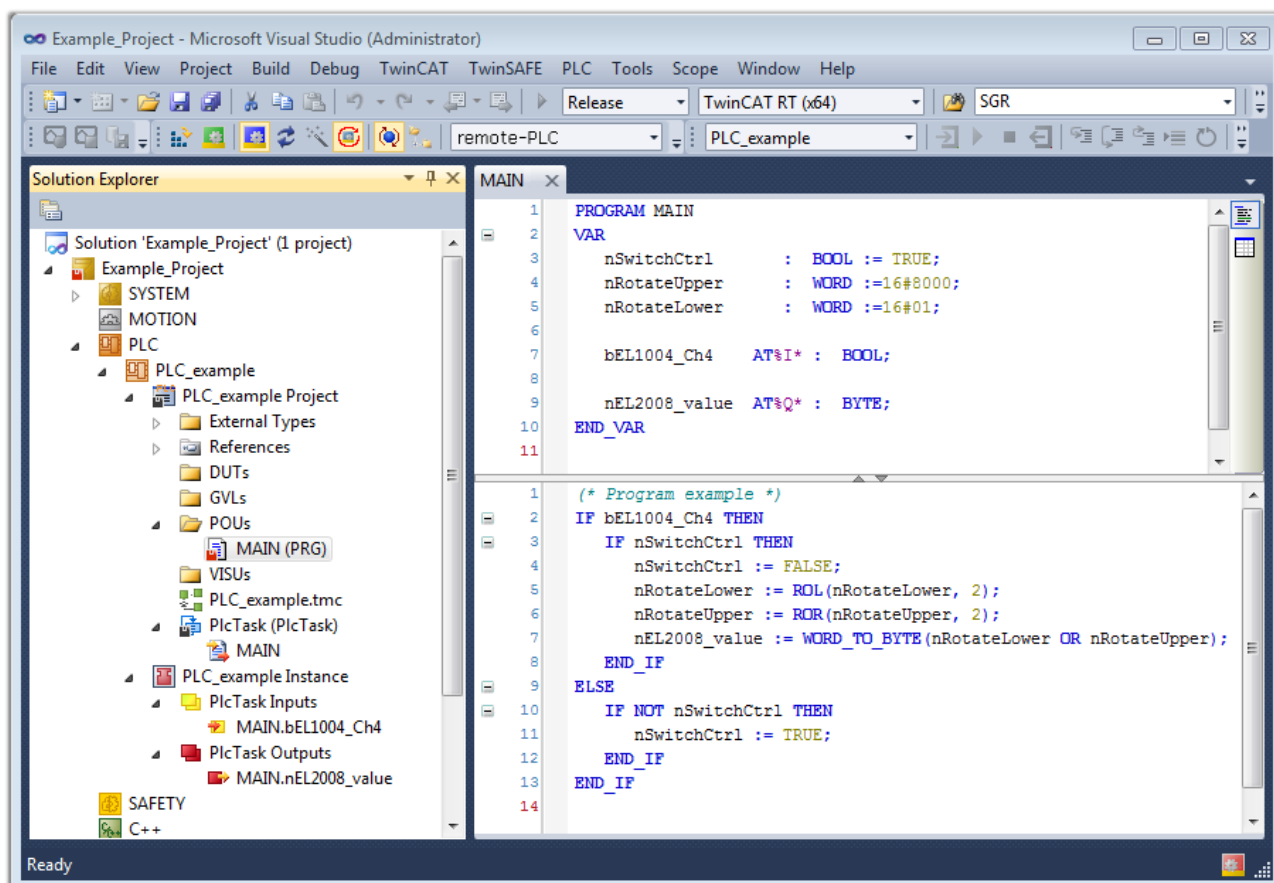
附图 51: 为 PLC 编程环境指定名称和目录

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：



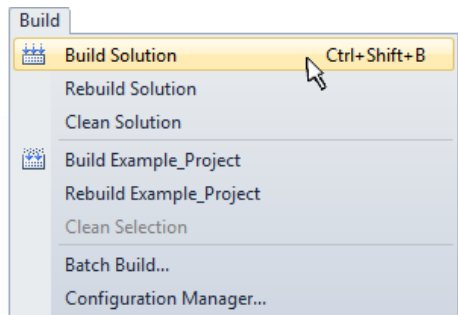
附图 52: 标准 PLC 项目的初始程序 “Main”

现在，已经为下一阶段的工作创建了示例变量和示例程序。



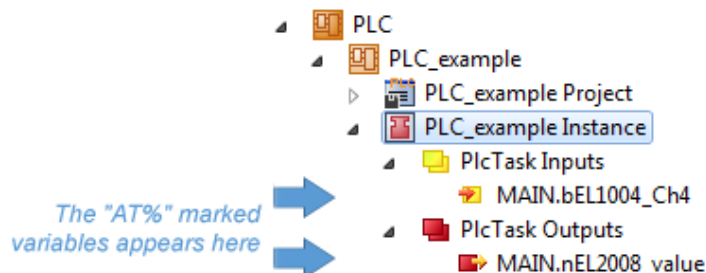
附图 53: 示例程序编译后，包含未分配地址的变量

现在，控制程序被创建为一个项目文件夹，接下来是编译过程：



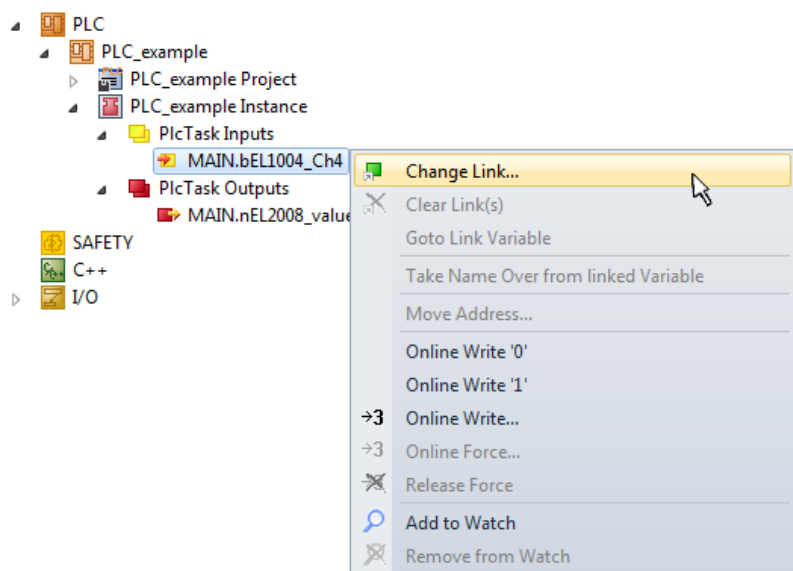
附图 54: 开始编译程序

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：



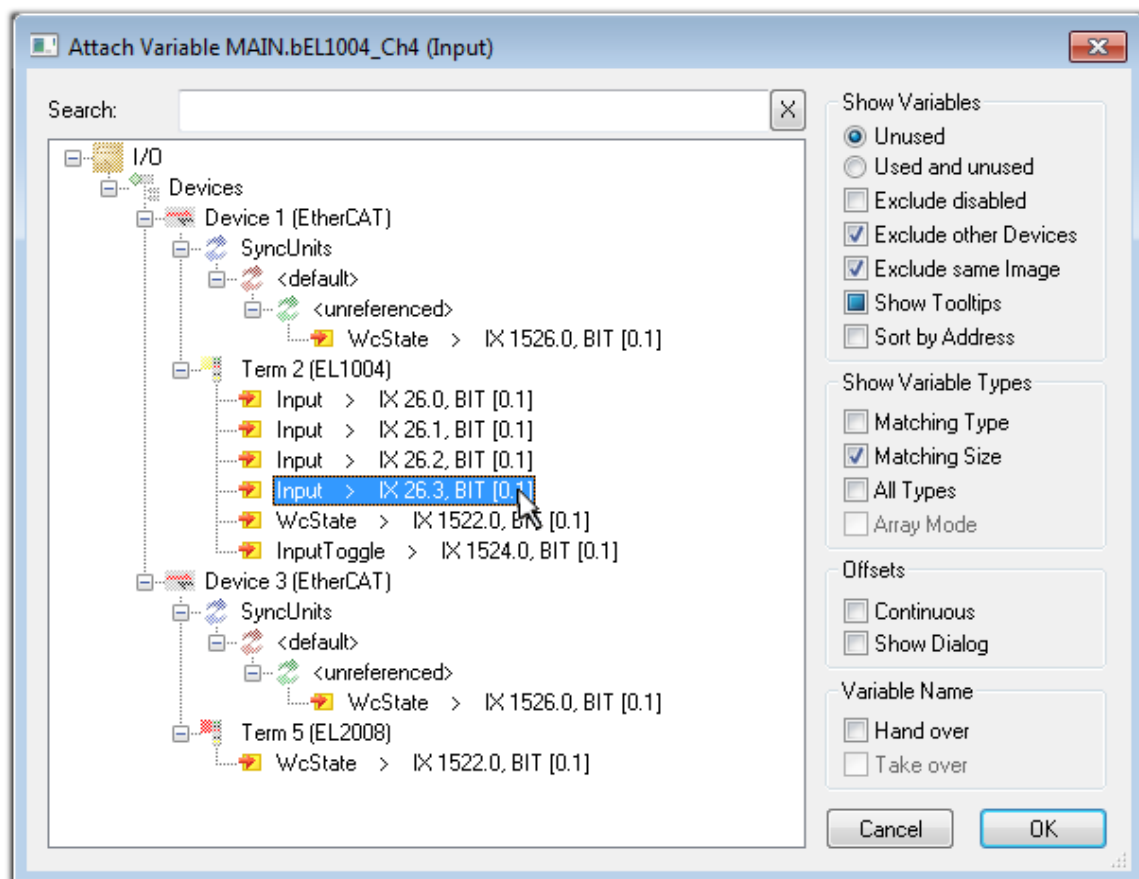
变量分配

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：



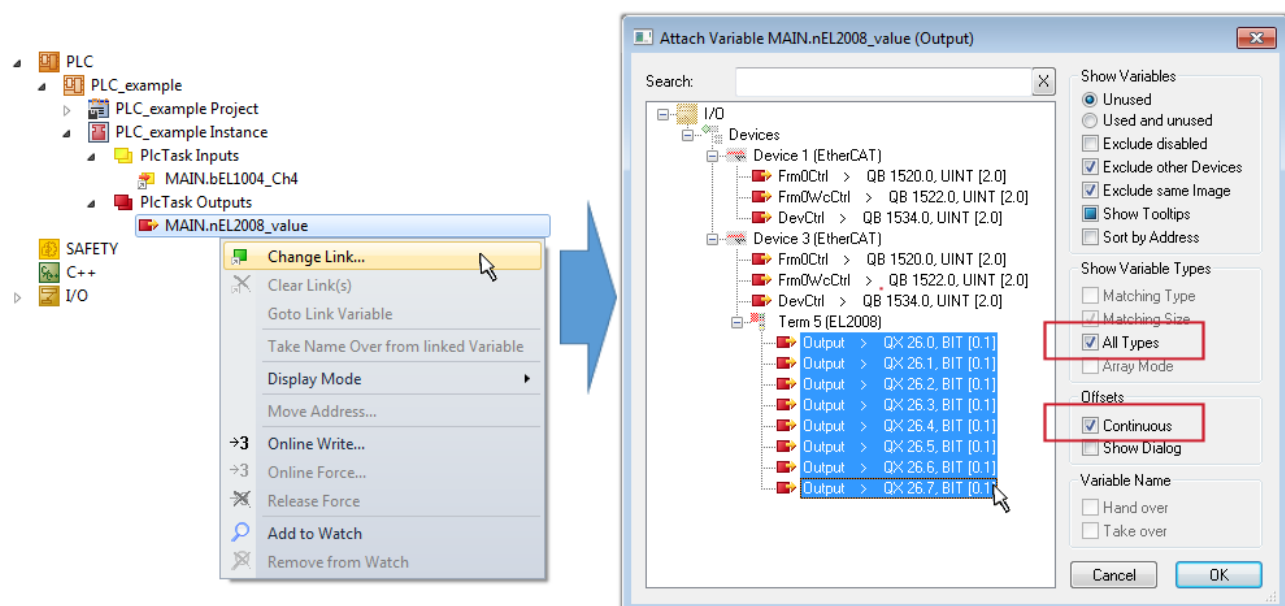
附图 55: 在 PLC 变量和过程对象之间建立链接

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：



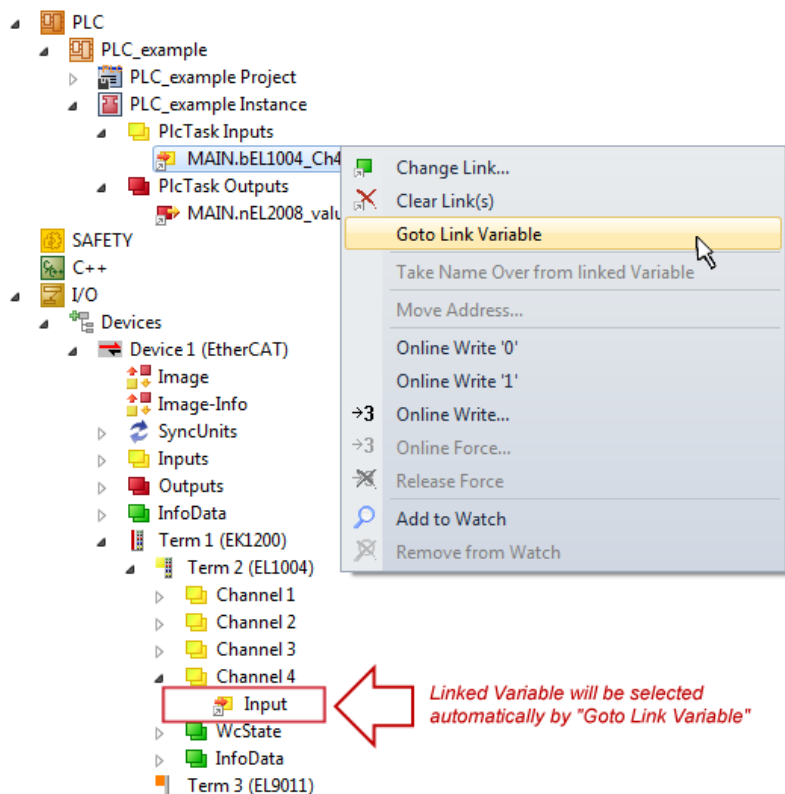
附图 56: 选择 BOOL 类型的 PDO

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 57: 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 58: “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

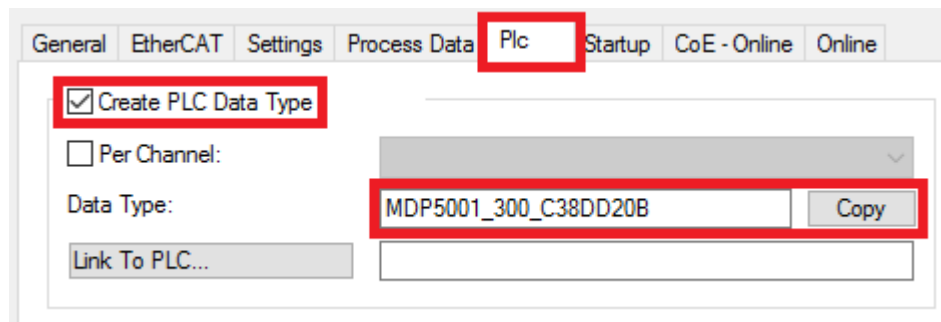


以下类型的变量分配方式只适用于 TwinCAT V3.1.4024.4 及以上版本，并且只能用于带微处理器的端子模块。

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。

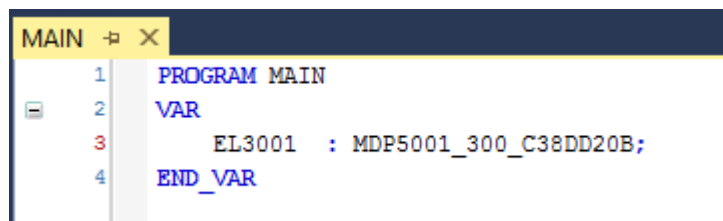
下面以 EL3001 单通道模拟量输入端子模块 -10...+10 V 为例，说明其操作步骤：

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。



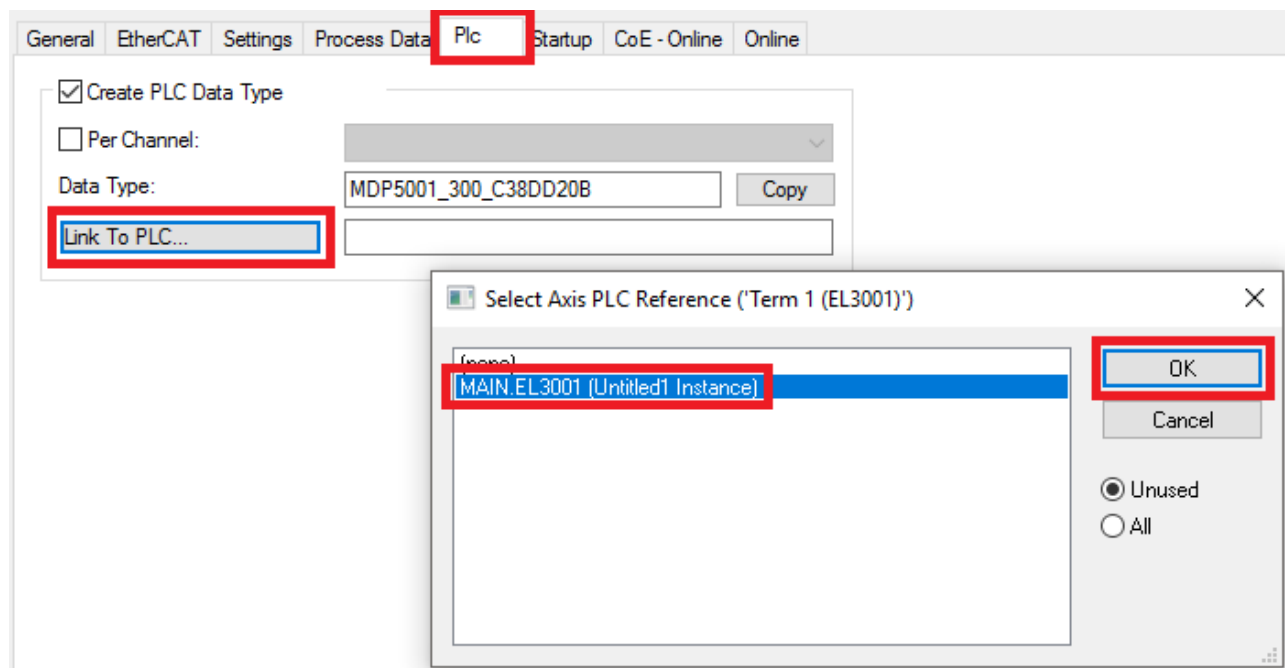
附图 59: 创建一个 PLC 数据类型

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



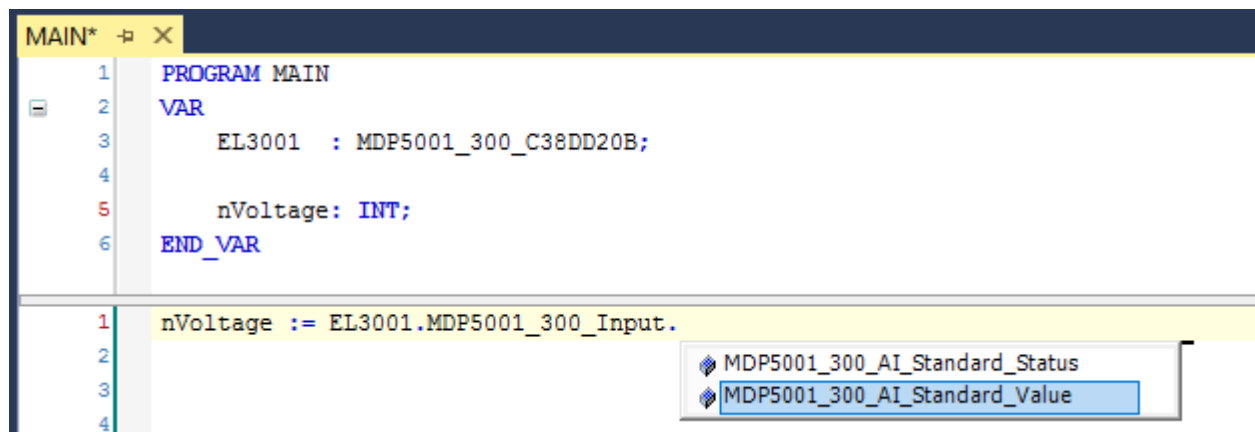
附图 60: 结构体的实例

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 61: 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 62: 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：

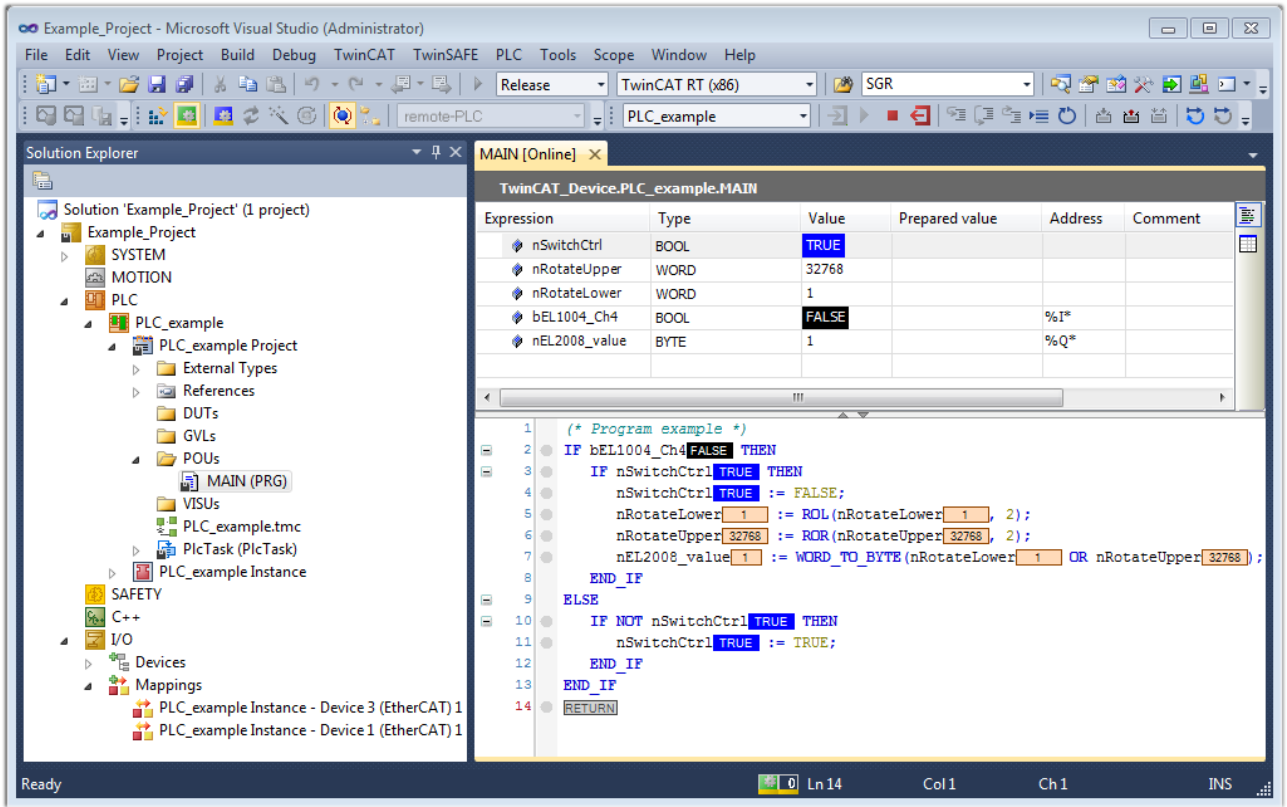
-  Mappings
 -  PLC_example Instance - Device 3 (EtherCAT) 1
 -  PLC_example Instance - Device 1 (EtherCAT) 1

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 63: 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）。

5.2 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>。

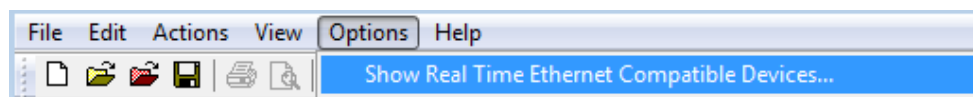
5.2.1 TwinCAT real-time 实时驱动程序的安装

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：

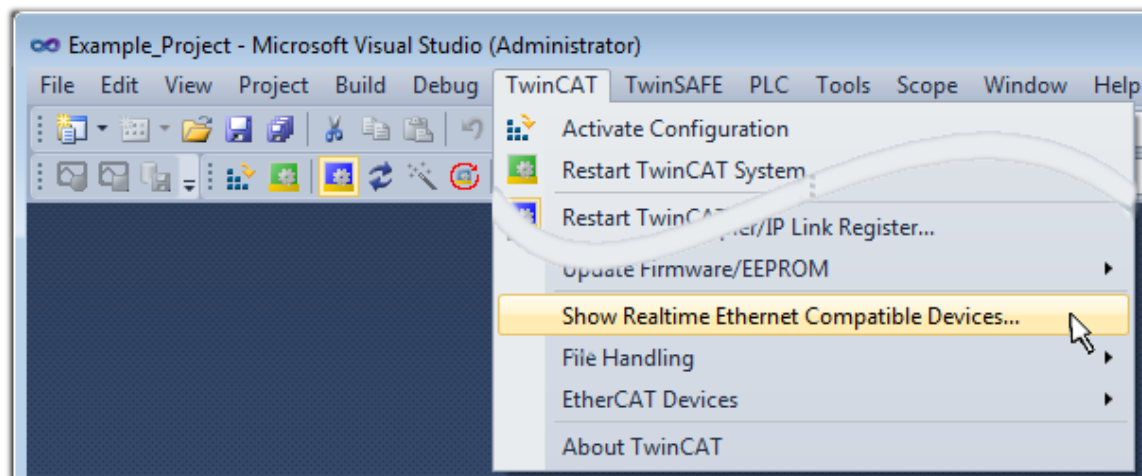
A：通过 TwinCAT 适配器对话框

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。



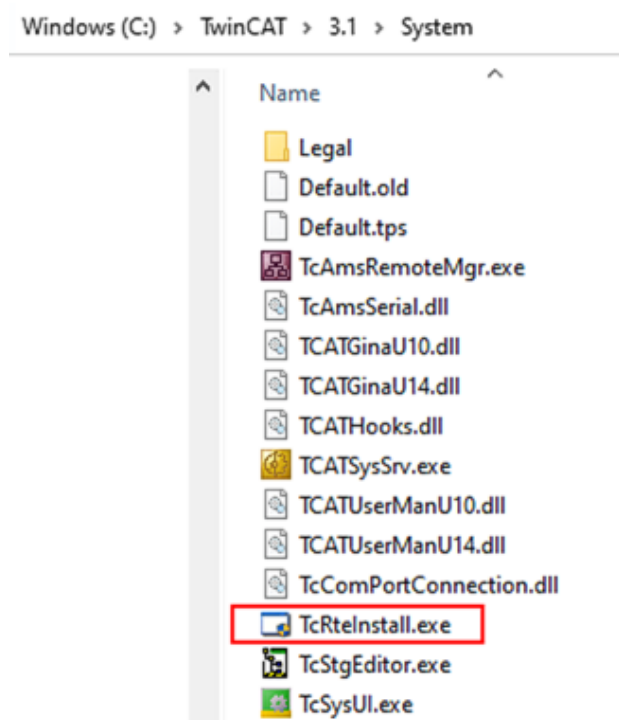
附图 64: System Manager “选项” (TwinCAT 2)

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



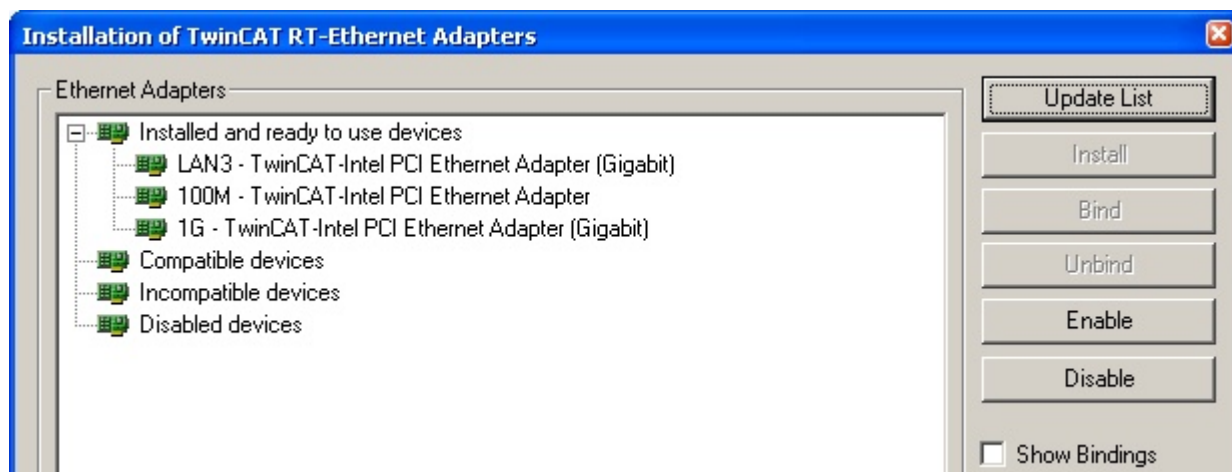
附图 65: 在 VS Shell 下调用 (TwinCAT 3)

B : 通过 TwinCAT 目录下的 TcRteInstall.exe



附图 66: TwinCAT 目录下的 TcRteInstall

两种情况下均出现以下对话框：



附图 67: 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。

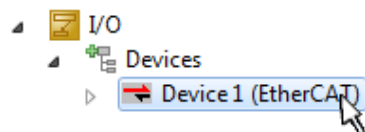
关于未签名驱动程序的 Windows 警告可以忽略。

或者，首先可以插入一个 EtherCAT 设备，如离线配置创建 [► 84] 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：

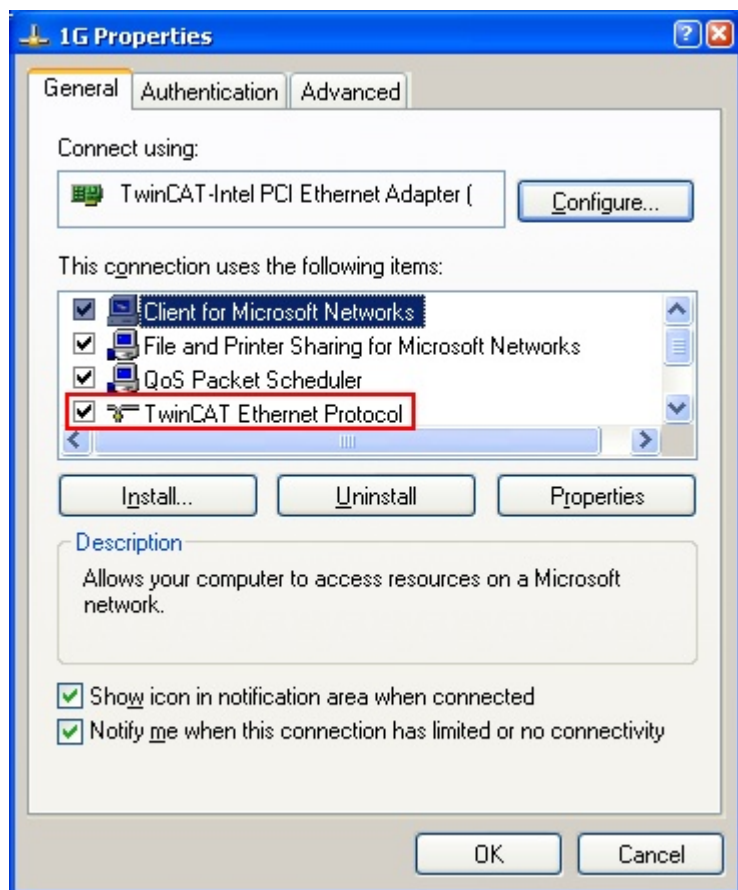


附图 68: EtherCAT 设备属性（TwinCAT 2）：点击“Adapter”选项卡的“Compatible Devices...”

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：

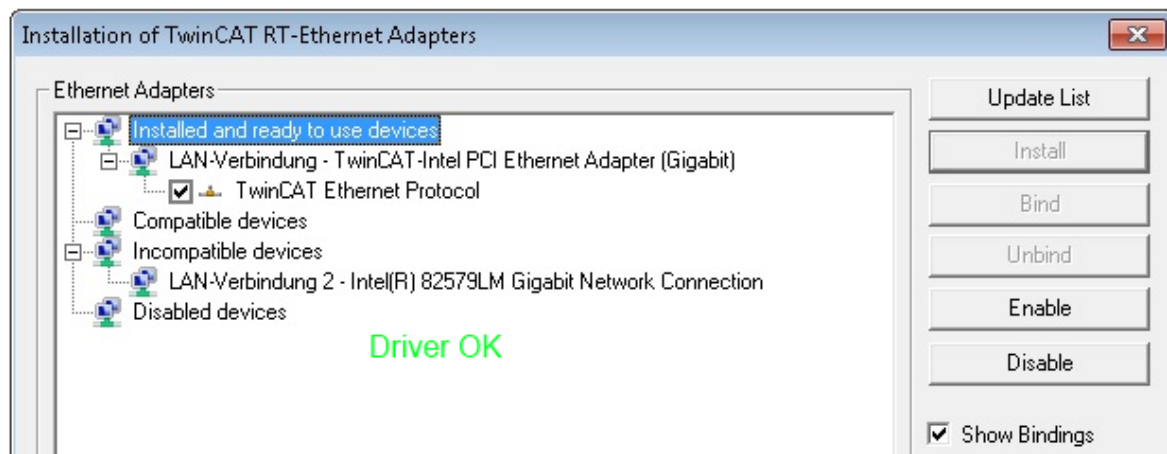


安装后，Windows 的网络接口概览中显示驱动程序已激活（Windows 开始→系统属性→网络）



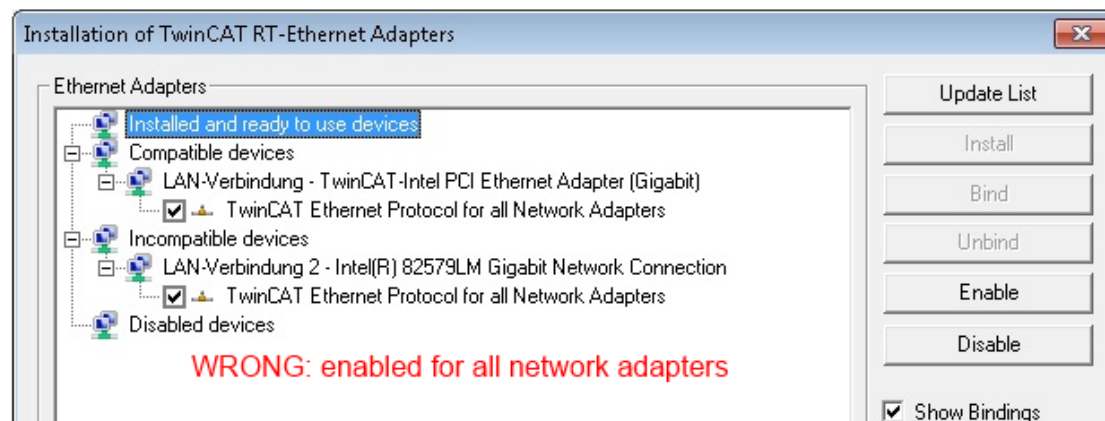
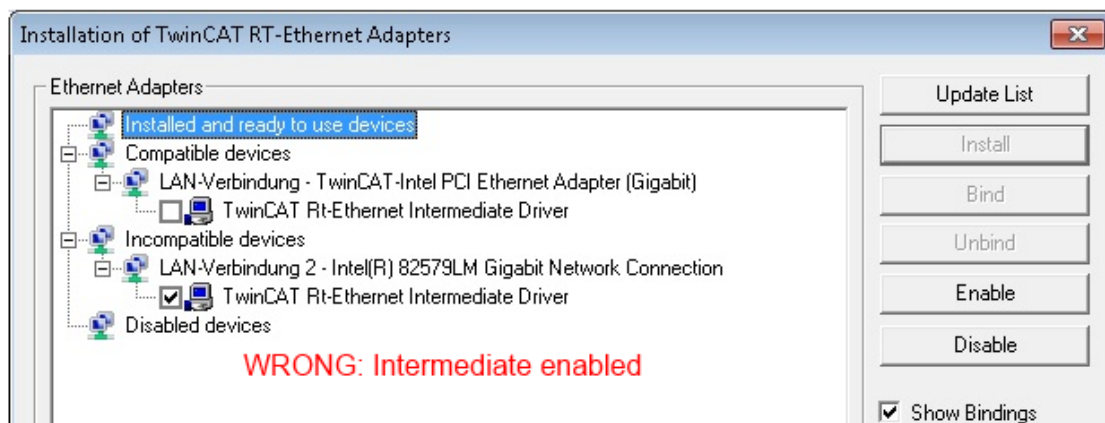
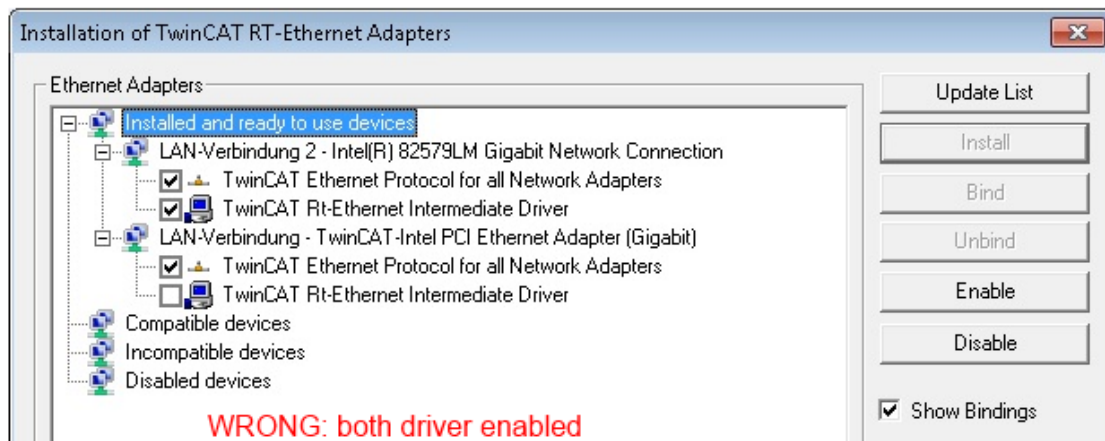
附图 69: Windows 的网络接口属性

驱动程序的正确设置如下：



附图 70: 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：

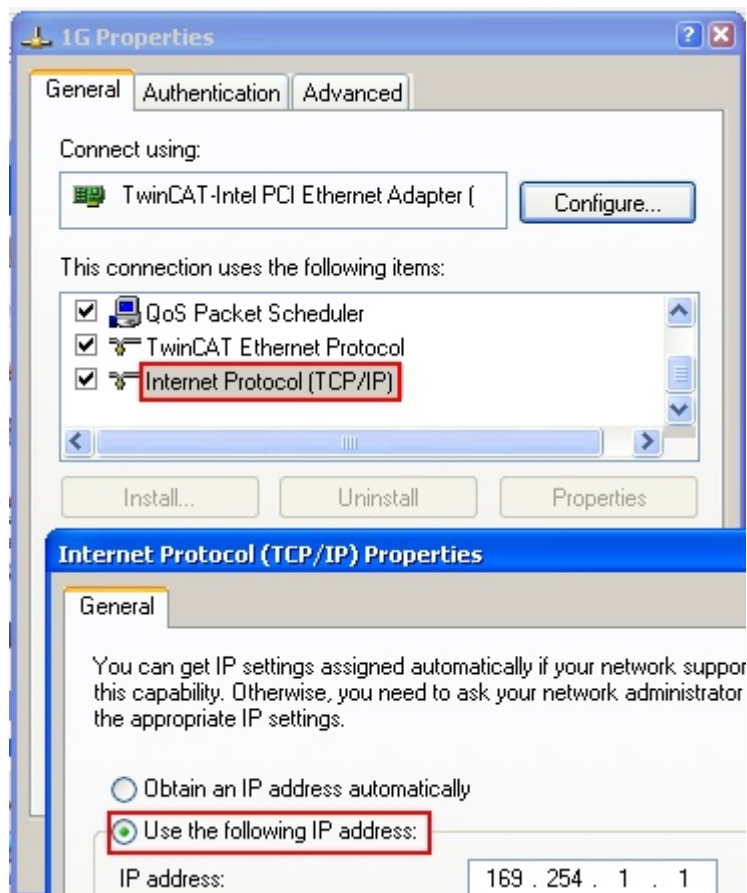


附图 71: 以太网端口驱动程序的错误设置

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 72: 以太网端口的 TCP/IP 设置

5.2.2 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 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”

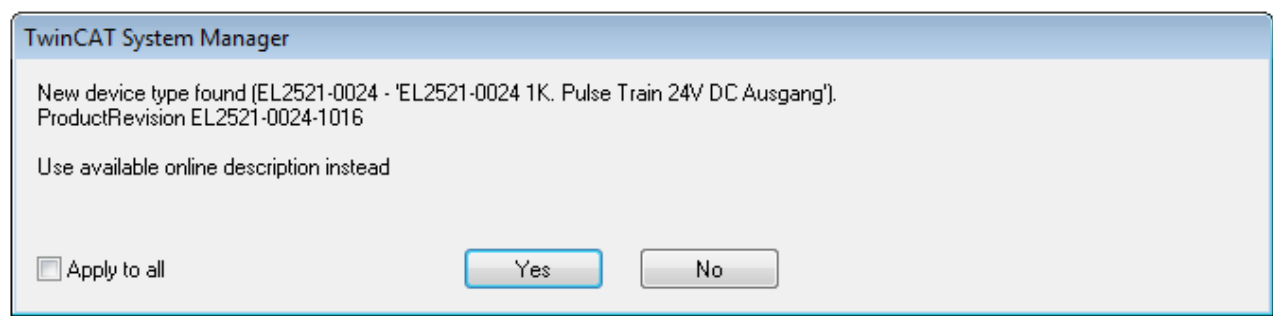
Name
(EL2521-0025-1018)
Revision

附图 73: 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。

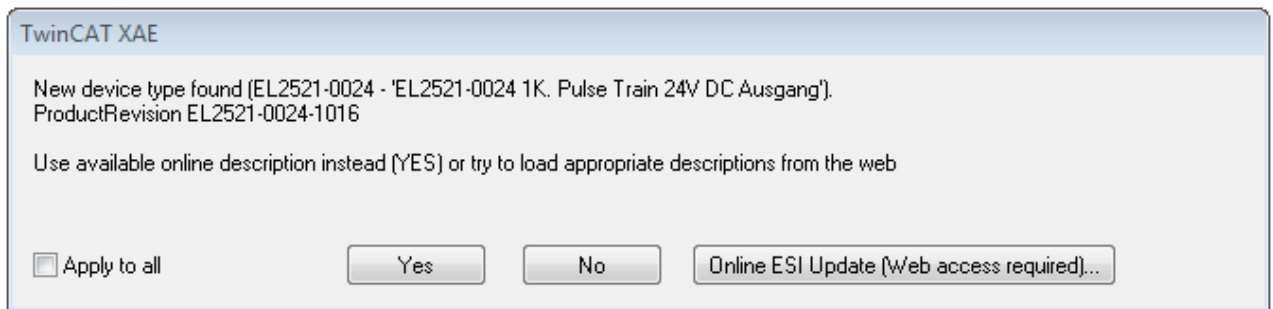
在线描述

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 74: 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 75: 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 \[► 84\]](#)”一章。

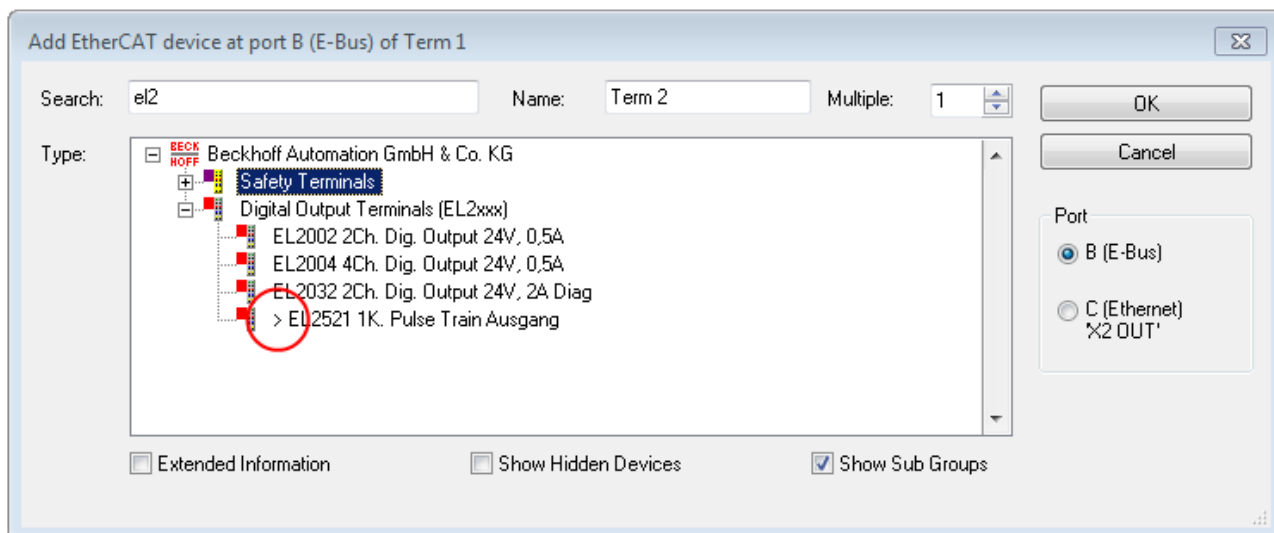
如果使用在线描述，System Manager 会从 EtherCAT 从站的 EEPROM 中读取一份设备描述。在复杂的从站中，EEPROM 的大小可能不足以容纳完整的 ESI，此时配置中的 ESI 就会不完整。因此，建议这种情况下优先使用离线 ESI 文件。

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 76: 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 77: 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

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：

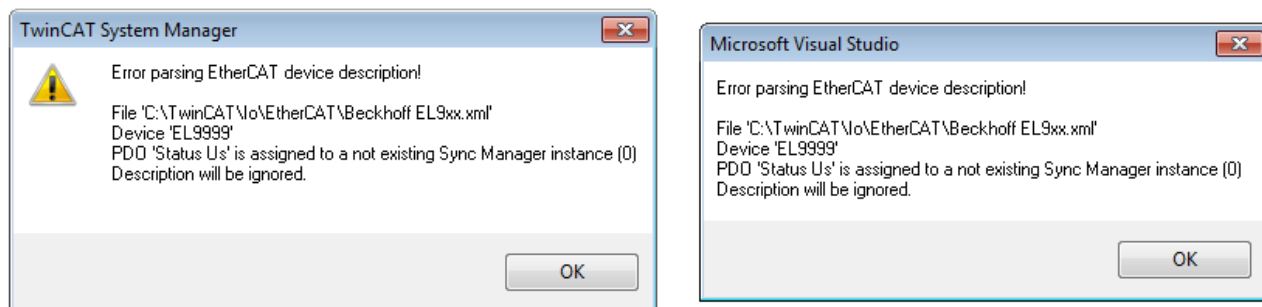
`C:\User\[USERNAME]\AppData\Roaming\Beckhoff\TwinCAT3\Components\Base\EtherCATCache.xml`

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

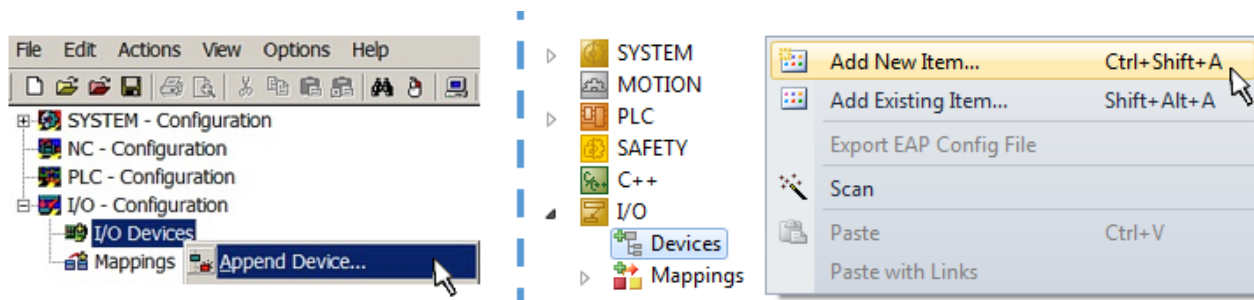
可能的原因包括：

- *.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

5.2.3 创建 OFFLINE 配置

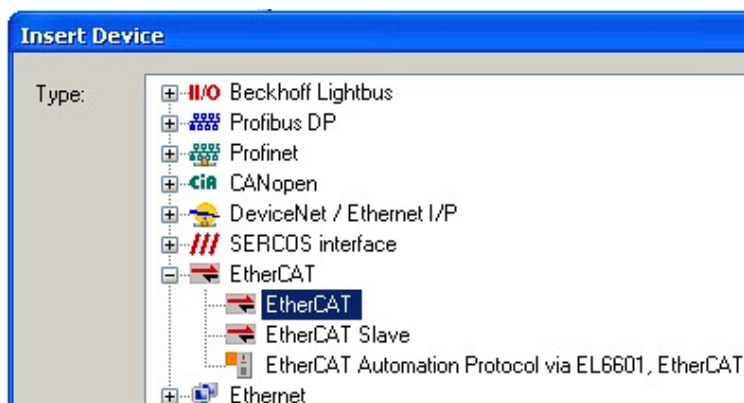
创建 EtherCAT 设备

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



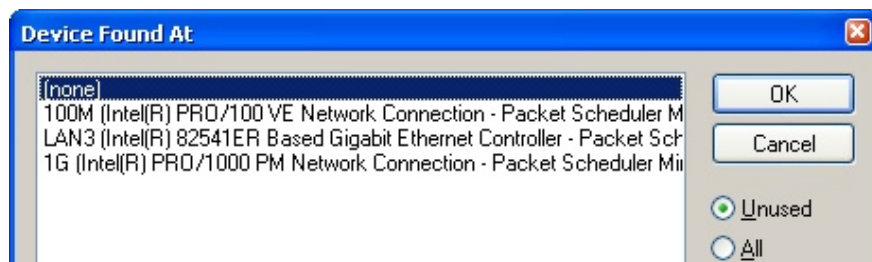
附图 79: 添加 EtherCAT 设备（左：TwinCAT 2；右：TwinCAT 3）

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”。



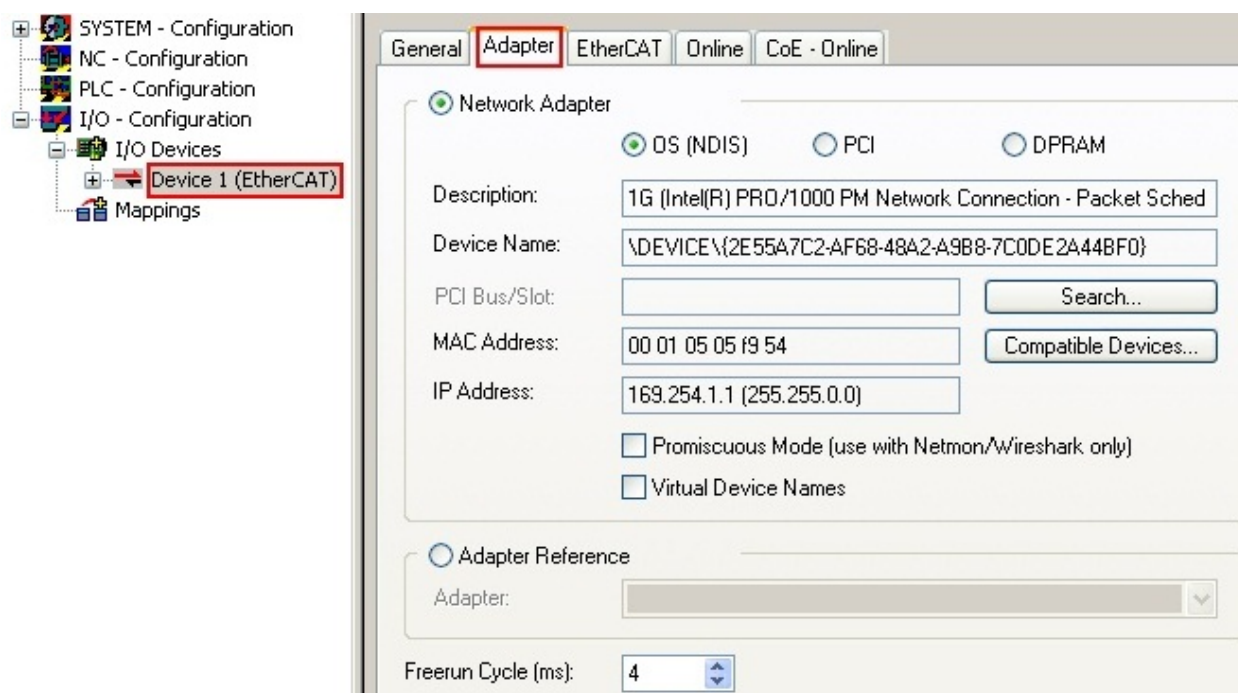
附图 80: 选择 EtherCAT 连接（TwinCAT 2.11, TwinCAT 3）

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。



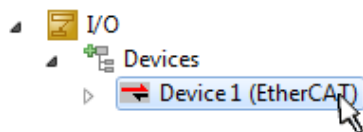
附图 81: 选择以太网端口

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 82: 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:



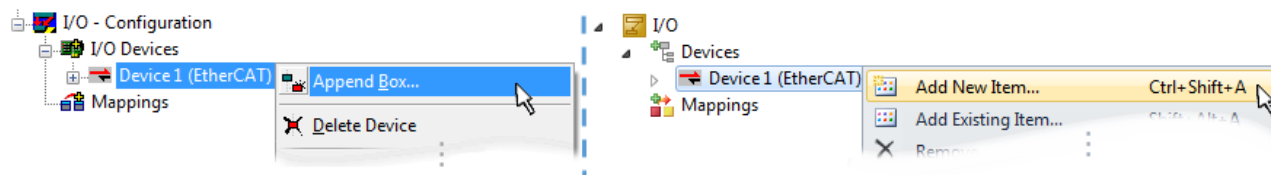
i

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 [▶ 75]。

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

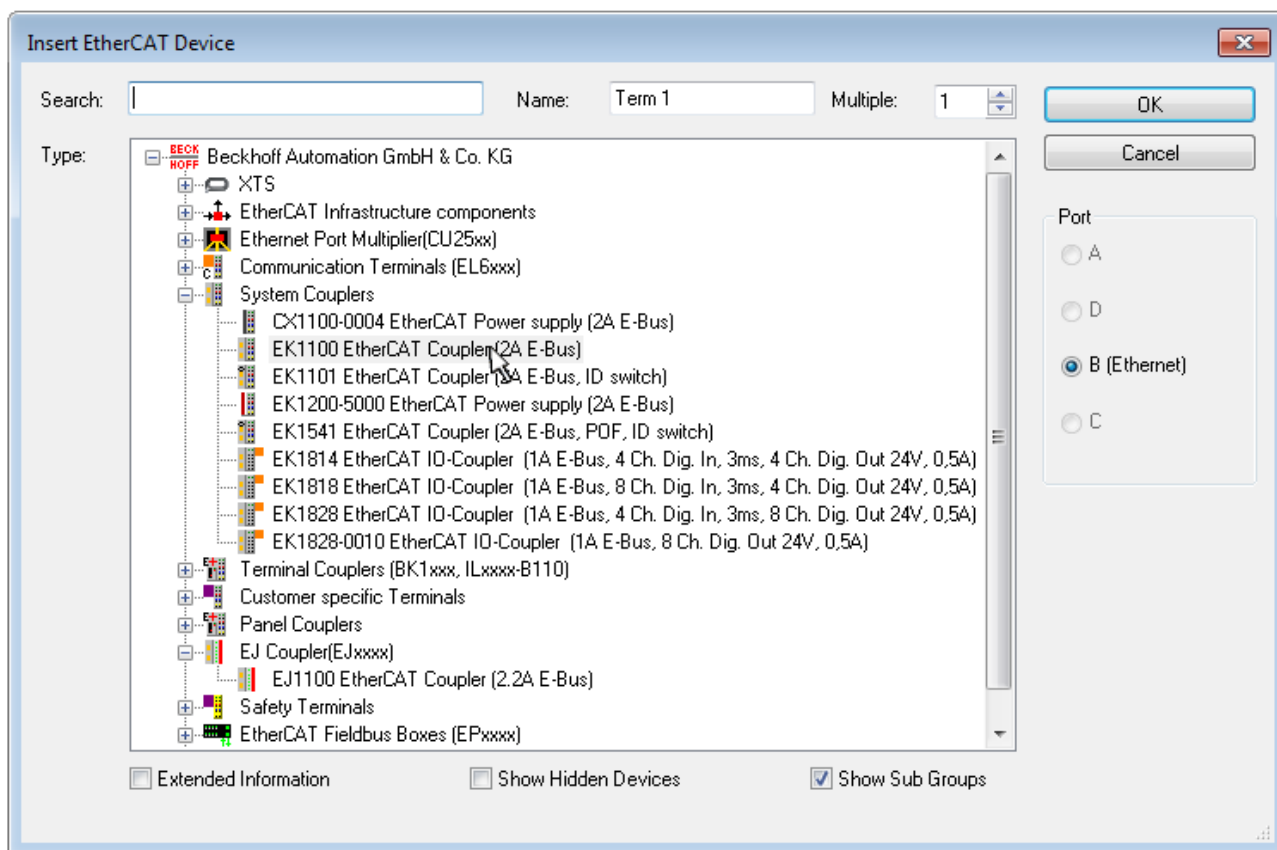
弹出选择新设备的对话框 对话框中只显示已有 ESI 文件的设备。

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

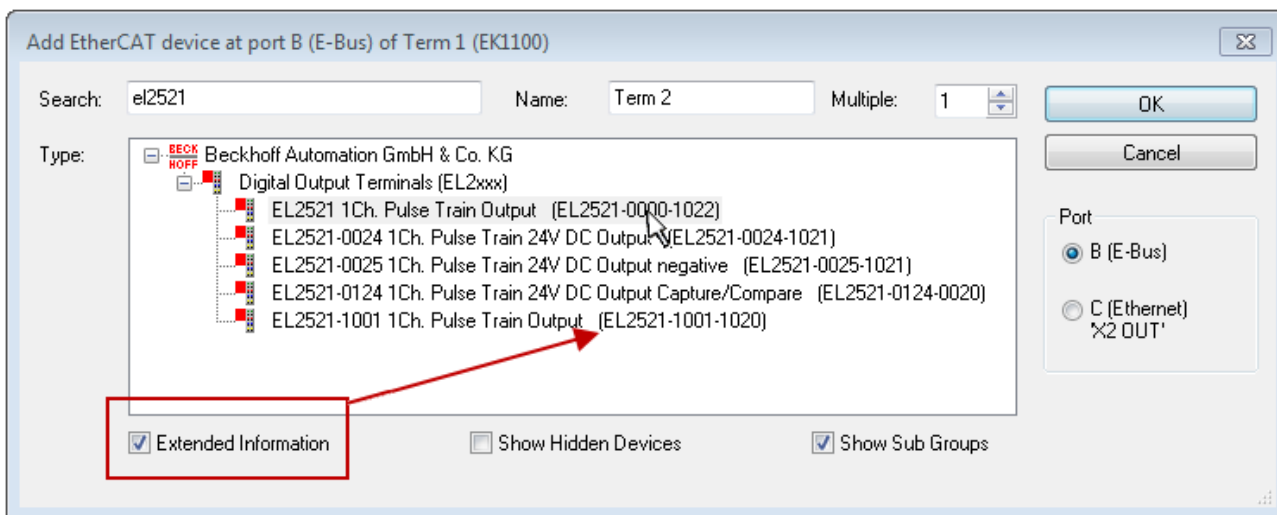
- “Ethernet”：基于电缆的 100BASE-TX：耦合器、盒模块、带 RJ45/M8/M12 连接器的设备
- “E-Bus”：LVDS “端子模块总线”，EtherCAT 插拔式模块（EJ），EtherCAT 端子模块（EL/ES），各种模块化模块

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



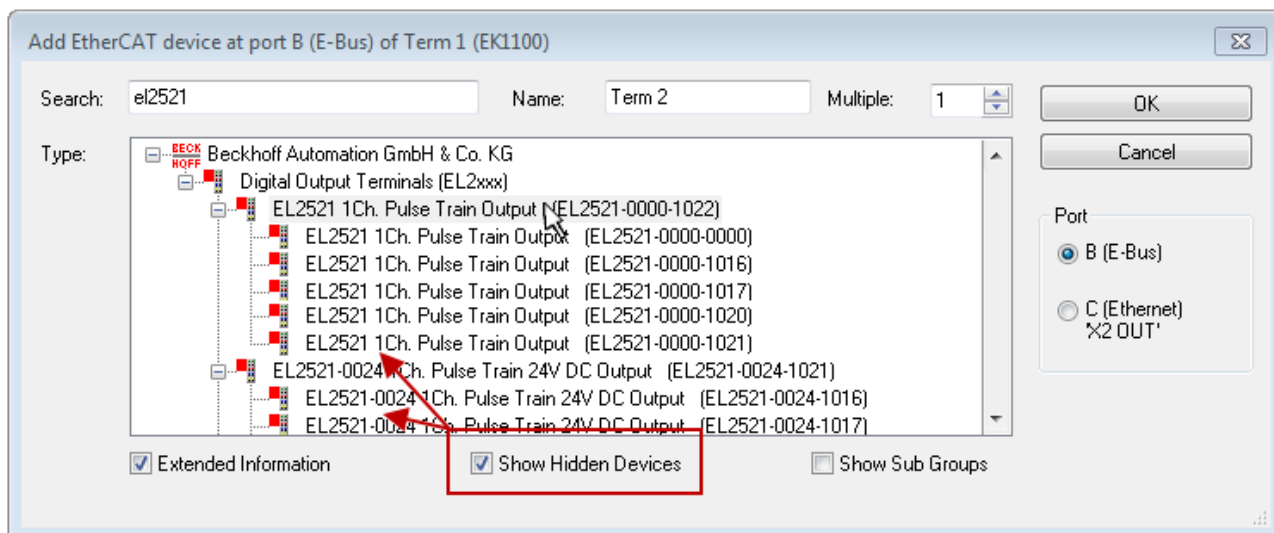
附图 84: 新增 EtherCAT 设备的选择对话框

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 85: 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 86: 显示以前的版本

修订版本的设备选择 - 兼容性

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

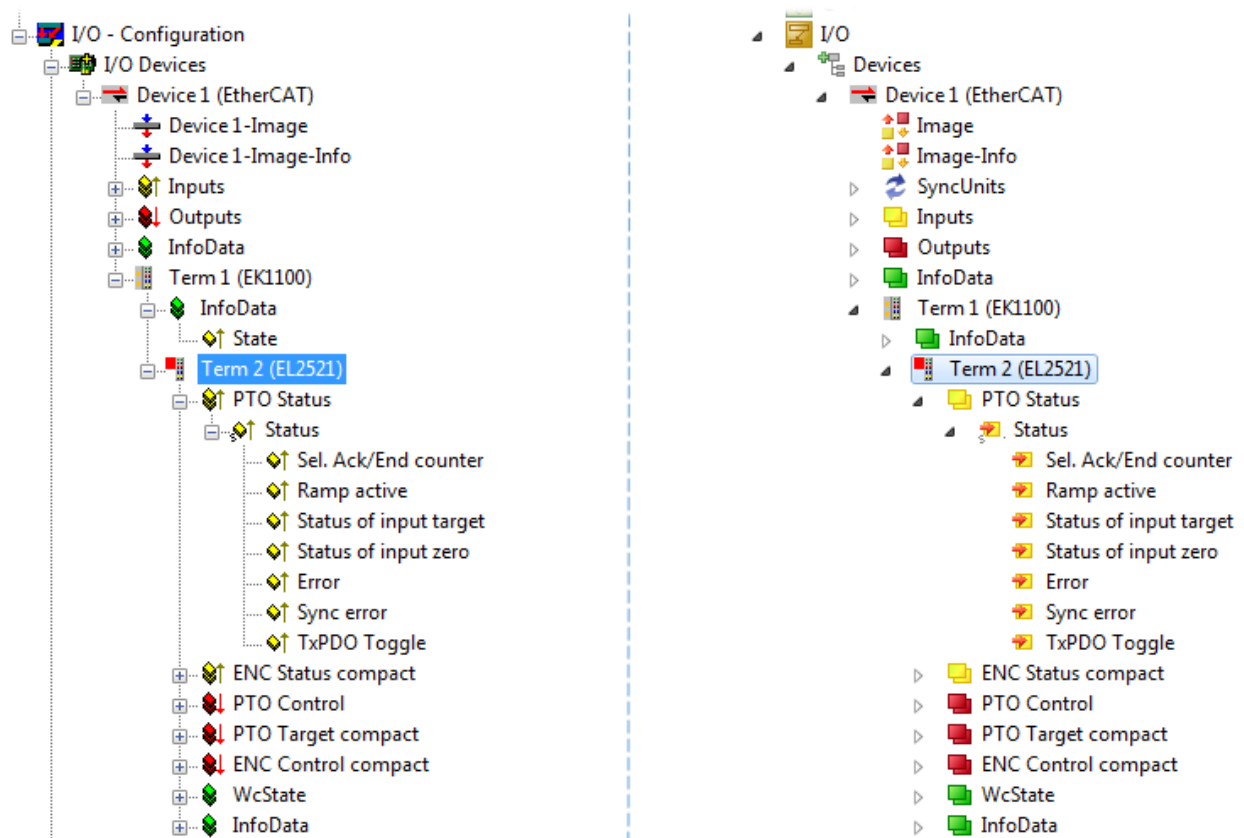
如果在配置中指定了EL2521-0025-**1018**，则在实践中可以使用EL2521-0025-**1018**或更高版本（-1019, -1020）。

Name
(EL2521-0025-1018)
Revision

附图 87: 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 88: TwinCAT 树中的 EtherCAT 端子模块 (左: TwinCAT 2; 右: TwinCAT 3)

5.2.4 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。

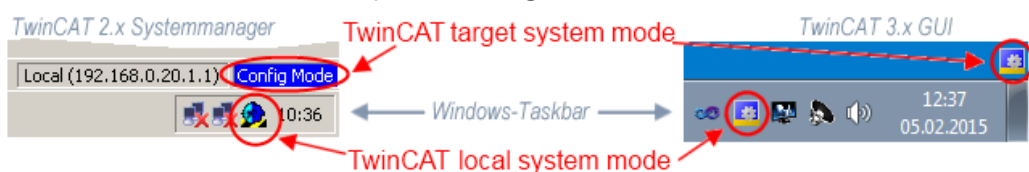
以下方法可以将TwinCAT 设置成配置模式：

- 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

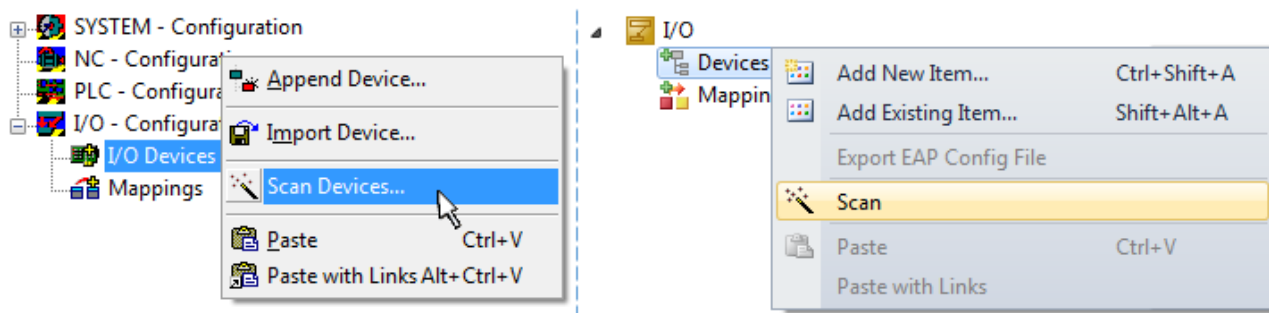
i 在 RUN 模式（生产运行）下，在线搜索不可用。注意 TwinCAT 编程系统和 TwinCAT 目标系统之间的区别。

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。



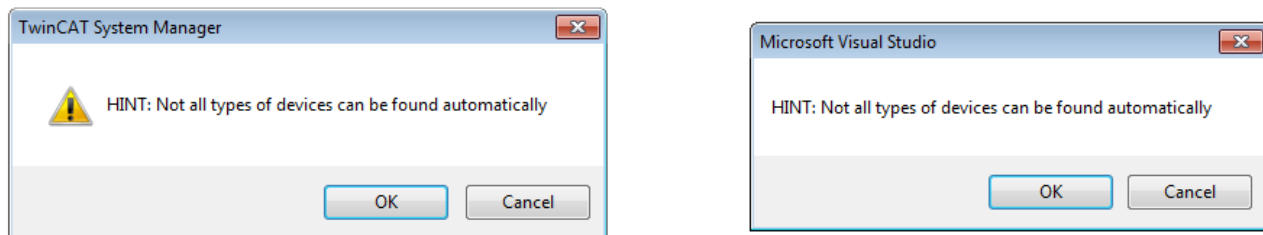
附图 89: 本地/目标系统差异（左：TwinCAT 2；右：TwinCAT 3）

右键单击配置树中的“I/O Devices”可以打开搜索对话框。



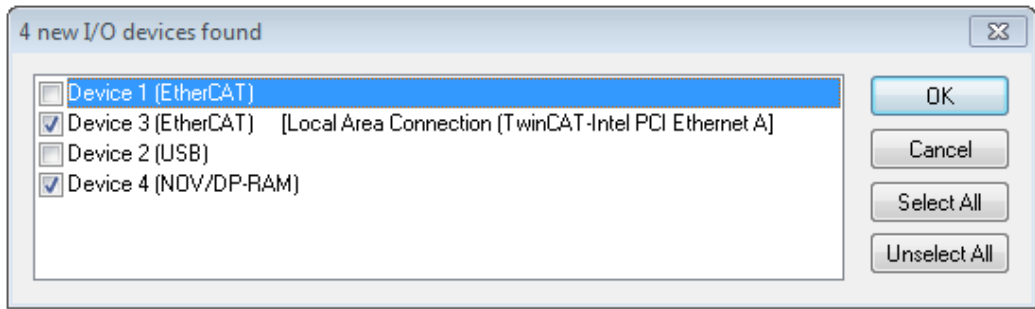
附图 90: Scan Devices（扫描设备）（左：TwinCAT 2；右：TwinCAT 3）

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 NOVRAM、现场总线卡、SMB 等。然而，并非所有设备都能自动找到。



附图 91: 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 92: 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”。

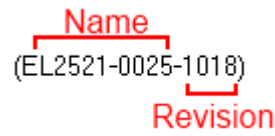
● **选择以太网端口**

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 [► 75]。

检测/扫描 EtherCAT 设备

● **在线扫描功能**

i 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。



附图 93: 默认设置示例

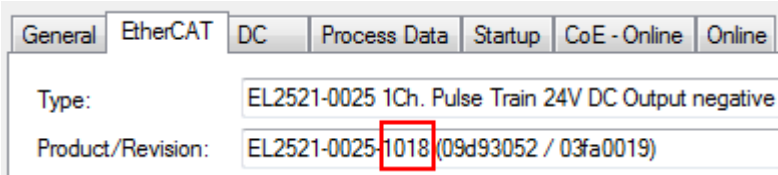
注意

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 [► 93]。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 IO 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：



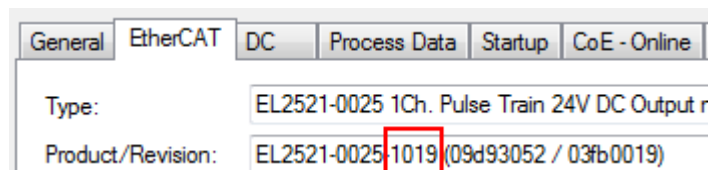
附图 94: 安装修订版本-1018的EtherCAT 端子模块，

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[► 93]。

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 95: 检测修订版本 1019 的 EtherCAT 端子模块

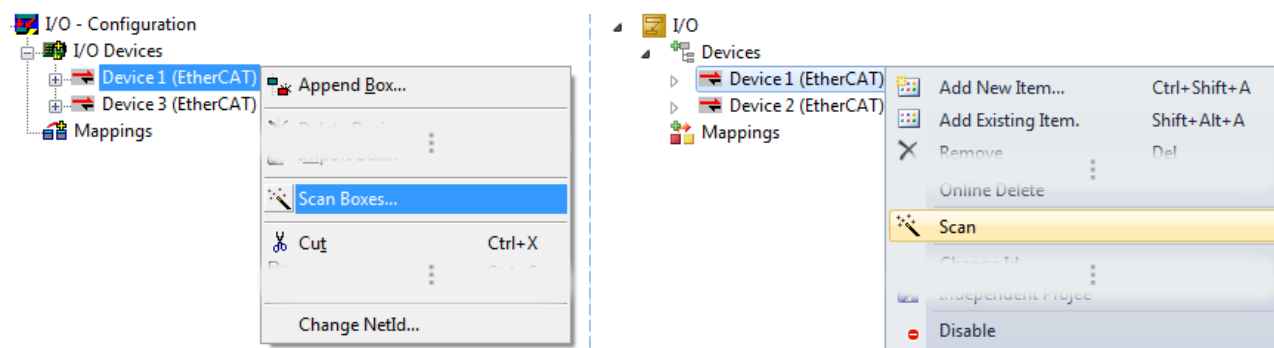
调试工程师通常不会注意到这一点。TwinCAT 也不会发出任何信号，因为实际上是从头创建了一个新的配置。然而，根据兼容性规则，这意味着不应该将 EL2521-0025-**1018** 的备件安装到这台机器上（即使这在绝大多数情况下还是可以使用的）。

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



附图 97: 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。



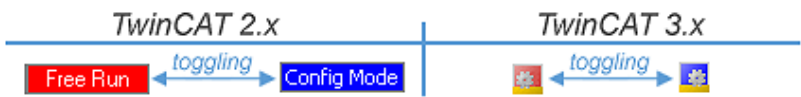
附图 98: TwinCAT 2 的扫描进度示例

配置已建立，然后可以切换到在线状态（OPERATIONAL）。



附图 99: 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。

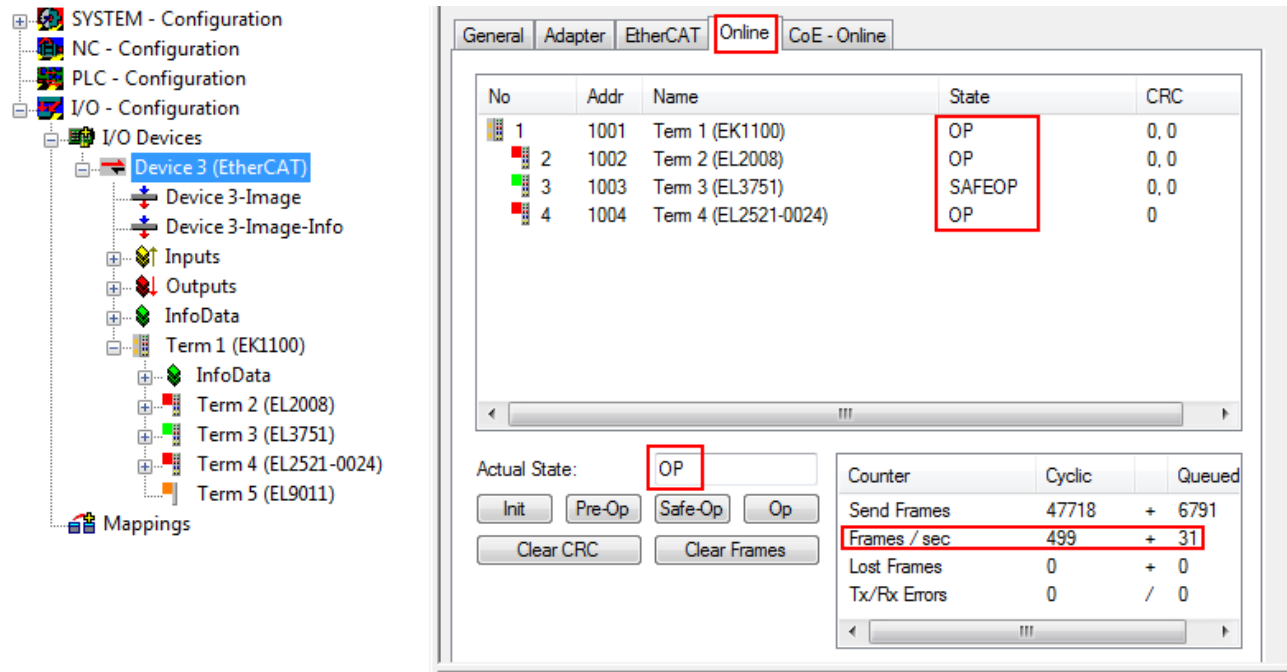


附图 100: 显示在状态栏下方的“Free Run”和“Config Mode”来回切换



附图 101: 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。



附图 102: 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

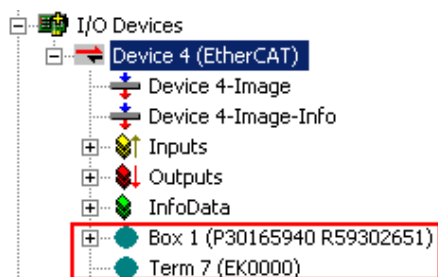
至此，配置工作就完成了。该配置可以按照[手动流程 \[► 84\]](#)中的描述进行修改。

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 103: 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。

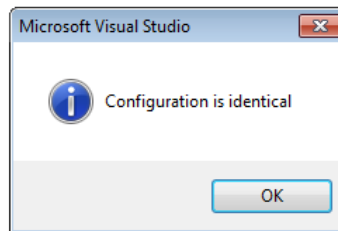
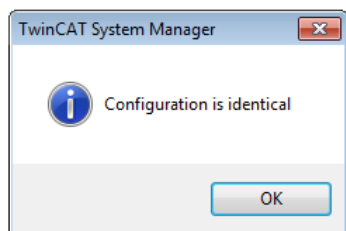
扫描现有配置

注意

比较后修改配置

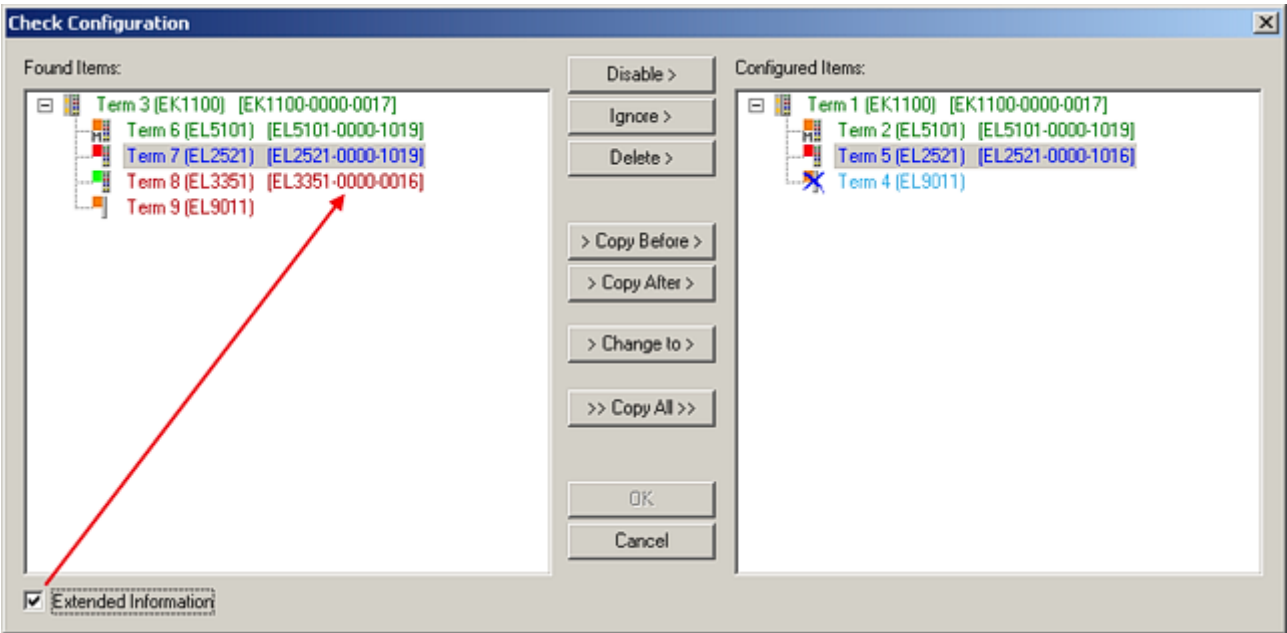
通过这种扫描（TwinCAT 2.11 或 3.1），目前只对设备属性中的供应商（制造商）、设备名称和修订版本进行比较！务必谨慎执行“ChangeTo（更改为）”和“Copy（复制）”操作，认真考虑倍福 IO 兼容性规则（见前文）。然后，原来配置的设备版本被扫描发现的修订版本所取代；这可能会影响设备支持的过程数据和功能。

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 104: 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 105: 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	- 此 EtherCAT 从站在另一侧不存在。 - 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。

● 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

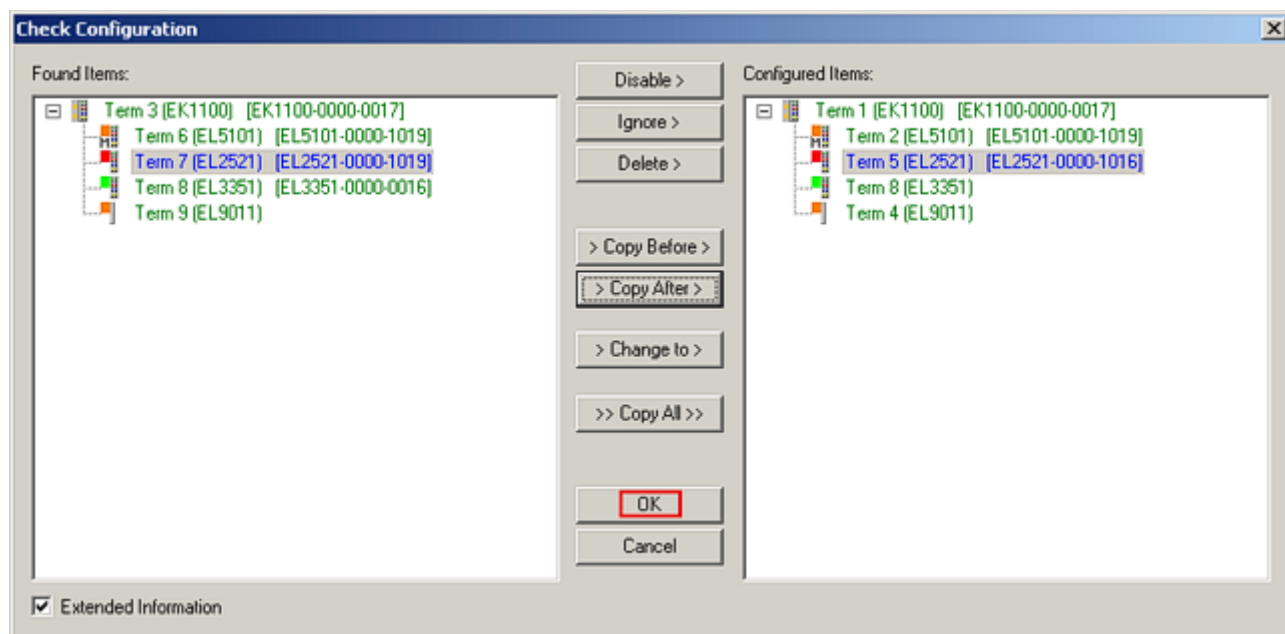
如果在配置中指定了EL2521-0025-**1018**，则在实践中可以使用EL2521-0025-**1018**或更高版本（-**1019**，-**1020**）。

Name
(EL2521-0025-1018)
Revision

附图 106: 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...

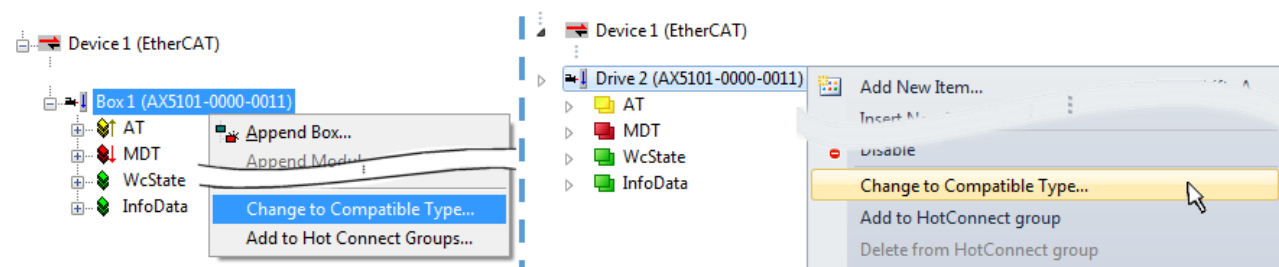


附图 107: 更正对话框，有修改项

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 108: 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”：

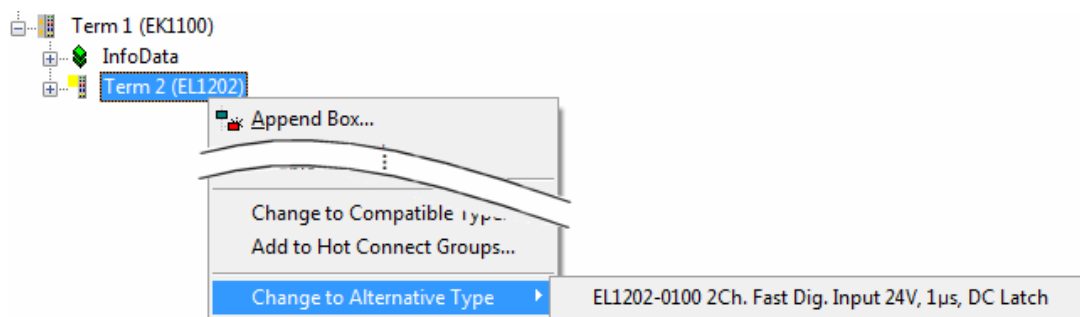
- 物理层（例如RJ45、Ebus...）
- FMMU（允许实际数量比配置的多）
- Sync Manager（SM，允许实际数量比配置的多）

- EoE (属性 MAC, IP)
- CoE (属性 SdoInfo, PdoAssign, PdoConfig, PdoUpload, CompleteAccess)
- FoE
- PDO (过程数据: Sequence, SyncUnit SU, Sync Manager 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

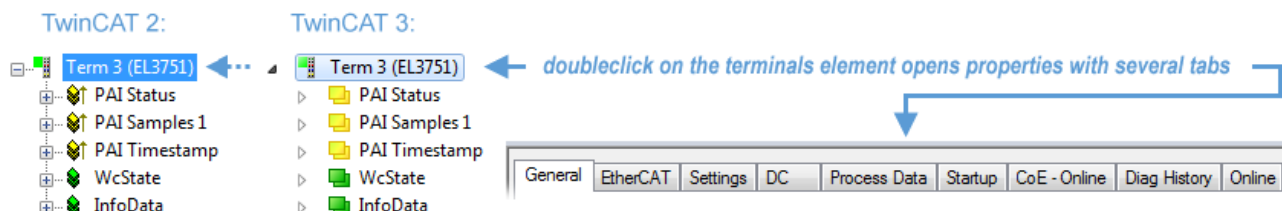


附图 109: TwinCAT 2 对话框 Change to Alternative Type

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。

5.2.5 EtherCAT 设备的配置

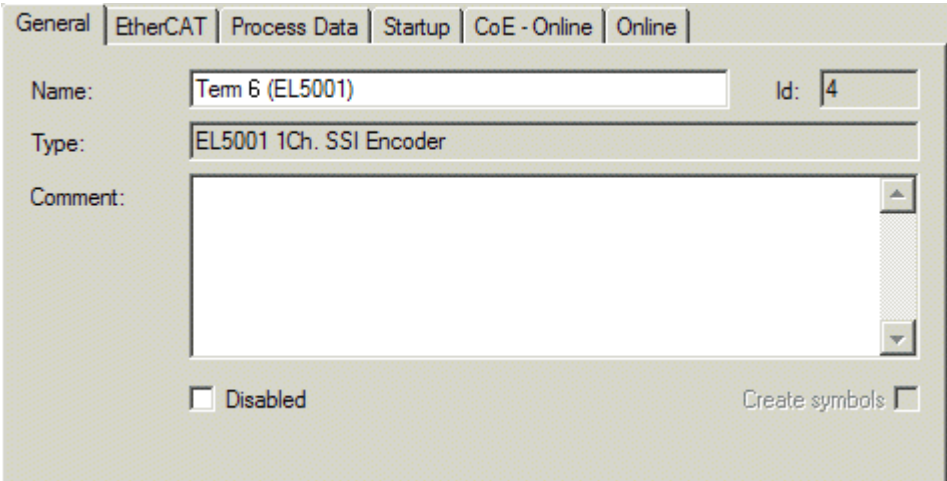
在 TwinCAT 2 System Manager 的左侧窗口或 TwinCAT 3 开发环境的 Solution Explorer (解决方案浏览器) 中, 分别点击树结构中希望配置的端子模块 (在示例中: EL3751 Term 3)。



附图 110: 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)。

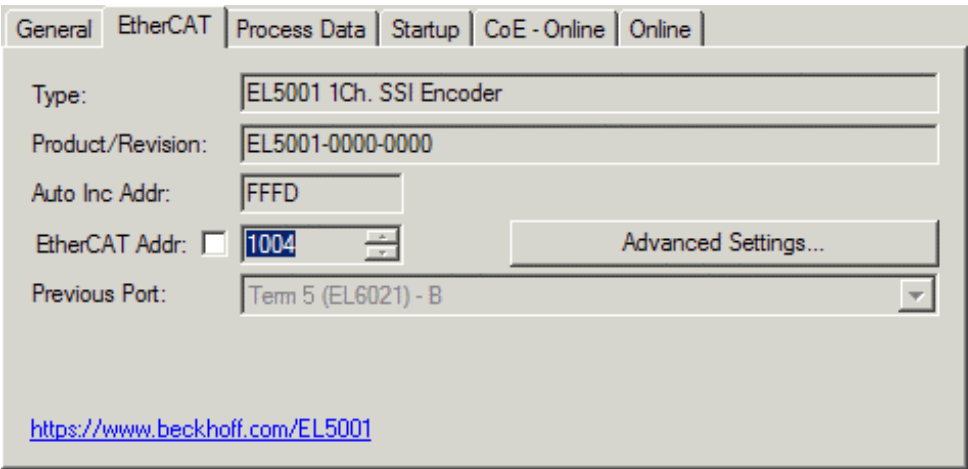
“General(常规)”选项卡



附图 111: “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 112: “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 _{hex} 。每增加一个从站，地址就减 1 (FFFF _{hex} 、FFFE _{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。

SM	Size	Type	Flags
0	246	MbxOut	
1	246	MbxIn	
2	0	Outputs	
3	5	Inputs	

Index	Size	Name	Flags	SM	SU
0x1A00	5.0	Channel 1	F	3	0

Index	Size	Offs	Name	Type	Default (hex)
0x3101:01	1.0	0.0	Status	BYTE	
0x3101:02	4.0	1.0	Value	UDINT	
		5.0			

附图 113: “Process Data (过程数据)” 选项卡

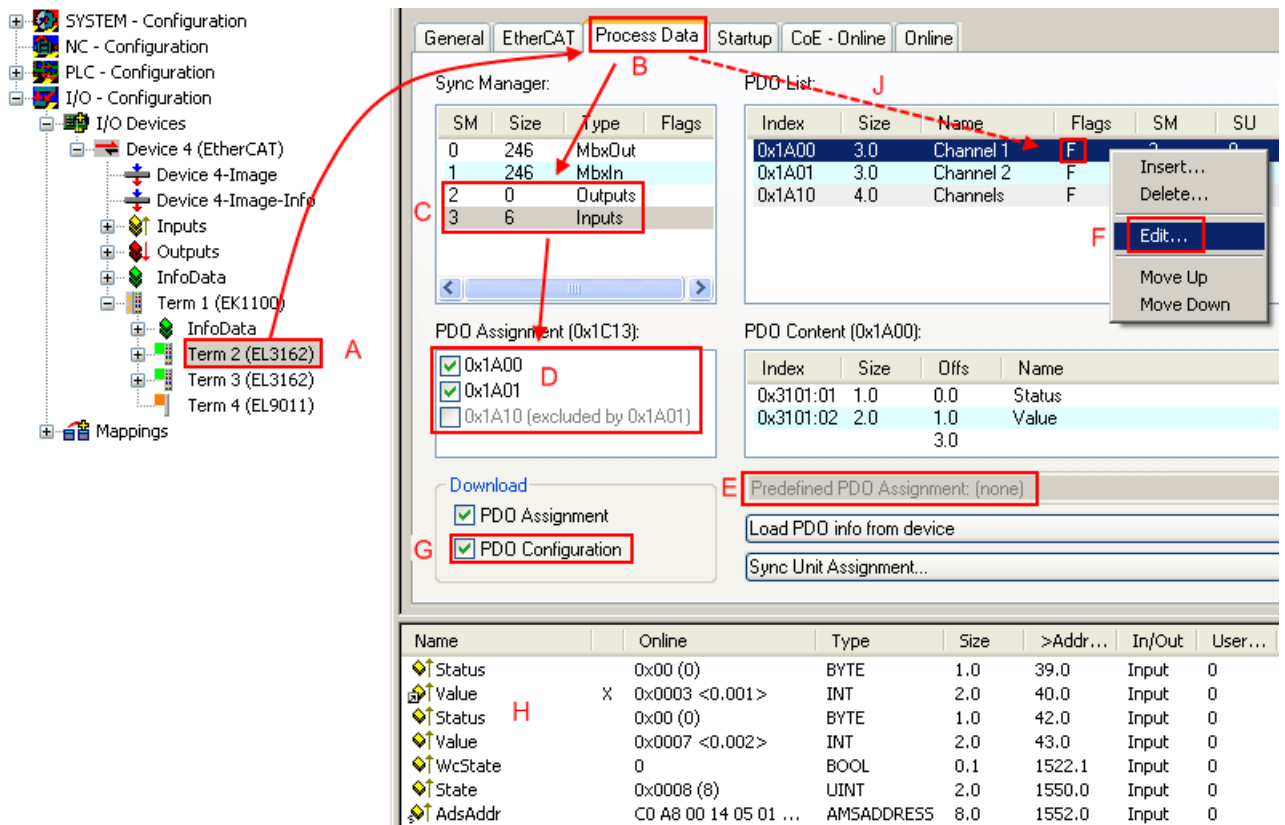
EtherCAT从站在每个周期内传输的过程数据（PDO）是应用程序期望周期性更新的用户数据，或者是被发送到从站的用户数据。为此，EtherCAT 主站 (Beckhoff TwinCAT) 在启动阶段对每个EtherCAT从站进行了参数设置，以定义其希望传输到该从站或从该从站传输的过程数据 (位/字节大小、数据源位置、传输类型)。如果配置错误，将会使从站启动失败。

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 114: 配置过程数据

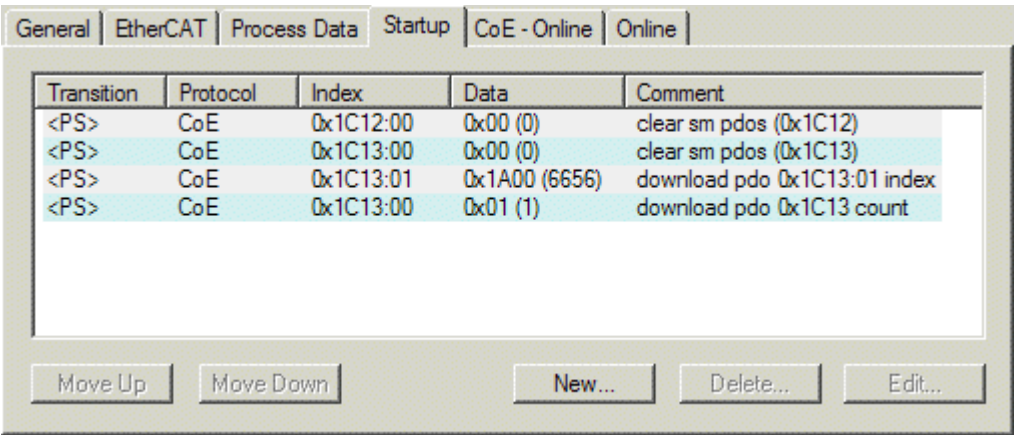
手动修改过程数据

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 [► 104]。

“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相同顺序发送到从站。



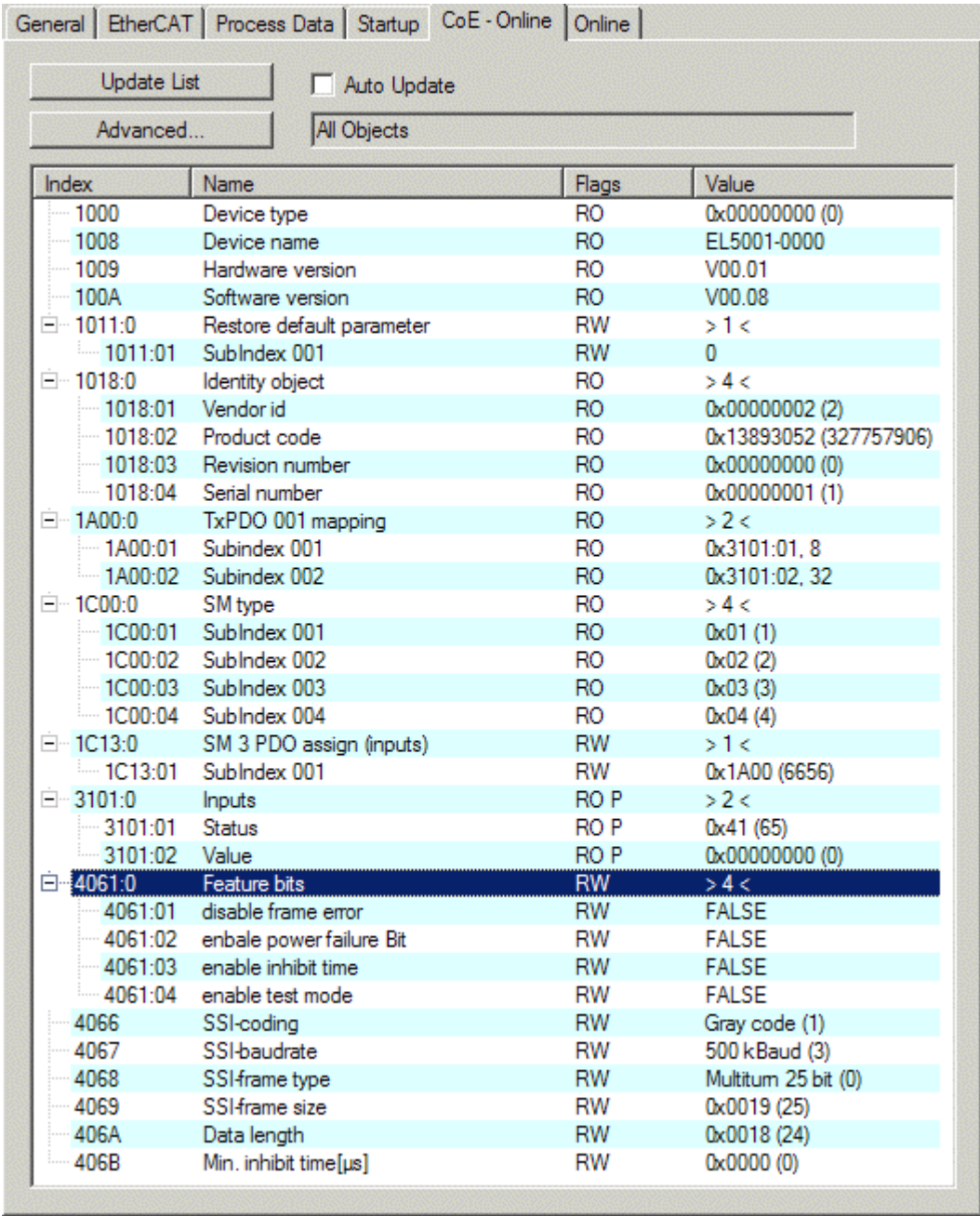
附图 115: “Startup (启动)” 选项卡

列	Description
Transition	发送请求的过渡期。这可以是 • 从Pre-OP到Safe-OP（PS）的过渡，或 • 从Safe-OP到运行（SO）的过渡。 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

- Move Up**
该按钮可将所选请求在列表中向上移动一个位置。
- Move Down**
该按钮可将所选请求在列表中向下移动一个位置。
- New**
该按钮可添加一个新的邮箱下载请求，将在启动时发送。
- Delete**
该按钮可以删除选定的条目。
- Edit**
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 116: “CoE - Online” 选项卡

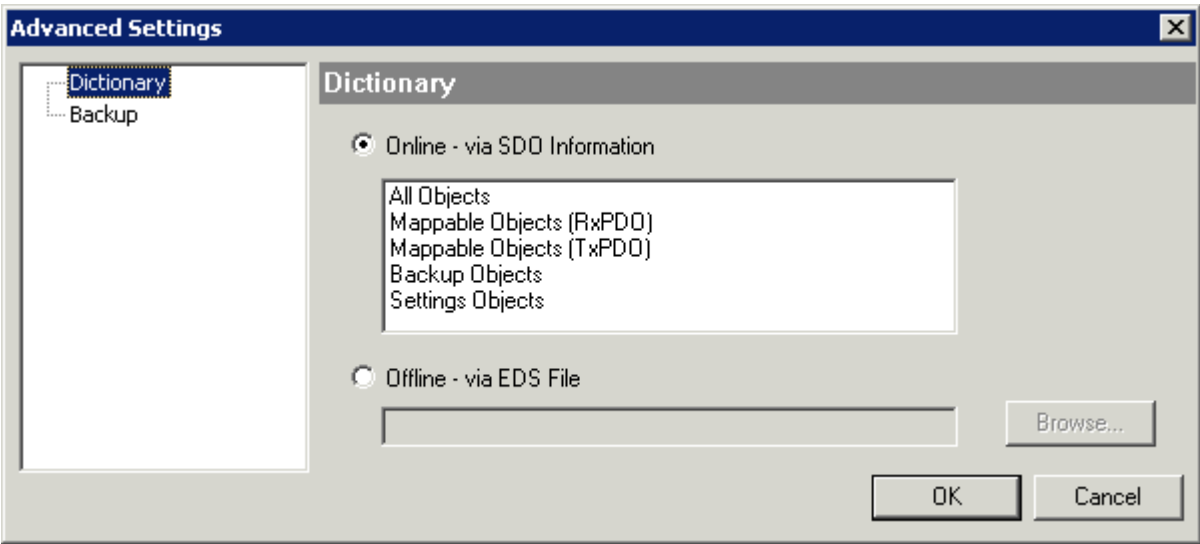
对象列表显示

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。

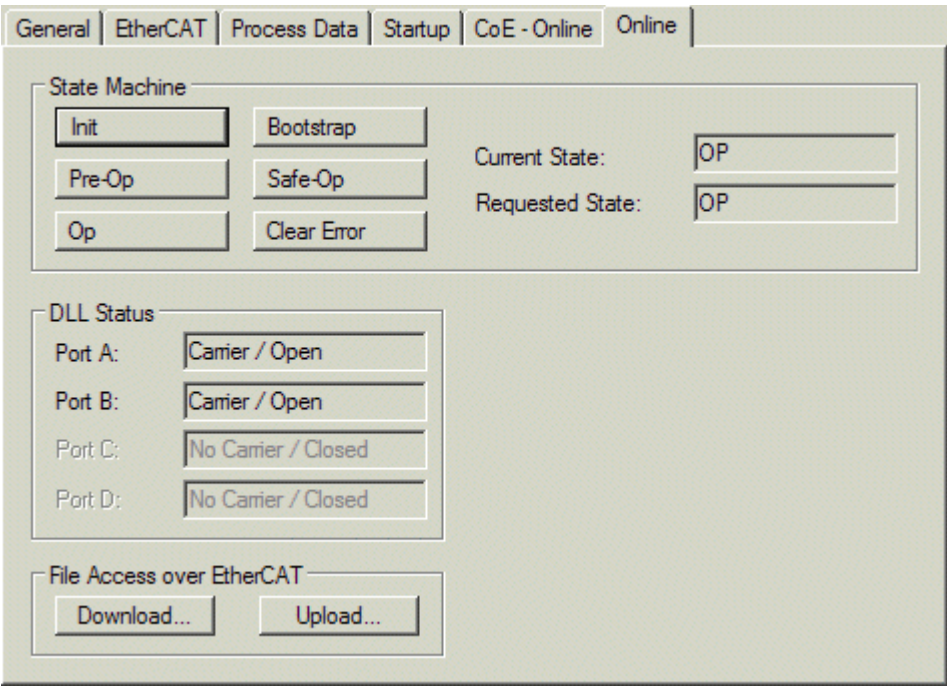
Advanced Advanced 按钮可打开 Advanced Settings 对话框。在这里，你可以指定哪些对象会显示在列表中。



附图 117: “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 118: “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：

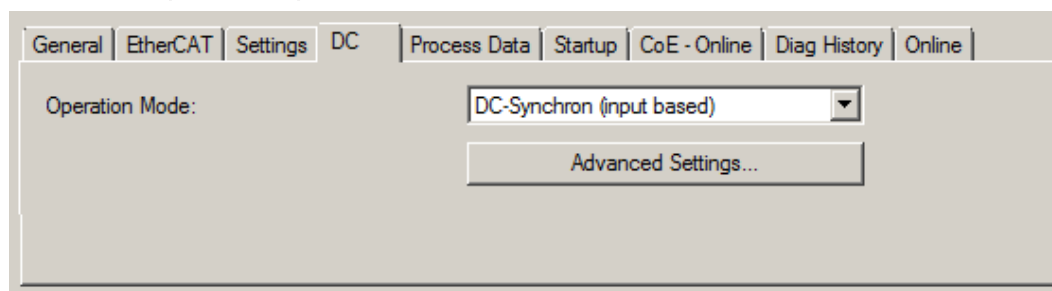
Status	Description
No Carrier / Open	端口没有通讯信号，但端口处于打开状态。
No Carrier / Closed	端口没有通讯信号，且端口处于关闭状态。
Carrier / Open	端口有通讯信号，且端口处于打开状态。
Carrier / Closed	端口有通讯信号，但端口处于关闭状态。

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 119: “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

5.2.5.1 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 [▶ 102])，

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	<table border="1"> <tr> <td>F</td><td>固定内容：该 PDO 内容固定，System Manager (系统管理器) 无法更改。</td></tr> <tr> <td>M</td><td>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 PDO Assignment 列表中删除。</td></tr> </table>	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 PDO Assignment 列表中删除。
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 PDO Assignment 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 \[► 99\]](#) 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。

5.3 EtherCAT 从站的一般调试说明

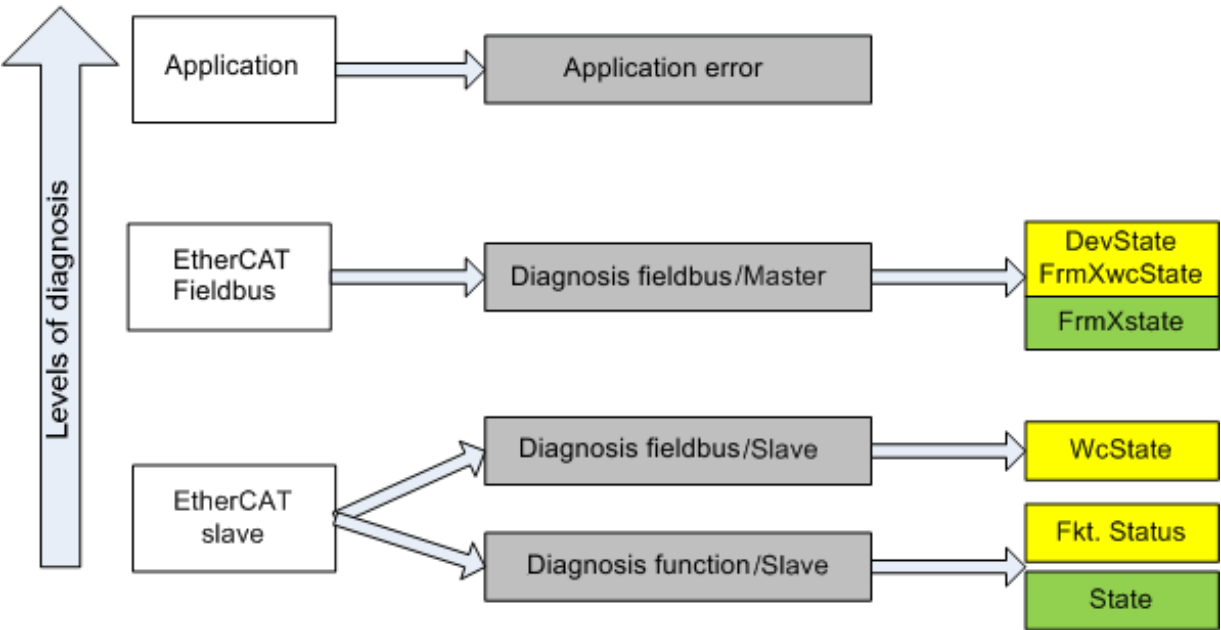
该摘要简单介绍了 TwinCAT 下的 EtherCAT 从站运行的若干方面。关于详细信息，可查看相应章节，例如 [EtherCAT 系统文档](#)。

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 120: 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。

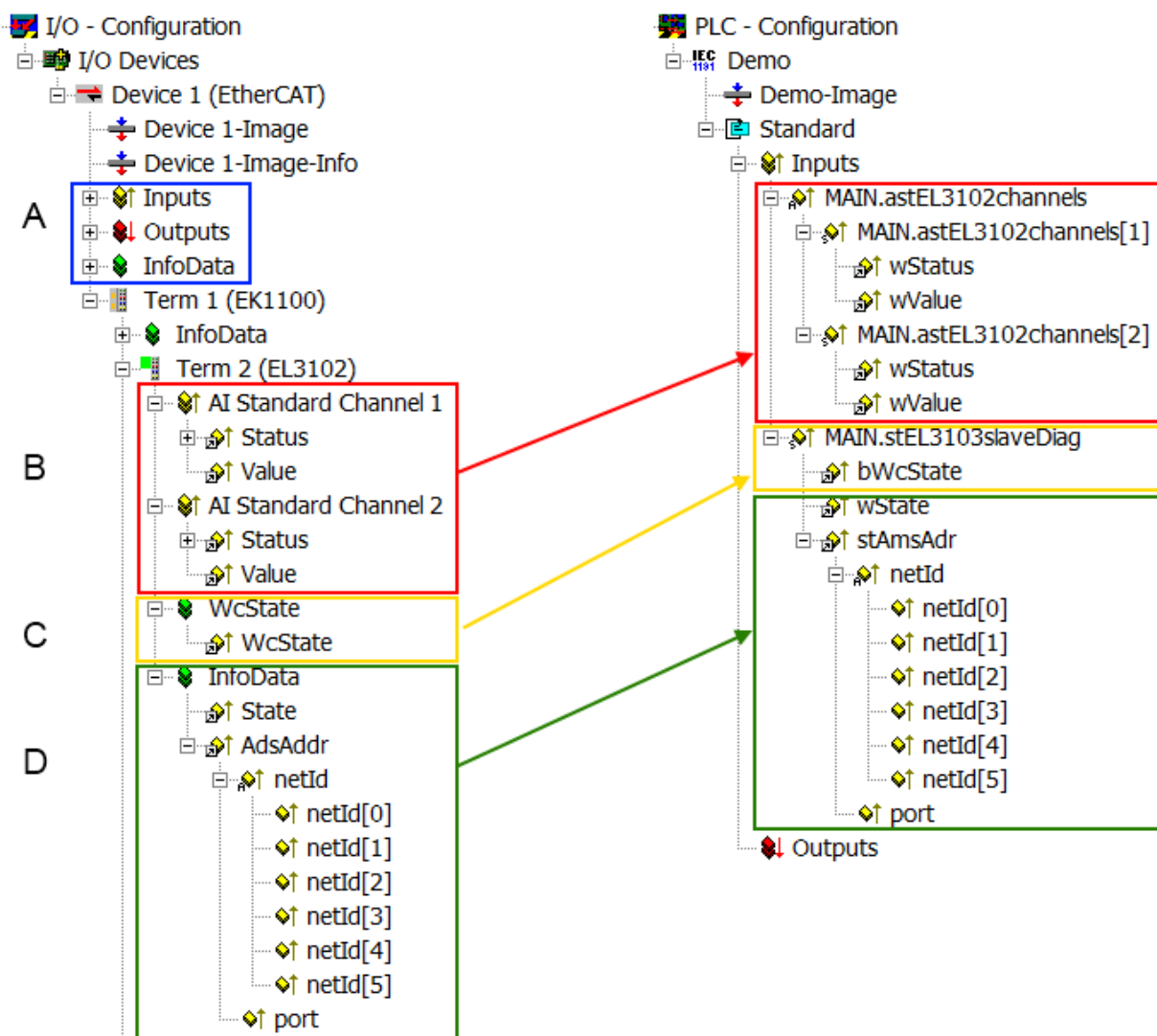
以及

- 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 121: 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： - 主站中的 CoE 用于与/通过从站进行通信 - TcEtherCAT.lib 功能 - 执行在线扫描
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 - 位符号可参见设备手册 - 其他设备可能提供更多的信息，或者没有典型的从站信息	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。	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。

代码	功能	实施	应用/评估
	1. 2.		
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 - 除了系统启动时，很少/从不改变。 - 本身是非周期性确定的（例如 EtherCAT 状态）	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：

Index	Name	Flags	Value
6010:0	AI Inputs Ch.2	RO	> 17 <
6401:0	Channels	RO	> 2 <
8000:0	AI Settings Ch.1	RW	> 24 <
8000:01	Enable user scale	RW	FALSE
8000:02	Presentation	RW	Signed (0)
8000:05	Siemens bits	RW	FALSE
8000:06	Enable filter	RW	FALSE
8000:07	Enable limit 1	RW	FALSE
8000:08	Enable limit 2	RW	FALSE
8000:0A	Enable user calibration	RW	FALSE
8000:0B	Enable vendor calibration	RW	TRUE

附图 122: EL3102，CoE 字典



EtherCAT 系统文档

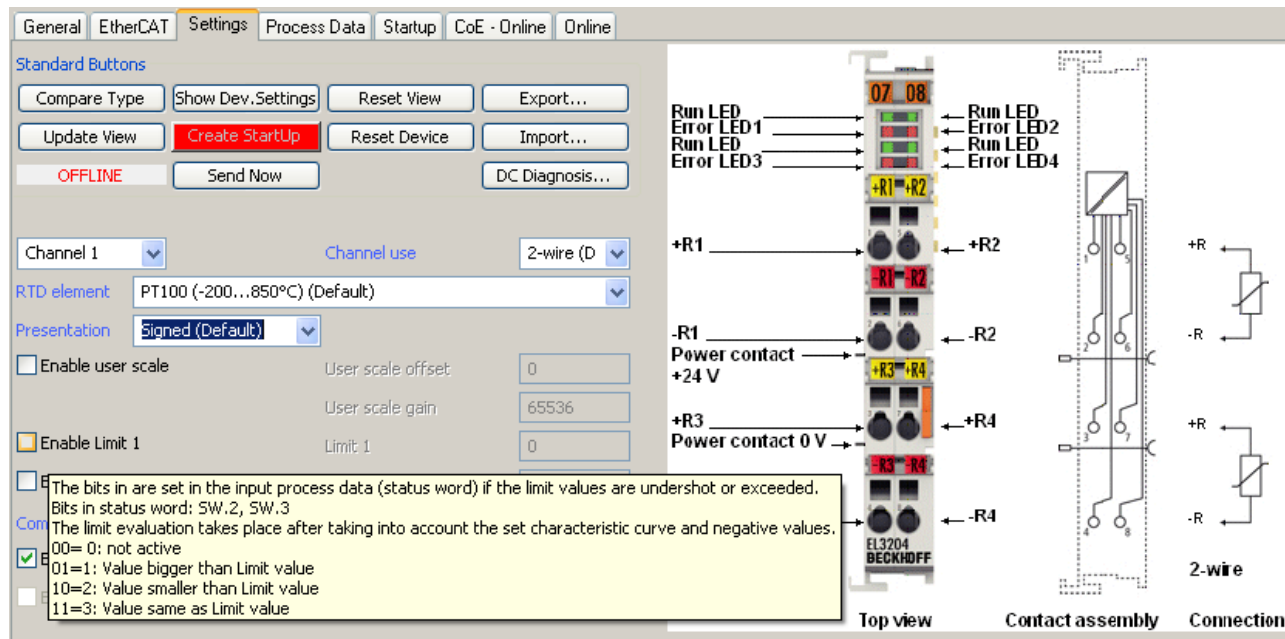
必须遵守 [EtherCAT 系统文档](#)（EtherCAT 基础知识-->CoE 接口）中的全面描述！

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 123: 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 [▶ 26]”。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。

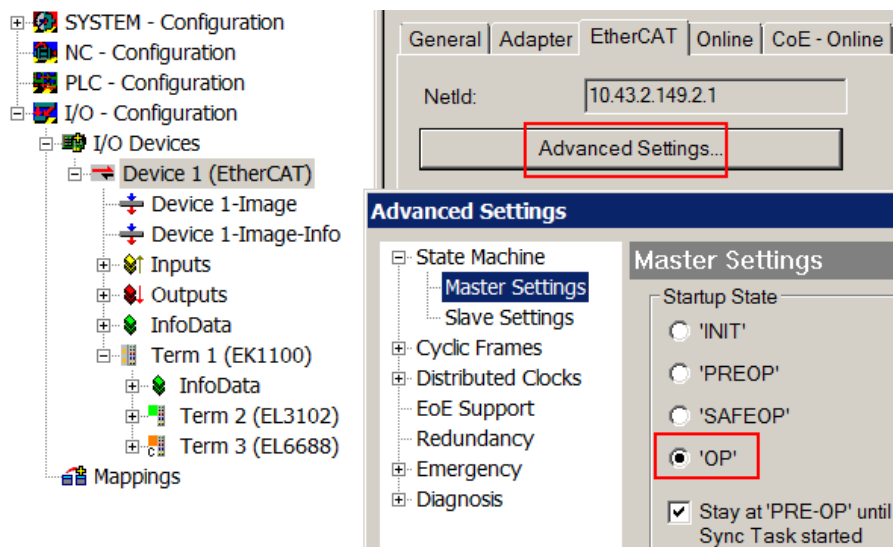
EtherCAT 主站本身在启动时必须经过这些例行的步骤，直到目标状态 OP。

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

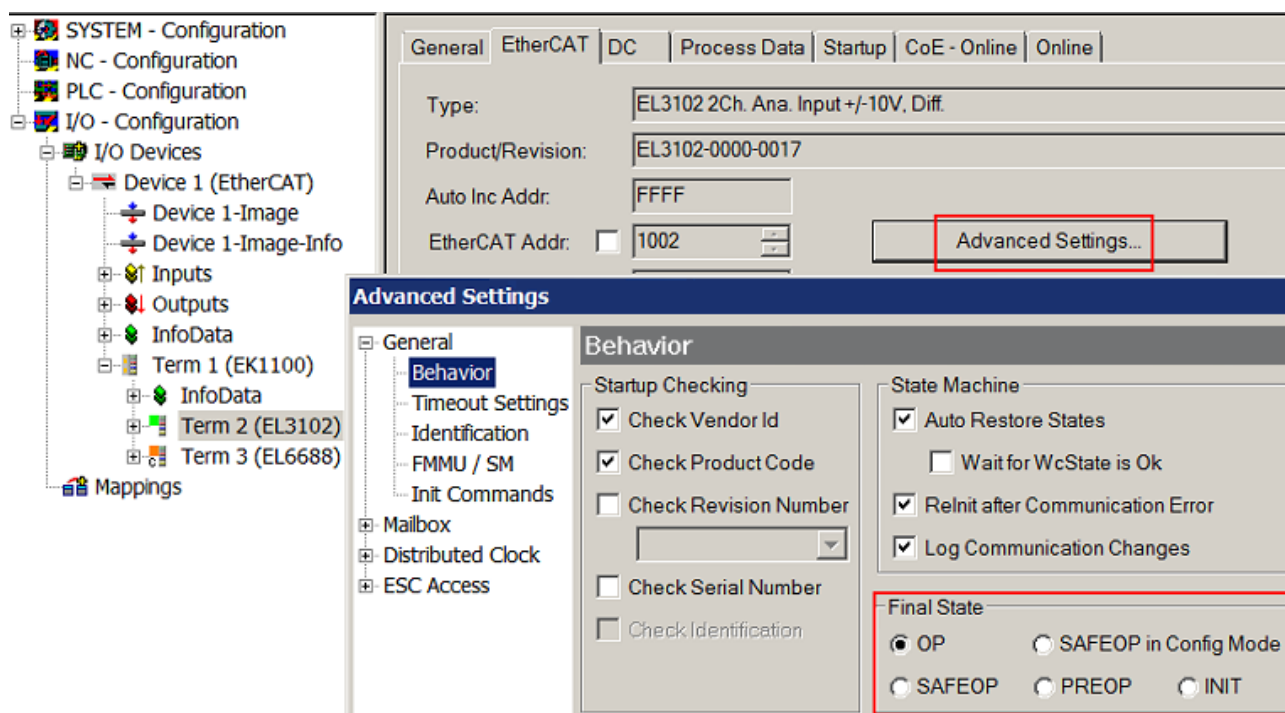
EtherCAT 主站的 advanced settings（高级设置）的标准设置如下：

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



附图 124: 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 125: 从站的默认目标状态

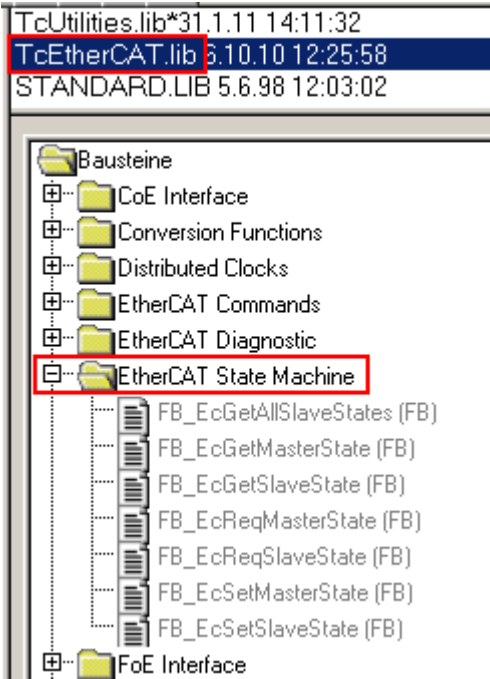
手动控制

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 126: PLC 功能块

关于 E-Bus 电流的说明

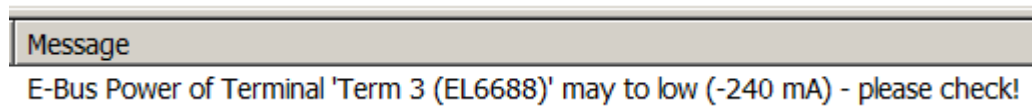
EL/ES 端子模块置于 DIN 导轨上，紧跟在耦合器后面。总线耦合器可以向后续的EL 端子模块供给 5V 的 E-bus 系统电压；耦合器原则上可以负担最多 2 A 的 E-Bus 电流。关于每个 EL 端子模块需要消耗多少 E-bus 电流的信息，可参见倍福公司网站和产品目录。如果后续的端子模块需要的电流超过了耦合器可以提供的电流，则必须在 I/O 站的适当位置插入E-Bus电源端子模块（例如 EL9410）。

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 127: 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：



附图 128: 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！

5.4 CoE 对象概览

5.4.1 标准 CoE 对象

标准 CoE 对象

索引（十六进制）1000 至 1018:04 的对象称为标准 CoE 对象。它们包含有关相应设备和相应端子模块的通用信息，例如设备名称、软件和硬件版本以及制造商特定 ID（序列号、版本号、供应商 ID 和产品代码）。

索引（十六进制）	名称	含义	数据类型	标志	默认值
1000:0	设备类型	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	0x00001389 (5001 _{dec})

索引（十六进制）	名称	含义	数据类型	标志	默认值
1008:0	设备名称	EtherCAT 从站的设备名称	STRING	RO	EL6695-0000

索引（十六进制）	名称	含义	数据类型	标志	默认值
1009:0	硬件版本	EtherCAT 从站的硬件版本	STRING	RO	00

索引（十六进制）	名称	含义	数据类型	标志	默认值
100A:0	软件版本	EtherCAT 从站的固件版本	STRING	RO	02

索引（十六进制）	名称	含义	数据类型	标志	默认值
1018:0	标识	识别从站的信息	UINT8	RO	0x04 (4 _{dec})
1018:01	供应商 ID	EtherCAT 从站的供应商 ID	UINT32	RO	0x00000002 (2 _{dec})
1018:02	产品代码	EtherCAT 从站的产品代码	UINT32	RO	0x1A243052 (438579282 _{dec})
1018:03	修订版本	EtherCAT 从站的修订版本号；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	0x00000064 (100 _{dec})
1018:04	序列号	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 _{dec})

5.4.2 端子模块特定 CoE 对象

端子模块特定 CoE 对象

EL6695 桥接端子模块的端子模块特定 CoE 对象从索引（十六进制）0x10F4 开始。对象 0x10F4 和 0x10F5 包含外部同步的状态和设置。

索引 (十六进制)	名称	含义	数据类型	标志	默认值
10F4:0	外部同步状态	有关同步状态的信息	UINT8	RO	0x13 (19 _{dec})
10F4:01	同步模式	同步模式 0 = 无同步 1 = 副边为同步主站 2 = 原边为同步主站 3 = 两侧均为同步主站 - 不允许使用此状态, 应通过适当的 EtherCAT 主站设置来更改, 即仅在一侧激活 PDO 时间戳映射。	BIT2	RO P	0x00 (0 _{dec})
10F4:0E	Control value update toggle	当新的控制值可用时, 该位切换	BOOLEAN	RO P	0x00 (0 _{dec})
10F4:0F	Time stamp update toggle	当提供新的 DC 数据时, 该位切换	BOOLEAN	RO P	0x00 (0 _{dec})
10F4:10	External device not connected	0 = 另一侧已连接至其 EtherCAT 现场总线 1 = 另一侧未连接至其 EtherCAT 现场总线	BOOLEAN	RO P	0x00 (0 _{dec})
10F4:11	Internal time stamp	当前侧的分布式时钟时间	UINT64	RO P	-
10F4:12	External time stamp	另一侧 (远程侧) 的分布式时钟时间	UINT64	RO P	-
10F4:13	DC 主时钟的控制值	用于修正低优先级参考时钟的偏移量	INT32	RO P	0x00000000 (0 _{dec})

索引 (十六进制)	名称	含义	数据类型	标志	默认值
10F5:0	外部同步设置	用于同步 EtherCAT 桥接端子模块的设置	UINT8	RO	0x12 (18 _{dec})
10F5:01	同步主站	0: 同步主站位于原边 1: 同步主站位于副边	BOOLEAN	RW (PREO P)	0x00 (0 _{dec})
10F5:02	32 位时间戳	0: 64 位时间戳 1: 32 位时间戳	BOOLEAN	RW	0x00 (0 _{dec})
10F5:11	控制间隔时间 (ms)	用于计算控制值的间隔时间 (ms)	UINT16	RW	0x0000 (0 _{dec})
10F5:12	附加系统时间	用于计算控制值的附加 DC 时间	UINT64	RW	0x0000000000000000 (0 _{dec})

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1608:0	RxPDO-Map	PDO 映射 RxPDO 1 (声明的输出过程数据的 PDO 映射)	UINT8	RW	0x00 (0 _{dec})
..1608:FF	-	-	-	-	-
...					
161F:00...					
161F:FF					

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1801:0	TxPDO-Par External Sync Compact	PDO 参数 TxPDO 2	UINT8	RO	0x09 (9 _{dec})
1801:06	不包括 TxPDOs	指定不得与 TxPDO 2 一起传输的 TxPDO (TxPDO 映射对象的索引 (十六进制))	OCTET-STRING [4]	RO	02 1A 03 1A
1801:07	TxPDO 状态	如果无法正确读入相关输入数据, 则 TxPDO State 被置位	BOOLEAN	RO P	0x00 (0 _{dec})
1801:09	TxPDO 切换	每次更新相应的输入数据时, 都会进行 TxPDO-Toggle	BOOLEAN	RO P	0x00 (0 _{dec})

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1802:0	TxPDO-Par External Sync	PDO 参数 TxPDO 3	UINT8	RO	0x09 (9 _{dec})
1802:06	不包括 TxPDOs	指定不得与 TxPDO 3 一起传输的 TxPDO (TxPDO 映射对象的索引 (十六进制))	OCTET-STRING [4]	RO	01 1A 03 1A
1802:07	TxPDO 状态	如果无法正确读入相关输入数据, 则 TxPDO State 被置位	BOOLEAN	RO P	0x00 (0 _{dec})
1802:09	TxPDO 切换	每次更新相应的输入数据时, 都会进行 TxPDO-Toggle	BOOLEAN	RO P	0x00 (0 _{dec})

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1803:0	TxPDO-Par External Sync (32 Bit)	PDO 参数 TxPDO 4	UINT8	RO	0x09 (9 _{dec})
1803:06	不包括 TxPDOs	指定不得与 TxPDO 4 一起传输的 TxPDO (TxPDO 映射对象的索引 (十六进制))	OCTET-STRING [4]	RO	01 1A 02 1A
1803:07	TxPDO 状态	如果无法正确读入相关输入数据, 则 TxPDO State 被置位	BOOLEAN	RO P	0x00 (0 _{dec})
1803:09	TxPDO 切换	每次更新相应的输入数据时, 都会进行 TxPDO-Toggle	BOOLEAN	RO P	0x00 (0 _{dec})

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1A01:0	TxPDO-Map External Sync Compact	PDO 映射 TxPDO 1	UINT8	RW	0x05 (5 _{dec})
1A01:01	子索引 (十六进制) 001	12 位对齐	UINT32	RW	0x0000:00, 12
1A01:02	子索引 (十六进制) 002	PDO 映射条目 (对象 0x1801, 条目 0x09)	UINT32	RW	0x1801:09, 1
1A01:03	子索引 (十六进制) 003	PDO 映射条目 (对象 0x1801, 条目 0x07)	UINT32	RW	0x1801:07, 1

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1A01:04	子索引 (十六进制) 004	1 位对齐	UINT32	RW	0x0000:00, 1
1A01:05	子索引 (十六进制) 005	PDO 映射条目 (对象 0x10F4 (外部同步状态), 条目 0x10)	UINT32	RW	0x10F4:10, 1

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1A02:0	TxPDO-Map External Sync	PDO 映射 TxPDO 2	UINT8	RW	0x09 (9 _{dec})
1A02:01	子索引 (十六进制) 001	PDO 映射条目 (对象 0x10F4 (外部同步状态), 条目 0x02)	UINT32	RW	0x10F4:01, 2
1A02:02	子索引 (十六进制) 002	10 位对齐	UINT32	RW	0x0000:00, 10
1A02:03	子索引 (十六进制) 003	PDO 映射条目 (对象 0x1802, 条目 0x09)	UINT32	RW	0x1802:09, 1
1A02:04	子索引 (十六进制) 004	PDO 映射条目 (对象 0x1802, 条目 0x07)	UINT32	RW	0x1802:07, 1
1A02:05	子索引 (十六进制) 005	PDO 映射条目 (对象 0x10F4 (外部同步状态), 条目 0x0F (时间戳更新切换))	UINT32	RW	0x10F4:0F, 1
1A02:06	子索引 (十六进制) 006	PDO 映射条目 (对象 0x10F4 (外部同步状态), 条目 0x10 (外部设备未连接))	UINT32	RW	0x10F4:10, 1
1A02:07	子索引 (十六进制) 007	PDO 映射条目 (对象 0x10F4 (外部同步状态), 条目 0x11 (内部时间戳))	UINT32	RW	0x10F4:11, 64
1A02:08	子索引 (十六进制) 008	PDO 映射条目 (对象 0x10F4 (外部同步状态), 条目 0x12 (外部时间戳))	UINT32	RW	0x10F4:12, 64
1A02:09	子索引 (十六进制) 009	PDO 映射条目 (对象 0x10F4 (外部同步状态), 条目 0x13 (保留))	UINT32	RW	0x10F4:13, 32

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1A03:0	TxPDO-Map External Sync	PDO 映射 TxPDO 3	UINT8	RW	0x09 (9 _{dec})
1A03:01	子索引 (十六进制) 001	PDO 映射条目 (对象 0x10F4 (外部同步状态), 条目 0x02)	UINT32	RW	0x10F4:01, 2
1A3:02	子索引 (十六进制) 002	10 位对齐	UINT32	RW	0x0000:00, 10
1A3:03	子索引 (十六进制) 003	PDO 映射条目 (对象 0x1803, 条目 0x09)	UINT32	RW	0x1803:09, 1
1A3:04	子索引 (十六进制) 004	PDO 映射条目 (对象 0x1803, 条目 0x07)	UINT32	RW	0x1803:07, 1
1A3:05	子索引 (十六进制) 005	PDO 映射条目 (对象 0x10F4 (外部同步状态), 条目 0x0F (时间戳更新切换))	UINT32	RW	0x10F4:0F, 1

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1A3:06	子索引 (十六进制) 006	PDO 映射条目 (对象 0x10F4 (外部同步状态), 条目 0x10 (外部设备未连接))	UINT32	RW	0x10F4:10, 1
1A3:07	子索引 (十六进制) 007	PDO 映射条目 (对象 0x10F4 (外部同步状态), 条目 0x11 (内部时间戳))	UINT32	RW	0x10F4:11, 32
1A3:08	子索引 (十六进制) 008	PDO 映射条目 (对象 0x10F4 (外部同步状态), 条目 0x12 (外部时间戳))	UINT32	RW	0x10F4:12, 32
1A3:09	子索引 (十六进制) 009	PDO 映射条目 (对象 0x10F4 (外部同步状态), 条目 0x13 (保留))	UINT32	RW	0x10F4:13, 32

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1A04:0	活动 TxPDOs-Map	PDO 映射 TxPDO 4	UINT32	RW	0x05 (5 _{dec})
1A04:01	子索引 (十六进制) 01	PDO 映射条目 (对象 0xF640:01 (远程写入周期 u16Count))	UINT32	RW	10 01 40 F6
1A04:02	子索引 (十六进制) 02	PDO 映射条目 (对象 0xF630:01 活动 TxPdo 信息 PDO 1-8)	UINT32	RW	10 01 30 F6
1A04:03	子索引 (十六进制) 03	PDO 映射条目 (对象 0xF630:02 活动 TxPdo 信息 PDO 9-16)	UINT32	RW	10 02 30 F6
1A04:04	子索引 (十六进制) 04	PDO 映射条目 (对象 0xF630:03 活动 TxPdo 信息 PDO 17-24)	UINT32	RW	10 03 30 F6
1A04:05	子索引 (十六进制) 05	PDO 映射条目 (对象 0xF630:04 活动 TxPdo 信息 PDO 25-32)	UINT32	RW	10 04 30 F6

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1A05:0	FoE Info-Map	PDO 映射 TxPDO 5	UINT8	RW	0x01 (1 _{dec})
1A05:01		PDO 映射条目 (对象 0xF650:01 活动 TxPdo 信息 PDO 25-32)	UINT32	RW	10 01 50 F6

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1A08:0	TxPDO-Map	PDO 映射 TxPDO 1 (声明的输入过程数据的 PDO 映射)	UINT8	RW	0x00 (0 _{dec})
...1A08:FF	-	-	-	-	-
...1A1F:00...					
1A1F:FF					

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1C00:0	Sync Manager type	同步管理器类型通道 (邮箱/过程数据, 读取/写入)	UINT8	RO	0x04 (4 _{dec})
1C00:01	子索引 (十六进制) 001	同步管理器类型通道 1: 邮箱写入	UINT8	RO	0x01 (1 _{dec})

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1C00:02	子索引 (十六进制) 002	同步管理器类型通道 2: 邮箱读取	UINT8	RO	0x02 (2 _{dec})
1C00:03	子索引 (十六进制) 003	同步管理器类型通道 3: 过程数据写入 (输出)	UINT8	RO	0x03 (3 _{dec})
1C00:04	子索引 (十六进制) 004	同步管理器类型通道 4: 过程数据读取 (输入)	UINT8	RO	0x04 (4 _{dec})

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1C12:0	RxPDO 分配	PDO 分配输出 (从 FW09 开始: 62, 而非 9)	UINT8	RW	0x00 (0 _{dec})
1C12:01	子索引 (十六进制) 001	第 1 个分配的 RxPDO (包含相关 RxPDO 映射对象的索引 (十六进制))	UINT16	RW	0x0000 (0 _{dec})

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1C13:0	TxPDO 分配	PDO 分配输入 (从 FW09 开始: 62, 而非 9)	UINT8	RW	0x01 (1 _{dec})
1C13:01	子索引 (十六进制) 001	第 1 个分配的 TxPDO (包含相关 TxPDO 映射对象的索引 (十六进制))	UINT16	RW	0x1A01 (6657 _{dec})
1C13:02	子索引 (十六进制) 002	第 2 个分配的 TxPDO (包含相关 TxPDO 映射对象的索引 (十六进制))	UINT16	RW	0x0000 (0 _{dec})

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1C32:0	SM 输出参数	输出的同步参数	UINT8	RO	0x20 (32 _{dec})
1C32:01	同步模式	当前的同步模式: 0: Free Run 1: 与 SM 2 事件同步 2: DC 模式 - 与 SYNC0 事件同步 3: DC 模式 - 与 SYNC1 事件同步	UINT16	RW	0x0000 (0 _{dec})
1C32:02	循环时间	循环时间 (单位: ns): Free Run: 本地定时器的周期 与 SM 2 事件同步: 主站周期 DC 模式: SYNC0/SYNC1 循环时间	UINT32	RW	0x00000000 (0 _{dec})
1C32:03	偏移时间	从 SYNC0 事件到输出的时间 (单位: ns, 仅 DC 模式)	UINT32	RW	0x00000000 (0 _{dec})
1C32:04	支持同步模式	支持的同步模式: 位 0 = 1: 支持 Free Run 位 1 = 1: 支持与 SM 2 事件同步 位 2-3 = 01: 支持 DC 模式 位 4-5 = 10: SYNC1 事件下的输出偏移 (仅 DC 模式) 位 14 = 1: 动态周期 (在写入 1C32:08 时开始测量)	UINT16	RO	0xC007 (49159 _{dec})

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1C32:05	最短循环时间	最短循环时间 (单位: ns)	UINT32	RO	0x00000000 (0 _{dec})
1C32:06	计算并复制时间	SYNC0 和 SYNC1 事件之间的最小时间 (单位: ns, 仅 DC 模式)	UINT32	RO	0x00000000 (0 _{dec})
1C32:08	命令	0: 本地循环时间的测量停止 1: 本地循环时间的测量开始 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 _{dec})
1C32:09	最长延迟时间	从 SYNC1 事件到输出的时间 (单位: ns, 仅 DC 模式)	UINT32	RO	0x00000000 (0 _{dec})
1C32:0B	缺失 SM 事件计数器	OPERATIONAL 期间缺失的 SM 事件数量 (仅 DC 模式)	UINT16	RO	0x0000 (0 _{dec})
1C32:0C	周期超限计数器	OPERATIONAL 期间周期超时的次数 (周期没有及时完成或下一个周期开始得太早)	UINT16	RO	0x0000 (0 _{dec})
1C32:0D	偏移过短计数器	SYNC0 和 SYNC1 事件之间的间隔次数太小 (仅 DC 模式)	UINT16	RO	0x0000 (0 _{dec})
1C32:20	同步错误	在上一周期同步出错 (输出太晚; 仅 DC 模式)	BOOLEAN	RO	0x00 (0 _{dec})

索引 (十六进制)	名称	含义	数据类型	标志	默认值
1C33:0	SM 输入参数	输入的同步参数	UINT8	RO	0x20 (32 _{dec})
1C33:01	同步模式	当前的同步模式: 0: Free Run 1: 与 SM 3 事件同步 (无输出可用) 2: DC - 与 SYNC0 事件同步 3: DC - 与 SYNC1 事件同步 34: 与 SM 2 事件同步 (输出可用)	UINT16	RW	0x0000 (0 _{dec})
1C33:02	循环时间	如 1C32:02	UINT32	RW	0x00000000 (0 _{dec})
1C33:03	偏移时间	从 SYNC0 事件到读取输入的时间 (单位: ns, 仅 DC 模式)	UINT32	RW	0x00000000 (0 _{dec})
1C33:04	支持同步模式	支持的同步模式: 位 0: 支持 Free Run 位 1: 支持与 SM 2 事件同步 (输出可用) 位 1: 支持与 SM 3 事件同步 (无输出可用) 位 2-3 = 01: 支持 DC 模式 位 4-5 = 01: 基于本地事件的输入偏移 (输出可用) 位 4-5 = 10: 基于 SYNC1 事件的输入偏移 (无输出可用)	UINT16	RO	0xC007 (49159 _{dec})

索引 (十六进制)	名称	含义	数据类型	标志	默认值
		位 14 = 1: 动态周期 (从写入 1C32:08 或 1C33:08 开始测量)			
1C33:05	最短循环时间	如 1C32:05	UINT32	RO	0x00000000 (0 _{dec})
1C33:06	计算并复制时间	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 _{dec})
1C33:08	命令	如 1C32:08	UINT16	RW	0x0000 (0 _{dec})
1C33:09	最长延迟时间	从 SYNC1 事件到读取输入的时间 (单位: ns, 仅 DC 模式)	UINT32	RO	0x00000000 (0 _{dec})
1C33:0B	缺失 SM 事件计数器	如 1C32:11	UINT16	RO	0x0000 (0 _{dec})
1C33:0C	周期超限计数器	如 1C32:12	UINT16	RO	0x0000 (0 _{dec})
1C33:0D	偏移过短计数器	如 1C32:13	UINT16	RO	0x0000 (0 _{dec})
1C33:20	同步错误	如 1C32:32	BOOLEAN	RO	0x00 (0 _{dec})

5.4.3 配置文件特定 CoE 对象



读取 CoE 对象 0x6000 和 0x7000

读取 CoE 对象 0x6000 和 0x7000 时给出的数据并非真实的过程数据。实际数据仅可通过 PDO 分配间接读取。

索引 (十六进制)	名称	含义	数据类型	标志	默认值
6000:0	输入数据	声明的输入过程数据 (动态创建)	UINT8	RO	0x00 (0 _{dec})
6000:01					
...					
6000:FF					

索引 (十六进制)	名称	含义	数据类型	标志	默认值
7000:0	输出数据	声明的输出过程数据 (动态创建)	UINT8	RO	0x00 (0 _{dec})
7000:01					
...					
7000:FF					

索引 (十六进制)	名称	含义	数据类型	标志	默认值
F000:0	模块化设备配置文件	模块化设备配置文件的常规信息	UINT8	RO	0x02 (2 _{dec})
F000:01	模块索引 (十六进制) 间距	通道之间对象 Index (Hex) 的间隔	UINT16	RO	0x0010 (16 _{dec})
F000:02	最大模块数量	通道数量	UINT16	RO	0x0001 (1 _{dec})

索引 (十六进制)	名称	含义	数据类型	标志	默认值
F008:0	代码字	目前保留	UINT32	RW	0x00000000 (0 _{dec})

索引 (十六进制)	名称	含义	数据类型	标志	默认值
F010:0	模块列表	最大子索引 (十六进制)	UINT8	RW	0x01 (1 _{dec})
F010:01	子索引 (十六进制) 001	-	UINT32	RW	0x00000000 (0 _{dec})

当前为 64 个 PDO，每个 Tx-PDO 占 2 位

位 0 → 1=映射存在 (p)

位 1 → 1=映射激活 (a)

索引 (十六进制)	名称
F630	Active TxPdo
映射:	a p a p a p a p a p a p a p a p
0	0
1	1A07 1A06 1A05 1A04 1A03 1A02 1A01 1A00
2	1A0F 1A0E 1A0D 1A0C 1A0B 1A0A 1A09 1A08
3	1A17 1A16 1A15 1A14 1A13 1A12 1A11 1A10
4	1A1F 1A1E 1A1D 1A1C 1A1B 1A1A 1A19 1A18
...	
8	1A3F 1A3E 1A3D 1A3C 1A3B 1A3A 1A39 1A38

索引 (十六进制)	位	名称	含义	注释
F800:0	-	Device Config		
F800:01	0x8000	AoE	1 = 协议禁用	
	0x4000	EoE		
	0x2000	FoE		
	0x1000	SoE		
	0x0800	VoE		
	0x0400	其它		
	0x0200	-		如果为 0，则系统将检查 MBX 协议 ECAT-EEPROM 是否存在
	...			
	0x0002	-		抑制错误消息
	0x0001	-		1: 如果未经路由: 抑制错误消息
	...			
0xF800:02	0x0100	FoE Buffer	Enable	
	0x2000	Watchdog (看门狗) 模式		EL6695 的 Watchdog (看门狗) 默认通过 AL 事件 “Watchdog (看门狗)” 触发。通过将 0x2000 中的相应位单独置位，即可改为使用 WriteCycleCounter 作为触发。在某些 EtherCAT 主站下运行时，可能需要进行此设置。

索引 (十六进制)	位	名称	含义	注释
	0x0002	OP State	禁用对另一侧的依赖	禁用运行状态对副边的依赖 (仅限设备仿真)
	0x0001	OP State		禁用运行状态对原边的依赖 (仅限设备仿真)
0xF800:03	0x2000		SW 软重启	在从任意状态切换为 Init 状态期间重新启动 (A → I)
	...			

索引 (十六进制)	名称	含义	数据类型	标志	默认值
F820:0	ADS Server Settings	目标 Net ID / Port	对象	RW	0x02 (2 _{dec})
F820:01	Net ID	目标 Net ID	数组 [0..5]	RW	0x00, ..
F820:02	端口	目标端口	UINT16	RW	0x0000

索引 (十六进制)	名称	含义	数据类型	标志	默认值
F821:0	EL6695 ADS Settings	源 Net ID / Port	对象	RO	0x02 (2 _{dec})
F821:01	Net ID	源 Net ID	数组 [0..5]	RO	0x00, ..
F821:02	端口	源端口	UINT16	RO	0x0000

索引 (十六进制)	名称	含义	数据类型	标志	默认值
FA20:0	设备诊断	包含有关状态、CPU 和堆负载的诊断信息，以及有关发送数据包和内部复制时间的信息	UINT16	RD	0x1D (29 _{dec})
FA20:01	状态		UINT16	RD	
FA20:02	CPU usage [%]		UINT16	RD	
FA20:03	Heap Usage [%]		UINT16	RD	
FA20:04	AOE Packets		UINT16	RD	
FA20:05	EOE Packets		UINT16	RD	
FA20:06	FOE Packets		UINT16	RD	
FA20:07	SOE Packets		UINT16	RD	
FA20:08	VOE Packets		UINT16	RD	
FA20:09	Other Packets		UINT16	RD	
FA20:0A	Mbx Info		UINT16	RD	
FA20:0B	PD Copy time (my)	时间 (μs)	UINT16	RD	
FA20:0C	PD Copy time (remote)		UINT16	RD	
FA20:0D..	Info 2		UINT16	RD	
FA20:1D	Info 18		UINT16	RD	

6 功能和运行模式

根据过程数据设置，EL 端子模块的运行模式决定了基本的运行原则。

6.1 基本功能原理

EtherCAT 桥接端子模块 EL6695 可以实现 EtherCAT 网段与两个可能不同的主站之间的同步实时数据交换。还支持通过诸如 AoE、FoE、EoE、VoE 等各种非周期性协议进行异步通信。

EL6695 提供三项基本功能：

- 同步 PDO 数据交换
- 异步数据交换
- 分布式时钟同步

在同步过程数据的同步数据交换期间，EtherCAT 主站将预定义的过程数据从桥接端子模块的一侧复制到另一侧。两侧的状态变量提供关于数据缺失以及对方状态的信息。

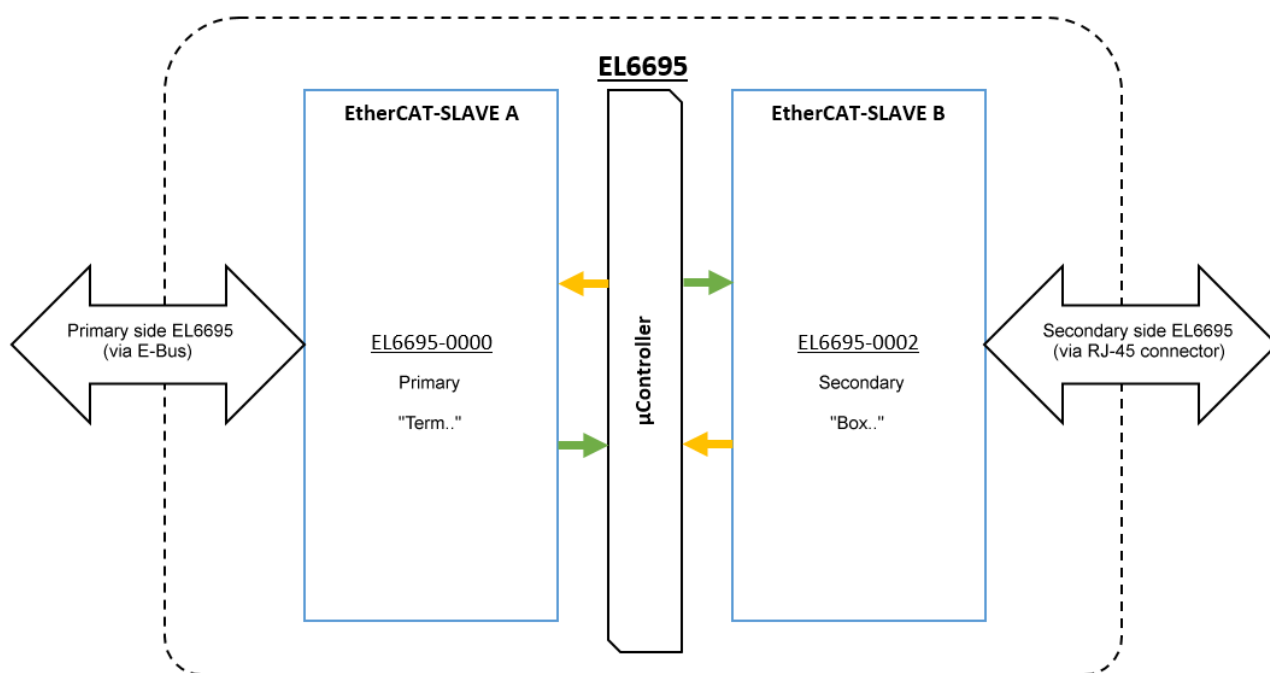
分布式时钟可以双向同步。重要的是要确定哪一侧包含更高层级的参考时钟，即所谓的主时钟。然后，EL6695 将时间值传输至另一侧的次级 EtherCAT 主站，以便该主站（如果支持此功能）能够调整其 EtherCAT 系统时间。

EL6695 和 EL6692（该产品将继续销售）不同的是，它具有扩展功能，例如灵活的 CoE 配置和设备仿真选项，特别是数据吞吐率显著提升。

TwinCAT 3 的 TwinCAT System Manager 中有一个用户友好的配置接口（TwinCAT 2 版本正在准备中）。此扩展插件对于一般运行并非绝对必要，但仿真模式需要它。

EL6695 的内部电源由原边（E-bus）或副边（RJ45）冗余提供。只要至少在其中一侧供电，EL6695 两侧的 EtherCAT 上即可完全正常运行。副边（RJ45）通过外部连接供电，原边通过 E-bus 供电。如果两侧均供电，则优先使用 24 V 电源。

EL6695 实际上在一个外壳中包含两个完整的 EtherCAT 从站：原边（E-bus）的 EL6695-0000，以及副边带有网络电缆连接（RJ45）的 EL6695-0002。

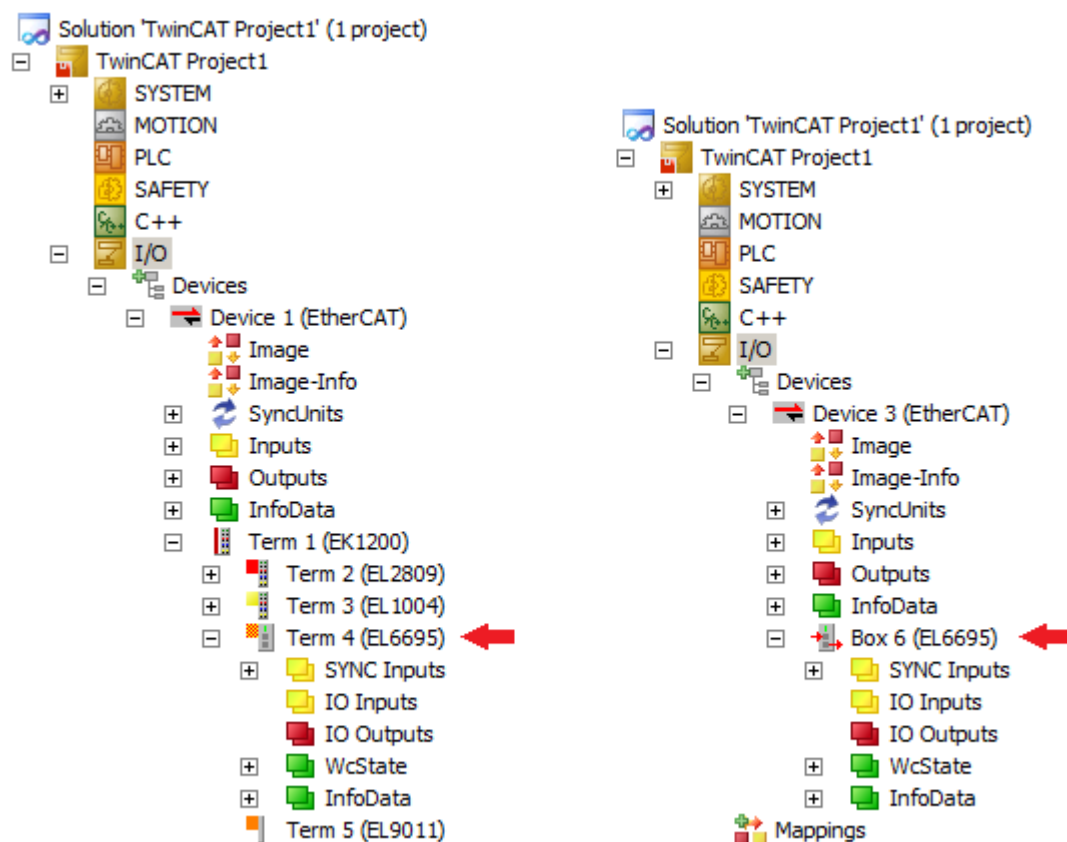


附图 129: 桥接端子模块的基本功能

就功能而言，桥接端子模块的两侧是等效的，即所有功能均可从两侧操作或使用。尽管如此，出于定位原因，仍继续使用 E-bus 侧的术语“原边”和 RJ45 侧的术语“副边”，尽管对于 EL6695 而言，原边/副边并不表示任何权重或等级关系。

6.1.1 通过 TwinCAT 访问 EL6695

上下文菜单（右键点击，然后选择 **Add New Item...**（添加新项目...））可用于创建配置或将 EL6695 桥接端子模块集成到 TwinCAT 项目中。或者，右键点击相应的 EtherCAT 设备，从设备 n（EtherCAT）的上下文菜单中选择 **Scan**（扫描）功能，该功能将自动读取所有已连接的设备和端子盒，并将其集成到结构中（请参见图示）：



附图 130: “Scan” 之后：端子模块网段中的 EL6695（左侧）以及通过 EtherCAT（通过 RJ-45 X1）作为 Box（右侧）

在 E-bus 侧，EL6695 可视为端子模块网段中的“端子模块”。通过 RJ45 接口，它在配置中表示为“Box”（请参见 EtherCAT 从站）。读取/创建后，EL6695 桥接端子模块的初始状态如下：

- Free Run => DC 支持禁用
- 没有预先配置的过程数据变量
- 没有分布式时钟信息

6.2 与 EL6692 的兼容性

EL6695 本质上应被视为新开发的产品，尽管它支持与 EL6692 兼容的运行模式，在此模式下它使用相同的默认过程数据（状态/诊断）。但请注意，由于其时间特性不同，EL6695 与 EL6692 并非完全可互换兼容。

通过选择 0x1A02 或 0x1A03 中的诊断数据，可以显示 EL6692 中常见的诊断数据，例如 SyncMode、TxPdoToggle、TxPDO state，“Timestamp update toggle” 和 “External device not connected”。

Name	Typ	Grö...	>Adr...	Ein/...	User.
◆↑ Sync Mode	BIT2	0.2	39.0	Eing...	0
◆↑ TxPDO toggle	BOOL	0.1	40.4	Eing...	0
◆↑ TxPDO state	BOOL	0.1	40.5	Eing...	0
◆↑ Timestamp update toggle	BOOL	0.1	40.6	Eing...	0
◆↑ External device not connected	BOOL	0.1	40.7	Eing...	0

附图 131: EL6695 兼容模式（默认 PDO）

6.3 状态机 EL6695

EL6695 桥接端子模块的两侧（原边和副边）均支持以下 EtherCAT 状态：INIT、PreOP、SafeOP、OP、BOOTSTRAP。

一侧的状态变化不会影响另一侧的状态。在仿真模式下，情况并非如此，请参见该部分。

i INIT 状态

EL6695 桥接端子模块处理相应侧的单侧“正常”INIT 状态请求，而不影响另一侧。在端子模块的基本配置发生更改后（例如通过 FW 更新、更改设备仿真、更改部分对象字典），端子模块将在**两侧**重新启动。

6.4 周期性过程数据 PDO

为便于区分，为 EL6695 引入两个术语

- “对称 PDO 映射”：两侧创建相同大小和顺序的过程数据
- “选择性 PDO 映射”：一侧将最大容量的过程数据加载到 EL6695 中，另一侧的主站从中选择一个子集并请求将其作为周期性过程数据。

关键数据（截至 2015 年 6 月，FW04，TwinCAT 3.1 b4018.4）：

- 每个方向最多 255 个变量（如果手动创建 PDO，则通过 TwinCAT 管理 0x1A08:nn 和 0x1608:nn 中的最多 255 个条目）。
- 最大 PDO 总大小：MTU（~1500 字节），即每个方向一个以太网帧，对应约 1408 字节用户数据。根据要求可提供更大容量，需要在 TwinCAT 中进行适配。
- 最小 PDO 大小为 1 字节，不允许使用位级 PDO。

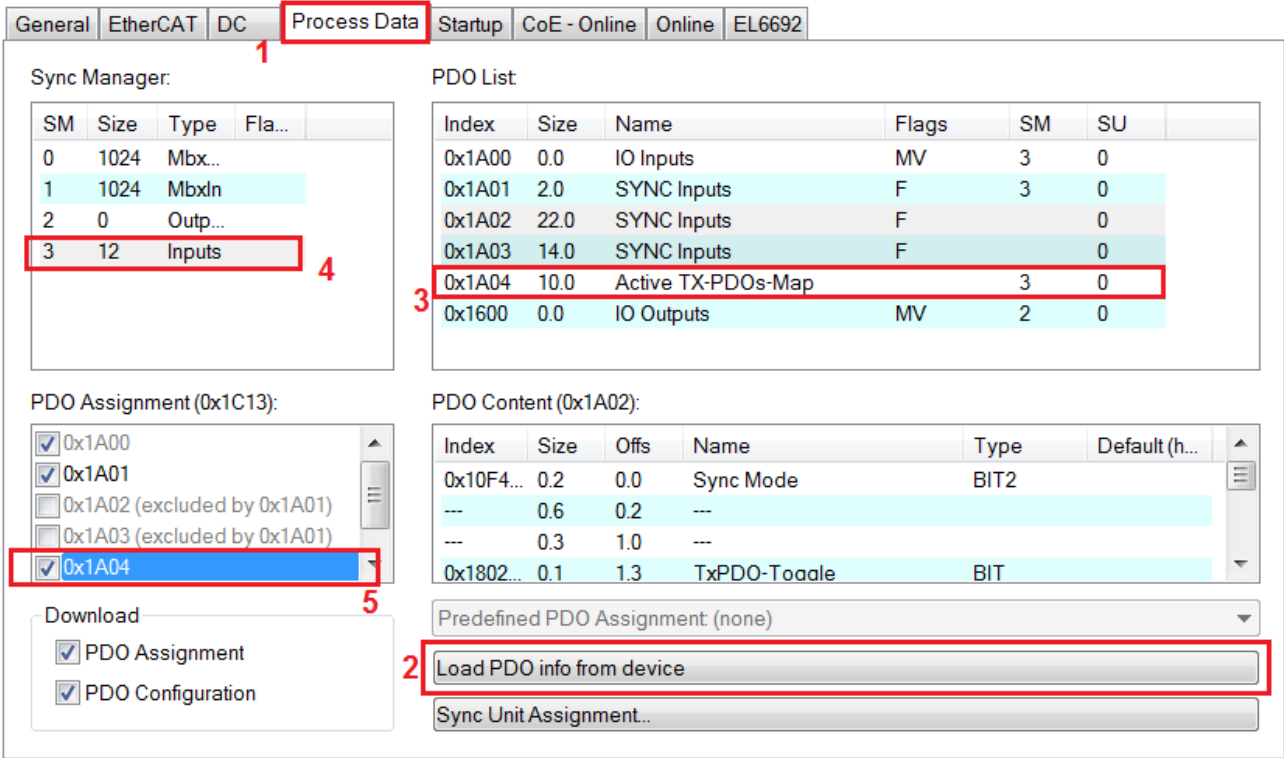
配置更改仅在相应的 EL6695 侧重启后生效。

对于每个 6000 类型的对象，系统会在另一侧搜索相应的 7000 类型对象 – 例如 6001:05 → 7001:05。另一侧的对象（7000 类型）的数量必须等于或大于 6000 类型的对象。就此而言，也可以复制 7000 的子集 → 6000，请参见“选择性 PDO 映射”。

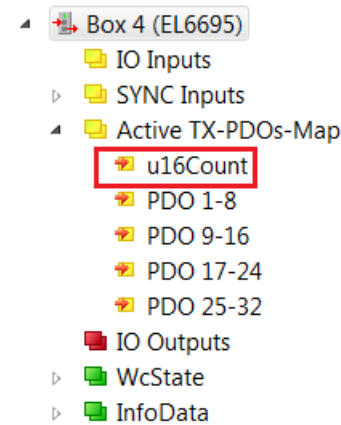
6.4.1 流程控制

流程控制需由应用层自行处理，也就是说 EL6695 模块不采用握手机制。然而，在两种 PDO 模式下都可以显示一个计数器，该计数器在每次来自另一侧的写入访问时递增：CoE 对象 0xF640。为确保该计数器可作为周期性 PDO 进行映射，应通过启动条目按如下方式进行映射：

为了能够映射 CoE 对象 0xF640，应首先显示所有在线 CoE 对象，如“通信基础” / “CoE 接口”章节中的“在线/离线列表”部分 [► 30]所述。然后可以通过“Process Data（过程数据）”选项卡重新加载设备的 PDO 数据，以便 PDO 列表显示“活动 Tx PDO 映射”条目。如果现在在 Sync Manager（同步管理器）下选择“Inputs”条目，则可以在“PDO 分配”下勾选对象 0x1A04，从而将 CoE 对象 0xF640 的计数器映射到过程映像中。



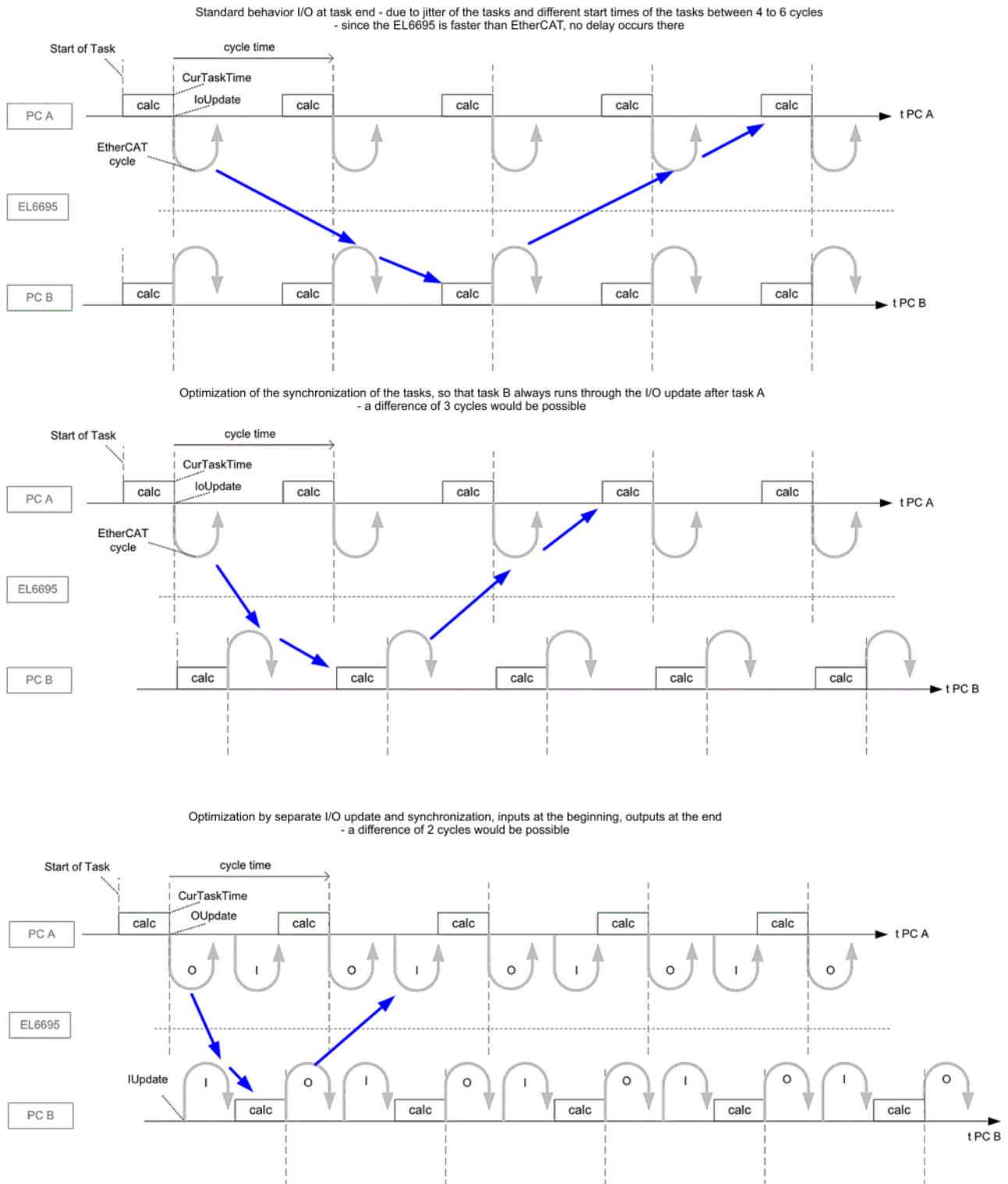
附图 132: 从设备加载 PDO 数据会创建 “Active TX PDOs Map” 条目



附图 133: PDO 结构

数据吞吐率 (示例)

下图展示了 EL6695 在标准配置下以及采用两种优化方法（即通过同步优化和通过包含同步的独立 IO 更新优化）时的数据吞吐率：



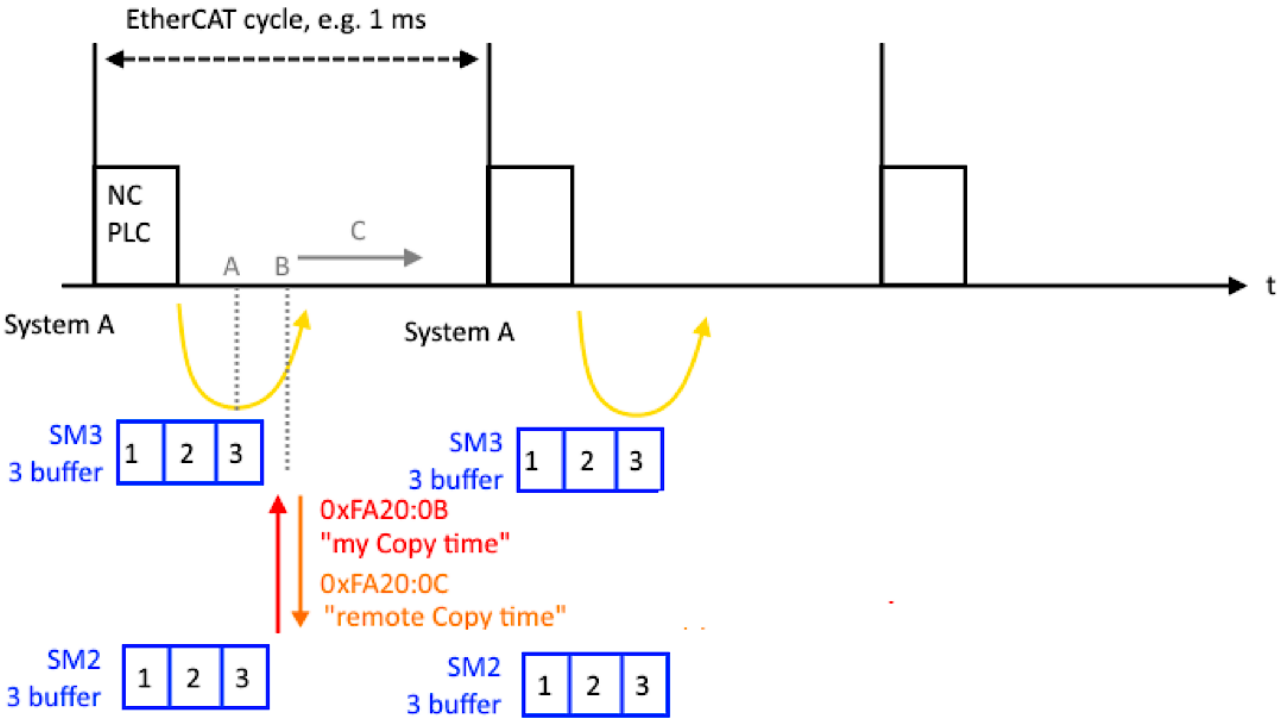
附图 134: 数据吞吐量：标准配置及优化后

将配置的过程数据从 EtherCAT 的一侧传输到另一侧所需的时间取决于数据量。

内部数据传输由一侧的读取操作触发。下图“EL6695 数据传输顺序”说明了该顺序：

- Sync Manager（同步管理器）2 和 3 以 3 缓冲模式运行，
- 如果 EtherCAT 主站通过 EtherCAT 帧（A）从 Sync Manager（同步管理器）3 的缓冲区 3 获取数据，则在此读取过程完成后（B），EL6695 立即开始将另一侧的新数据复制到下一个空闲缓冲区（在本示例中为缓冲区 1）。此过程所需的“Copy Time（复制时间）”可在线读取。
名称“本侧”/“远程”指的是“我”侧，具体取决于当前在线的 CoE 是原边还是副边。

- 数据将一直保留在那里，直到被 EtherCAT A 取走为止。
- 如果周期较长，相对陈旧的数据可能会在那里保留将近整个 EtherCAT 周期。传输时序可通过 CoE 设置进行调整。通过这种延迟可以对系统进行优化，使得系统 B 存储在 SM2 中的数据很快就被内部传输到 A 侧的 SM3，这样后续经过的 EtherCAT 帧 A 所获取的数据就相对较新。
重要的是确保 A 侧和 B 侧的系统抖动最小。
通过以下参数可以设置偏移量
 - CoE 0x1C33:03 ShiftTime, 单位: ns
 - CoE 0xF830:01 (仅在线 CoE 中可用) , 延迟单位: μ s



附图 135: EL6695 数据传输顺序

- 以下示例展示了采用如下配置的测量结果：
- 原边的 PLC 发送一个 PDO 集合
 - EL6695 将该集合传输到另一侧
 - 原边的 PLC 获取数据，并可能以修改后的形式重新发送
 - EL6695 将数据传输到另一侧
 - 原边的 PLC 获取数据，并统计至此为止的 PLC 循环次数

在本示例中，未进行任何优化（PDO 延迟、DC 同步）。

PDO 字节数	任务周期时间	双向的任务周期数 (平均值)	由此产生的传输时间 (单向)	EL6695 内部输入和输出数据的复制时间；来自 CoE 对象 0xFA20 设备诊断 (平均值)
200	50 μ s	4.4	141.1 μ s	14.3 μ s
	100 μ s	3.03	151.5 μ s	14.3 μ s
1400	150 μ s	8.9	667.5 μ s	42.9 μ s
	200 μ s	6.0	600 μ s	42.3 μ s

为实现硬耦合的周期性数据传输，建议对 EtherCAT 的两侧进行分布式时钟耦合，以避免数据传输过程中的拍频效应。

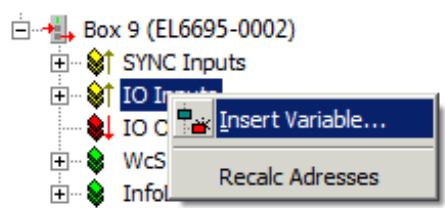
注意：实际上，EL6695 总是会涉及两个具有各自周期的现场总线。需要优化时序（DC 同步，调整偏移时间），以实现 PC A → PC B 的传输时间缩短至与直接 Realtime Ethernet 链路上的发布者/订阅者模式相当。即便如此，也只能针对一个方向进行优化，此外还存在内部传输时间带来的延迟。

6.4.2 对称 PDO 映射

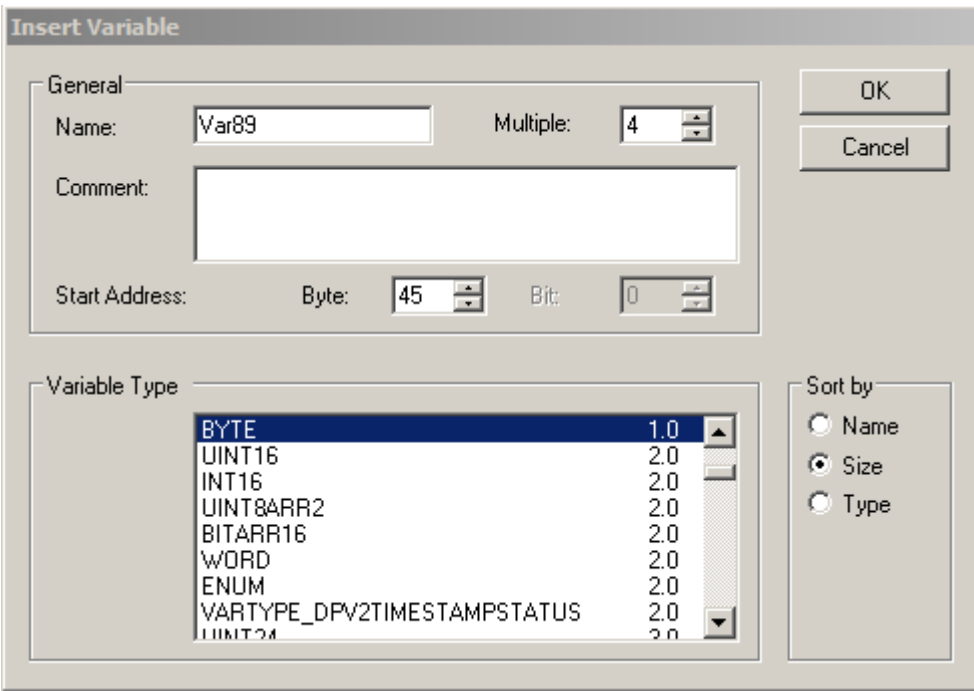
创建通用过程数据/简单数据交换（对称 PDO 映射）

程序：

右键点击 EL6695 目录树中的 **IO Inputs (IO 输入)** 或 **IO Outputs (IO 输出)**，以显示上下文菜单。选择 **Insert Variable...**（插入变量...）（TwinCAT 2）或 **Add New Item**（添加新项目）（TwinCAT 3），以创建新变量/过程数据（请参见下图）。

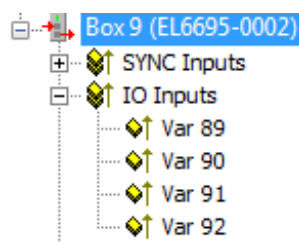


附图 136: IO 输入的上下文菜单：Adding new variables（添加新变量）



附图 137: 用于添加新变量的对话框（此处为 4 x BYTE），名称以 Var89 开头

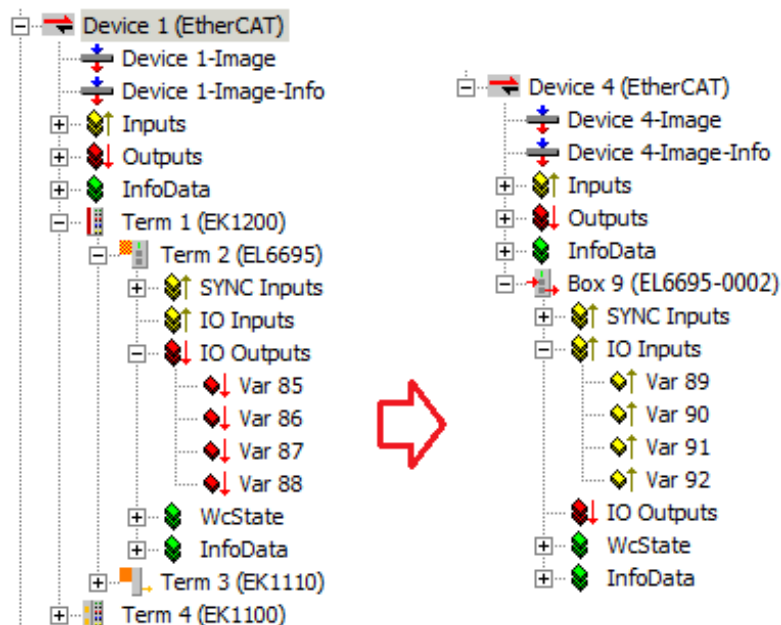
现在，在此对话框中可以为变量分配名称，并从多种可用的数据类型中进行选择。通过 **Multiple（多个）** 选择框可以同时声明多个相同类型的变量，并可以设置定义的起始地址。确认上述对话框后，将显示如下所示的过程映像。



附图 138: EL6695 副边新添加的变量

变量名称会自动递增，如下所示。输出变量也是这样创建。

现在应在原边（即 EL6695 的“远程侧”）创建四个相同类型的合适输出变量，用于数据交换。这些变量的名称不必相同，即它们可以称为 Var85 至 Var88。



附图 139: 创建过程数据

一旦也在 EL6695 原边的输出中添加 4 x BYTE 变量，则当这些变量被写入后（例如通过 PLC 程序），所有值将通过四个输入变量连续输出。例如，如果将一个值写入变量 Var86，则该值会被写入 EL6695 远程侧映射中顺序相应的输出变量 Var90（只要端子模块处于 OP 状态）。

两个控制器 PLC1 到 PLC2 的传输方向

通常，该端子模块将用于在两个控制器之间传输数据。由于桥接端子模块将输出映射为输入、将输入映射为输出，因此产生以下传输方式：

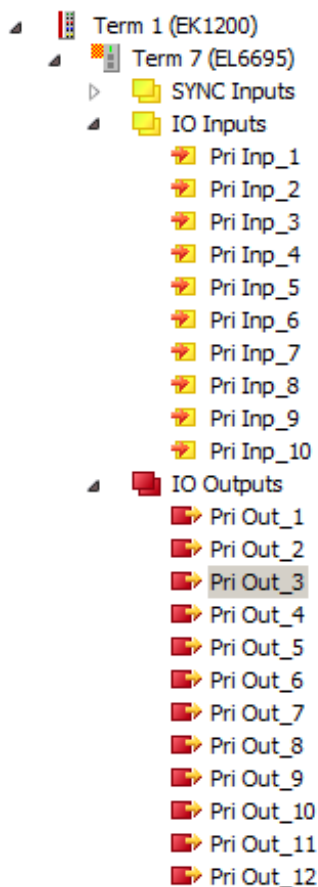
- 输出 (PLC1) → 输出 EL6695 ⇒ 输入 EL6695-0002 远程侧 → 输入 (PLC2)
- 输入 (PLC1) → 输入 EL6695 ⇒ 输出 EL6695-0002 远程侧 → 输出 (PLC2)

在每种情况下，EL6695 桥接端子模块均按字节顺序将输出值“映射”为输入值，反之亦然（无需其他配置或设置）。因此，所使用的变量的数据类型在每一侧都必须相同。

通过配置接口/扩展插件进行自动配置：

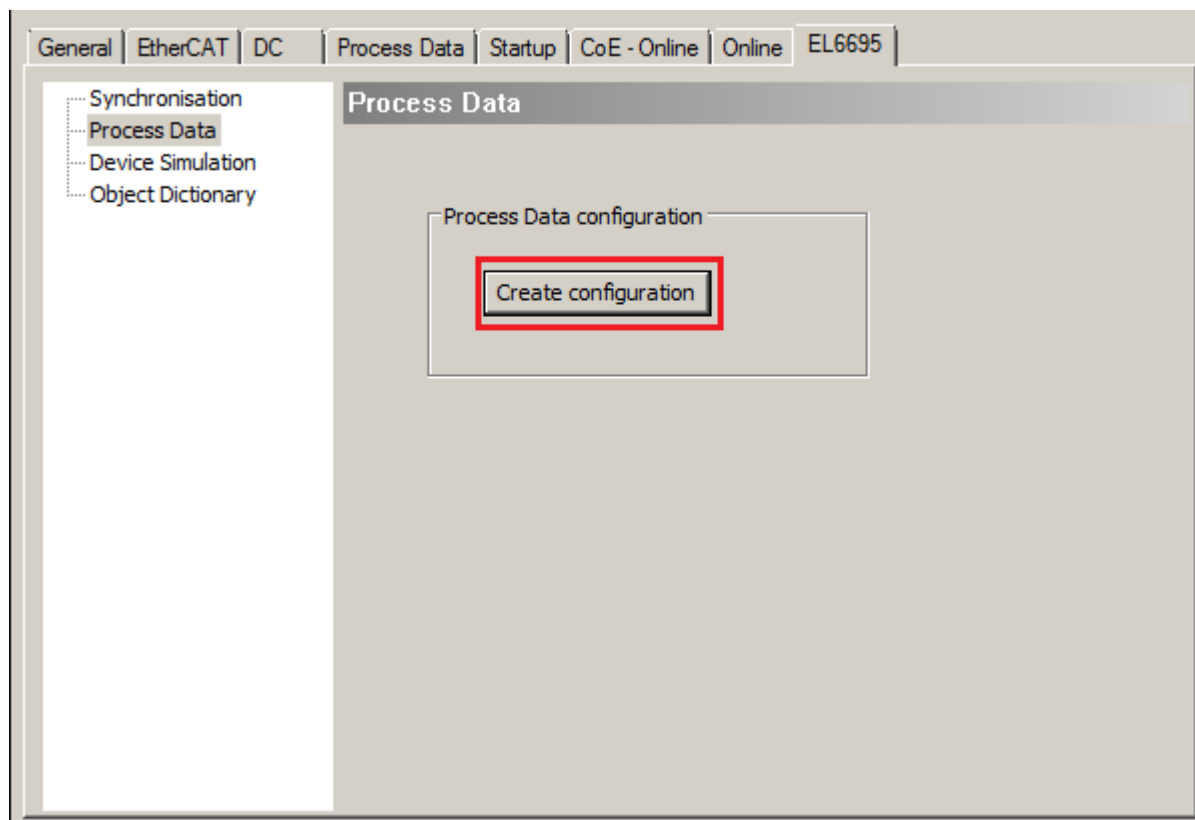
EL6695 扩展插件提供 Create Configuration（创建配置）按钮，可以在另一侧在线读取这一侧已经创建的变量，从而无需单独创建这些变量。显然，建议在过程映像较大时使用此功能，并且该功能仅通过 [EL6695] 的插件提供。此外，EL6695 必须处于就绪状态，并且可从两侧“在线”访问。下文将描述 TwinCAT 3 的操作步骤，在原边创建的变量自动镜像到副边：

- 前提条件：TwinCAT 处于“Free Run” / “配置模式”（请参见右下角符号，），EL6695 存在于原边的“设备”（端子模块）下方
- **A)** 在原边创建变量（例如 10 个输入变量和 12 个输出变量）：



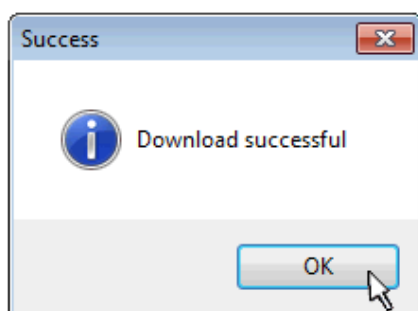
附图 140: EL6695 原边的示例应用变量

- **B)** 在原边选择该端子模块；按下 [EL6695] 选项卡中“过程数据”下的 [Create configuration]（创建配置）按钮后，将为另一侧创建一组“镜像”变量，在本示例中由端子模块在内部为副边创建；因此这些变量尚不可见

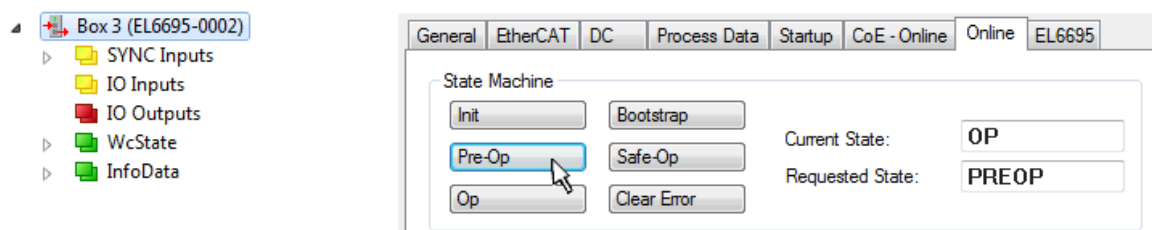


附图 141: EL6695 选项卡中“Process Data（过程数据）”下的 [Create configuration]（创建配置）

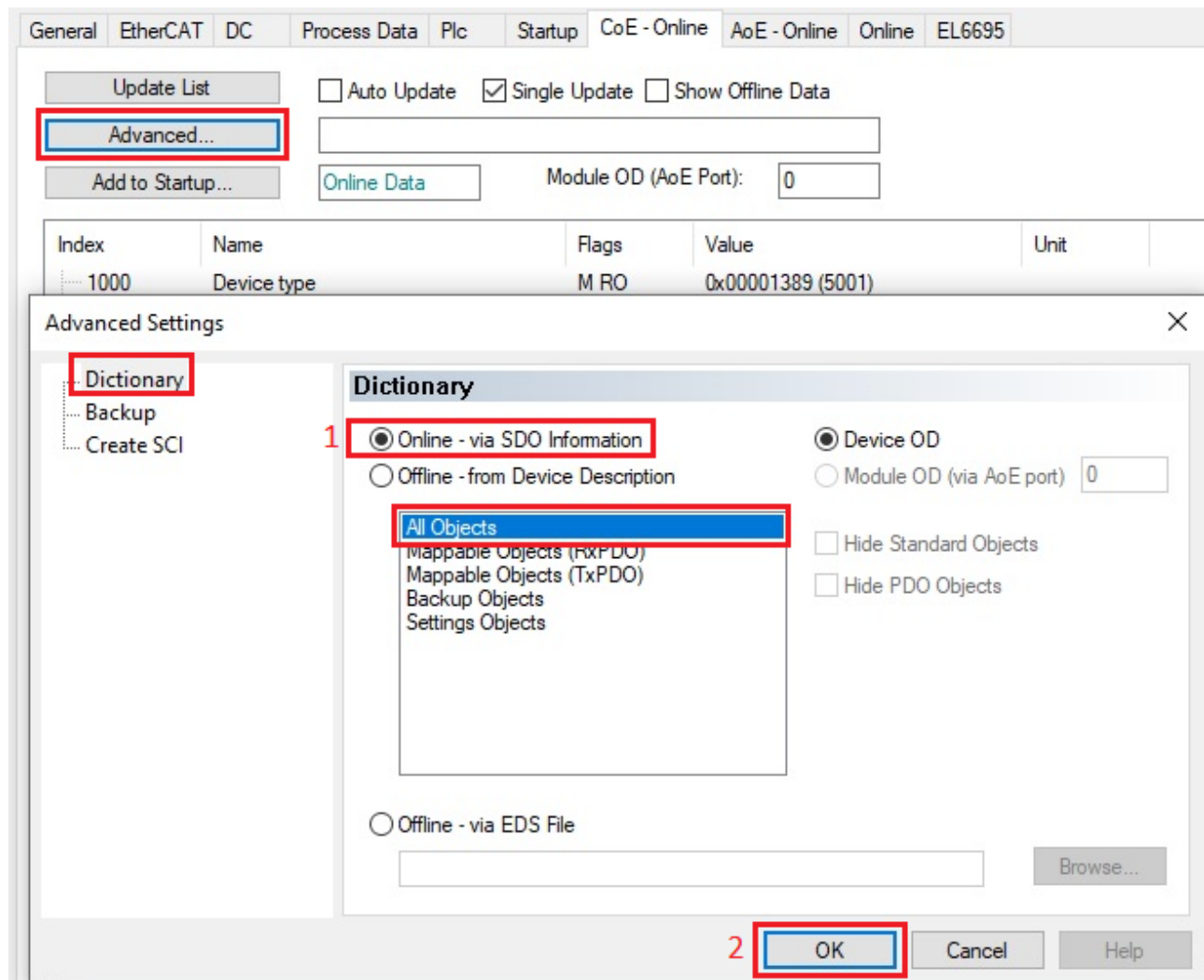
- C) 此操作的成功执行应分别进行确认：



- D) 现在必须将原边通过 INIT 设置为 PreOp 和 OP。
- E) 后续步骤在副边完成。为了使端子模块通过内部初始化接管变量数据，必须在副边的设备上通过扫描过程重新读取该端子模块。它可能处于 Error PreOp 状态，因为原边和副边的 PDO 不匹配。务必手动将其设置为 PreOp 状态（而非 OP 状态），以避免发送 Startup List。因此，不得激活 FreeRun（自由运行）模式。

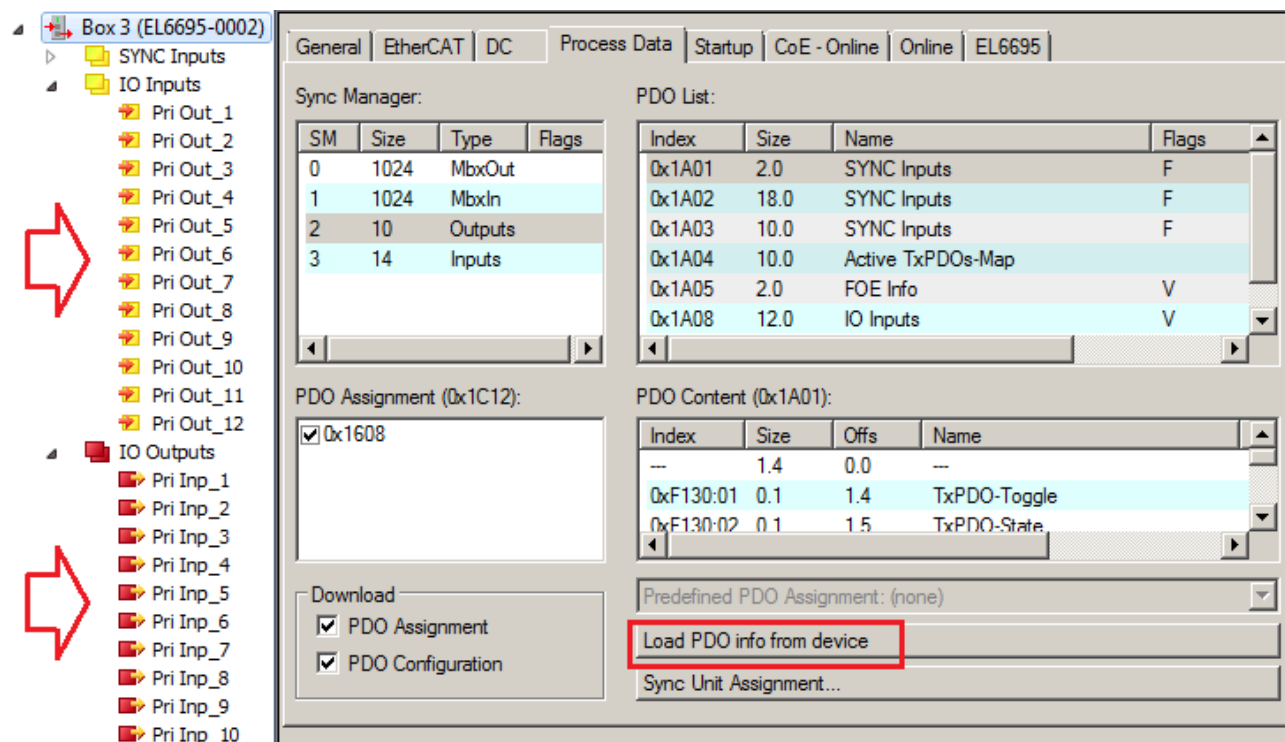


- F) 首先，为了进行后续操作，必须将对象字典结构从端子模块传输到 TwinCAT 中。这可以通过 [CoE - Online]（CoE - 在线）选项卡中的“Advanced...（高级…）”选项来完成 并按图示执行以下步骤（1 和 2）：



附图 142: 从设备加载对象字典结构

- **G** 在 [Process Data] (过程数据) 选项卡下, 使用 [Load PDO info from device] (从设备加载 PDO 信息) 从另一侧创建“镜像”变量配置, 随后该配置在“解决方案资源管理器” (左侧) 中可见:



附图 143: 通过 “Load PDO info from device（从设备加载 PDO 信息）” 将原边的变量镜像适配传输到副边

- **H)** 如有必要，使用 TwinCAT 菜单中的 “Reload Devices（重新加载设备）”（）重新加载配置，并激活 FreeRun（自由运行）模式。

根据此处描述的操作步骤，如果副边存在一组 I/O 变量（即 PDO 配置），则也可以为原边创建变量。

限制变量

此操作步骤应使用以下 PDO/变量类型：
Byte、UINT、UDINT

无 TwinCAT 环境下的使用

如果另一侧没有配备 [Load PDO info from device]（从设备加载 PDO 信息）功能的 TwinCAT EtherCAT 主站，则必须在该侧复制 PDO 上传机制：

- 读取 PDO 分配对象 0x1C12、0x1C13
- 然后读出 PDO 映射对象 0x160x（输出）、0x1A0x（输入），以及
- 相应地读出 PDO 0x60xx、0x70xx

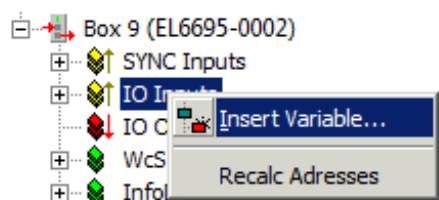
● 如果无法实施或不打算采用扫描过程

I 为了在副边创建变量，在执行 [Create configuration]（创建配置）的步骤 **B)** 和 **C)** 之后，必须将原边的端子模块手动设置为 INIT 状态，然后再次设置为 OP 状态（“Online（在线）”选项卡：点击 Init，然后依次点击：Safe-Op、Pre-Op、Op）。然后，在副边可以改为插入相应的 EL6695 变体（-0002）。此外，其余流程步骤与上述 **E)** 之后的描述相同。

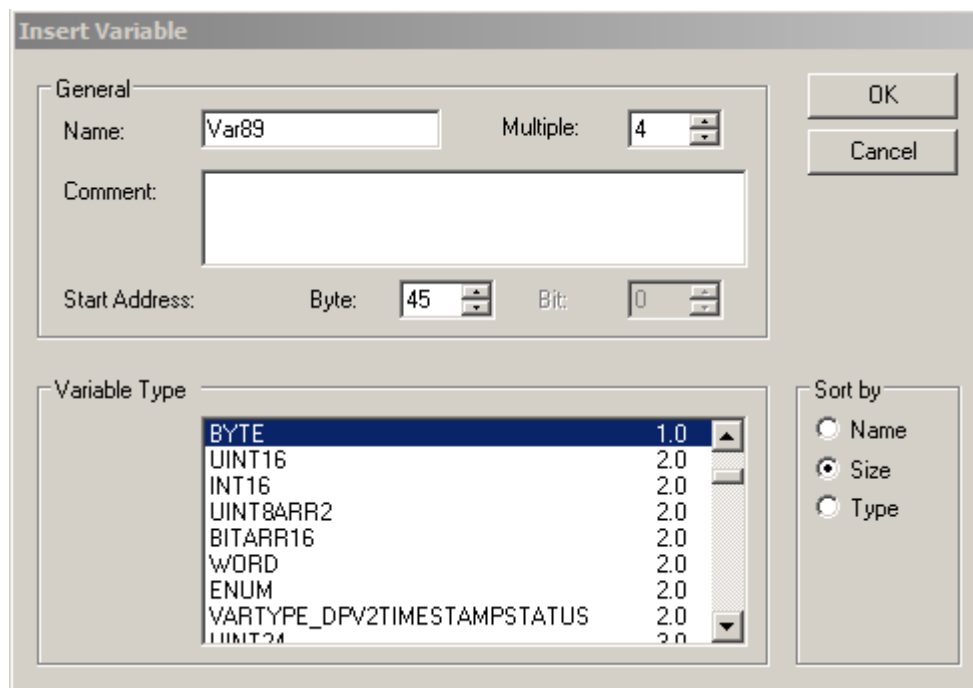
6.4.2.1 创建通用过程数据/简单数据交换（对称 PDO 映射）.

程序：

右键单击 EL6695 目录树中的 **IO Inputs（IO 输入）** 或 **IO Outputs（IO 输出）**，以显示上下文菜单。选择 **Insert Variable...（插入变量...）**（TwinCAT 2）或 **Add New Item（添加新项目）**（TwinCAT 3），以创建新变量/过程数据（请参见下图）。

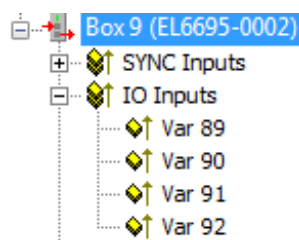


附图 144: IO 输入的上下文菜单：插入新变量



附图 145: 用于添加新变量的对话框（此处为 4 x BYTE），名称以 Var89 开头

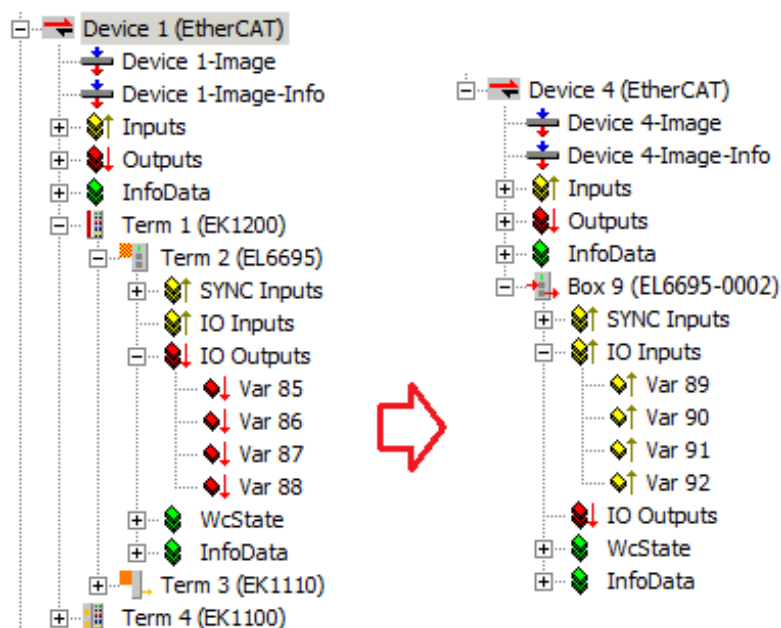
现在，在此对话框中可以为变量分配名称，并从多种可用的数据类型中进行选择。通过 **Multiple（多个）** 选择框可以同时声明多个相同类型的变量，并可以设置定义的起始地址。确认上述对话框后，将显示如下所示的过程映像。



附图 146: EL6695 副边新添加的变量

变量名称会自动递增，如下所示。输出变量也是这样创建。

现在应在原边（即 EL6695 的“远程侧”）创建四个相同类型的合适输出变量，用于数据交换。这些变量的名称不必相同，即它们可以称为 Var85 至 Var88。



附图 147: 创建过程数据

如果也在 EL6695 原边的输出中添加 4 x BYTE 变量，则当这些变量被写入后（例如通过 PLC 程序），所有值将通过四个输入变量连续输出。例如，如果将一个值写入变量 Var86，则该值会被写入 EL6695 远程侧映射中相应的输出变量 Var90（只要端子模块处于 OP 状态）。

从两个控制器 PLC1 到 PLC2 的传输方向

桥接端子模块通常用于在两个控制器之间传输数据。由于桥接端子模块将输出映射为输入、将输入映射为输出，因此产生以下传输方式：

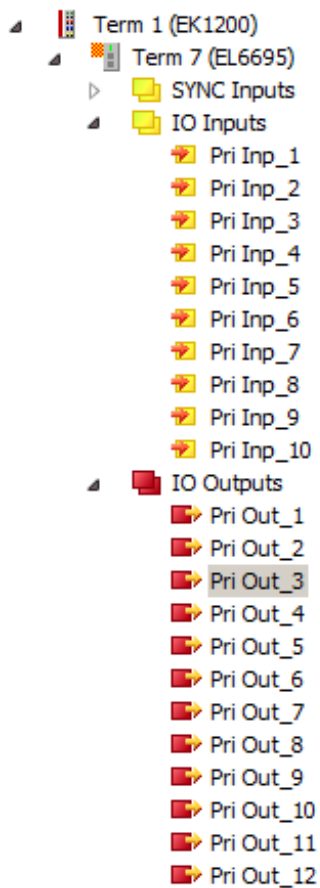
- 输出（PLC1）→ 输出 EL6695 ⇒ 输入 EL6695-0002 远程侧 → 输入（PLC2）
- 输入（PLC1）→ 输入 EL6695 ⇒ 输出 EL6695-0002 远程侧 → 输出（PLC2）

EL6695 桥接端子模块按字节顺序将输出值“映射”为输入值，反之亦然（无需任何附加配置或设置）。因此，每一侧的数据类型必须相同。

6.4.2.2 通过配置接口/扩展插件进行自动配置

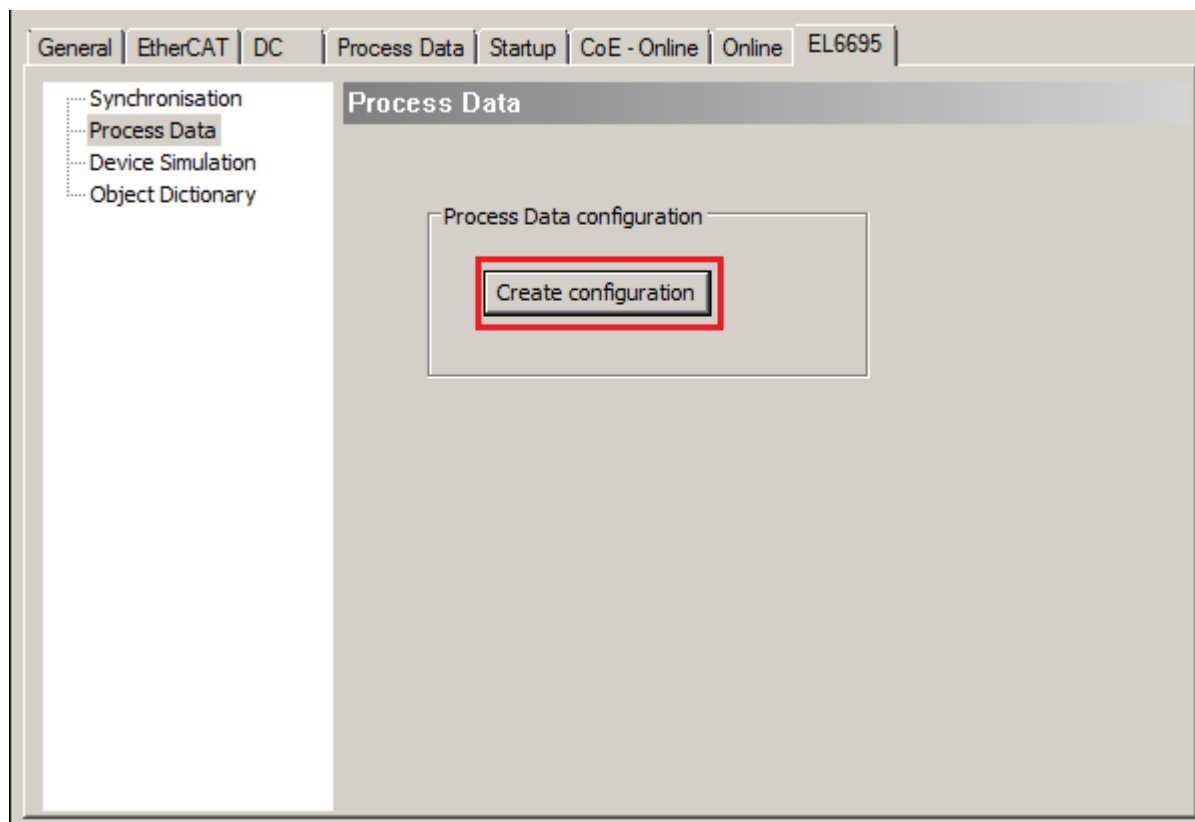
带有 [Create configuration]（创建配置）按钮的 EL6695 扩展插件可使创建的变量在另一侧在线可读，从而无需在另一侧显式创建这些变量。建议使用此功能来映射全面的过程映像，并且该功能仅通过 [EL6695] 插件提供。此外，EL6695 必须从两侧均处于就绪状态，即“在线”可访问。以下展示了 TwinCAT 3 的操作步骤，在原边创建的变量自动“镜像”到副边：

- 前提条件：TwinCAT 处于“Free Run” / “配置模式”（请参见右下角符号 、），“Device”（term）下方有一个 EL6695 的原边。
- A) 已在原边创建变量（例如 10 个输入变量和 12 个输出变量）：



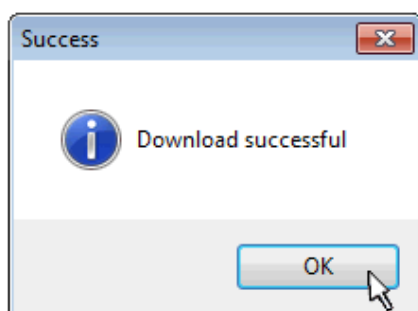
附图 148: EL6695 原边已创建变量的示例

- **B)** 在原边勾选该端子模块；通过 [EL6695] 选项卡中“Process Data（过程数据）”下的 [Create configuration]（创建配置）按钮，现在将为另一侧（在本示例中为副边）在端子模块内部创建一组“镜像”变量；因此该变量此时尚不可见

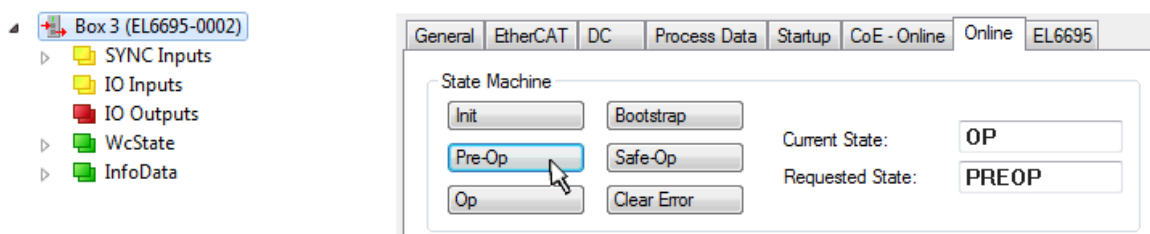


附图 149: EL6695 选项卡中“Process Data（过程数据）”下的 [Create configuration]（创建配置）

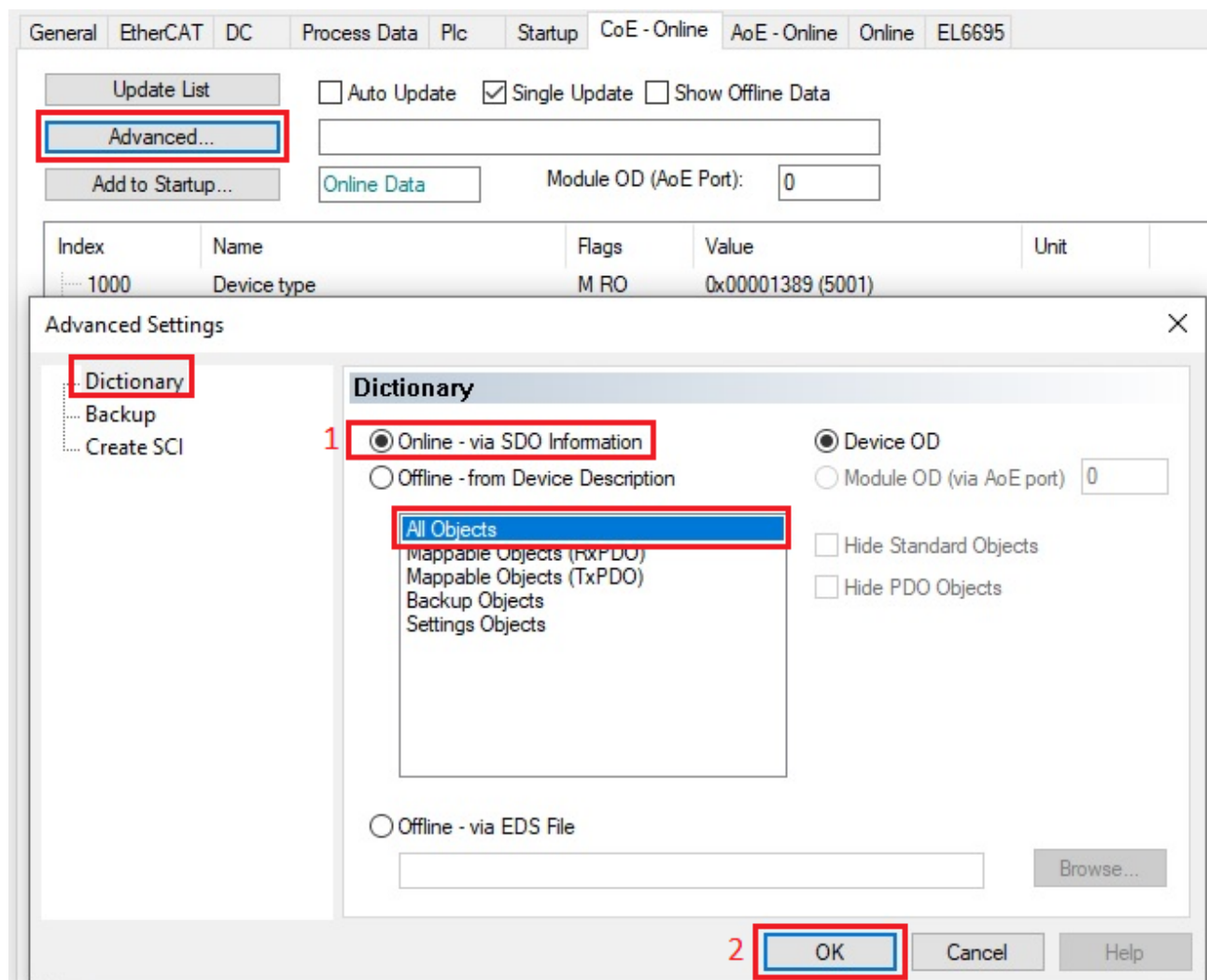
- C) 此操作成功完成后会相应地进行确认：



- D) 现在必须让原边执行一次从 INIT 到 PreOp 和 OP 的设置。
- E) 后续步骤在副边进行。为了使端子模块通过内部初始化接管变量数据，必须在副边的设备上通过扫描过程重新读取该端子模块。它可能处于 Error PreOp 状态，因为原边和副边的 PDO 不匹配。务必手动将其移至 PreOp 状态（而非 OP 状态），以避免发送 Startup List。因此，不得激活自由运行模式。

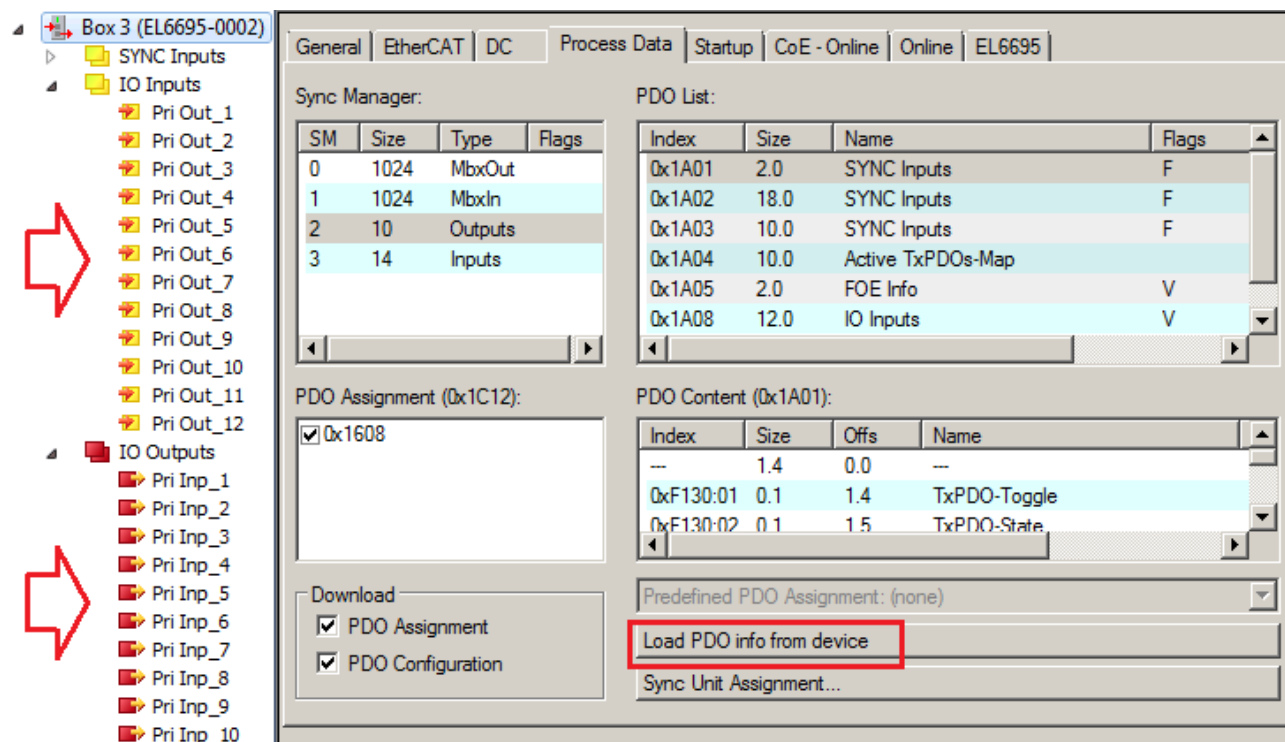


- F) 为了进行后续操作，必须先将端子模块的对象字典结构传输到 TwinCAT 中。这可以通过 [CoE - Online]（CoE - 在线）选项卡中的“Advanced...（高级…）”选项来完成 并按图示执行以下步骤（1 和 2）：



附图 150: 从端子模块加载对象字典结构

- **G** 在 [Process Data]（过程数据）选项卡下，使用 [Load PDO info from device]（从设备加载 PDO 信息）从另一侧生成“镜像”变量配置，随后该配置在“解决方案资源管理器”（左侧）中可见：



附图 151: 使用 “Load PDO info from device（从设备加载 PDO 信息）” 将变量以镜像形式从原边传输到副边

- **H)** 如有必要，使用 TwinCAT 菜单中的 “Reload Devices（重新加载设备）”（）重新加载配置，并启用自由运行模式

根据此处所示的过程，如果副边存在一组 I/O 变量（即 PDO 配置），则也可以为原边自动生成变量。

变量限制

此操作步骤应使用以下 PDO/变量类型：

Byte、UINT、UDINT

无 TwinCAT 环境下的使用

如果另一侧没有配备 [Load PDO info from device]（从设备加载 PDO 信息）功能的 TwinCAT EtherCAT 主站，则必须在该侧复制 PDO 上传机制：

- 读取 PDO 分配对象 0x1C12、0x1C13
- 然后读取 PDO 映射对象 0x160x（输出）、0x1A0x（输入），以及
- 读取 PDO 0x60xx、0x70xx

● 如果无法实施或不打算进行扫描

i 为了在副边不进行扫描的情况下生成变量，在执行 [Create configuration]（创建配置）的步骤 **B)** 和 **C)** 之后，必须将原边的端子模块手动设置为 INIT 状态，然后再设置回到 OP 状态（“Online（在线）”选项卡：点击 Init，然后依次点击：Safe-Op、Pre-Op、OP）。然后，在副边可以插入相应的 EL6695 变体（-0002），以代替使用 “add Item（添加项目）” 进行扫描。否则，其余流程步骤与上述 **E)** 之后的描述相同。

● 使用示例程序

i 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。

→ 示例配置：<https://infosys.beckhoff.com/content/1033/el6695/Resources/3521535883.zip>

6.4.3 选择性 PDO 映射

选择性 PDO 映射

用户可以通过选择性 PDO 映射配置自定义的周期性数据交换。为此，在端子模块的一侧创建一套“完整”的过程数据。现在可以在另一侧定义该集合的一个子集，用于周期性读取。此过程被称为选择性 PDO 映射。

默认的 PDO 分配和映射会被移除，用户必须创建自定义的 PDO 分配和映射。为此，了解 PDO 分配或映射的结构非常重要。

基本上（标准情况下），有两个 CoE 对象 0x1C12 “Rx PDO Assign”和 0x1C13 “Tx PDO Assign”，其中定义了副边和原边的数据输入（RxPDO）和数据输出（TxPDO）的分配。它们包含对 CoE 对象 0x1608 “IO RxPDO 映射”和 0x1A08 “IO TxPDO 映射”的引用，这两个对象中定义了输入和输出的映射。所有这四个 CoE 对象（0x1C12/0x1C13 和 0x1608/0x1A08）都具有 RW 标志（RW = 读取/写入）。两个映射对象 0x1608 和 0x1A08 初始引用 CoE 对象 0x7000 “PD 输出”和 0x6000 “PD 输入”（过程数据输入/输出），其中存储了过程数据（结构）。“初始”指的是当用户在原边或副边添加新的输入或输出数据时，TwinCAT 进行的自动配置。这也可以在 [Startup]（启动）选项卡中查看。所有对象引用的链接都是在这里创建的，在从 PREOP 转换到 SAFEOP（P → S）的过程中。两个“终极”CoE 对象 0x6000 和 0x7000 为“RO”（只读），因为在设备内部，输入和输出数据的复制过程是在 OP 状态下执行的。

Index	Name	Flags	Value
1000	Device type	RO	0x00001389 (5001)
1008	Device name	RO	EL6695 <SECONDARY>
1009	Hardware version	RO	00
100A	Software version	RO	02
1018:0	Identity	RO	> 4 <
10F4:0	External synchronization status	RO	> 18 <
1608:0	IO RxPDO-Map 8	RW	> 1 <
1608:01	SubIndex 001	RW	0x7000:01, 8
1801:0	TxPDO-Par External Sync Compact	RO	> 6 <
1802:0	TxPDO-Par External Sync	RO	> 6 <
1803:0	TxPDO-Par External Sync(32 Bit)	RO	> 6 <
1A01:0	TxPDO-Map External Sync Compact	RO	> 5 <
1A02:0	TxPDO-Map External Sync	RO	> 8 <
1A03:0	TxPDO-Map External Sync(32 Bit)	RO	> 8 <
1A04:0	Active TxPDOs-Map	RW	> 5 <
1A05:0	FOE Info-Map	RW	> 1 <
1C00:0	Sync manager type	RO	> 4 <
1C12:0	RxPDO assign	RW	> 1 <
1C12:01	SubIndex 001	RW	0x1608 (5640)
1C13:0	TxPDO assign	RW	> 2 <
1C32:0	SM output parameter	RO	> 32 <
1C33:0	SM input parameter	RO	> 32 <
7000:0	PD Outputs	RO	> 1 <
7000:01	SubIndex 001	RO P	0x00 (0)
F000:0	Modular device profile	RO	> 2 <

附图 152: 通过 0x1C12 将 1 字节变量基本映射到 0x7000 “PD 输出”

（初始）链接对象概览：

- **RxPDO 分配：** 0x1C12 → RxPDO 映射 0x1608 → 0x7000 PD 输出
- **TxPDO 分配：** 0x1C13 → TxPDO 映射 0x1A08 → 0x6000 PD 输入

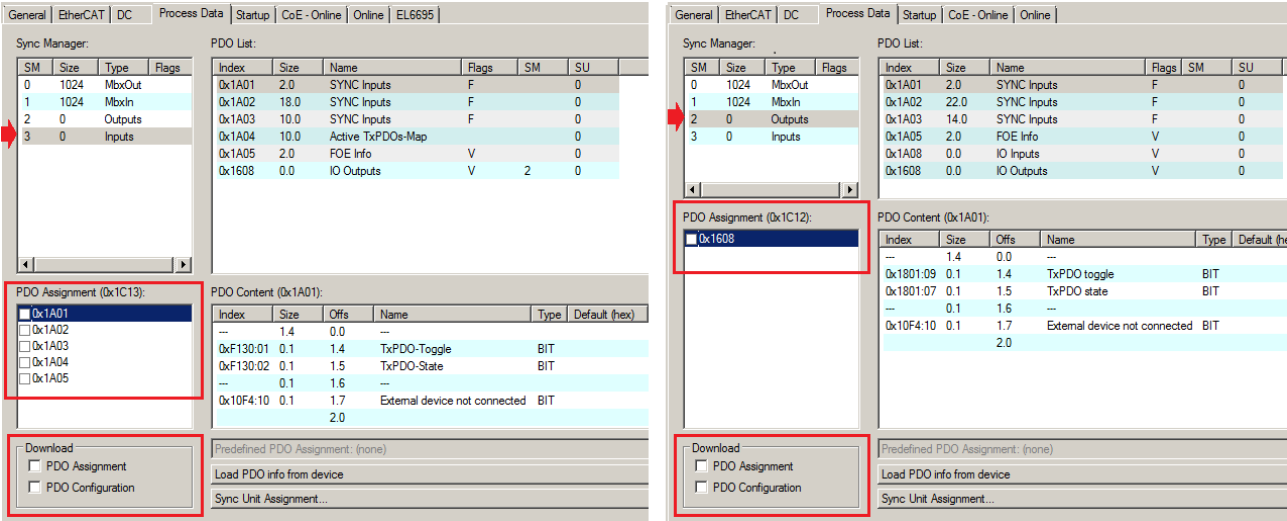
CoE 对象映射

i 0x1608 指向 0x7000，0x1A08 指向 0x6000。对象 0x7000 和 0x6000 分别镜像到端子模块的另一侧。

例如，如果在原边存在对象 0x7000:0E PD 输出，则应在副边创建一个对象 0x6000:0E PD 输入，用于数据交换。

TwinCAT 下的操作步骤

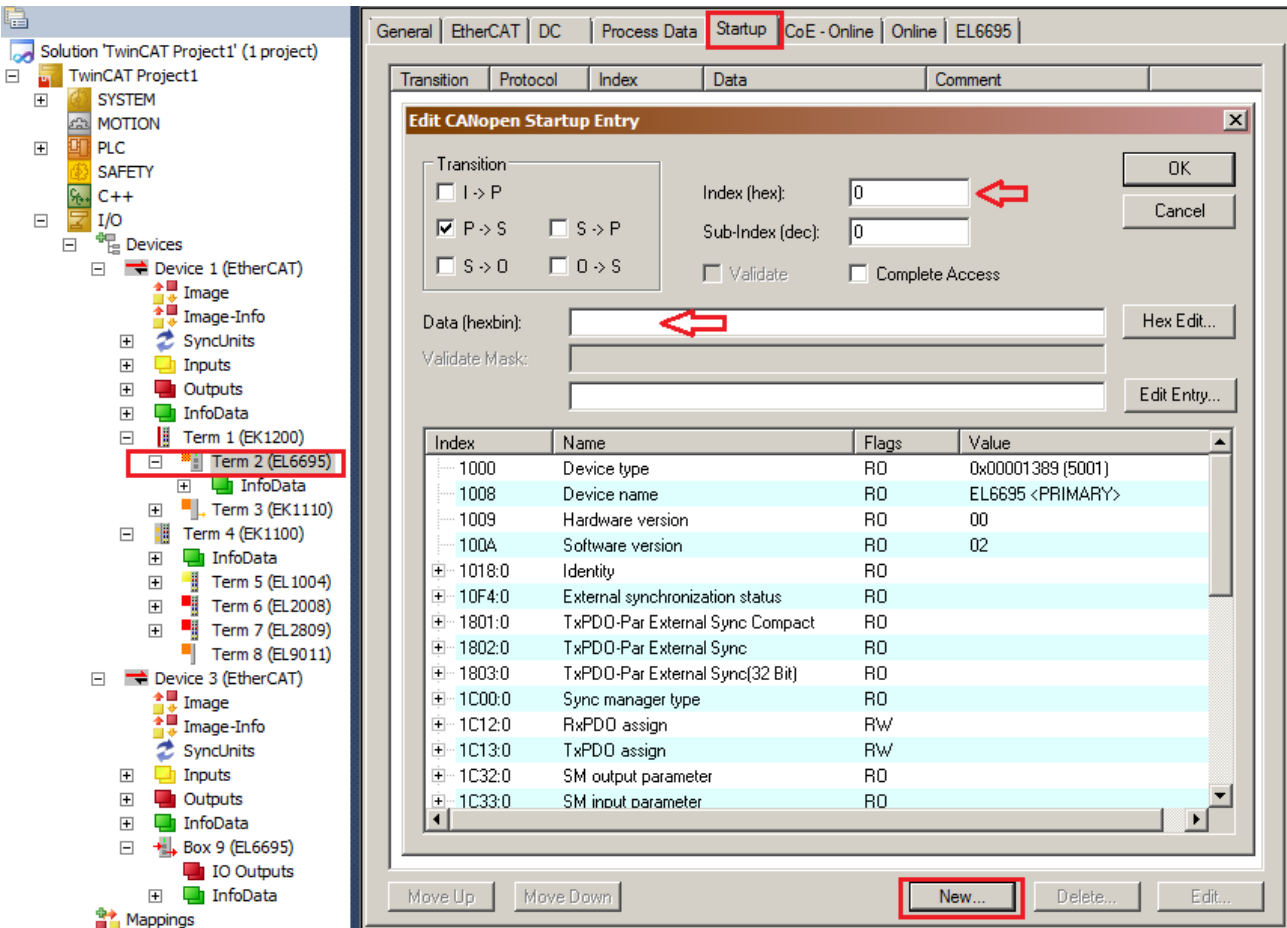
要创建选择性映射，应首先取消勾选端子模块“Process Data（过程数据）”选项卡中 outputs, inputs 和 downloads 下的所有选项，以禁用现有的 PDO 配置及其分配。



附图 153: 选择性映射中的过程数据

然后，“Startup”选项卡中的现有条目也会被删除，端子模块不再具有过程映像。然后，按如下所述分两步重新定义过程映像：

务必在端子模块的“Startup”选项卡中创建新条目。选择该选项卡右下角的“New...（新建…）”。



附图 154: 创建新的输出变量（在本示例中：EL6695 的原边）

作为示例，总共创建四个输出变量，大小分别为 8、16、32 和 8 [建议顺序]：

为此，在本示例中按如下方式创建 0x1614 对象（可从 0x1600 到 0x161F 范围内选择）（例如，勾选“Complete Access（完全访问）”；输入“IO RxPDO Map（IO RxPDO 映射）”作为注释）：

Edit CANopen Startup Entry

Transition:
☐ I -> P
☒ P -> S ☐ S -> P
☐ S -> O ☐ O -> S

Index (hex): 1614
 Sub-Index (dec): 0
☐ Validate ☒ Complete Access

Data (hexbin): 04 00 08 01 00 70 10 02 00 70 20 03 00 70 08 04 00 70
 Validate Mask:
 Comment: RxPDO-Map

Index	Name	Flags	Value
1000	Device type	RO	0x00001389 (5001)
1008	Device name	RO	EL6695 <PRIMARY>
1009	Hardware version	RO	00
100A	Software version	RO	02
+ 1018:0	Identity	RO	
+ 10F4:0	External synchronization status	RO	
+ 1801:0	TxPDO-Par External Sync Compact	RO	
+ 1802:0	TxPDO-Par External Sync	RO	
+ 1803:0	TxPDO-Par External Sync(32 Bit)	RO	
+ 1C00:0	Sync manager type	RO	
+ 1C12:0	RxPDO assign	RW	
+ 1C13:0	TxPDO assign	RW	
+ 1C32:0	SM output parameter	RO	
+ 1C33:0	SM input parameter	RO	

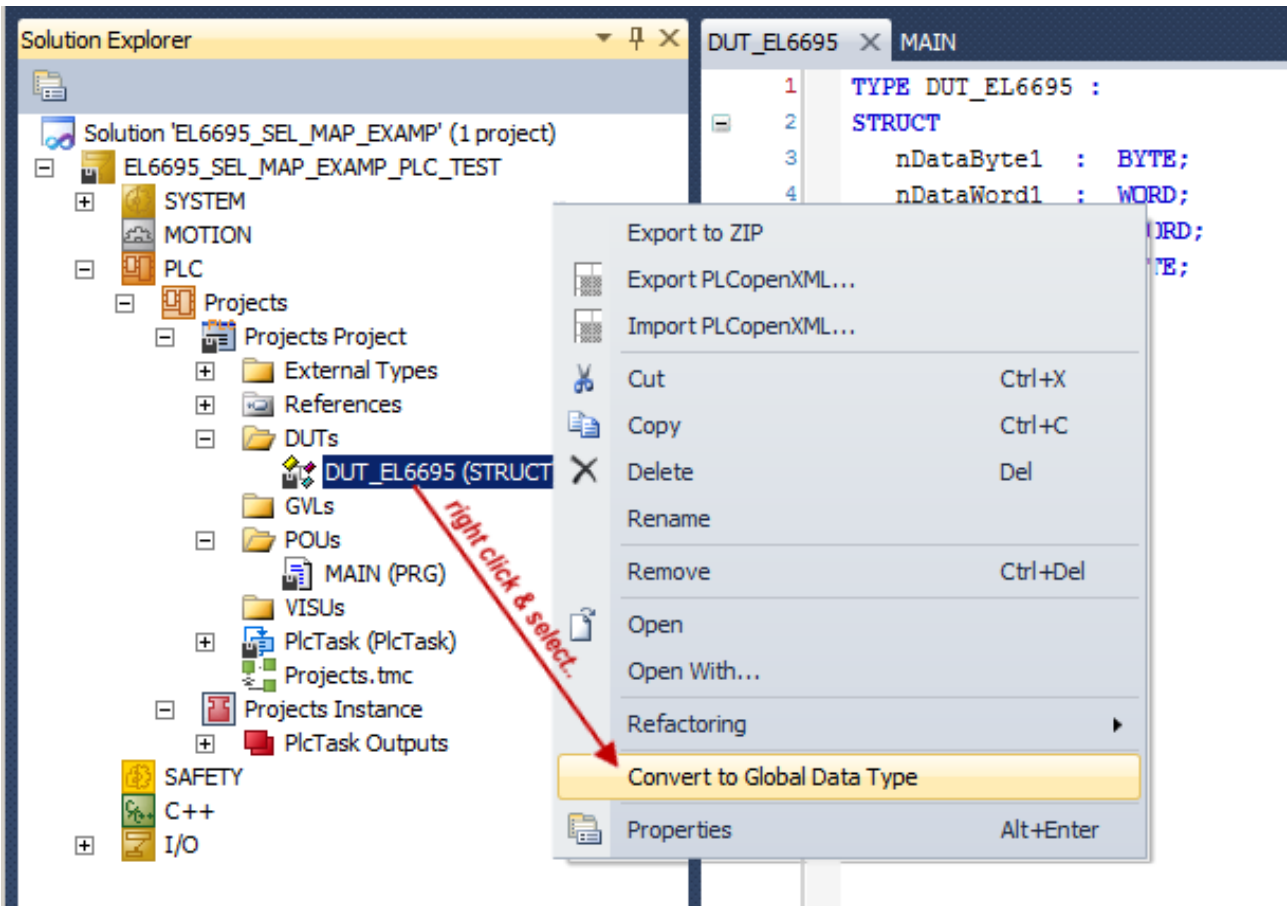
附图 155: 设置 0x1614 对象

- 输入: **04 00 08 01 00 70 10 02 00 70 20 03 00 70 08 04 00 70**

条目数据的含义:

04 00	子索引号
08 01 00 70	对象 7000, 索引 1, 长度 8
10 02 00 70	对象 7000, 索引 2, 长度 16
20 03 00 70	对象 7000, 索引 3, 长度 32
08 04 00 70	对象 7000, 索引 4, 长度 8

例如, 这可以表示程序对象或结构体的数据构成。为了能够将结构体变量与 EL6695 的 PDO 链接, 首先必须将其转换为 TC3 下的全局数据类型:



附图 156: 转换为全局数据类型

第二步，通过“Edit CANopen Startup Entry（编辑 CANopen 启动条目）”对话框进行以下条目设置：

Edit CANopen Startup Entry

Transition:
☐ I -> P
☒ P -> S ☐ S -> P
☐ S -> O ☐ O -> S

Index (hex):
 Sub-Index (dec):
☐ Validate ☒ Complete Access

Data (hexbin): Hex Edit...
 Validate Mask:
 Comment: Edit Entry...

Index	Name	Flags	Value
1000	Device type	M RO	0x00001389 (5001)
1008	Device name	RO	EL6695-0000
1009	Hardware version	RO	
100A	Software version	RO	
+ 1018:0	Identity	RO	> 4 <
+ 10F4:0	External synchronization status	RO	> 24 <
+ 1801:0	TxPDO-Par External Sync Compact	RO	> 6 <
+ 1802:0	TxPDO-Par External Sync	RO	> 6 <
+ 1803:0	TxPDO-Par External Sync(32 Bit)	RO	> 6 <
+ 1A01:0	TxPDO-Map External Sync Compact	RO	> 5 <
+ 1A02:0	TxPDO-Map External Sync	RO	> 8 <
+ 1A03:0	TxPDO-Map External Sync(32 Bit)	RO	> 8 <
+ 1C00:0	Sync manager type	RO	> 4 <
+ 1C32:0	SM output parameter	RO	> 32 <

附图 157: 为 RxPDO 创建对象 0x1C12

因此，要创建的 CoE 启动条目是索引 0x1C12，子索引为 0（对象的起始子索引）。勾选“Complete Access（完全访问）”，以便输入的值不会被解释为 WORD、DWORD 等数据类型。分配过程将在从 PREOP 转换到 SAFEOP 期间进行；在此，仅保留（或应设置）勾选“P → S”。在“Comment（注释）”下可以添加说明性注释，该注释随后将在 CoE 概览中作为对象名称显示。在“Data (hexbin)（数据（十六进制二进制））”中指定 CoE 对象的内容。

前两个块定义对象的子索引数量（高/低字节交换：01 00 → 子索引数量 = 1）。随后是第一个子索引 01 的内容 14 16 → 0x1614。

- 输入：**01 00 14 16**

现在，这就是上述引用之一，在本示例中指向已创建的“RxPDO Map（RxPDO 映射）”对象（0x1614）。

现在，[Startup]（启动）选项卡应如下所示：

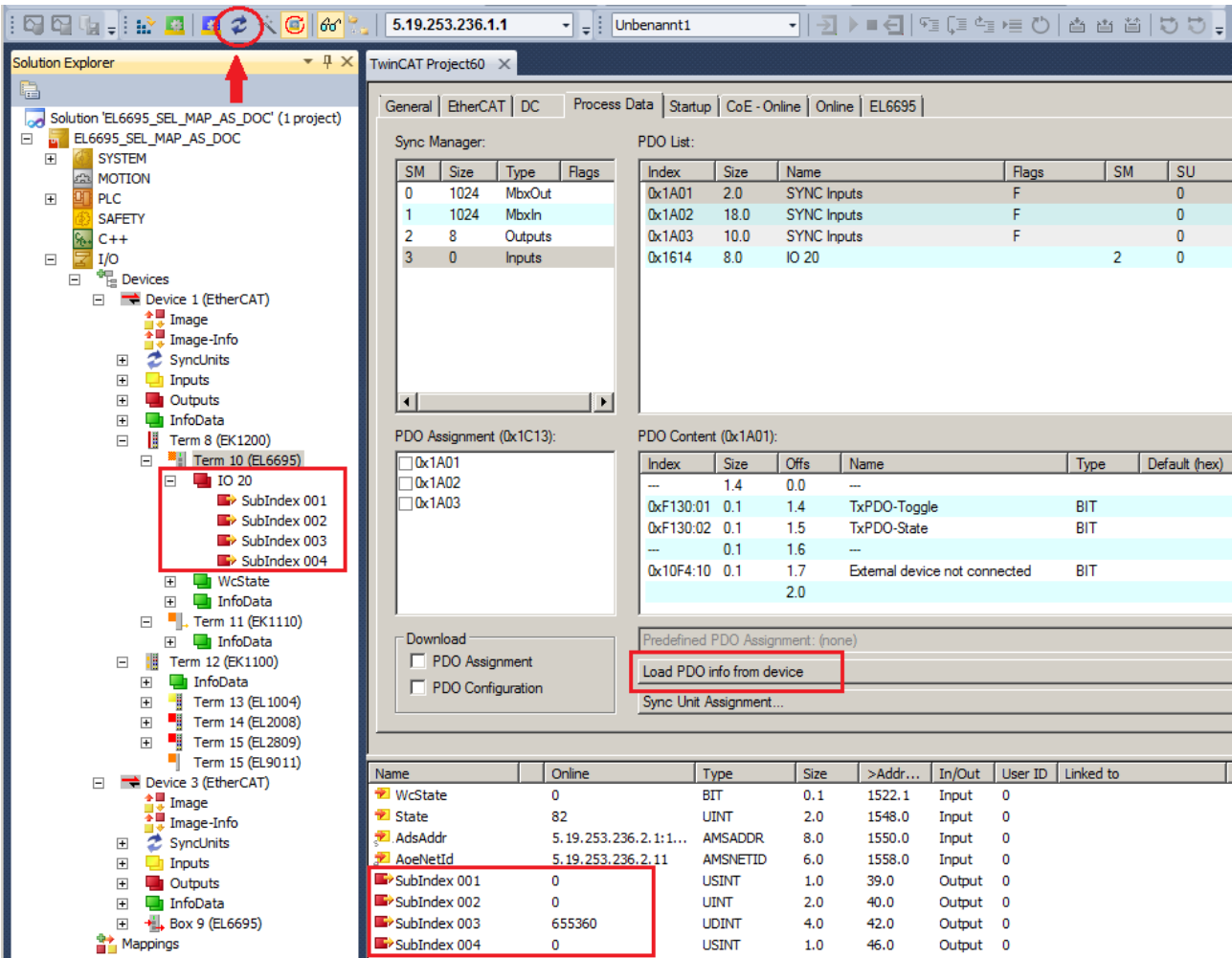
General EtherCAT DC Process Data Startup CoE - Online Online EL6695					
Transition	Protocol	Index	Data	Comment	
PS	CoE	0x1614 C 0	04 00 08 01 00 70 10 02 00 70 20 03 00 70 08 04 00 70	RxPDO-Map	
PS	CoE	0x1C12 C 0	01 00 14 16	RxPDO assign - SubIndex 000	

附图 158: 原边的新 PDO 映射和分配对象

重要的是，“RxPDO 分配”（通过对象 0x1C12）始终位于 RxPDO 映射条目列表的末尾。它可能包含多个引用（根据需要）。例如，一个数据集可能如下所示：06 00 03 16 1A 16 1E 16 08 16 04 16 .. 等。

使用以下步骤，在配置中显示已创建的变量：

1. 在菜单中选择: [TwinCAT] → [Reload Devices] (重新加载设备)
2. 在端子模块的“Process Data (过程数据)”选项卡中, 选择“Load PDO info from device (从设备加载 PDO 信息)”
3. 通过“Update List (更新列表)”显示 CoE 在线对象



附图 159: 新创建的过程数据: 对象 0x1608

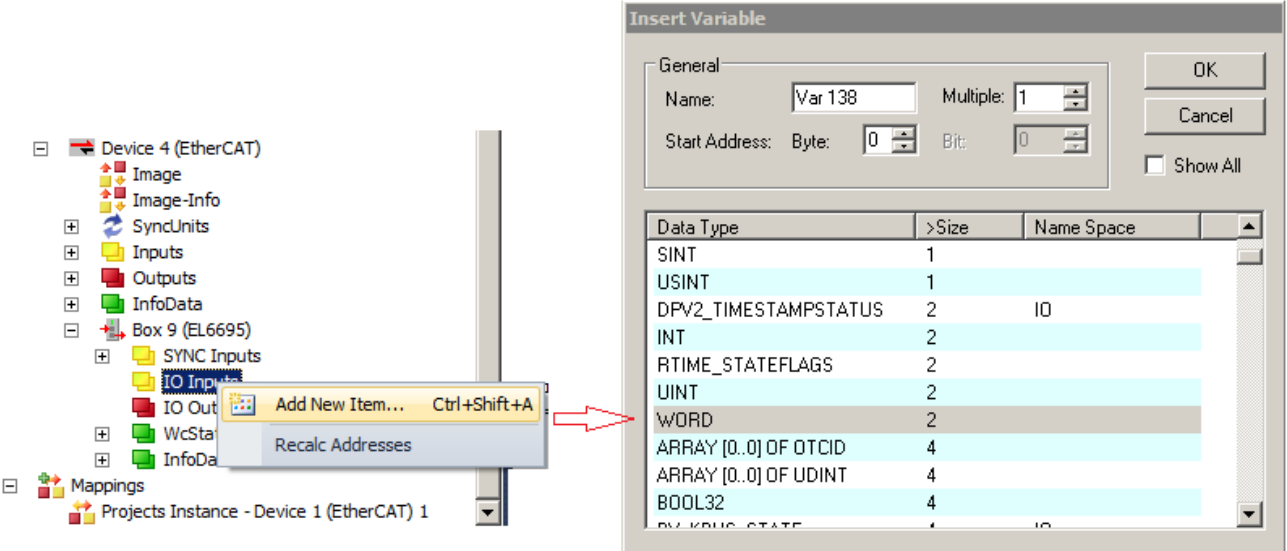
Index	Name	Flags	Value
+ 1803:0	TxPDO-Par External Sync(32 Bit)	RO	> 6 <
+ 1C00:0	Sync manager type	RO	> 4 <
- 1C12:0	RxPDO assign	RW	> 1 <
1C12:01	SubIndex 001	RW	0x1614 (5652)
+ 1C13:0	TxPDO assign	RW	> 0 <
+ 1C32:0	SM output parameter	RO	> 32 <
+ 1C33:0	SM input parameter	RO	> 32 <
+ F000:0	Modular device profile	RO	> 2 <
F008	Code word	RW	0x00000000 (0)
+ F010:0	Module list	RO	> 1 <
+ F130:0	TxPDO-Parameter	RO	> 2 <
+ F630:0	Active TxPdo Info	RO	> 4 <
+ F640:0	Remote Write Cycles	RO	> 3 <
+ F650:0	FOE Info	RO	> 2 <
+ F800:0	Device Config	RO	> 3 <
+ F820:0	ADS Server Settings	RO	> 2 <
+ F821:0	EL6695 ADS Settings	RO	> 2 <
+ FA20:0	Device Diag	RO	> 29 <
- 1614:0	IO RxPDO-Map 20	RW	> 4 <
1614:01	SubIndex 001	RW	0x7000:01, 8
1614:02	SubIndex 002	RW	0x7000:02, 16
1614:03	SubIndex 003	RW	0x7000:03, 32
1614:04	SubIndex 004	RW	0x7000:04, 8
+ 1A01:0	TxPDO-Map External Sync Compact	RO	> 5 <
+ 1A02:0	TxPDO-Map External Sync	RO	> 8 <
+ 1A03:0	TxPDO-Map External Sync(32 Bit)	RO	> 8 <
+ 1A04:0	Active TxPDOs-Map	RW	> 5 <
+ 1A05:0	FOE Info-Map	RW	> 1 <
- 7000:0	PD Outputs	RO	> 4 <
7000:01	SubIndex 001	RO P	0x00 (0)
7000:02	SubIndex 002	RO P	0x0000 (0)
7000:03	SubIndex 003	RO P	0x00000000 (0)
7000:04	SubIndex 004	RO P	0x00 (0)

附图 160: 在 [CoE Online] (CoE 在线) 中显示以引用的方式选择性创建的变量

创建选择性 PDO 变量

现在, 如果需要, 可以在 EL6695 桥接端子模块的另一侧 (在本示例中为副边) 镜像所选的“变量”。为此, 在“解决方案资源管理器”中添加所需的变量。

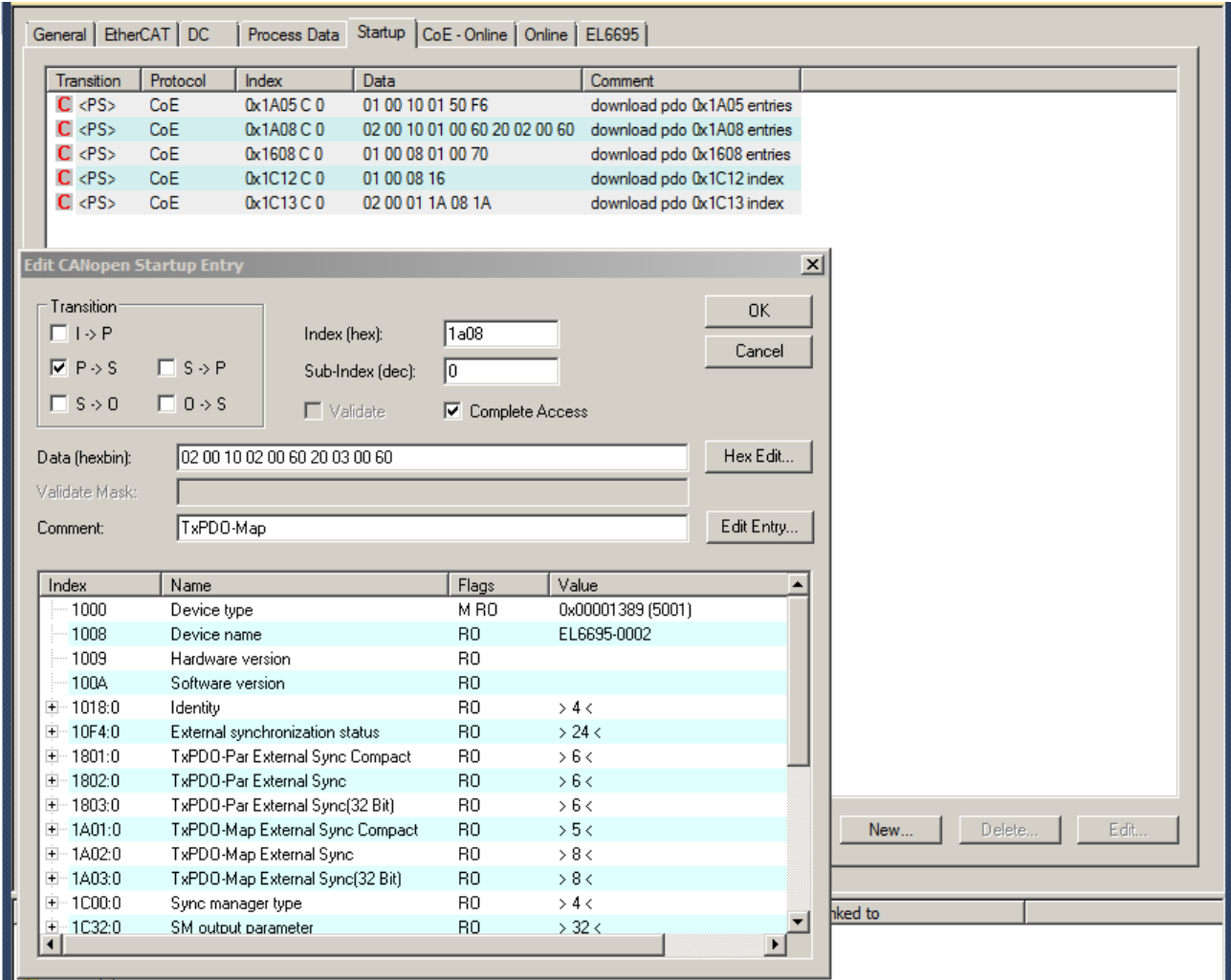
在本示例中, 原边具有 1-2-4-1 字节结构, 其中仅需将 2 字节和 4 字节变量类型“WORD”和“DWORD”作为整个过程数据的子集提供, 如下所示:



附图 161: 添加选择性 PDO 变量

DWORD 变量的操作步骤与此类似，如图所示。

然后，按如下方式完成启动条目设置，以适配 WORD 类型的子索引值 02 和 DWORD 类型的子索引值 03：



附图 162: 用于覆盖的启动条目 0x1A08（用于更正引用索引：01 → 02 和 02 → 03）

- 输入：02 00 10 02 00 60 20 03 00 60

条目数据的含义：

02 00	子索引号
10 02 00 60	对象 6000, 索引 2, 长度 16
20 03 00 60	对象 6000, 索引 3, 长度 32

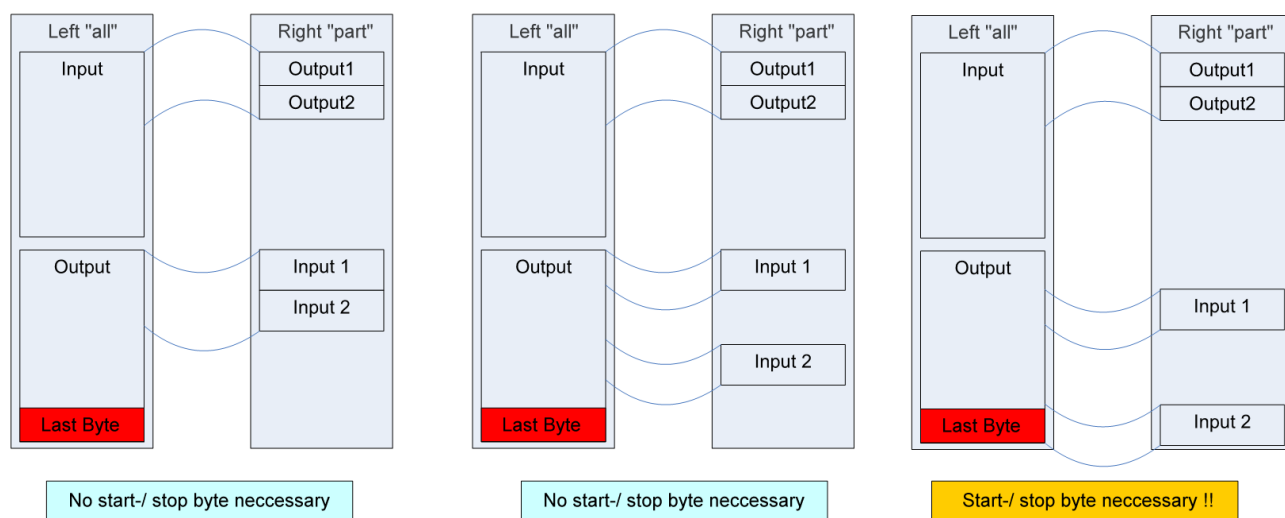
再次执行“Reload（重新加载）”指令并激活配置后，选择性 PDO 映射现在可供 PLC 程序使用。请注意，由于端子模块特定的系统特性，实时数据交换采用了一种特殊的缓冲机制，因此输出变量始终必须在开头和末尾各配置一个字节（即所谓的起始/停止字节）。这些“填充字节”变量不需要链接到 PLC 变量，因此也无需使用数据。

当 EL6695 用于其他 EtherCAT 主站时，对应的配置过程也要进行调整。

关于使用起始/停止字节的解释性说明

为了选择性映射输出数据，倍福通常建议在开头创建一个起始字节，在末尾创建一个停止字节。其技术原因在于过程数据的内部存储器管理，对最后一个字节的访问会触发内部切换到端子模块的下一个 EtherCAT 缓冲区；此时该输出数据包将不再支持读取访问。如果对输出缓冲区的读取访问是按“从上到下”选择性进行的，则没有问题。然而，如果变量是从不同部分依次读取的（大多数应用可能都是这种情况），并且要捕获最后一个字节，则缓冲区会被锁定，从而无法再获取更多数据。在实际过程数据之后添加一个“空”字节，可防止缓冲区意外切换。

该图说明，在许多情况下，起始/停止字节实际上并不是必需的：



附图 163: 使用起始/停止字节

● 插入起始和停止字节

为了避免在应用的整个使用寿命期间灵活使用 EL6695 以及进行任何配置更改时出现意外问题，倍福通常建议在输出变量的开头和末尾各配置一个字节。

● 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。

→ 选择性映射的示例程序，可按需索取：

<https://infosys.beckhoff.com/content/1033/el6695/Resources/1421824651.zip>

<https://infosys.beckhoff.com/content/1033/el6695/Resources/1421826315.zip>

6.4.4 FSoE 传输

FSoE 传输以周期性过程数据传输的方式实现，可通过两种不同的方式执行：

- 创建带安全模块/插槽的 ESI 并加载 EmulationMode（仿真模式），请参见该部分
- 配置到一个“External Device（外部设备）”的安全连接
- 注意：EL6695 不是需要认证许可的安全相关设备，而“仅仅”是传输链路中的一个元件。如果要对隐藏在 EL6695 “后面”的安全设备执行 ETG FSoE 测试，则 EL6695 的 PDO 配置当然应使 FSoE ETG 测试将 EL6695 识别为“安全类”设备。

如需更多信息，请联系倍福支持中心，并提供相关数据（TwinCAT 版本、所需性能值、周期、数据量等）

6.5 本地邮箱协议

由 EL6695 本地处理但不转发的非周期性数据。通常通过 EtherCAT 邮箱进行传输。

6.5.1 CoE - Can over EtherCAT

CoE - Can over EtherCAT

处理：本地处理，不转发

在 PLC 程序（ST）中，通过功能块 **FB_EcCoeSdoRead** 和 SDO（服务数据对象）访问，可以从 EtherCAT 从站的对象字典中读取数据。使用参数 `nSubIndex` 和 `nIndex` 选择要读取的对象。该块是（标准）库 `Tc2_EtherCAT` 的一部分（请参见 TwinCAT 3 Engineering → PLC → Libraries → TwinCAT 3 PLC Lib: `Tc2_EtherCAT` → CoE（TwinCAT 3 开发环境 → PLC → 库 → TwinCAT 3 PLC 库: `Tc2_EtherCAT` → CoE），网址：<http://infosys.beckhoff.com>）。



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。

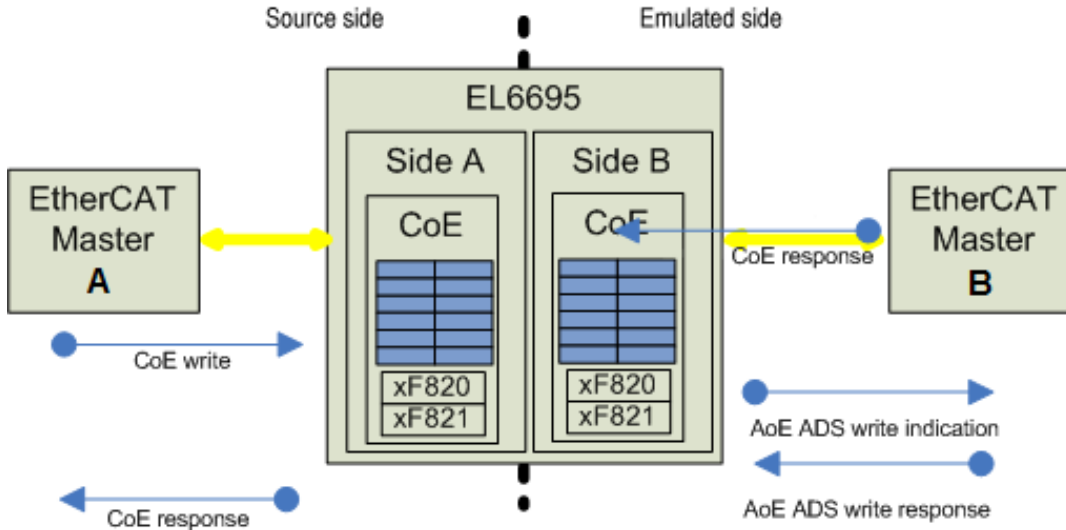
→ CoE 示例程序，可按需索取：

<https://infosys.beckhoff.com/content/1033/el6695/Resources/1421141259.zip>

6.6 具备传输功能的邮箱协议

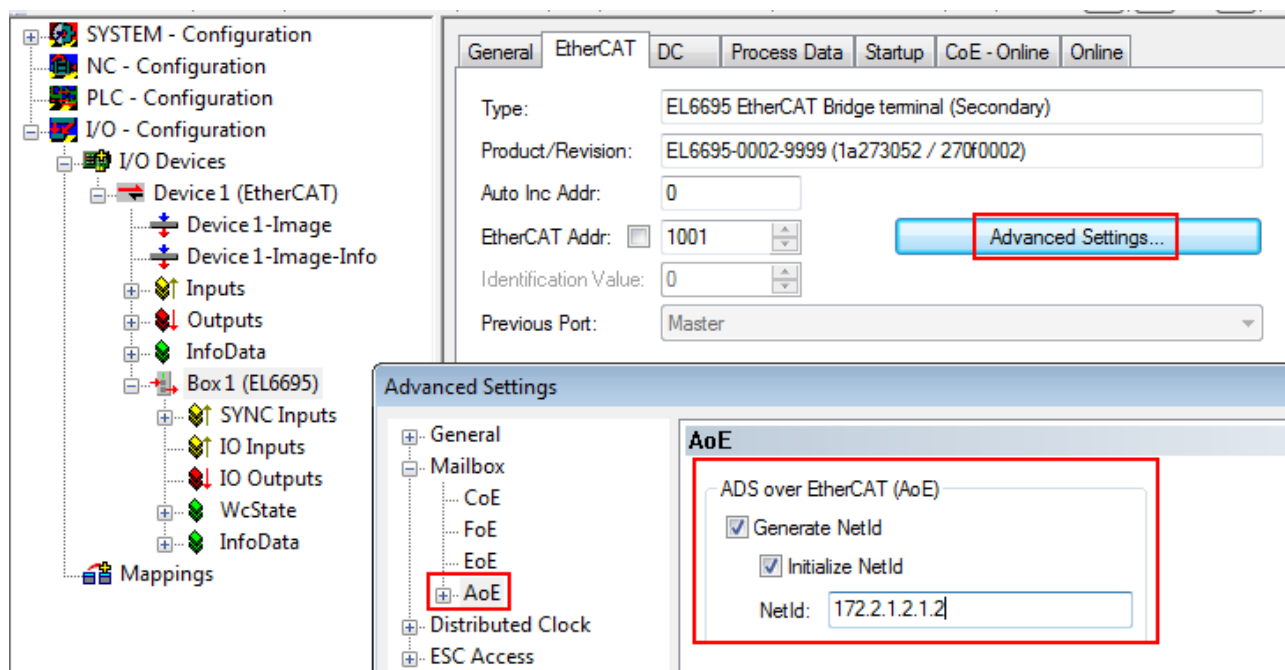
6.6.1 CoE 访问过程中的 AoE 应用

CoE 访问过程中的 AoE 应用



附图 164: 带 AoE ADS 操作的 CoE 写入访问的图示

- EL6695 在两侧各有一个 CoE 目录，其中至少有一侧被声明为用户特定侧。通常，在这种模式下，EL6695 不会在 CoE A 和 CoE B 之间建立连接。因此，两个控制器无法通过 CoE 非周期性地交换参数，并且一侧的写入访问无法通知另一侧的主站。因此，在 EL6695 中通过倍福 ADS 实现的数据交换机制如下：
 - 假设：至少有一侧是 TwinCAT 系统或支持倍福 AoE（ADS over EtherCAT）的 EtherCAT 主站。在本示例中，主站 B 是这样的系统，而主站 A 仅支持 CoE。
 - 主站 A 对 CoE A 的写入/读取访问被传输到 CoE B。
 - EL6695 从 CoE B 0xF820 读取目标 Ams-NetID，并向指定地址发送 AoE ADS 写入/读取指示，该地址处必须有服务接收并处理此指示。例如，在 TwinCAT PLC 中，这可能是来自 TcSystem.lib 的 FB ADSWRITEIND/ADSREADIND。
 - 然后从 PLC 使用 ADSWRITERES/ADSREADRES 将响应发送回 EL6695。使用来自 CoE A 0xF821 的自身 AMS Net ID 作为目标地址。在启动期间，EtherCAT 主站应将其作为 AoE Init Cmd 加载到 EL6695 中。
 - 现在，EL6695 可以将 CoE 写入/读取响应返回给主站 A
 - 在此过程中应考虑传输时间。
 - 此 AoE 机制在存储 AMS NetID 的 CoE 区域中运行。



附图 165: 启用 NetID

● 需要存储 AMS 地址

i 若采用部分对象字典机制——即对象字典（OD）的部分条目由其他设备交互并追加至本机对象字典，则必须为相应（新增）的对象存储对应的 AMS 地址（通过对象 0xF820 [► 122]）。

- “部分对象字典”模式：
 - 对象字典（OD）的各个部分被下载并插入到 OD 中。
- “带镜像对象的部分对象字典”模式：
 - 对象字典（OD）的各个部分被加载到 EL6695，并插入到 OD 中。并且会在远程 OD 中创建镜像对象。该模式可在 CoE A 与 CoE B 之间建立直接数据链接，无需借助上文所述的 AoE 机制。然而，该模式仅覆盖 0x8nnn 参数范围：
 - CoE A 0x8000 – 0x87FF 被镜像到 CoE B 0x8800 – 0x8FFF
 - CoE B 0x8000 – 0x87FF 被镜像到 CoE A 0x8800 – 0x8FFF
 - 因此，0x88xy 对象为“只读”（RO）

● 已修改的 CoE 对象

i ● 原始名称为字符串“Beckhoff EL6695”，存储在 CoE 对象 0x10E1 “OEM Device Name”中。此内容不可更改。

对象字典（OD）还通过 0x1C13 及后续对象定义过程数据。源侧与仿真侧的过程数据长度必须相同！

● OD 以断电保持的方式存储在 EL6695 中，每侧最大容量为 128 kbytes。

● 源侧必须在线/存在且已定义 EL6695 的行为，然后才能对主站 B 的仿真侧进行访问。如果主站 B 先启动，EL6695 将响应为 PreOp_ERR，表示过程数据尚未定义。一旦主站 A 随后启动且 EL6695 已定义完成，主站 B 必须运行“clear ERR”，并重新启动 B 侧进入 OP 状态。该行为可通过 CoE 对象 0xF800:02 的位 1（MASK_RUN_PRI_UNC）和位 2（MASK_RUN_SCND_UNC）进行配置（请参见附录）。

- “complete”、“partial”、“mirrored”模式通过 EL6695 中 *.coe 文件的内容进行设置

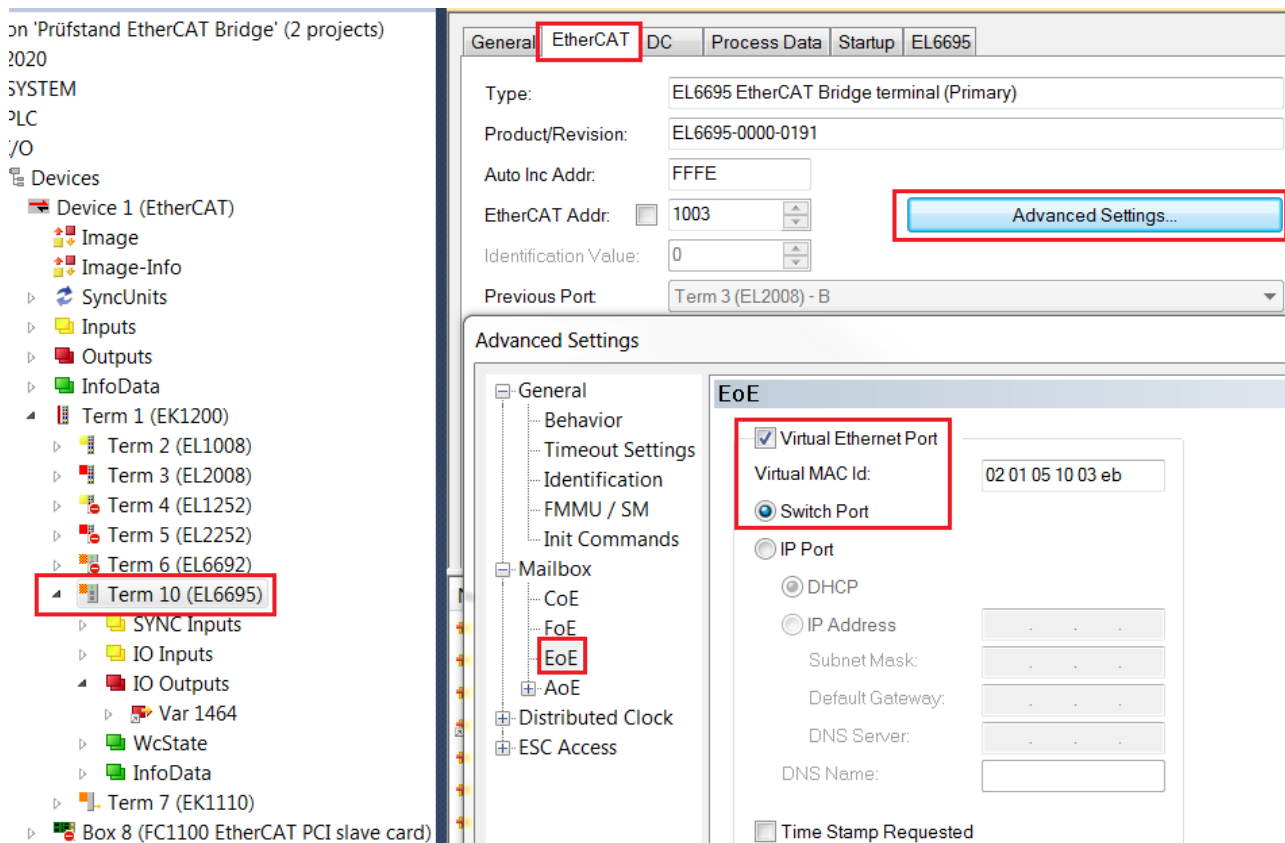
6.6.2 EoE - Ethernet over EtherCAT

处理：转发

支持主/从侧的邮箱大小不同

由于 TwinCAT 及所连接的 EtherCAT 环境充当虚拟网卡，Windows 负责处理 IP 帧（例如 192.168.2.1）到 EL6695 的路由，以便将这些帧传输至另一侧的 EtherCAT 系统。基本原理：请参见 EL6601/EL6614 的文档。

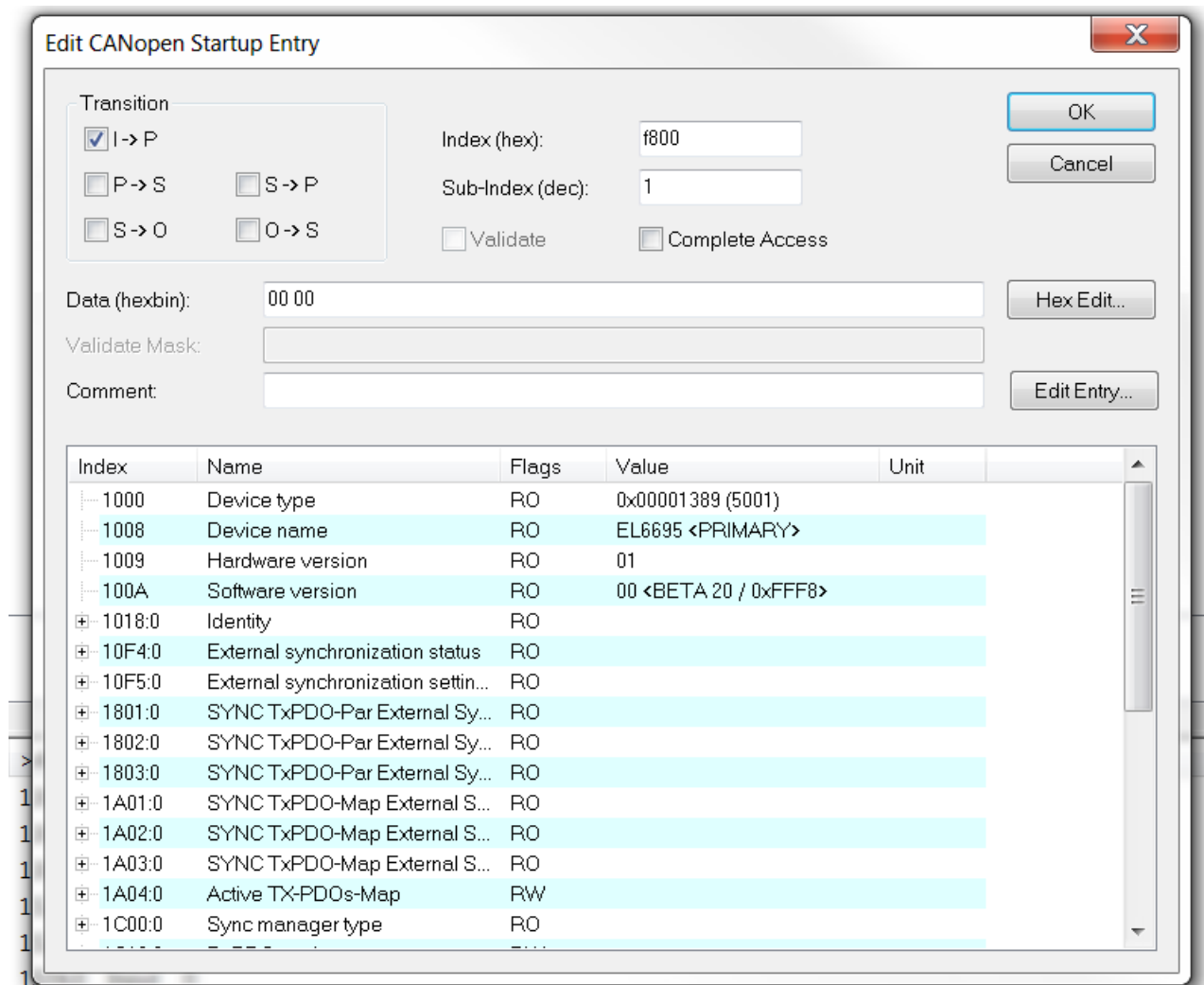
要在 EL6695 中启用 EoE，应在端子模块的 EtherCAT 选项卡的 Advanced Settings（高级设置）中的 EoE 部分，勾选“Virtual Ethernet Port（虚拟以太网端口）”选项并选择“Switch Port（交换端口）”。



附图 166: EoE 配置

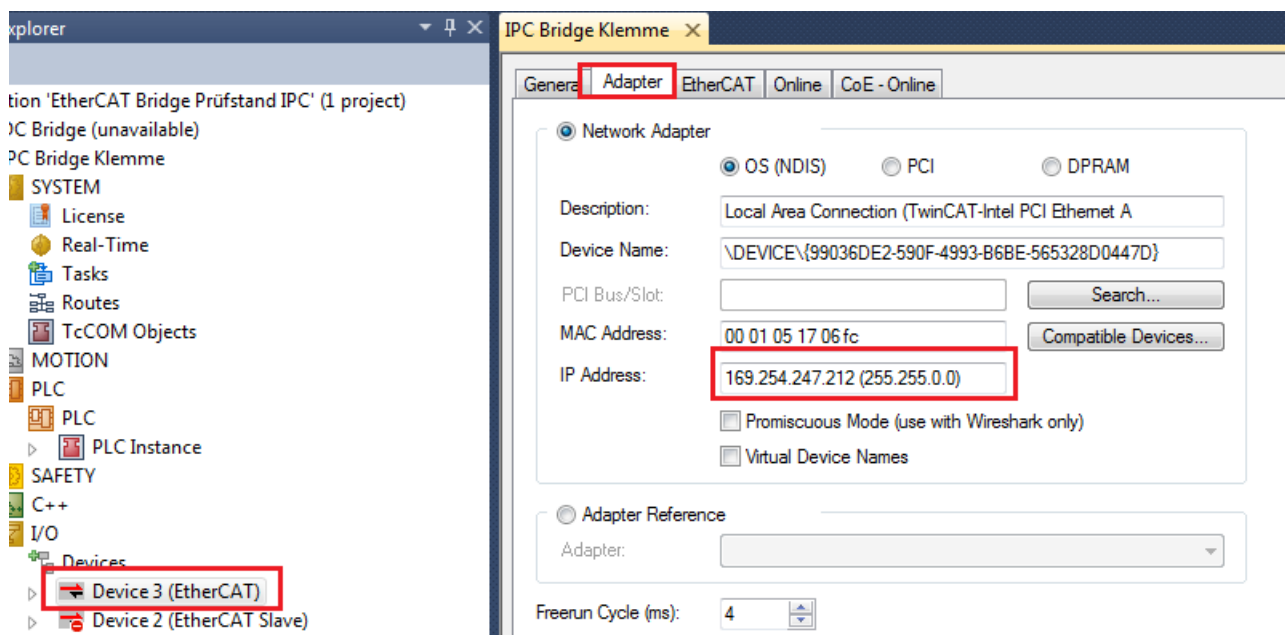
● 通过对象 0xF800（Device Config）激活 EoE

如有需要，应通过端子模块的 CoE 对象 0xF800（Device Config）检查 EoE 的激活情况。如果在条目 0xF800:01（Config 1）中设置了位 14（例如填入值 0x4000），则 EoE 邮箱协议将被阻止（请参见附录）。如需激活，应将该位设置为零，或者将完整的 16 位值设置为 0x0000。或者，也可以通过 startup 条目来完成（请参见图示 [▶ 159]）。



附图 167: 验证对象 0xF800 的正确值 (0x0000) : EoE 已启用

为确保 EoE 通信正常工作，两个 EtherCAT 主站的 IP 地址必须在同一子网内。

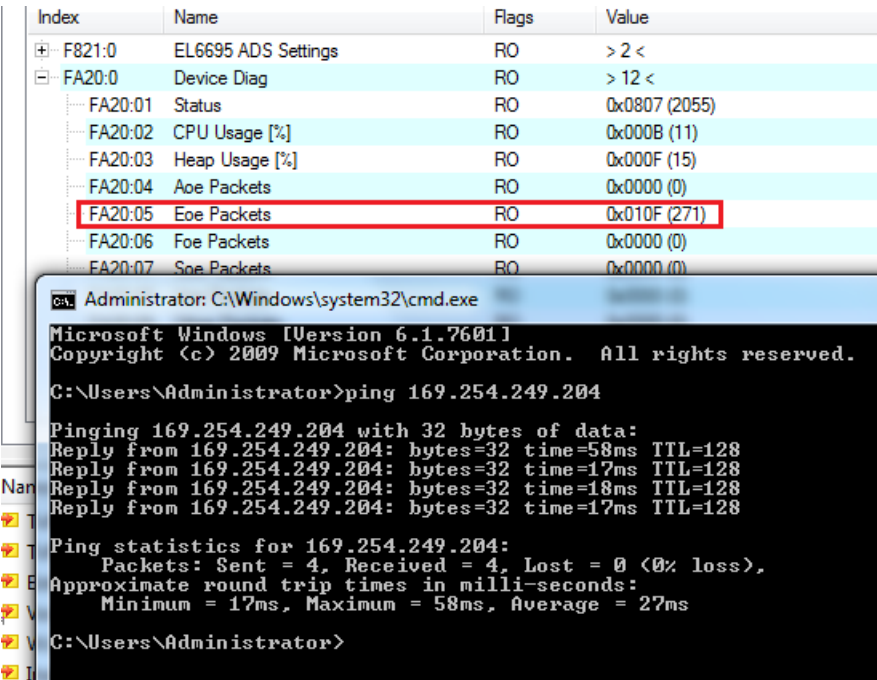


附图 168: 检查 EoE IP 地址

在本示例中，主站 1 的 IP 为 169.254.247.212，子网为 255.255.0.0。

主站 2 可以使用 IP 169.254.249.204，但也必须位于子网 255.255.0.0 内。

然后，就可以在主站 1 的命令行中执行“ping”命令连上主站 2。



附图 169: 通过命令行从主站 1 向主站 2 执行“ping”命令

在 EL6695 的 CoE 对象中，EoE 数据包随后会在对象 0xFA20:05 中递增计数。在本示例中，使用 ping 命令发送四个数据包。

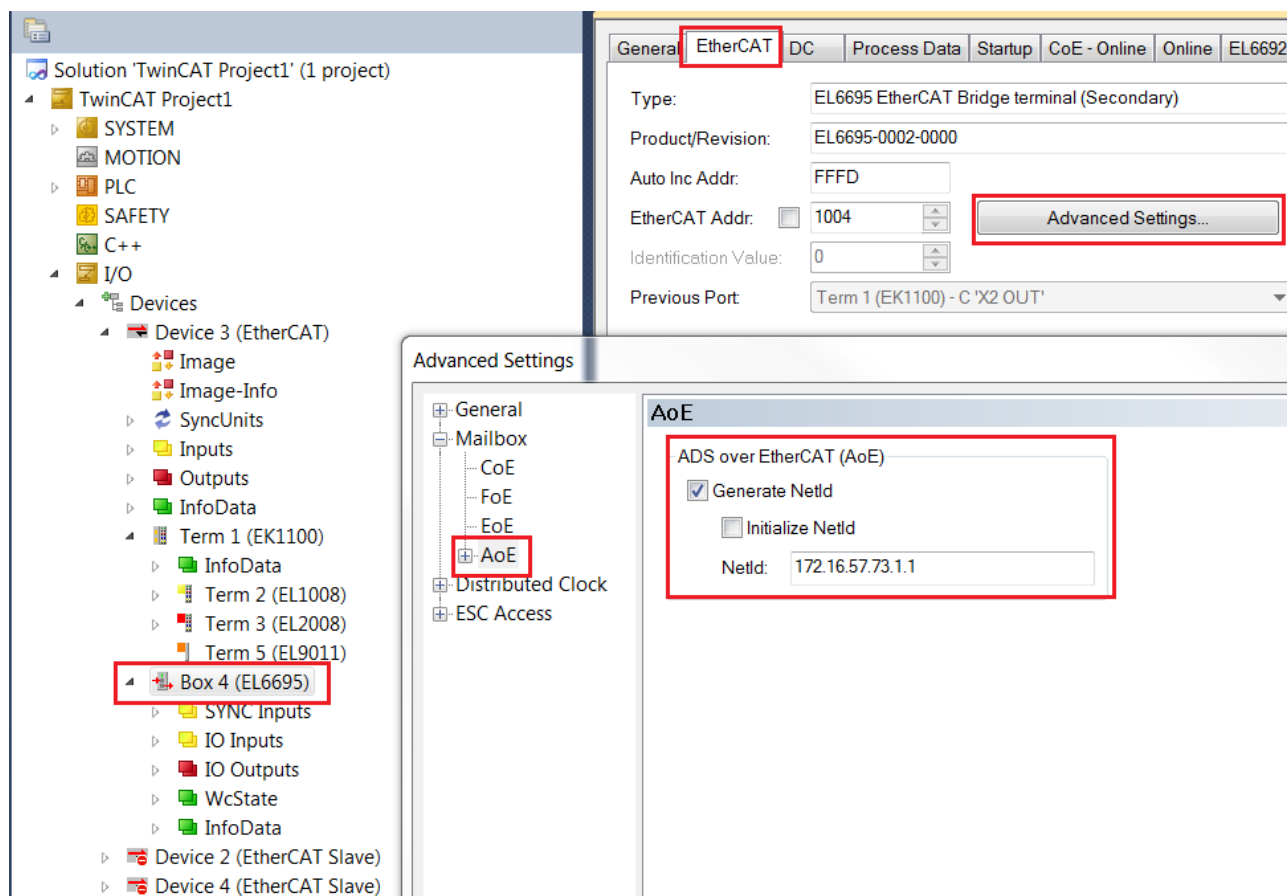
6.6.3 AoE - ADS over EtherCAT

AoE - ADS over EtherCAT

处理：转发

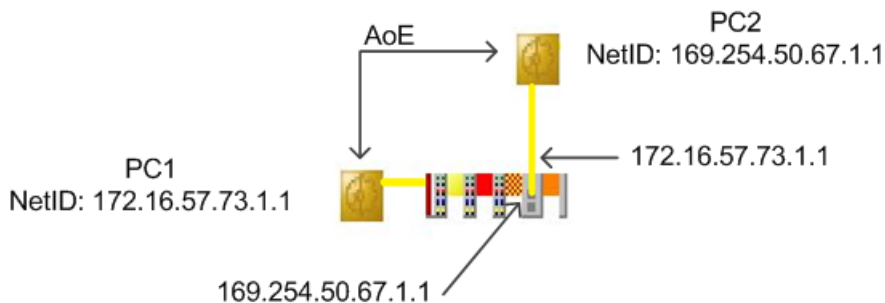
支持主/从侧的邮箱大小不同

ADS（自动化设备规范）是倍福开发并公开的一种协议，用于基于硬件或软件的设备之间的数据交换。这些 ADS 报文的结构可在 Beckhoff Information System（倍福信息系统）或 ETG 标准中查看。EL6695 可通过 AoE（ADS over EtherCAT）将 ADS 报文直接传输到另一侧，无需底层的 IP 通道。为此，在两侧必须均启用基于 AoE 的通信。



附图 170: ADS 帧路由的设置

为了通知 TwinCAT 要如何或通过哪个通道路由 ADS 帧，这里的 NetID 应设置为对方系统的地址。上述示例基于 PC2 的视角。



附图 171: 为 AoE – ADS over EtherCAT 设置 NetID



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。

→ 示例程序，可按需索取：

<https://infosys.beckhoff.com/content/1033/el6695/Resources/1421139595.zip>

6.6.4 FoE – Filetransfer over EtherCAT

FoE – Filetransfer over EtherCAT

本质上，该功能用于通过各参与主站（PLC）的两块关联内存区域实现数据访问。数据可以作为一种数据流从原边（E-bus 侧）传输到副边（RJ45 接口），反之亦然。

每个存储区域的最大容量为 32768 字节。

EL6695 中的处理过程取决于 EtherCAT 状态：

- BOOT: FoE 用于本地 FW 更新
- 所有其他状态（INIT、PreOP、SafeOP、OP）：取决于 CoE 密码
 - 无密码：FoE 转发到远程站点
 - 有密码：根据密码进行本地保存。由此实现特殊功能。



FoE 请求/无密码

FoE 请求必须由对侧接受并处理。TwinCAT 版本（3.10 b4014 和 2.11 b2245）不支持此功能。两侧的邮箱大小必须相同。

缓冲式 FoE 运行模式

如果在对象 0xF800（Device Config）的子索引 02 中置位了 [MASK_BUFFER_FOE] = 位 8，则允许使用不同的邮箱大小。

6.6.4.1 在 TwinCAT 下使用 FoE 的要求

TwinCAT 中的编程

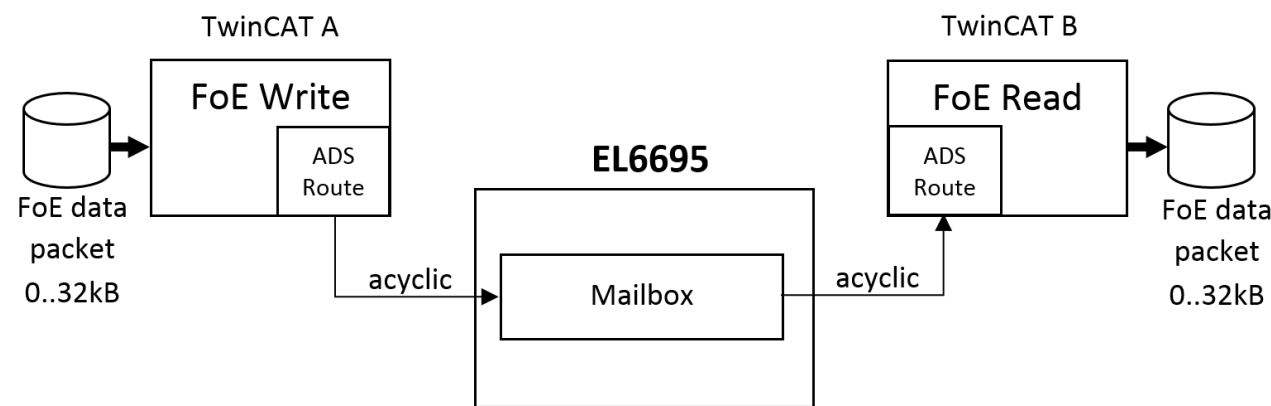
编程时，需要集成包含用于访问 EtherCAT 主站和从站设备的功能块的库“Tc2_EtherCAT”（“Tc2_..”适用于 TwinCAT 3）。其中所包含的以下功能块可实现针对数据或文件传输的以下访问：

- FUNCTION_BLOCK FB_EcFoeOpen（打开通信端口）
- FUNCTION_BLOCK FB_EcFoeAccess（数据的写入/读取）
- FUNCTION_BLOCK FB_EcFoeClose（关闭通信端口）

有关更多详细信息，请访问倍福信息页面（infosys.beckhoff.com）。

注意：FoE 传输本身无法检查文件的正常结束，因此也无法确认数据传输是否完整。建议通过其他渠道传输文件大小或文件完整性信息，例如在文件内部加入可验证的元信息。

通过 EL6695 进行的 FoE 数据交换遵循以下模式：



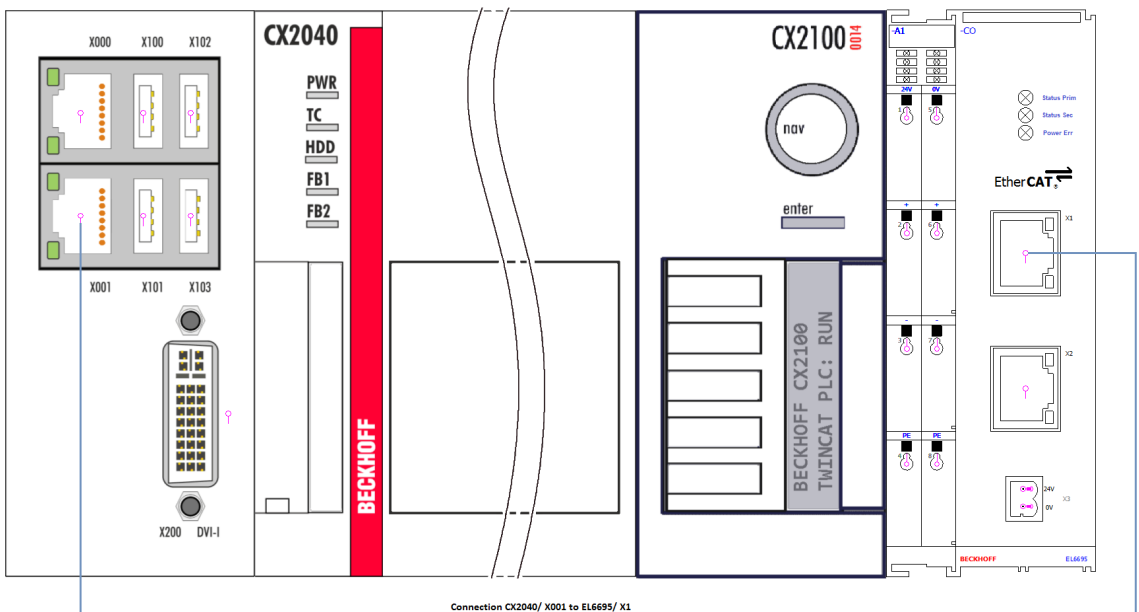
附图 172: 通过 EL6695 进行的 FoE 数据交换

- FoE 数据集每个最大为 32 kB 的，均可通过 EL6695 进行传输
- 使用的是通过 EL6695 邮箱（默认 1024 字节）进行的非周期性 EtherCAT 通信
- EL6695 仅作为数据缓冲区使用，其自身不执行任何 FoE 处理

6.6.4.2 配置举例

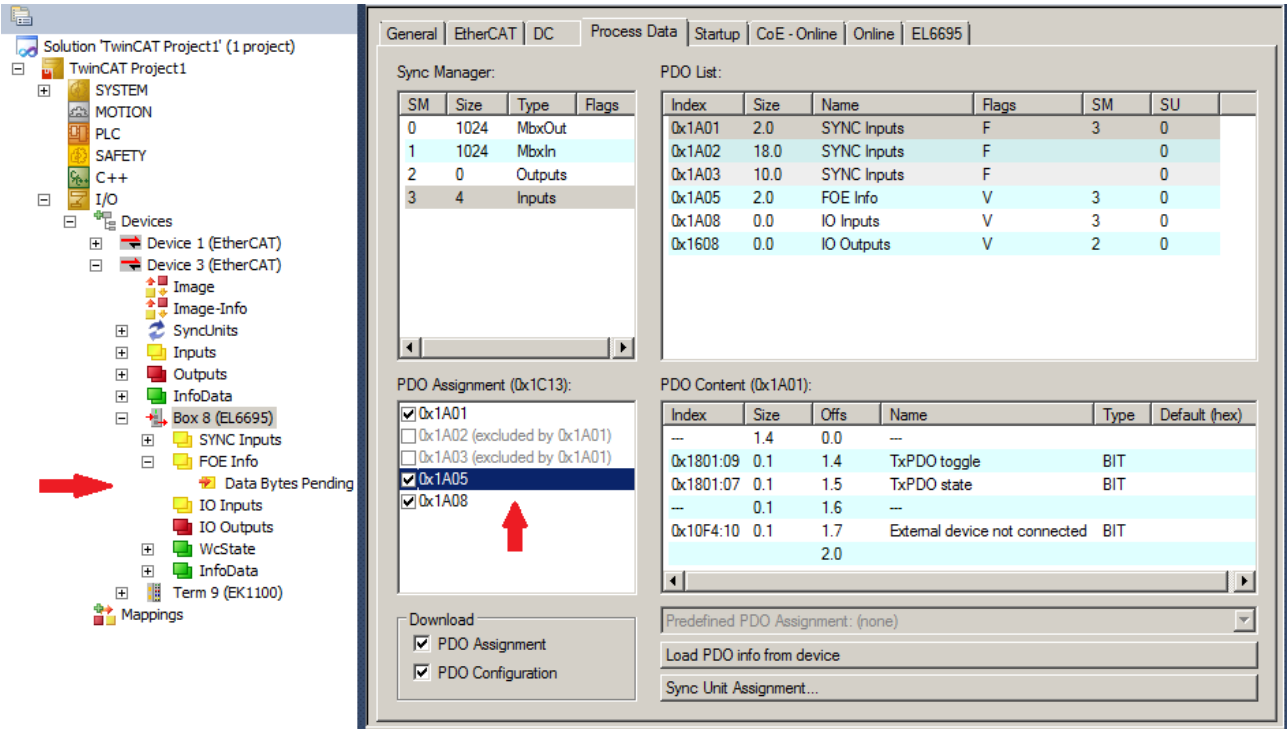
配置举例

下面通过一个编程示例来说明 FoE 的功能。下图 [► 164]展示了包含 CX2040 嵌入式控制器（含 CX2100-0014 电源模块）以及下游 EL2809、EL1004 和 EL6695 的物理结构，其中 CX PC 的以太网接口 X001 连接至 EL6695 桥接端子模块的“上方” RJ45 接口 X1：



附图 173: 采用 CX2040（含 CX2100-0014 电源模块）及 EL6695 的配置方案

此外，需要在系统管理器中添加 CoE 对象 0x1A05，以便在接收侧能够根据信息对象“Data Bytes Pending（待处理数据字节数）”来判断是否已接收到数据或文件。为此，必须在 process data（过程数据）上激活相应的复选框。图中的展示仅针对副边，在本示例中代表接收侧。此外，TwinCAT 示例程序中包含的可公开访问的变量“ScndFoeBytesToRead”需要与“Data Bytes Pending”进行关联。



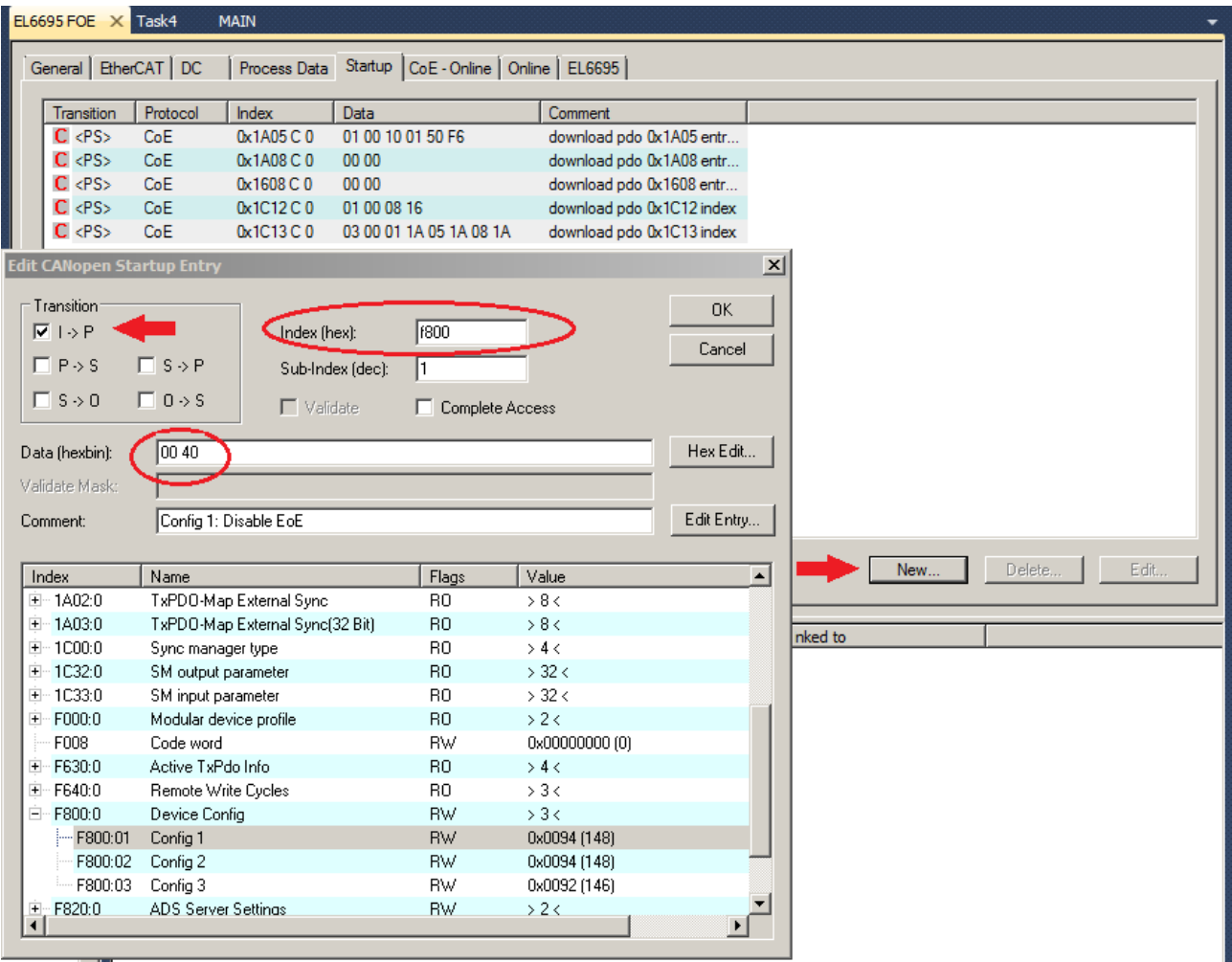
附图 174: 在副边添加 0x1A05（Data Bytes Pending）

Init – 启动配置

根据对象 0xF800 的配置定义（请参见 CoE/参数目录 – 配置文件特定对象），在从 Init 状态转换到 PreOp 状态（I → P）时，需要在 0xF800 中输入两个值：

- 0xF800:01 中的值 0x4000（用于停用 EoE）
- 0xF800:02 中的值 0x0100（用于在 EL6695 上预留 FoE 缓冲区内内存的特殊模式）

下面展示使用系统管理器（TC3.1）用户界面的操作步骤，在原边或副边所标记的端子模块或端子盒的“Startup（启动）”选项卡中进行设置：



附图 175: 在设备配置 0xF800:01 中输入值 0x4000（注释 = 配置 1: 禁用 EoE）

此处需要为对象 0xF800 的子索引 01 输入值 0x4000（类型 00 40），以停用 EoE。这一点之所以必要，仅仅是因为在本示例中，端子模块的原边与副边相连，且两侧都连接到同一个 EtherCAT 主站。

● 在 FoE 下禁用 EoE 协议

I 如果原边与副边连接在同一条 EtherCAT 线路中，则必须通过设置端子模块中对象 0xF800:01 的位 14（0x4000）来禁用 EoE 协议，否则该端子模块将被阻塞（由于重复的 ARP 以太网请求）。

此外，必须为 0xF800:02 输入值 0x0100（类型 00 01）（注释 = 配置 2: 启用 FoE 缓冲区）。无论在 EL6695 桥接端子模块的原边还是副边进行此设置都无关紧要，因为这些设置最终也会被另一侧采纳。

● 更改设备配置 0xF800

I 对象 0xF800 在原边和副边均可配置，并且最终也会被另一侧采纳。

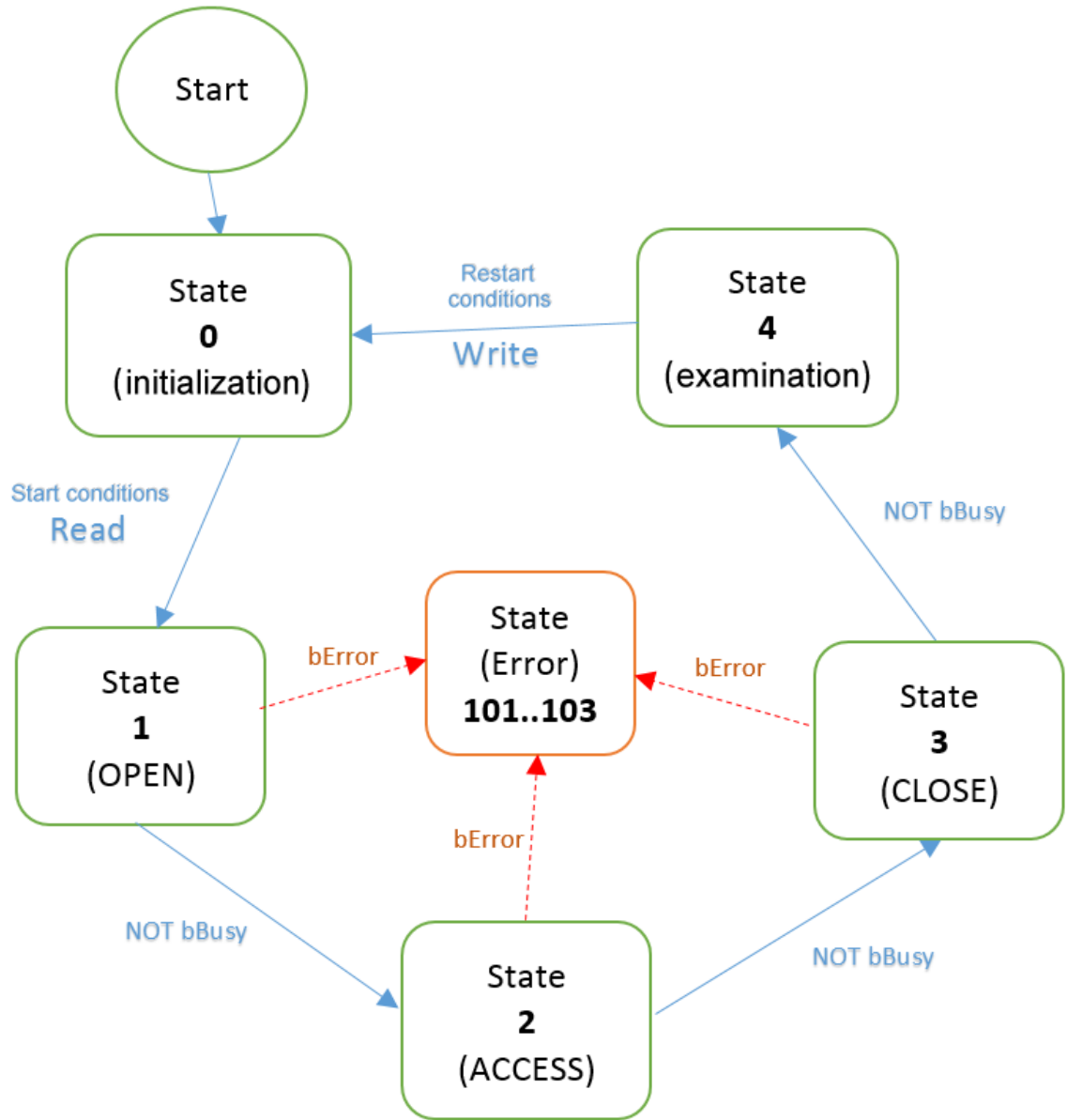
注意必须选择正确的转换：P → S 必须关闭，I → P 必须激活。完成上述设置后，效果应如下图所示：

General EtherCAT DC Process Data Startup CoE - Online Online EL6695					
Transition	Protocol	Index	Data	Comment	
 <PS>	CoE	0x1A05 C 0	01 00 10 01 50 F6	download pdo 0x1A05 entries	
 <PS>	CoE	0x1A08 C 0	00 00	download pdo 0x1A08 entries	
 <PS>	CoE	0x1608 C 0	00 00	download pdo 0x1608 entries	
 <PS>	CoE	0x1C12 C 0	01 00 08 16	download pdo 0x1C12 index	
 <PS>	CoE	0x1C13 C 0	03 00 01 1A 05 1A 08 1A	download pdo 0x1C13 index	
 IP	CoE	0xF800:01	0x4000 (16384)	Config 1: Disable EoE	
 IP	CoE	0xF800:02	0x0100 (256)	Config 2: Enable FoE Buffer	

附图 176: 在设备配置 0xF800 中输入两个值

有关程序示例的说明

附件中的程序示例旨在说明 FoE 数据传输，该示例基于下文所示的写入和读取访问的状态图。



附图 177: 示例程序状态图：通过“OPEN”、“ACCESS”和“CLOSE”写入和读取随机数据

在程序中定义的 **bEnabled** 标志用于控制程序，并用作启动条件或重启条件。由于附加了声明 `AT%I*`，该标志可以与输入端子模块的“实际”输入进行关联，从而“从外部”控制程序，例如通过连接的按钮（up +）来实现。

- 读取启动条件：**bEnabled** 必须为 TRUE（例如，触发外接按钮）；此外，需检查“ScndFoeBytesToRead”，以确认是否有传入数据，且其值必须 > 0。
- 写入重启条件：**bEnabled** 必须再次变为 FALSE（例如，再次松开外部连接的按钮）。

此外，在写入数据的状态机处于 State 1 期间，会检测标志 **bEnabled** 是否为 TRUE，只有当该标志为 TRUE 后才能用 OPEN 启动写入过程。

由于附加了声明 `AT%Q*`，**bDataEqual** 可以与数字量输出模块的输出变量进行关联。通过这种方式，可以在最后查看是否正确完成了数据传输。

0 至 4 的各个状态作为 **iWrState** 编程如下：

- **[iWrState = 0]**：执行初始化：
 - 设置 FALSE → **bDataEqual**
 - 在 **aWrBuffer** 中生成随机数据
 - 准备调用 OPEN 功能块：设置 FALSE → **bExecute**
 - 设置下一个状态：1 → **iWrState**
- **[iWrState = 1]**：查询输入变量 **bEnabled**，以判断是否开始调用 OPEN 功能块。如果 **bEnabled** = TRUE，则使用以下参数调用 OPEN 功能块：
 - EC_MasterNetId_Wr → **sNetId**
 - EL6695_WR_EcAddr → **nPort**
 - 16#12345678 → **dwPass**
 - 1 → **eMode**（写入标识符）
 - TRUE → **bExecute**
 - T#10S → **tTimeout**
 - 重复此调用，直到查询 **bBusy** 标志返回 FALSE；随后执行以下操作：
 - 设置下一个状态：2 → **iWrState**；如果 **bBusy** = TRUE，则不进行状态变更。本示例中不考虑由 OPEN 功能块（用于读取/写入）的 **bError** 标志的错误处理，任何其他功能块（用于读取/写入的 ACCESS、CLOSE）也是如此。
 - 准备调用 ACCESS 功能块：设置 FALSE → **bExecute**
- **[iWrState = 2]**：使用以下参数调用 ACCESS 功能块：
 - fbWrFoeOpen.hFoe → **hFoe**
 - ADR(aWrBuffer) → **pBuffer**
 - TRUE → **bExecute**
 - T#14S → **tTimeout**
 - 重复此调用，直到查询 **bBusy** 标志返回 FALSE；随后执行以下操作：
 - 设置下一个状态：3 → **iWrState**
 - 准备调用 CLOSE 功能块：设置 FALSE → **bExecute**
- **[iWrState = 3]**：使用以下参数调用 CLOSE 功能块：
 - fbWrFoeOpen.hFoe → **hFoe**
 - TRUE → **bExecute**
 - T#14S → **tTimeout**
 - 重复此调用，直到查询 **bBusy** 标志返回 FALSE；随后执行以下操作：
 - 设置下一个状态：4 → **iWrState**
- **[iWrState = 4]**：在此状态下，仅查询输入变量 **bEnabled** 是否再次变为 FALSE。随后，程序将重新从起始状态开始执行：

- 如果 **bEnabled = FALSE**，则设置下一个状态：0 → **iWrState**

在下次启动时，会再次生成新的随机数，再行比对时读取到的数值应与这些随机数匹配一致。为此提供了 **bDataEqual** 标志，它可以指示这一点，我们将在下文中看到。

读取访问的状态图原则上类似，并且在 State 4 中包含一个用于将写入值与读取值进行数据比较的程序段。

0 至 4 的各个状态作为 **iRdState** 编程如下：

- **[iRdState = 0]**：在此状态下，查询输入变量 **bEnabled** 是否为 TRUE。随后，程序从查询输入变量“ScndFoeBytesToRead”开始，该变量关联至 [FoE Info].Data Bytes Pending（对象 0x1A05），用于判断接收侧是否有数据“存在”。
 - 如果 **bDataEqual = TRUE**，则查询 **ScndFoeBytesToRead**
 - 如果 **ScndFoeBytesToRead = TRUE**，则设置下一个状态：1 → **iRdState**
 - 为准备调用 OPEN 功能块进行设置：如果 **ScndFoeBytesToRead = TRUE**，则设置 FALSE → **bExecute**
- **[iRdState = 1]**：查询输入变量 **bEnabled**，以判断是否开始调用 OPEN 功能块。如果 **bEnabled = TRUE**，则使用以下参数调用 OPEN 功能块：
 - EC_MasterNetId_Rd → **sNetId**
 - EL6695_RD_EcAddr → **nPort**
 - 16#12345678 → **dwPass**
 - 0 → **eMode**（读取标识符）
 - TRUE → **bExecute**
 - T#10S → **tTimeout**
 - 重复此调用，直到查询 **bBusy** 标志返回 FALSE；随后执行以下操作：
 - 设置下一个状态：2 → **iRdState**
 - 准备调用 ACCESS 功能块：设置 FALSE → **bExecute**
- **[iRdState = 2]**：使用以下参数调用 ACCESS 功能块：
 - fbRdFoeOpen.hFoe → **hFoe**
 - ADR(aRdBuffer) → **pBuffer**
 - TRUE → **bExecute**
 - T#14S → **tTimeout**
 - 重复此调用，直到查询 **bBusy** 标志返回 FALSE；随后执行以下操作：
 - 设置下一个状态：3 → **iRdState**
 - 准备调用 CLOSE 功能块：设置 FALSE → **bExecute**
- **[iRdState = 3]**：使用以下参数调用 CLOSE 功能块：
 - fbRdFoeOpen.hFoe → **hFoe**
 - TRUE → **bExecute**
 - T#14S → **tTimeout**
 - 重复此调用，直到查询 **bBusy** 标志返回 FALSE；随后执行以下操作：
 - 设置下一个状态：4 → **iRdState**
- **[iRdState = 4]**：在此状态，会对两个存储区域进行最终检查，如果两个存储区域相同，则将 **bDataEqual** 标志设置为 TRUE。建议将此标志作为输出变量进行“呈现”，以便通过输出端子模块将其转接至输出端，诸如此类。
 - 设置下一个状态：0 → **iRdState**

使用示例程序



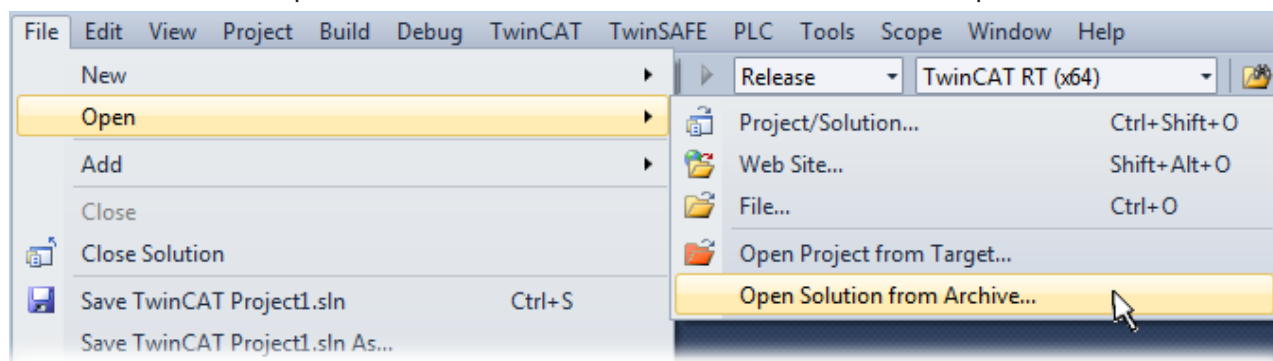
本文件包含我们产品在某些应用领域的应用案例。这里提供的应用说明是基于我们产品的典型特征，仅作为示例。本文件中的说明明确地不涉及具体的应用。因此，客户有责任评估和决定产品是否适用于特定的应用。我们对本文件所包含的源代码的完整性和正确性不承担任何责任。我们保留在任何时候修改本文件内容的权利，对错误和遗漏的信息不承担任何责任。

→ 下载 FoE 示例程序：

<https://infosys.beckhoff.com/content/1033/el6695/Resources/1421822987.zip>

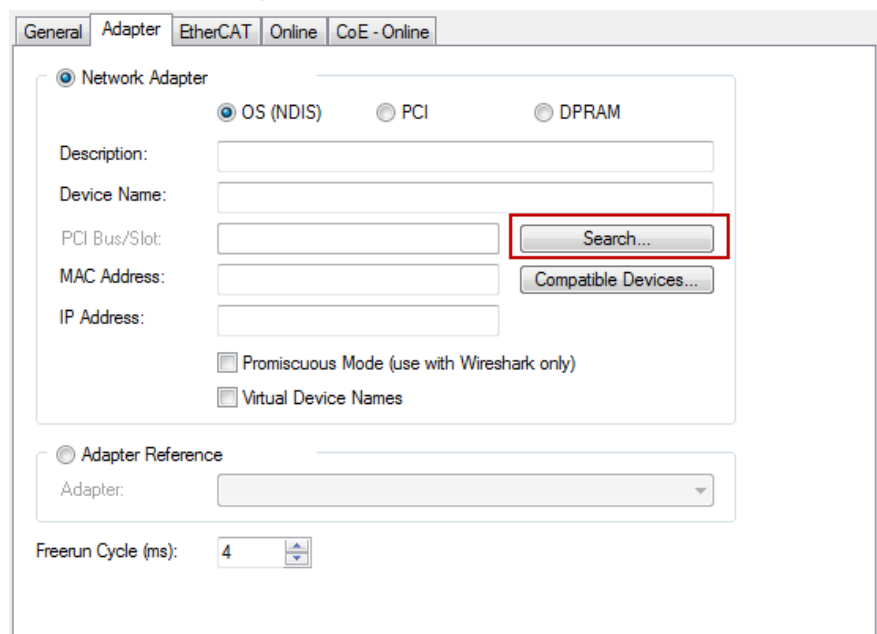
启动示例程序的准备工作 (tnzipped 文件/TwinCAT 3)

- 点击下载按钮将 Zip 压缩包保存到本地硬盘，然后在临时文件夹中解压 *.tnzip 压缩包文件。



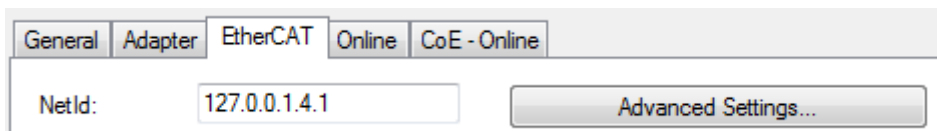
附图 178: 打开 *.tnzip 压缩包

- 选择 .tnzip 文件（示例程序）。
- 在弹出的窗口中，选择用于存储项目的目标目录。
- 有关 PLC 调试程序和启动程序的一般说明，请参阅端子模块文档资料或 EtherCAT 系统文档资料。
- 示例中的 EtherCAT 设备通常应声明为您当前的系统。在“Solutionexplorer”中选择 EtherCAT 设备后，选择“Adapter”选项卡并点击“Search...”：



附图 179: 搜索示例 EtherCAT 配置的现有硬件配置

- 检查 NetId：EtherCAT 设备的“EtherCAT”选项卡显示配置的 NetId：



前四个数字必须与目标系统的项目 NetId 相同。可在上述 TwinCAT 环境中查看项目 NetId，打开下拉菜单可选择目标系统（在文本字段中点击右键）。数字块置于目标系统每个计算机名称旁的括号中。

- 修改 NetId：右键单击解决方案资源管理器中的“EtherCAT device”，打开一个上下文菜单，其中会显示“Change NetId...” 必须进行选择。必须输入目标计算机 NetId 的前四个数字；最后两个值通常都是 4.1。

示例：

- 项目的 NetId： myComputer (123.45.67.89.1.1)
- 通过“Change NetId...”输入： 123.45.67.89.4.1

6.6.4.3 示例：FoE 数据吞吐率

下文展示了使用 EL6695 进行 FoE 吞吐率测定的示例。这些值应被视为该布局方式的示例以及粗略参考值。如有必要，需另行测定在各具体应用中实际可达到的吞吐率。

本文档并不包含用于该种目的的 TwinCAT 程序；它基于指定的 FoE 示例程序 [► 170]。

数据吞吐率被定义为数据量与所需时间之比（单位时间内的数据量）。使用已描述的 FoE 访问功能块，可将指定数据量写入 EL6695 或从中读取。EL6695 被设计用于在两个不同的 TwinCAT 系统之间建立通信。不过，使用该端子模块也可以在一个 TwinCAT 系统内部建立原边与副边之间的连接，例如，当其副边的 X1 接口连接来自 EK1100 耦合器输出 X2 接口时。此时，在一个 TwinCAT 系统中也会存在两个独立的 EtherCAT 网段（请参见示例配置 [► 164]）。

时间测量通过 FB_EcFoeAccess 功能块间接进行，依据是每次完整执行读取访问所需的任务周期数（即写入的字节数与读取的字节数相同）（请参见关于程序示例的说明 [► 167]）。如果有数据待处理，则待读取端子模块侧的“DataBytesPending” PDO > 0，并标志着读取访问的开始。读取访问的结束由已知的写入数据量决定（对于两个独立的系统，文件大小必须已知）。

在任务周期时间为 1 ms 的情况下，数据吞吐率的简单计算如下：

{数据量 [字节]} / {任务周期数} = 吞吐率 [kB/s]

测量顺序

- 测试环境可确定端子模块完成一次读取访问所需的任务周期数。向端子模块写入多组不同容量、单包数据长度各异的数据集，并对每组数据集进行完整、反复读取操作。由此，对每一次单独的读取访问所对应的任务周期数进行了计数。
- 此外，还测定了测试程序在未经过中间连接的 EL6695 时的 FoE 吞吐率。由此得出测试程序的“虚拟”时间需求。
- 端子模块实际所需的任务周期数（以及由此得出的数据吞吐率）则由“实际”所需任务周期数（带 EL6695）与虚拟任务周期数之间的差值决定。
- 整个吞吐率测试包含数千个独立测试，涉及不同的数据量及数据包大小。

示例中的配置：

- C6015 + EK1100 + EL6695（原边） + EL9011
- EK1100 + EL6695-0002（副边）

示例的关键数据：

- 循环时间 = 1 ms
- 测试运行涵盖了所有 FoE 和所有数据包大小。由于在定义的关键范围之外，吞吐率变化不大，因此为下面的压缩示意图选择了以下限值：
- 通过 FoE 传输的数据量（FoE 文件大小）：从 1167 字节到 32767 字节，分 50 步，每步 632 字节

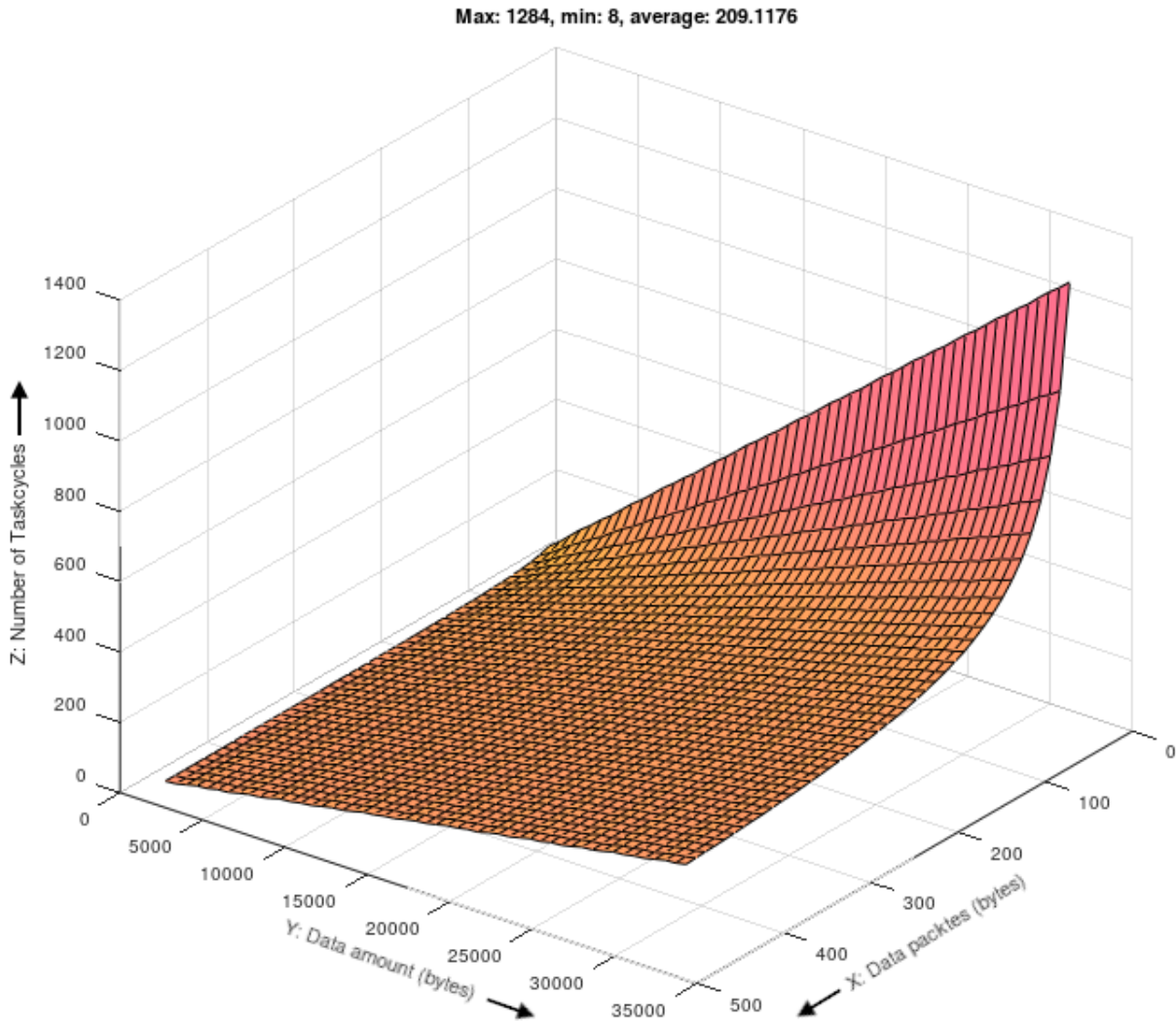
- 数据包（即 FoE 文件所划分成的数据包）：从 30 字节到 470 字节，分 40 步，每步 11 字节

结果：

数据包大小（按步长递增）= X 轴

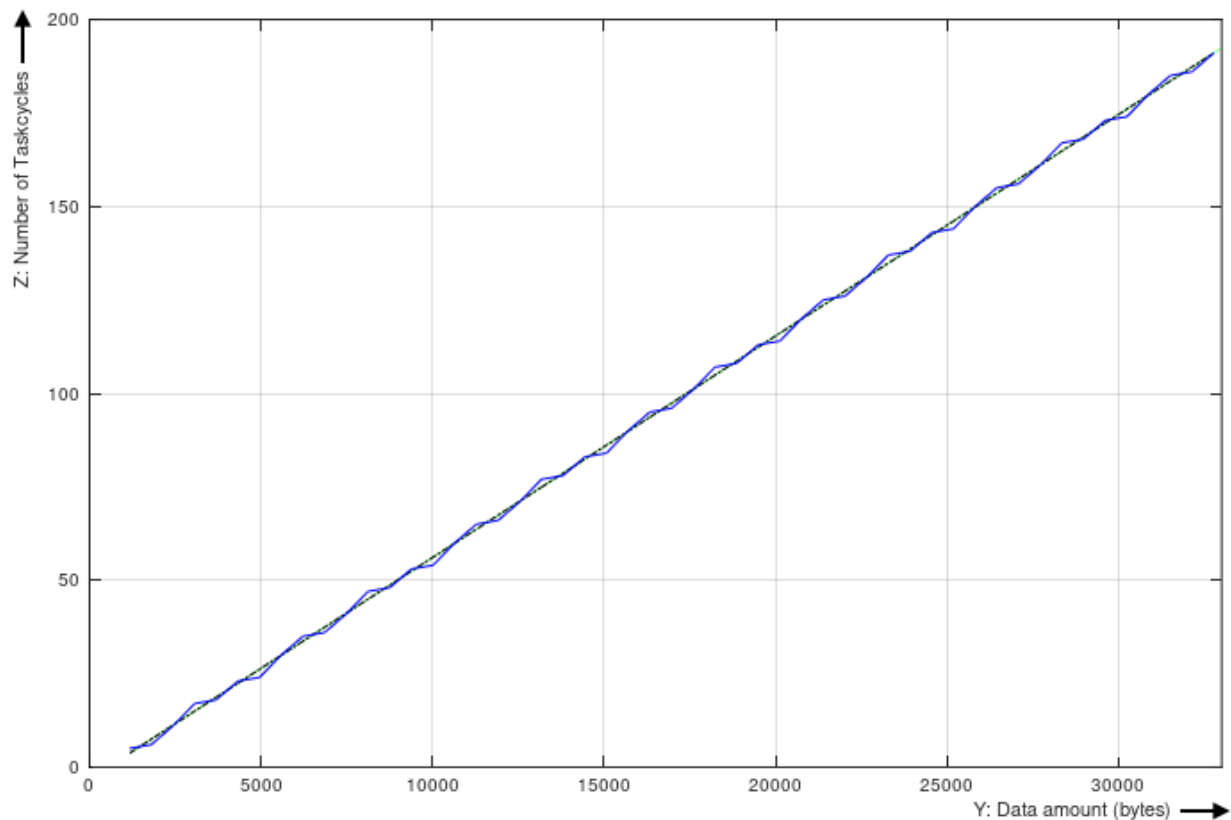
数据量（按步长递增）= Y 轴

所需任务周期数 = Z 轴；即 $Z(X, Y)$



附图 180: 使用 EL6695 的 FoE 吞吐率测试：任务周期数取决于数据包大小和数据量

然而，这也包含了由于顺序数据访问而产生的与系统相关的虚拟值（请参见上文）。如果将这些虚拟值相应地记录下来，并从每个 X/Y 测试点的 Z 轴各值中减去，则会得到一条不再依赖于数据包大小的近似线性曲线：



附图 181: 关于 EL6695 的 FoE 数据吞吐率示例

在线性近似下，此 FoE 示例得出的吞吐率约为 **171 kB/s**。

6.6.5 VoE - vendor-specific protocol over EtherCAT

VoE 可以实现用户特定协议的实施，从而使得内部邮箱能够用于一种特定的重新定义的数据传输。

如需更多信息，请联系倍福支持中心，并提供相关数据（TwinCAT 版本、所需性能值、周期、数据量等）

6.6.6 SoE - Servo Drive Profile over EtherCAT

如需更多信息，请联系倍福支持中心，并提供相关数据（TwinCAT 版本、所需性能值、周期、数据量等）

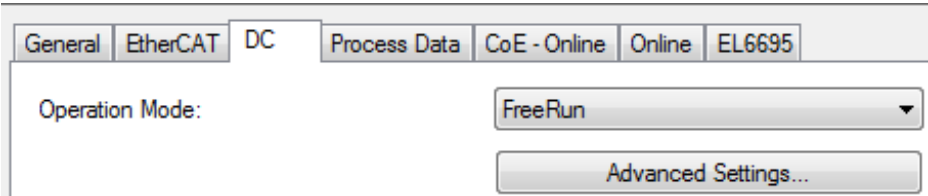
6.7 分布式时钟



EtherCAT 主站中的 DC 支持

EL6695 的分布式时钟单元具有 DC 系统时间，但没有同步/锁存装置。该初始化子程序支持 TwinCAT 3 Build 4018.4 及以上版本、TwinCAT 2 Build 2248 及以上版本。

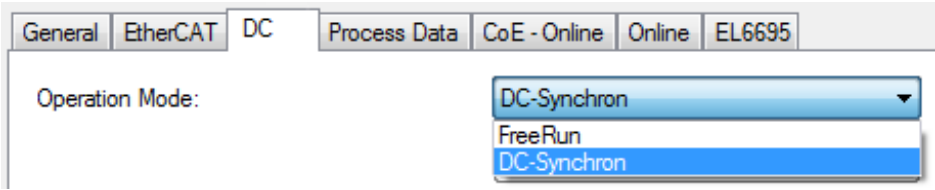
可在该选项卡下查看 EL6695 的配置。对于数据交换，其默认设置为“无同步”（“FreeRun，自由运行”）。



附图 182: 分布式时钟：无同步

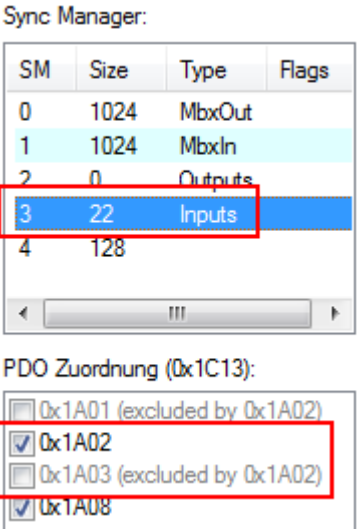
由于 EL6695 包含两个完整的 EtherCAT 从站，其两个分布式时钟模块在底层是相互独立的。EL6695 支持两种 DC 同步机制：

- 与其前代产品（EL6692）一样，它可以将内部/外部时间戳信息提供给同步从站侧，从而为该侧的 EtherCAT 主站提供调整其存储的 real-time /EtherCAT 周期的选项。两个控制方向均支持。
 - TwinCAT 下的应用：将 EL6695 设置为 DC 模式，以便 TwinCAT 将其用作参考时钟



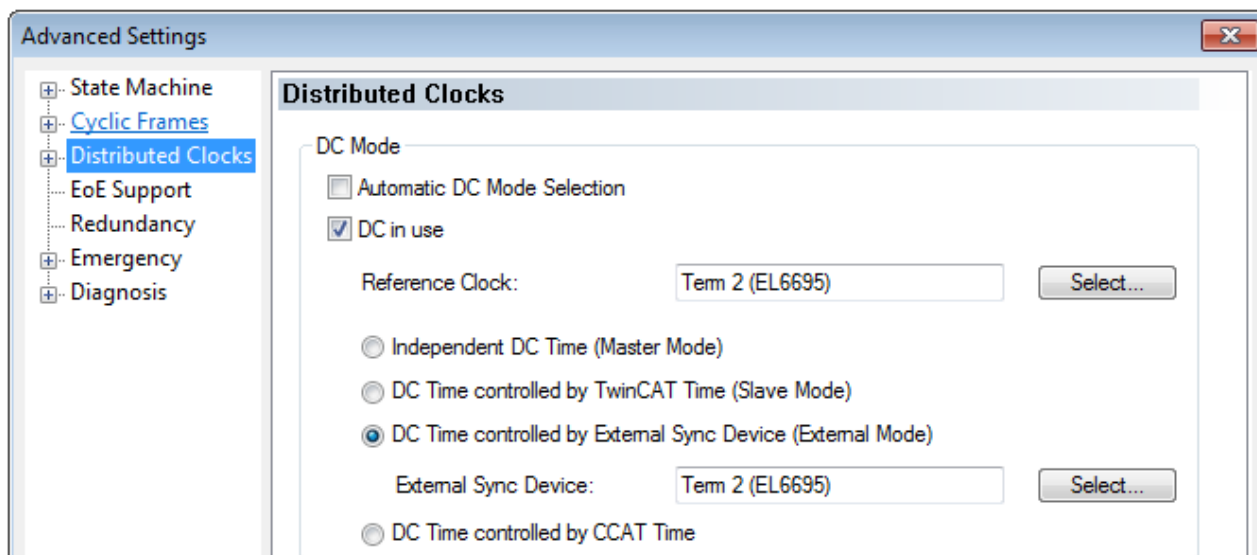
附图 183: 分布式时钟：同步

- 显示并激活时间戳 PDO（64 位使用 0x1A02，32 位使用 0x1A03）。这样，TwinCAT 就能检测到该端子模块可用作外部参考时钟，并读取时间戳。



附图 184: 显示时间戳 PDO 0x1A02 或 0x1A03

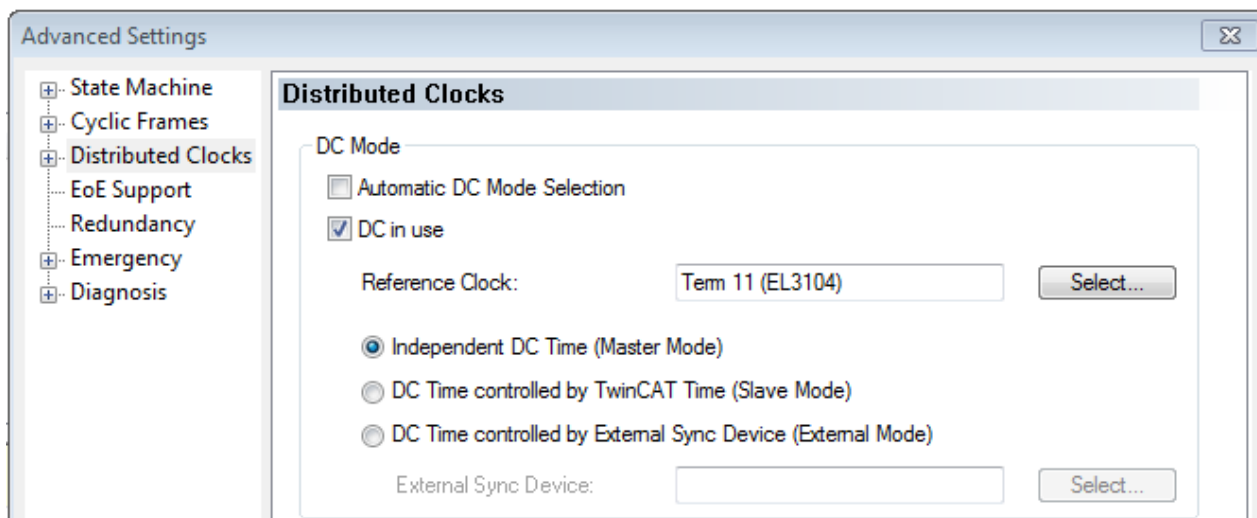
- 然后这个外部参考时钟就可以在“EtherCAT DC 主站设置”中被选择：



附图 185: 分布式时钟：选择外部参考时钟（在本示例中位于原边）

此设置仅在“SyncSlave（同步从站）”侧需要配置。

例如，在“SyncMaster（同步主站）”侧可以输入一个 EL3104（支持 DC 功能）：



附图 186: 分布式时钟：选择 EL3104 作为“SyncMaster”（在本示例中位于副边）

- 如果 EL6695 在两个 EtherCAT 系统中都是第一个支持 DC 的从站，则可以实现两个桥接侧的直接 DC 耦合，而无需通知 EtherCAT 主站。在这种情况下，两个 EtherCAT 系统/主站都将跟随 EL6695 时间，无需进行特殊配置。然而，两个系统时间之间仍存在一个恒定的偏移量。
 - 两个 EtherCAT 主站均从各自一侧设置 0x0920 DC 系统时间偏移量。EL6695 接受此设置，并确保 EL6695 中两个 ESC 的频率同步。

● 不支持本地控制过程

i 与 EL6692 不同，EL6695 不支持本地控制过程“ControlValue for DC MasterClock”。这个时间戳过程应当由 EtherCAT 主站来实现。

由于 EL6695 不在内部处理分布式时钟的时间信息，而只是将其转发到另一侧，因此不需要也不支持使用 ESC 中的特殊 DC 寄存器。在许多支持 DC 功能的 EtherCAT 设备中，主站在从 PreOp 状态转换到 SafeOp 状态期间，会向 DC 寄存器 0x09A0、0x0990、0x0980、0x09A8 执行写入操作。EL6695 没有这些寄存器，因此在启动期间尝试写入会导致 EtherCAT 主站发出错误消息，例如“Init Cmd failed: set DC cycle time”（初始化命令失败：设置 DC 周期）。

IP	PS	PI	SP	SO	SI	OS	OP	OI	IB	BI	CMD	Comment
	X										FPWR	set DC cycle time
	X										FPWR	set DC start time
	X										FPWR	set DC activation
	X										FPWR	set DC latch0 cfg

附图 187: 分布式时钟：错误消息

因此，对于 EL6695，EtherCAT 主站应避免写入这些寄存器，或者忽略写入命令的 WcState。在 EL6695-ESI 中，此 DC 特性通过标志 TimeLoopControlOnly = TRUE 来标识，EtherCAT 主站可以据此进行处理。

6.8 在线扫描

在 EL6695 中，每一侧都有 128 kbytes 的存储空间，用于存储可通过 FoE 进行写入和读取的任意文件。

此存储空间可用于保存仿真 ESI，以确保主站能够接收到完整的 ESI（EtherCAT 从站信息）并在其配置中使用

附录：示例

6.9 EL6692 兼容模式

EL6695 具有所谓的 EL6692 兼容模式。

这些模式的运行方式与常见的 EL6692 相同。该模式使得性能更高的 EL6695 能够用于为 EL6692 接口而设计的 EtherCAT 主站及应用软件。在这些模式下，基本上提供相同的默认过程数据。

6.9.1 带 PDO 映射与分配的模块操作

这种模块操作基本上是标准模式，对于新款 EL6695 也是如此。有关说明，请参见“对称 PDO 映射”部分。

6.10 EL6695 性能模式

6.10.1 PDO 映射的基本原理

已过时

还请参阅有关此

■ 对称 PDO 映射 [► 132]

6.10.2 无 Init 命令

此模式主要面向仅用于过程数据传输的外部主站。不得勾选“过程数据”选项卡中的自动 PDO 分配和映射 (automatic PDO assignment and mapping) 的复选框 (请参见选择性 PDO 映射 [► 144] 部分)。

然后可以在相应的端子模块侧声明变量。例如，如果声明了三个字节输出变量并重新启动设备 (在 TwinCAT 中：重新加载设备)，则 PD 输出对象的结构可以在相应的 CoE 对象 0x1608 中查看。其长度为 24 位，包含三个字节变量。

1608:0	IO RxPDO-Map 8	RW	> 1 <
1608:01	SubIndex 001	RW	0x7000:01, 24

附图 188: 24 位 CoE 对象 0x1608

在端子模块的另一侧，数据交换的分配与映射也应禁用，以确保端子模块能够正确解释数据，并且期望的是一个 24 位的条目，而不是三个 8 位的条目。此外，远程侧的过程数据长度必须完全匹配。在本示例中，远程侧的 CoE 对象 0x1A01 映射应予以禁用 (请参见对称 PDO 映射 [► 132] 部分)，否则会导致假设的输入数据量不同，从而使数据交换无法正常运行。

6.10.3 对象描述下载

EtherCAT 通信设备具备一个 CoE 目录，作为内部功能和通信的对象及参数的概览。EL6695 桥接端子上电初始状态下自带专属对象字典 (OD)，该对象字典适配 EL6695 本身。

通过对对象字典 (OD) 进行整体或部分修改，可以在 EL6695 端子模块上定义用户特定的设备。通过 TwinCAT System Manager 中的扩展插件，可以根据需要创建完整或部分的 OD，并将其保存为 *.coe 文件，这是倍福的内部文件格式。

6.10.4 设备仿真

在设备仿真过程中，在桥接侧会仿真一个与桥接端子模块不同的 EtherCAT 从站。从物理层面看，EL6695 桥接端子模块仍然是 EtherCAT 从站，并且始终会被重置回此原始状态。

要仿真的 EtherCAT 从站以 ESI 的形式加载到所选的 EL6695 侧。EL6695 接管这些数据，然后对外以标识数据、PDO 和 CoE 的形式仿真该从站。然而，这仅适用于形式上的呈现，不涉及 (时间相关的) 行为，因为要仿真行为需要修改 EL6695 的固件。被仿真从站的“行为”必须通过另一侧进行映射，并通过该侧提供或接收数据。

由于以此方式呈现的从站也必须在仿真模式下通过 EtherCAT ETG 一致性测试，因此用户必须确保在仿真模式下仅将有效/已认证/可认证的 ESI 加载到 EL6695。因此，必须遵守 ETG 规则 ETG1000.6 (映射) 和 ETG2000 (ESI 规范)。

EmulationMode (仿真模式) 可在单侧或双侧使用。

● 扩展插件 (TwinCAT 中的用户界面)



在处于仿真的从站中，所谓的扩展插件 (TwinCAT 中的用户界面) 不再可用。

如需更多信息，请联系倍福支持中心，并提供相关数据（TwinCAT 版本、所需性能值、周期、数据量等）

6.11 应用特定变量定义

如果 EL6695 端子模块在默认配置下不提供过程数据，则必须根据具体的应用要求用配置工具进行设置。如果所使用的 EtherCAT 主站不支持可变 PDO 映射，则用户可以创建自己的 ESI 设备描述文件，供特定的配置工具使用。



技术性

为符合相应的依据文档，必须遵守 ETG 规则 ETG1000.6（映射）和 ETG2000（ESI 规范）。



法律性

此外，设备描述必须通过 ETG 一致性测试；相应的用户/创建者必须确保这些拟发布的描述文件获得 ETG 的批准。

7 附录

7.1 EtherCAT AL 状态代码

详细信息请参见 [EtherCAT系统描述](#)。

7.2 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。以下概览显示一个固件可以运行的对应硬件版本。

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 [▶ 184] 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！

EL6695

硬件 (HW)	固件 (HW)	修订版本号	发布日期
02 – 13*	03	原边： EL6695-0000-0001 副边： EL6695-0002-0001	2015/02
	04	原边： EL6695-0000-0002 副边： EL6695-0002-0002	2015/06
		原边： EL6695-0000-0003 副边： EL6695-0002-0003	2015/06
	05	原边： EL6695-0000-0004 副边： EL6695-0002-0004	2015/07
	06		2015/08
	07	原边： EL6695-0000-0005 副边： EL6695-0002-0005	2017/10
	08		2018/04
	09	原边：	2021/07
	10	EL6695-0000-0006，	2022/02
	11	副边：	2023/12
	12*	EL6695-0002-0006	2024/01

*) 这是编制本文件时兼容的固件/硬件版本。请在倍福网页上查看是否有更多最新文档。

7.3 固件更新

本节介绍了倍福 ED/EF, 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和发布的。

- 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 NOVDRAM（掉电保持存储器选件）。

- 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。

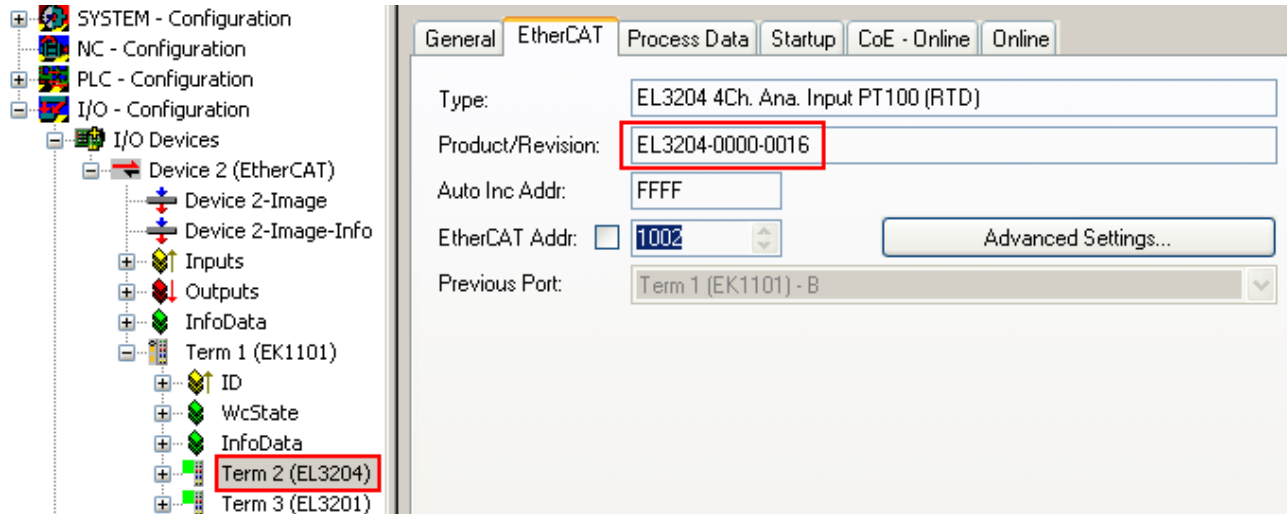
7.3.1 设备描述 ESI 文件/XML

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 189: 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](#)。

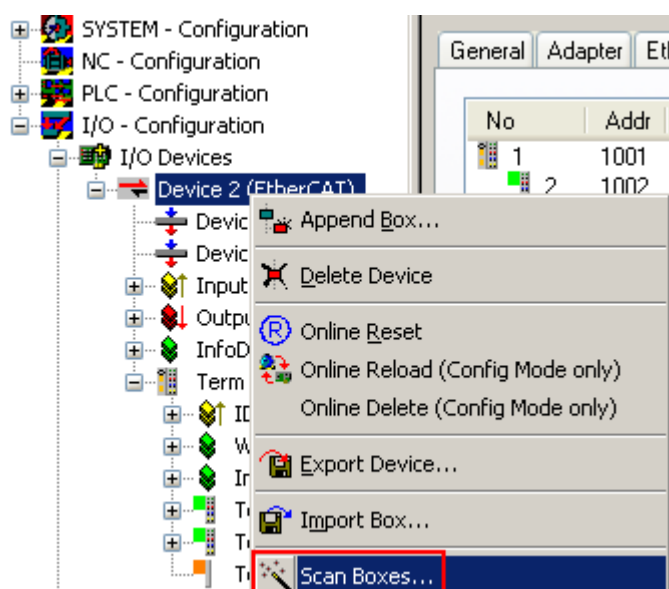
● XML/ESI 描述的更新



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：



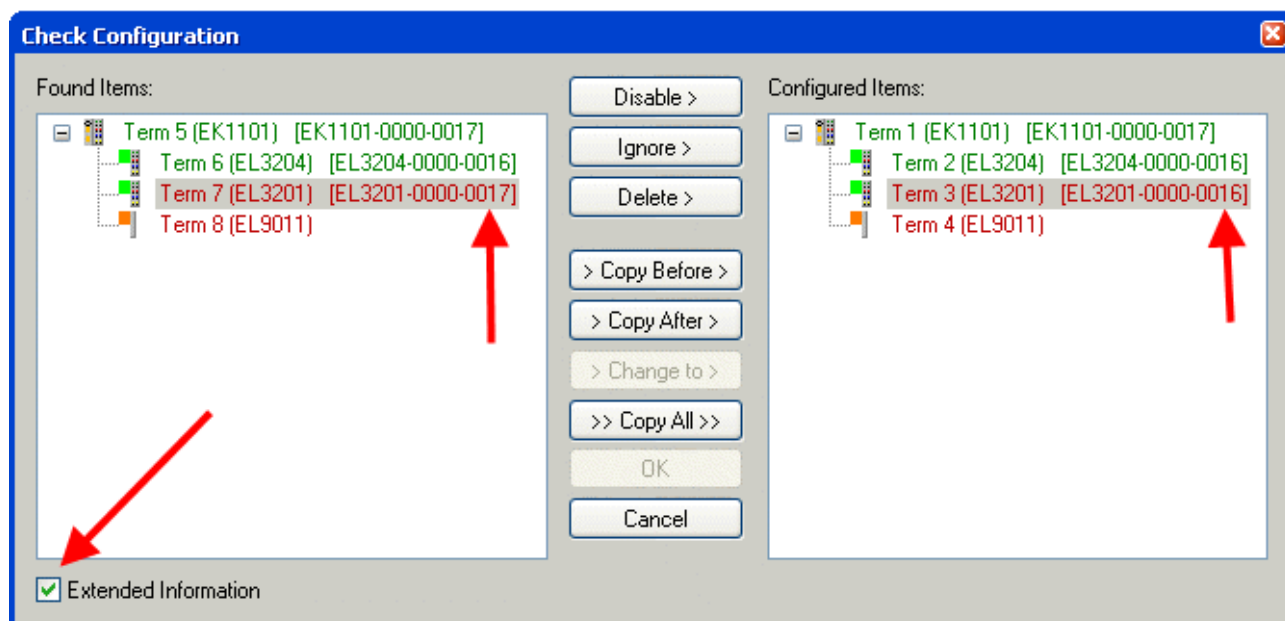
附图 190: 右键单击 EtherCAT Device 扫描下级从站

如果找到的内容与配置的内容相符，则显示



附图 191: 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。



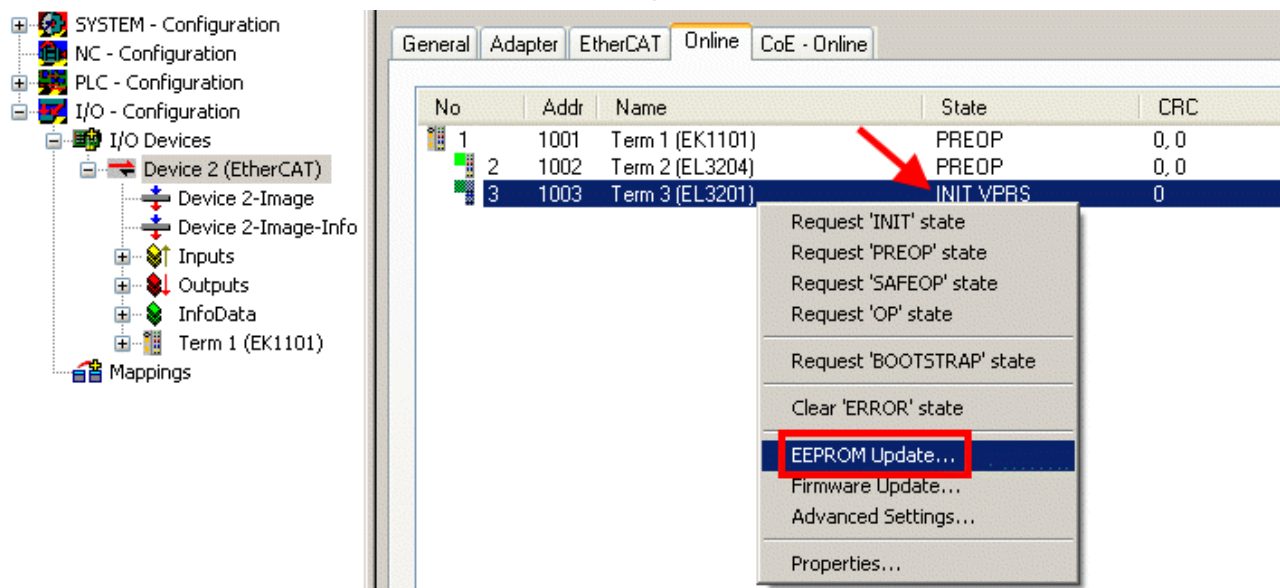
附图 192: 更改对话框

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

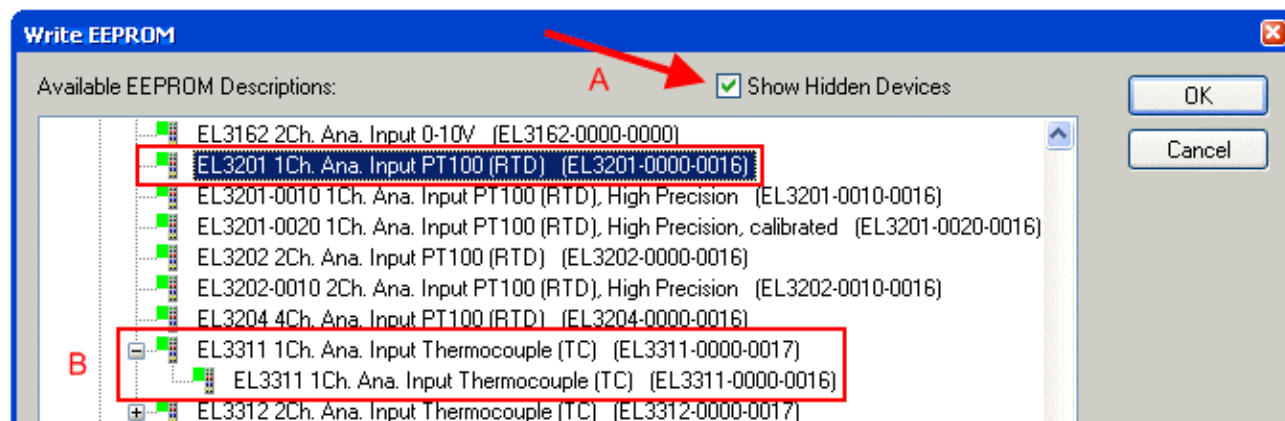
ESI/EEPROM 标识符可以在 TwinCAT 下按如下方式更新：

- 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 193: 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 194: 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。

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。

7.3.2 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

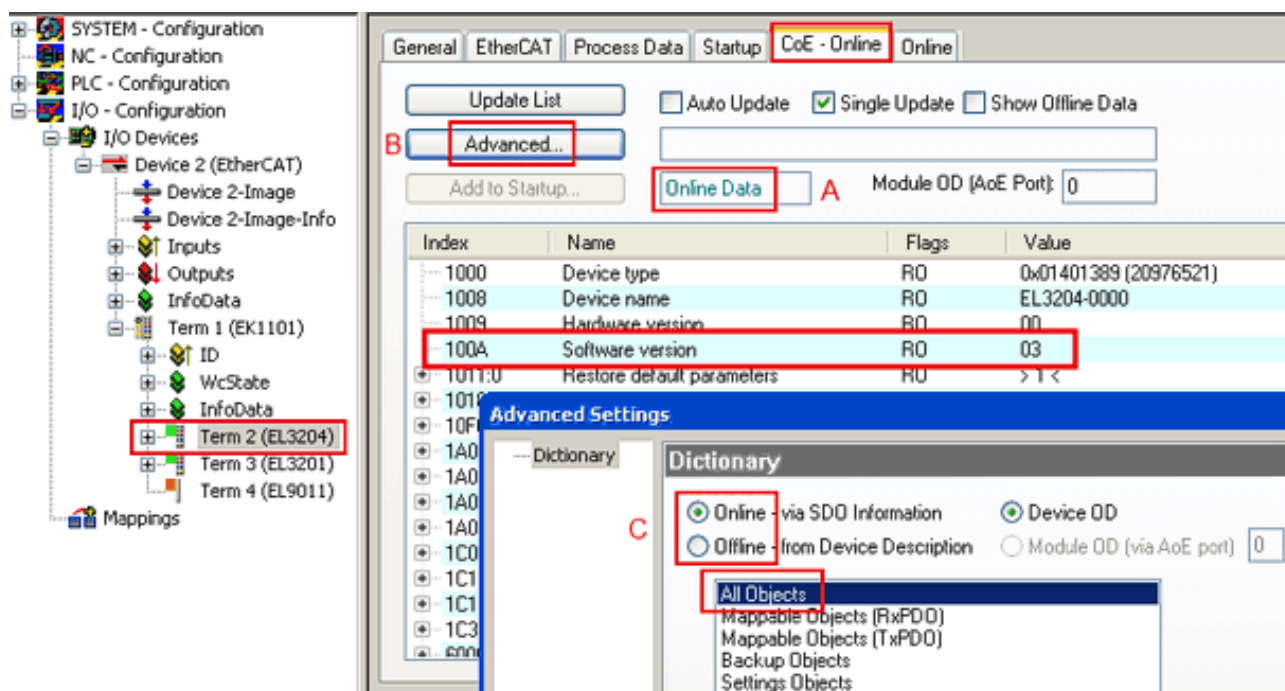
可用的 CoE 目录有两套：

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 EL5xx.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 195: 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。

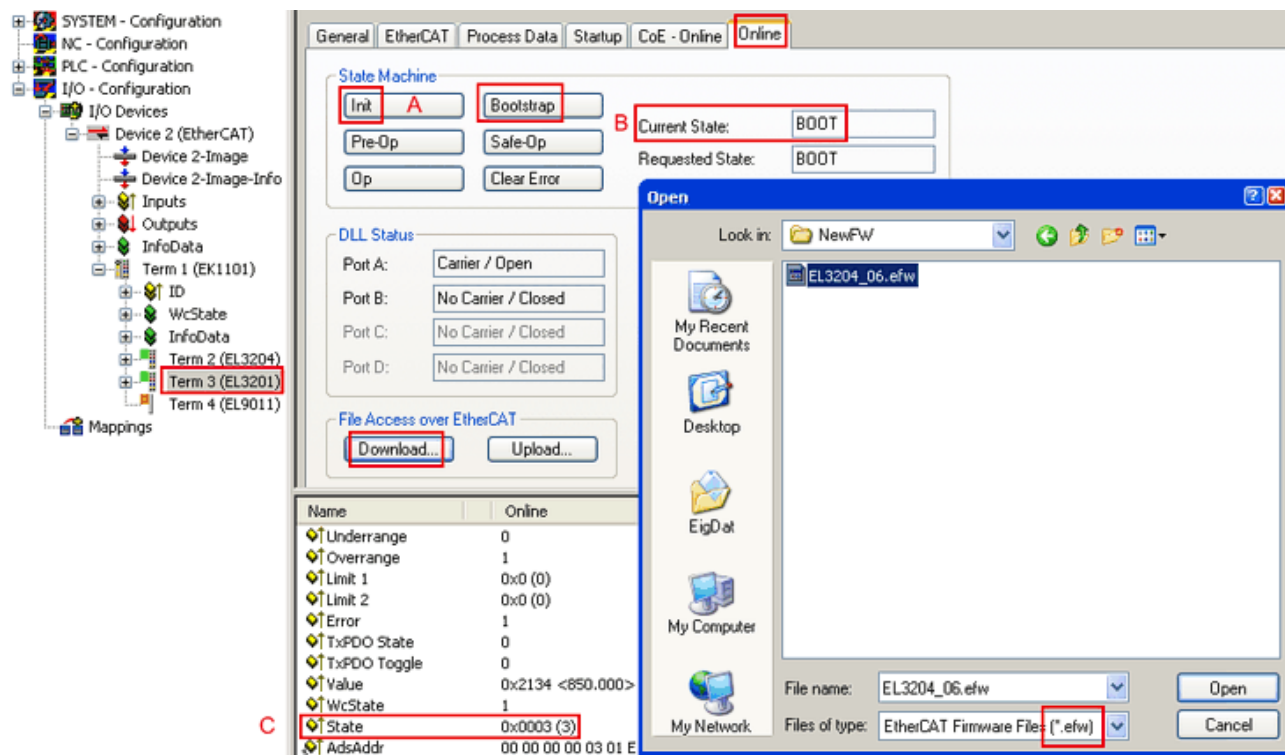
7.3.3 更新从站处理器的固件 *.efw

i

CoE 目录

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。

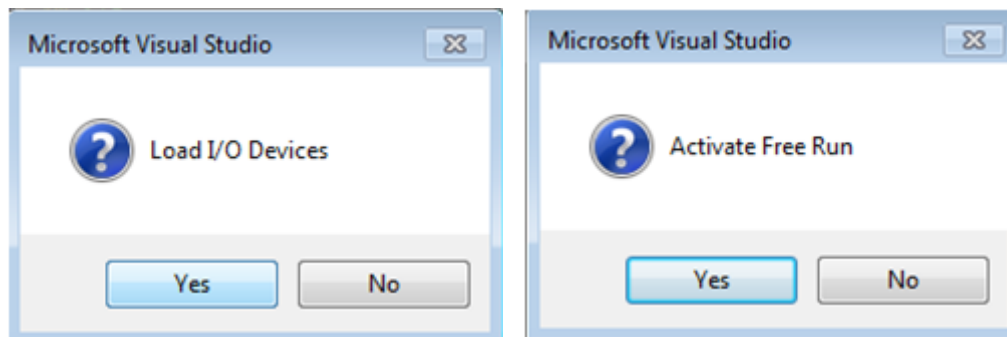
切换到 Online 选项卡，更新从站处理器的固件，参见图固件更新。



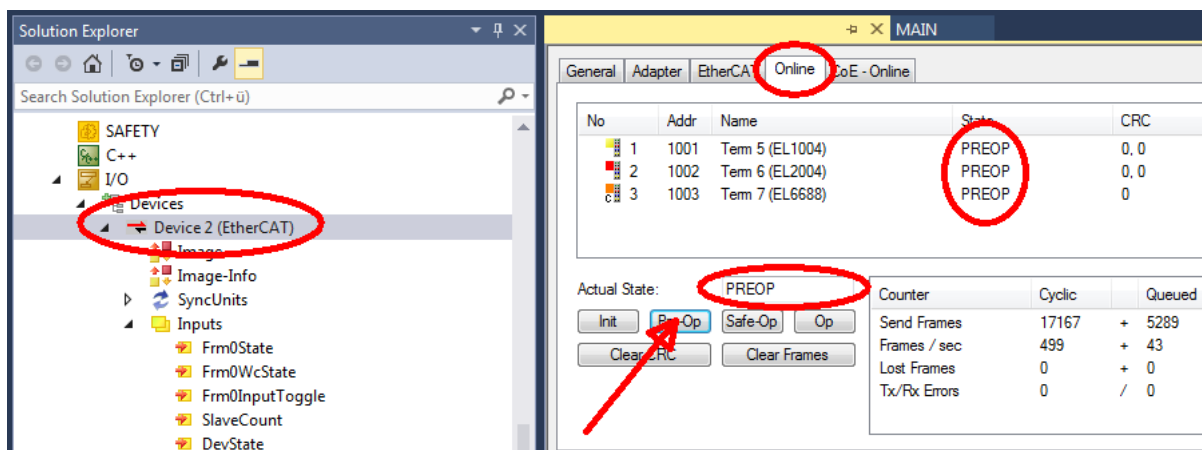
附图 196: 固件更新

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 ≥ 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。

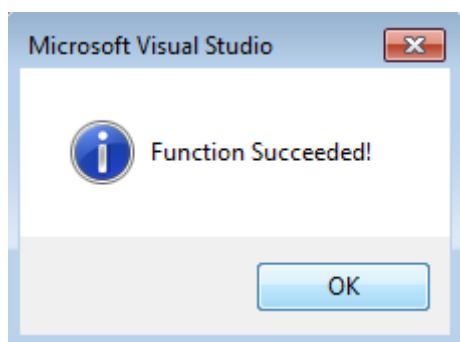


- 将 EtherCAT 主站切换到 PreOP



- 将从站切换到 INIT (A)
- 将从站切换到 BOOTSTRAP

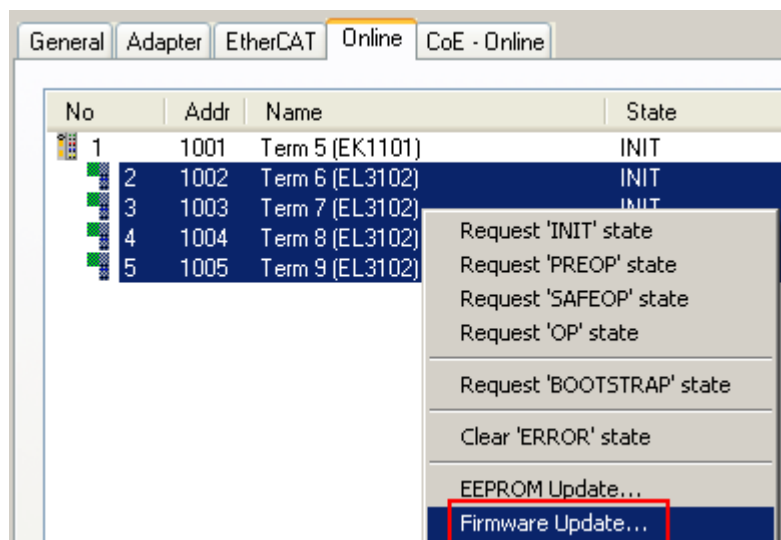
- 检查当前状态（B、C）
- 下载新的 *efw文件（一直等待，直到下载结束）。通常不需要密码。



- 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。

7.3.4 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。

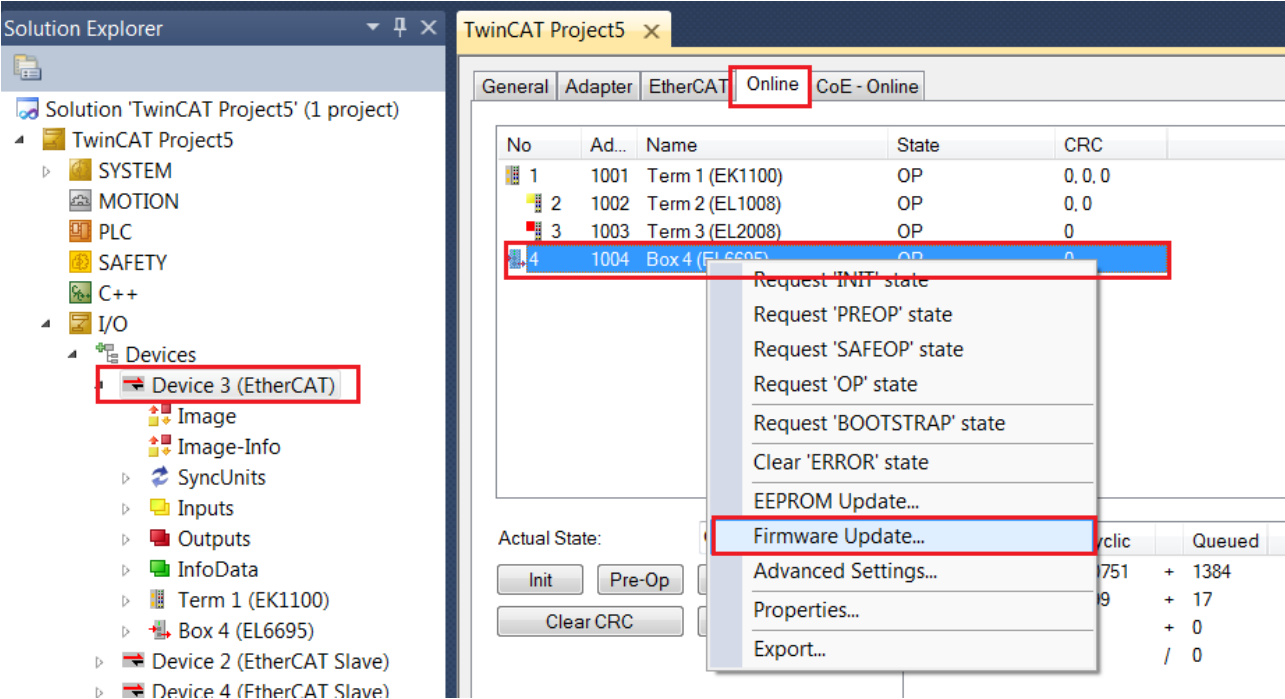


附图 197: 多重选择和固件更新

选择所需的从站，如上所述在 BOOTSTRAP 模式下进行固件更新。

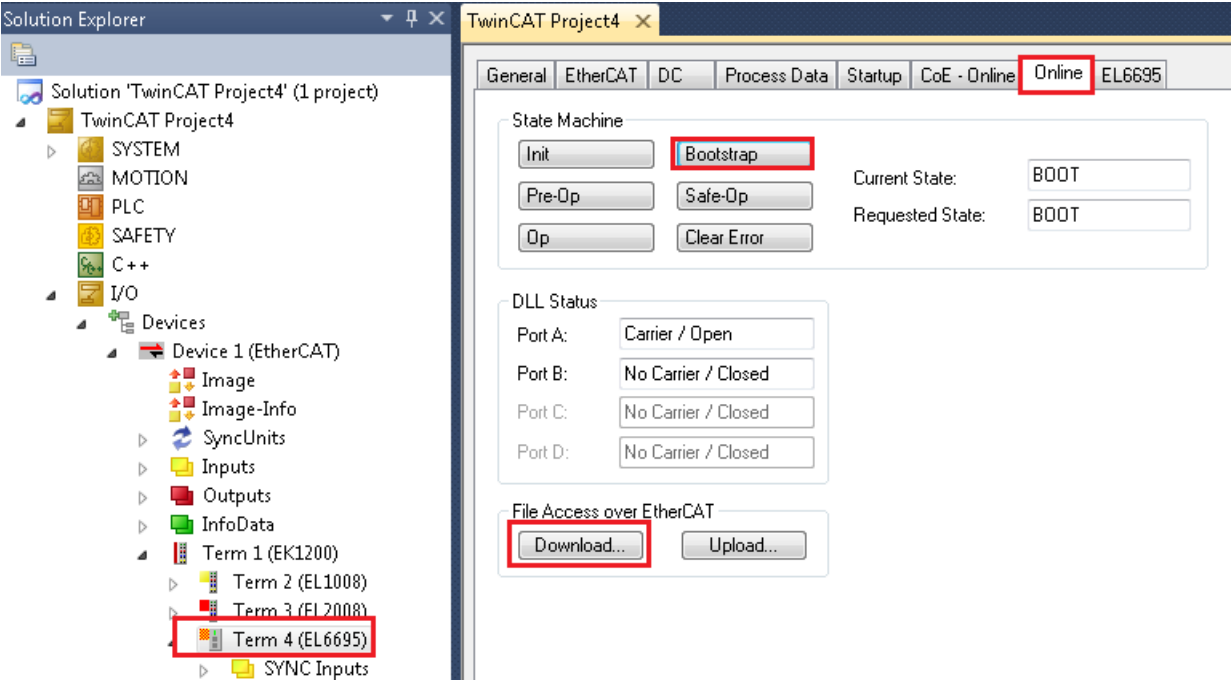
7.4 固件 (FW) /FPGA 更新的补充说明

通过两种方式可以进行固件更新：一种方法是通过“Online(在线)”选项卡从列表中选择设备 (EL6695)，然后在下拉菜单中选择“Firmware Update..”(固件更新..)。然后可以通过相应路径加载 *.efw 文件；不需要密码，但也可以选择设置密码。接下来，应将端子模块的两侧都设置为 INIT 状态，之后再将它们重新设置为 OP 状态。



附图 198: 通过“Online(在线)”选项卡进行固件更新

第二种方法是通过端子模块和“Online(在线)”选项卡直接进行固件更新。为此，在所选侧将端子模块设置为 BOOT 状态。



附图 199: 设置 BOOT 状态

然后，可以通过File Access over EtherCAT”下的“Download(下载)”按钮将 *.efw 文件下载到端子模块。然后将端子模块设置为 INIT 状态，再恢复为 OP 状态。另一侧也需要设置为 INIT 状态，再恢复为 OP 状态。

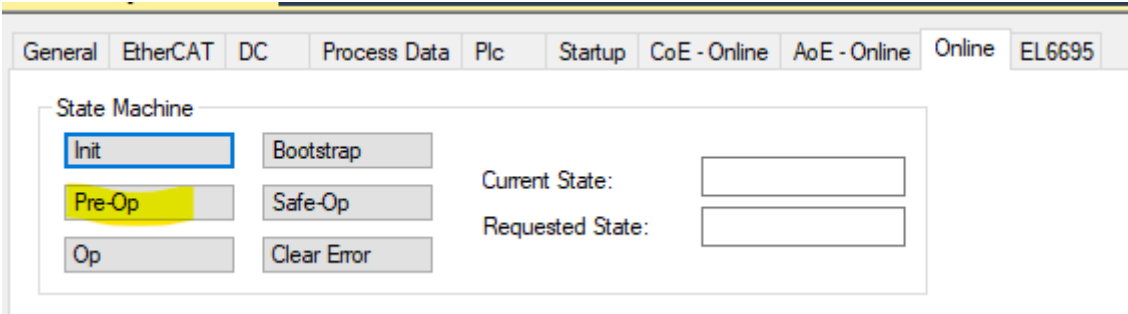
注意

FW 更新注意事项（测试版 FW）

- 更新前，请确保 EtherCAT 总线运行正常，无丢帧或链路丢失。
- 如果系统中有多于一个 EL6695 端子模块，一次只能对一个 EL6695 端子模块执行 FW 更新。否则设备将发生不可逆的损坏，并且该端子模块必须由倍福售后服务部门进行更换。
- 在更新期间必须避免电源故障。

对于 FPGA 更新，请按以下步骤操作：

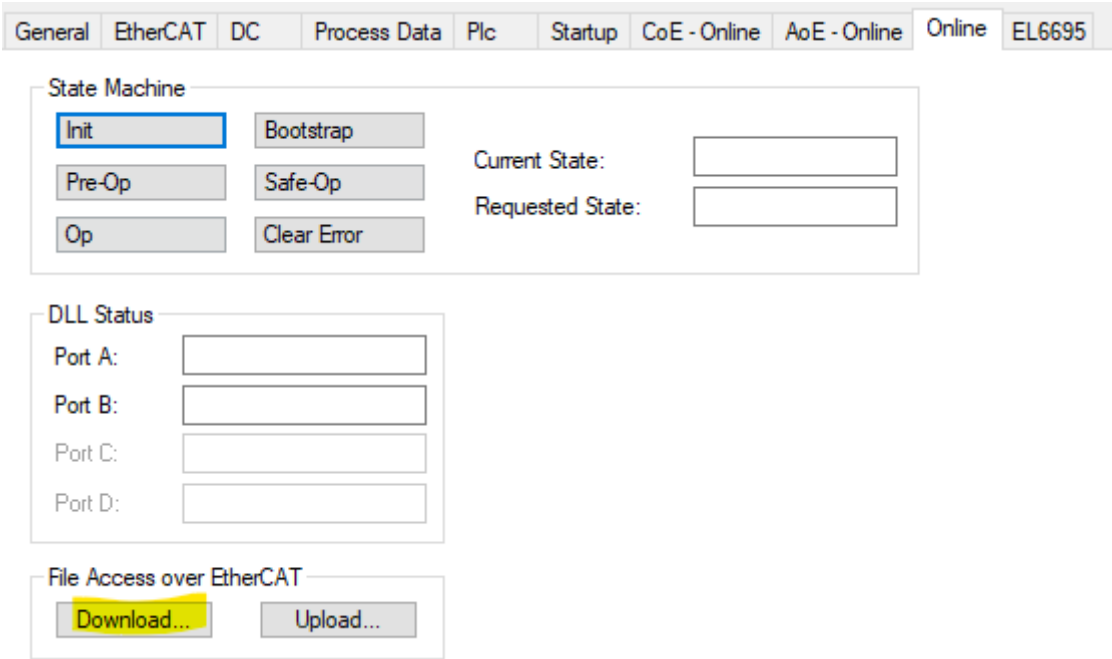
- 前提条件：EL6695 已位于配置树中的原边或副边，TwinCAT 处于配置模式/自由运行模式
- 在 Pre-OP 之后，切换端子模块：



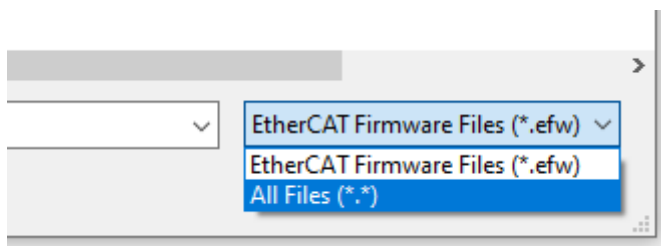
- 将值 0x1006695 写入 CoE 对象 0xF008 代码字：

F000:0	Modular device profile	RO	> 2 <
F008	Code word	RW	0x00000000 (0)
F630:0	Active TxPdo Info	RO	> 8 <

- 在 “Online（在线）” 选项卡中的 “File Access over EtherCAT” 下选择 Download：



- 选择 *.rbf FPGA 文件, “All Files (*.*) (所有文件 (*.*))” :



- 下载可能需要几分钟时间

7.5 恢复出厂状态

EL6695 端子模块支持两种重置方法：

- 通过命令
- 通过 CoE 对象 0x1011（FW07 及以下版本）

7.5.1 常规设备重置

- 通过在 CoE 对象 0xF008 中输入代码 0x33336695，可以重置已保存在 EL6695 端子模块中的设置。
- 将代码传输到对象 0xF008 后，您必须将 EL6695 设置为 INIT 模式。

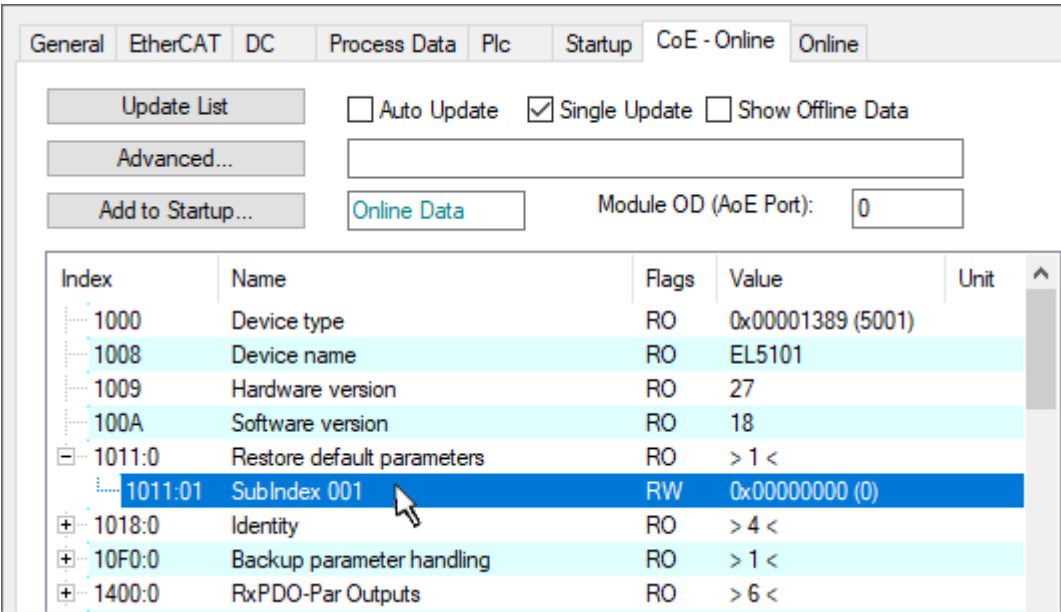
警告

与网络的连接会暂时中断

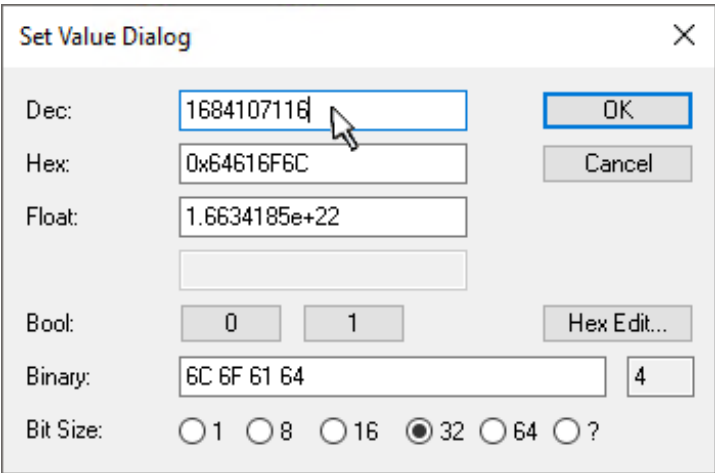
EL6695 在代码传输后会执行彻底系统重置，这会导致与网络的连接暂时中断。在生产环境中，这一点尤为重要，因为这种暂时但意外的中断可能会造成破坏性影响。

7.5.2 恢复出厂状态

某些 EtherCAT 设备（"从站"、"子设备"）以持久化（非易失）方式在本地保存参数或数据。要重置这些参数，可通过 EtherCAT 主站（如 TwinCAT）使用 CoE 对象 "恢复默认参数"子索引 001（见图：选择 "恢复默认参数" PDO）



附图 200: 选择 Restore default parameters



附图 201: 在 Set Value dialog 中输入一个恢复值

双击 SubIndex 001，进入设置值对话框。

目的是将持续存储在设备中,用户可以更改的功能控制 "参数"（设置、值）重置为默认状态。

易失性（非断电安全）"参数" 被重置

- 通过重新开机/关机
- 通过 INIT 更改 EtherCAT 状态

除非设备文件中另有说明。根据设备和 FW 版本的不同，可提供一个或两个复位功能；有关可用性，请参阅相应的设备文档。

注意

更改现有校验和
参数范围的任何现有校验和都将（当然）会因重置而改变。

复位级别 1：功能复位 "加载"

假设重置功能的使用频率较高，可能会多次使用，例如在调试期间。

将重置值 **1684107116** 输入字段 Dec 中，或将数值 **0x64616F6C** 输入字段 Hex 中(ASCII: “load”)，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。
例外：通过密码受写保护的對象
- 根据对象字典的大小，这个过程可能需要几毫秒到 > 1 秒。
- 只有直接对从站的 Online CoE 进行重置，才能成功恢复出厂值。在 Offline 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。
- 重置不需要单独确认。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。

注意

根据设备的不同，个别数值不会重置
根据设备的不同，有些值在 "加载" 时故意不重置，例如，如果这些值是调试或设备基本功能所必需的，而且确定这些值（删除后）需要花费大量精力。



替代的恢复值

对于某些较旧的终端（2007 年左右或更早创建的 FW），备份对象可通过替代还原值进行更改：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！

重置级别 2：数据重置 "擦除"

假设很少使用数据重置，例如在产品退役时。

将重置值 **2003398757** 输入字段 Dec 中，或将数值 **0x77697065** 输入字段 Hex 中(ASCII: "wipe")，并按 OK 确认(图：在 Set Value dialog 中输入恢复值)。

- 该功能的目的是完全重置（删除数据）出厂设置，以便之后无法读取用户特定设置。该功能的目的是完全重置删除数据出厂设置，之后无法读取用户特定设置
- 它会影响以下参数区域（如果设备上有的话）
 - CoE (参数)
 - SoE (参数)
 - FoE (设备上的文件)
 - 寄存器值（如果持续存储）
 - 用户存储器，如 NOVRAM（如有）
 - 其他数据区域,如果有的话
- 即使值/数据受到保护，例如有用户定义的写保护密码，也会被删除。
- 下列内容无法重置：
 - FW 版本和 EtherCAT 修订状态；设备上仍安装有任何更新版本
 - 电气健康数据，如设备统计数据（CoE: 0x1020）、各类运行时间计数器、可能与环境温度或负载条件有关的直方图或峰值, ...
 - 供应商校准数据受供应商密码保护，无法更改。供应商校准功能已启用供应商校准功能已启用

7.6 技术支持和服务

倍福公司及其合作伙伴在世界各地提供全面的技术支持和服务，对与倍福产品和系统解决方案相关的所有问题提供快速有效的帮助。

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
网址：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
网址：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
网址：www.beckhoff.com

Trademark statements

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

Third-party trademark statements

CANopen and CANopen FD are registered trademarks of CAN in AUTOMATION - International Users and Manufacturers Group e.V.

ControlNet®, DeviceNet and EtherNet/IP are trademarks of ODVA, Inc.

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

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

更多信息:
www.beckhoff.com/EL6695

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

