BECKHOFF New Automation Technology

Documentation | EN
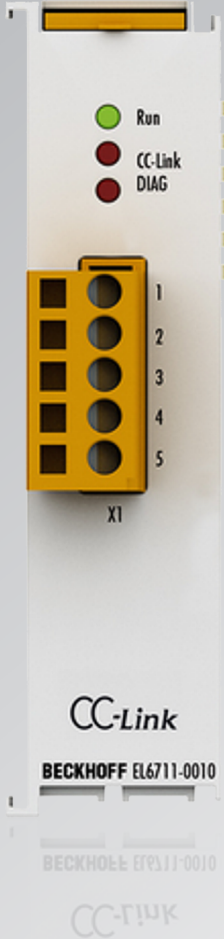
EL6711-0010

CC-Link Slave Terminal

Table of contents

1   Foreword

1.1   Notes on the documentation

Intended audience

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.
The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without prior announcement.

No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH. Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents: EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702 with corresponding applications or registrations in various other countries.

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany.

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.
The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.
Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

BECKHOFF

1.2   Guide through documentation

	NOTICE
	Further components of documentation The documentations named in the following table are further components of the complete documentation. These documentations are required for the use of EtherCAT Terminals.

No.	Title	Description
[1]	EtherCAT System Documentation	System description EtherCAT field bus • System overview • EtherCAT basics • Cable redundancy • Hot Connect • Distributed Clocks • Terminal configuration
[2]	Explosion protection for terminal systems	Notes on the use of Beckhoff terminal systems in potentially explosive atmospheres according to ATEX and IECEx
[3]	Infrastructure for EtherCAT / Ethernet	Technical recommendations and notes on design, execution and testing
[4]	Control Drawing I/O, CX, CPX	Connection diagrams and Ex marking

Version: 1.1   EL6711-0010

1.3 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation and drive engineering who are familiar with the applicable national standards.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER
Hazard with high risk of death or serious injury.

⚠ WARNING
Hazard with medium risk of death or serious injury.

⚠ CAUTION
There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE
The environment, equipment, or data may be damaged.

Information on handling the product

> **i** This information includes, for example:
> recommendations for action, assistance or further information on the product.

1.4   Documentation issue status

Version	Comment
1.1	• Chapter "Recommended mounting rails" removed • Update revision status
1.0	• Addenda, minor corrections • 1st public issue
0.3	• Addenda
0.2	• Minor corrections
0.1	• Preliminary documentation for EL6711-0010

1.5   Version identification of EtherCAT devices

1.5.1   General notes on marking

Designation

A Beckhoff EtherCAT device has a 14-digit designation, made up of

- family key
- type
- version
- revision

Example	Family	Type	Version	Revision
EL3314-0000-0016	EL terminal 12 mm, non-pluggable connection level	3314 4-channel thermocouple terminal	0000 basic type	0016
ES3602-0010-0017	ES terminal 12 mm, pluggable connection level	3602 2-channel voltage measurement	0010 high-precision version	0017
CU2008-0000-0000	CU device	2008 8-port fast ethernet switch	0000 basic type	0000

Notes

- The elements mentioned above result in the **technical designation**. EL3314-0000-0016 is used in the example below.

- EL3314-0000 is the order identifier, in the case of "-0000" usually abbreviated to EL3314. "-0016" is the EtherCAT revision.

- The **order identifier** is made up of
  - family key (EL, EP, CU, ES, KL, CX, etc.)
  - type (3314)
  - version (-0000)

- The **revision** -0016 shows the technical progress, such as the extension of features with regard to the EtherCAT communication, and is managed by Beckhoff.
  In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation.
  Associated and synonymous with each revision there is usually a description (ESI, EtherCAT Slave Information) in the form of an XML file, which is available for download from the Beckhoff web site. From 2014/01 the revision is shown on the outside of the IP20 terminals, see Fig. *"EL5021 EL terminal, standard IP20 IO device with batch number and revision ID (since 2014/01)"*.

- The type, version and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

1.5.2   Version identification of EL terminals

The serial number/ data code for Beckhoff IO devices is usually the 8-digit number printed on the device or on a sticker. The serial number indicates the configuration in delivery state and therefore refers to a whole production batch, without distinguishing the individual modules of a batch.

Structure of the serial number: **KK YY FF HH**

KK - week of production (CW, calendar week)
YY - year of production
FF - firmware version
HH - hardware version

Example with serial number 12 06 3A 02:

12 - production week 12
06 - production year 2006
3A - firmware version 3A
02 - hardware version 02

Fig. 1: EL2872 with revision 0022 and serial number 01200815

BECKHOFF

1.5.3 Beckhoff Identification Code (BIC)

The Beckhoff Identification Code (BIC) is increasingly being applied to Beckhoff products to uniquely identify the product. The BIC is represented as a Data Matrix Code (DMC, code scheme ECC200), the content is based on the ANSI standard MH10.8.2-2016.
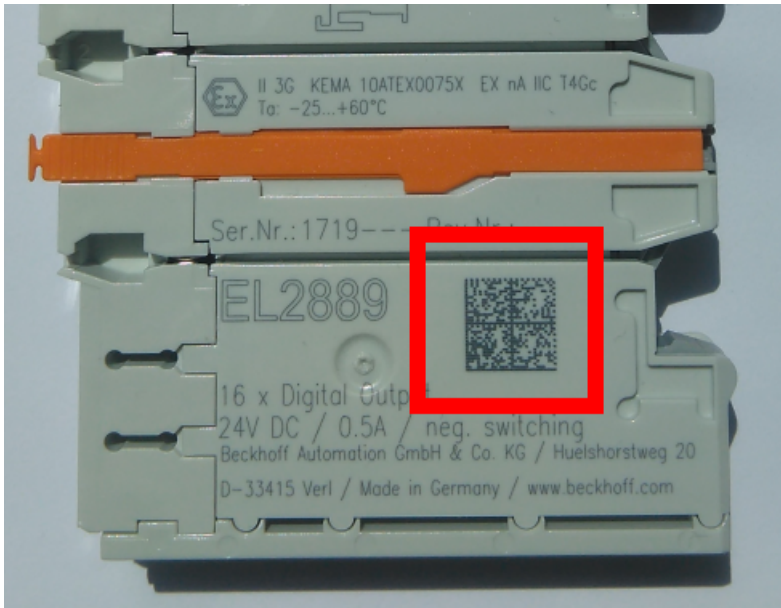

Fig. 2: BIC as data matrix code (DMC, code scheme ECC200)

The BIC will be introduced step by step across all product groups.

Depending on the product, it can be found in the following places:

- on the packaging unit
- directly on the product (if space suffices)
- on the packaging unit and the product

The BIC is machine-readable and contains information that can also be used by the customer for handling and product management.

Each piece of information can be uniquely identified using the so-called data identifier (ANSI MH10.8.2-2016). The data identifier is followed by a character string. Both together have a maximum length according to the table below. If the information is shorter, spaces are added to it.

Following information is possible, positions 1 to 4 are always present, the other according to need of production:

Posi-tion	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	**Beckhoff order number**	1P	8	1P072222
2	Beckhoff Traceability Number (BTN**)**	**Unique serial number, see note below**	SBTN	12	SBTNk4p562d7
3	Article description	**Beckhoff article description, e.g. EL1008**	1K	32	1KEL1809
4	Quantity	**Quantity in packaging unit, e.g. 1, 10, etc.**	Q	6	Q1
5	Batch number	Optional: Year and week of production	2P	14	2P401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	51S678294
7	Variant number	Optional: Product variant number on the basis of standard products	30P	32	30PF971, 2*K183
...					

Further types of information and data identifiers are used by Beckhoff and serve internal processes.

Structure of the BIC

Example of composite information from positions 1 to 4 and with the above given example value on position 6. The data identifiers are highlighted in bold font:

1P072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

Accordingly as DMC:

Fig. 3: Example DMC **1P**072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

BTN

An important component of the BIC is the Beckhoff Traceability Number (BTN, position 2). The BTN is a unique serial number consisting of eight characters that will replace all other serial number systems at Beckhoff in the long term (e.g. batch designations on IO components, previous serial number range for safety products, etc.). The BTN will also be introduced step by step, so it may happen that the BTN is not yet coded in the BIC.

NOTICE
This information has been carefully prepared. However, the procedure described is constantly being further developed. We reserve the right to revise and change procedures and documentation at any time and without prior notice. No claims for changes can be made from the information, illustrations and descriptions in this information.

1.5.4 Electronic access to the BIC (eBIC)

Electronic BIC (eBIC)

The Beckhoff Identification Code (BIC) is applied to the outside of Beckhoff products in a visible place. If possible, it should also be electronically readable.

Decisive for the electronic readout is the interface via which the product can be electronically addressed.

K-bus devices (IP20, IP67)

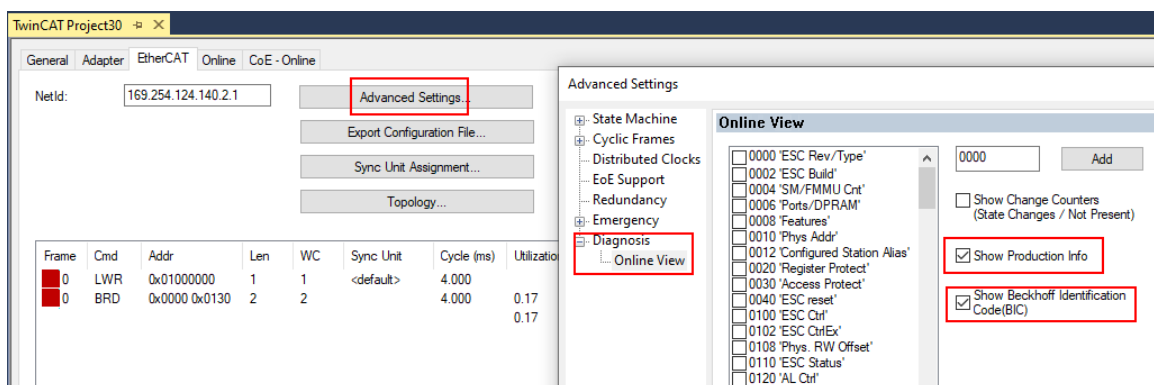Currently, no electronic storage and readout is planned for these devices.

EtherCAT devices (IP20, IP67)

All Beckhoff EtherCAT devices have a so-called ESI-EEPROM, which contains the EtherCAT identity with the revision number. Stored in it is the EtherCAT slave information, also colloquially known as ESI/XML configuration file for the EtherCAT master. See the corresponding chapter in the EtherCAT system manual (Link) for the relationships.

The eBIC is also stored in the ESI-EEPROM. The eBIC was introduced into the Beckhoff I/O production (terminals, box modules) from 2020; widespread implementation is expected in 2021.

The user can electronically access the eBIC (if existent) as follows:

- With all EtherCAT devices, the EtherCAT master (TwinCAT) can read the eBIC from the ESI-EEPROM
    - From TwinCAT 3.1 build 4024.11, the eBIC can be displayed in the online view.
    - To do this,
      check the checkbox "Show Beckhoff Identification Code (BIC)" under
      EtherCAT → Advanced Settings → Diagnostics:

    - The BTN and its contents are then displayed:

    - Note: as can be seen in the illustration, the production data HW version, FW version and production date, which have been programmed since 2012, can also be displayed with "Show Production Info".
    - Access from the PLC: From TwinCAT 3.1. build 4024.24 the functions *FB_EcReadBIC* and *FB_EcReadBTN* are available in the Tc2_EtherCAT Library from v3.3.19.0 for reading into the PLC..
- In the case of EtherCAT devices with CoE directory, the object 0x10E2:01 can additionally by used to display the device's own eBIC; the PLC can also simply access the information here:

◦ The device must be in PREOP/SAFEOP/OP for access:

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	1P158442SBTN0008ekp1KELM3704	Q1	2P482001000016
10F0:0	Backup parameter handling	RO	> 1 <		
10F3:0	Diagnosis History	RO	> 21 <		
10F8	Actual Time Stamp	RO	0x170bfb277e		

◦ The object 0x10E2 will be introduced into stock products in the course of a necessary firmware revision.

◦ From TwinCAT 3.1. build 4024.24 the functions *FB_EcCoEReadBIC* and *FB_EcCoEReadBTN* are available in the Tc2_EtherCAT Library from v3.3.19.0 for reading into the PLC.

• For processing the BIC/BTN data in the PLC, the following auxiliary functions are available in *Tc2_Utilities* from TwinCAT 3.1 build 4024.24 onwards

  ◦ F_SplitBIC: The function splits the Beckhoff Identification Code (BIC) sBICValue into its components based on known identifiers and returns the recognized partial strings in a structure ST_SplitBIC as return value.

  ◦ BIC_TO_BTN: The function extracts the BTN from the BIC and returns it as a value.

• Note: in the case of electronic further processing, the BTN is to be handled as a string(8); the identifier "SBTN" is not part of the BTN.

• Technical background
The new BIC information is additionally written as a category in the ESI-EEPROM during the device production. The structure of the ESI content is largely dictated by the ETG specifications, therefore the additional vendor-specific content is stored with the help of a category according to ETG.2010. ID 03 indicates to all EtherCAT masters that they must not overwrite these data in case of an update or restore the data after an ESI update.
The structure follows the content of the BIC, see there. This results in a memory requirement of approx. 50..200 bytes in the EEPROM.

• Special cases

  ◦ If multiple, hierarchically arranged ESCs are installed in a device, only the top-level ESC carries the eBIC Information.

  ◦ If multiple, non-hierarchically arranged ESCs are installed in a device, all ESCs carry the eBIC Information.

  ◦ If the device consists of several sub-devices with their own identity, but only the top-level device is accessible via EtherCAT, the eBIC of the top-level device is located in the CoE object directory 0x10E2:01 and the eBICs of the sub-devices follow in 0x10E2:nn.

PROFIBUS, PROFINET, DeviceNet devices etc.

Currently, no electronic storage and readout is planned for these devices.

2 Product overview
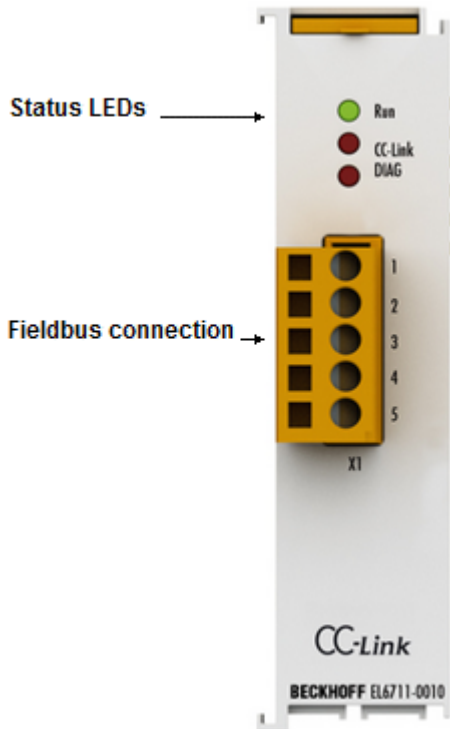
2.1 Introduction

Fig. 4: EL6711-0010

CC-Link slave terminal

As an EtherCAT slave, the EL6711-0010 CC-Link slave terminal enables data exchange between EtherCAT and CC-Link.

It is a slave on both the EtherCAT and CC-Link sides, so that an EtherCAT master can exchange data with a CC-Link master via this terminal. All CC-Link-specific settings such as station address, baud rate, data size and others are parameterized via the EtherCAT master.

On the CC-Link side, there is no need for complex configurations, and no further parameterization is required via the CC-Link master.

Quick links

- EtherCAT basics
- Basic Function Principles [▶ 111]
- Creation of the TwinCAT EL6711-0010 configuration [▶ 115]
- CoE object description and parameterization [▶ 124]

2.2   Technical data

Technical data	EL6711-0010				
Fieldbus	CC-Link				
CC-Link version	V 1.10 (or V 1.00) or V 2.00				
Function	Slave				
Number of fieldbus channels	1				
Max. process data size	According to CC-Link V 2.00, see table [▶ 17]				
Communication speed	156 kbps	625 kbps	2.5 Mbps	5 Mbps	10 Mbps
Max. total cable length	1200 m	900 m	400 m	160 m	100 m
Bus interface	1 x open-style connector, 5-pin				
Hardware diagnostics	Status LEDs				
Power supply for the electronics	internal				
Current consumption via E-bus	typ. 150mA				
Electrical isolation	500 V (E-Bus / CC-Link)				
Configuration	Via the EtherCAT master				
Weight	approx. 70 g				
Permissible ambient temperature at operation	0...+55 °C				
Permissible ambient temperature for storage	-25...+85 °C				
Permissible relative air humidity	95 % no condensation				
Dimensions (W x H x D)	24 mm x 100 mm x 52 mm				
Mounting [▶ 28]	on 35 mm mounting rail conforms to EN 60715				
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27				
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4				
Protect. rating	IP20				
Installation position	see notice [▶ 31]				
Approvals / markings*	CE, UKCA, EAC				

*) Real applicable approvals/markings see type plate on the side (product marking).

2.3   CC-Link system description

This chapter provides a brief general introduction to the CC-Link network.

CC-Link (Control and Communication Link) was developed by Mitsubishi Electronic Corporation in 1996 as a semi-open, proprietary network to allow its products to communicate with each other in a system. In 2000, Mitsubishi Electronic Corporation specified CC-Link as a completely open network and founded the CLPA (CC-Link Partner Association). Since CC-Link is an open network and the protocols are available to all interested parties, anyone can develop their own CC-Link interfaces. The intention is to increase the worldwide acceptance and interest.

CC-Link is a high-speed, deterministic field network that can process control and information data simultaneously at a maximum speed of 10 Mbps over a maximum distance of up to 100 meters. CC-Link is used for control of individual machines or entire factories and in building automation. Up to 64 stations can communicate with each other in a network. A distinction is made between CC-Link versions 1.00 (or 1.10) and version 2.00. The difference between the versions is in the amount of data transferred. CC-Link version 2.00 supports 8 times the amount of communication data in the system and per CC-Link station.

Source: CLPA, CC-Link Partner Association (https://www.cc-link.org/)

2.3.1   CC-Link specifications

Specifications					
Communication speed	10M/5M/2.5M/625k/156 kbps				
Communication method	Broadcast query method				
Synchronization method	Frame synchronization				
Coding method	NRZI coding				
Format of the transmission link	Bus format (conforms to EIA RS485)				
Transmission format	HDLC-compliant				
Max. number of devices that can be connected	64				
Number of slave stations	1 - 64				
Features	Technical data				
		Version 1.10	Version 2.00		
Maximum number of link addresses	decentralized I/O (RX, RY), bit	2048 bits each mixed*1: 4096 bits	8192 bits each, mixed*2: 16384 bits		
	Decentralized registers (RWr, RWw), Word	256 words each mixed: 512 words	2048 words each mixed*2: 4096 words		
Extended cyclic setting		-	Single	Double	Quadruple	Octuple
Number of link addresses per device	1 occupied station	RX, RY, (bit)	32 (mixed: 64)	32 (mixed: 64)	64 (mixed: 128)	128 (mixed: 256)
		RWr, RWw, (Word)	4 (mixed: 8)	8 (mixed: 16)	16 (mixed: 32)	32 (mixed: 64)
	2 occupied stations	RX, RY, (bit)	64 (mixed: 128)	96 (mixed: 192)	192 (mixed: 384)	384 (mixed: 768)
		RWr, RWw, (Word)	8 (mixed: 16)	16 (mixed: 32)	32 (mixed: 64)	64 (mixed: 128)
	3 occupied stations	RX, RY, (bit)	96 (mixed: 192)	160 (mixed: 320)	320 (mixed: 640)	640 (mixed: 1280)
		RWr, RWw, (Word)	12 (mixed: 24)	24 (mixed: 48)	48 (mixed: 96)	96 (mixed: 192)
	4 occupied stations	RX, RY, (bit)	128 (mixed: 256)	224 (mixed: 448)	448 (mixed: 896)	896 (mixed: 1792)
		RWr, RWw, (Word)	16 (mixed: 32)	32 (mixed: 64)	64 (mixed: 128)	128 (mixed: 256)
Maximum number of occupied stations	4 stations				

*1: "mixed" means "the same number of RX and RY are used or created".

Special feature of the decentralized I/Os (RX, RY), (BIT)

The I/O bits (RX, RY) of the decentralized device station (EL6711-0010) are divided into a user area and a system area. The size of the user area depends on the number of occupied stations and the extended cyclic setting. The system area always consists of the last 16 bits of the configured I/O bits (RX, RY). This must be taken into account during configuration!

Example 1:

For the extended cyclic setting "single" corresponds to the user area for:

1. 1 occupied station: 16 I/O bits (RX,RY)
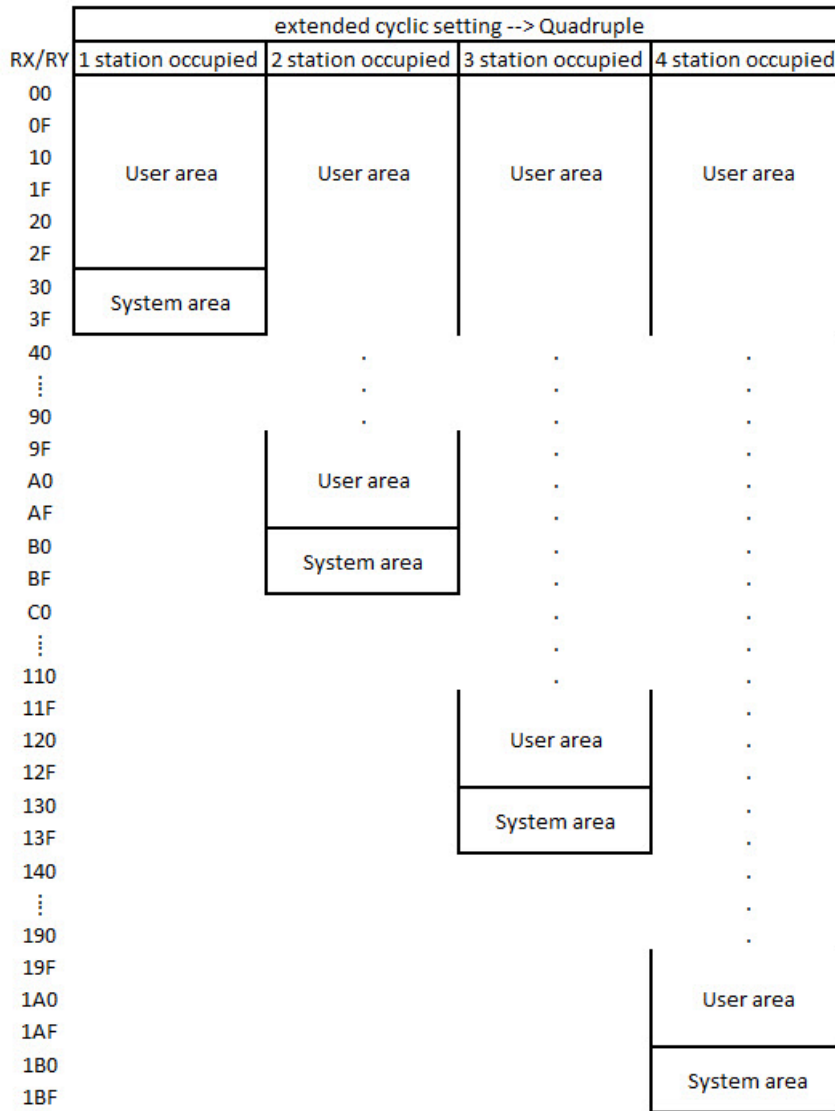2. 2 occupied stations: 48 I/O bits (RX,RY)

3. 3 occupied stations: 80 I/O bits (RX,RY)

4. 4 occupied stations: 112 I/O bits (RX,RY)

	extended cyclic setting --> single			
RX/RY	1 station occupied	2 station occupied	3 station occupied	4 station occupied
00 / 0F	User area	User area	User area	User area
10 / 1F	System area			
20 / 2F				
30 / 3F		System area		
40 / 4F				
50 / 5F			System area	
60 / 6F				
70 / 7F				System area

Example 2:

For the extended cyclic setting "quadruple" corresponds to the user area for:

1. 1 occupied station: 48 I/O bits (RX,RY)

2. 2 occupied stations: 176 I/O bits (RX,RY)

3. 3 occupied stations: 304 I/O bits (RX,RY)

4. 4 occupied stations: 432 I/O bits (RX,RY)

RX/RY	extended cyclic setting --> Quadruple			
	1 station occupied	2 station occupied	3 station occupied	4 station occupied
00 – 2F	User area	User area	User area	User area
30 – 3F	System area			
40 – 90		.	.	.
9F			.	.
A0 – AF		User area	.	.
B0 – BF		System area	.	.
C0 – 110			.	.
11F			.	.
120 – 12F			User area	.
130 – 13F			System area	.
140 – 190				.
19F				.
1A0 – 1AF				User area
1B0 – 1BF				System area

BECKHOFF

3 Basics communication

3.1 EtherCAT basics

Please refer to the EtherCAT System Documentation for the EtherCAT fieldbus basics.

3.2 EtherCAT State Machine

The state of the EtherCAT slave is controlled via the EtherCAT State Machine (ESM). Depending upon the state, different functions are accessible or executable in the EtherCAT slave. Specific commands must be sent by the EtherCAT master to the device in each state, particularly during the bootup of the slave.

A distinction is made between the following states:

- Init
- Pre-Operational
- Safe-Operational and
- Operational
- Boot

The regular state of each EtherCAT slave after bootup is the OP state.
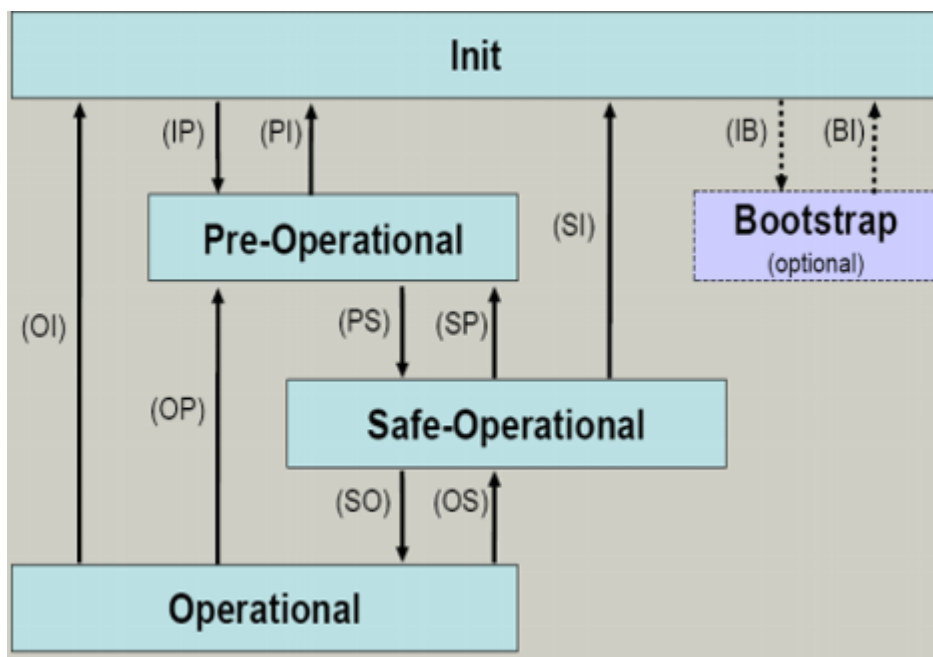
Fig. 5: States of the EtherCAT State Machine

Init

After switch-on the EtherCAT slave in the *Init* state. No mailbox or process data communication is possible. The EtherCAT master initializes sync manager channels 0 and 1 for mailbox communication.

Pre-Operational (Pre-Op)

During the transition between *Init* and *Pre-Op* the EtherCAT slave checks whether the mailbox was initialized correctly.

In *Pre-Op* state mailbox communication is possible, but not process data communication. The EtherCAT master initializes the sync manager channels for process data (from sync manager channel 2), the FMMU channels and, if the slave supports configurable mapping, PDO mapping or the sync manager PDO assignment. In this state the settings for the process data transfer and perhaps terminal-specific parameters that may differ from the default settings are also transferred.

Safe-Operational (Safe-Op)

During transition between *Pre-Op* and *Safe-Op* the EtherCAT slave checks whether the sync manager channels for process data communication and, if required, the distributed clocks settings are correct. Before it acknowledges the change of state, the EtherCAT slave copies current input data into the associated DP-RAM areas of the EtherCAT slave controller (ECSC).

In *Safe-Op* state mailbox and process data communication is possible, although the slave keeps its outputs in a safe state, while the input data are updated cyclically.

> ● **Outputs in SAFEOP state**
> The default set watchdog monitoring sets the outputs of the module in a safe state - depending on the settings in SAFEOP and OP - e.g. in OFF state. If this is prevented by deactivation of the watchdog monitoring in the module, the outputs can be switched or set also in the SAFEOP state.

Operational (Op)

Before the EtherCAT master switches the EtherCAT slave from *Safe-Op* to *Op* it must transfer valid output data.

In the *Op* state the slave copies the output data of the masters to its outputs. Process data and mailbox communication is possible.

Boot

In the *Boot* state the slave firmware can be updated. The *Boot* state can only be reached via the *Init* state.

In the *Boot* state mailbox communication via the *file access over EtherCAT* (FoE) protocol is possible, but no other mailbox communication and no process data communication.

3.3　CoE Interface

General description

The CoE interface (CAN application protocol over EtherCAT)) is used for parameter management of EtherCAT devices. EtherCAT slaves or the EtherCAT master manage fixed (read only) or variable parameters which they require for operation, diagnostics or commissioning.

CoE parameters are arranged in a table hierarchy. In principle, the user has read access via the fieldbus. The EtherCAT master (TwinCAT System Manager) can access the local CoE lists of the slaves via EtherCAT in read or write mode, depending on the attributes.

Different CoE parameter types are possible, including string (text), integer numbers, Boolean values or larger byte fields. They can be used to describe a wide range of features. Examples of such parameters include manufacturer ID, serial number, process data settings, device name, calibration values for analog measurement or passwords.

The order is specified in two levels via hexadecimal numbering: (main)index, followed by subindex. The value ranges are

- Index: 0x0000 …0xFFFF (0...65535$_{dec}$)
- SubIndex: 0x00…0xFF (0...255$_{dec}$)

A parameter localized in this way is normally written as 0x8010:07, with preceding "0x" to identify the hexadecimal numerical range and a colon between index and subindex.

The relevant ranges for EtherCAT fieldbus users are:

BECKHOFF

- 0x1000: This is where fixed identity information for the device is stored, including name, manufacturer, serial number etc., plus information about the current and available process data configurations.
- 0x8000: This is where the operational and functional parameters for all channels are stored, such as filter settings or output frequency.

Other important ranges are:

- 0x4000: here are the channel parameters for some EtherCAT devices. Historically, this was the first parameter area before the 0x8000 area was introduced. EtherCAT devices that were previously equipped with parameters in 0x4000 and changed to 0x8000 support both ranges for compatibility reasons and mirror internally.
- 0x6000: Input PDOs ("input" from the perspective of the EtherCAT master)
- 0x7000: Output PDOs ("output" from the perspective of the EtherCAT master)

ⓘ Availability

Not every EtherCAT device must have a CoE list. Simple I/O modules without dedicated processor usually have no variable parameters and therefore no CoE list.

If a device has a CoE list, it is shown in the TwinCAT System Manager as a separate tab with a listing of the elements:
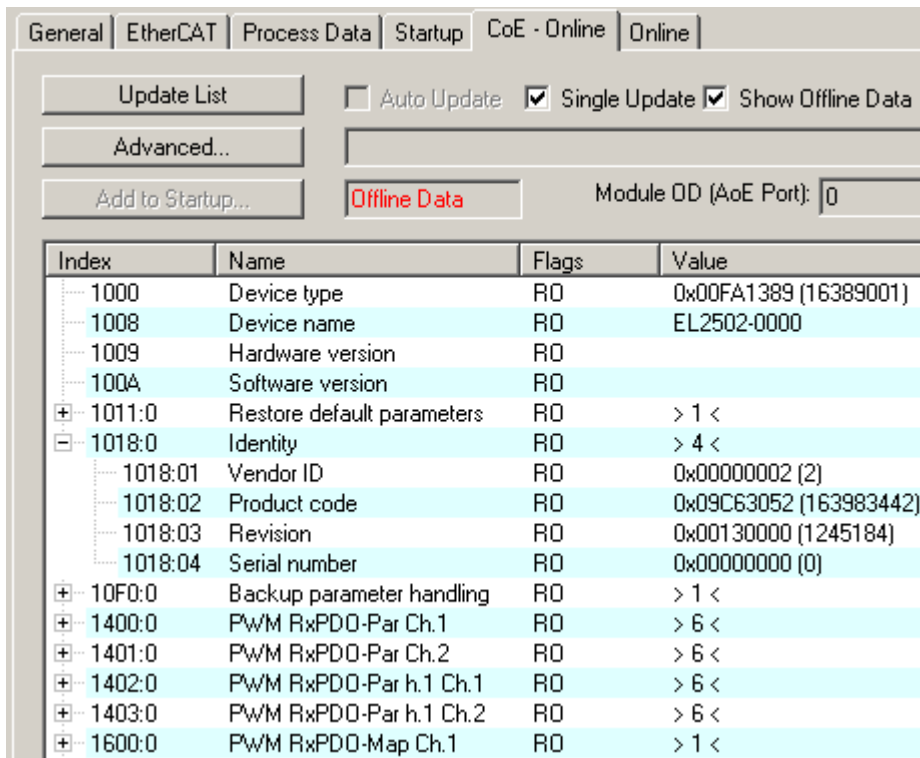
Fig. 6: "CoE Online" tab

The figure above shows the CoE objects available in device "EL2502", ranging from 0x1000 to 0x1600. The subindices for 0x1018 are expanded.

Version: 1.1 EL6711-0010

NOTICE

Changes in the CoE directory (CAN over EtherCAT), program access

When using/manipulating the CoE parameters observe the general CoE notes in chapter "CoE interface" of the EtherCAT system documentation:

- Keep a startup list if components have to be replaced,
- Distinction between online/offline dictionary,
- Existence of current XML description (download from the Beckhoff website),
- "CoE-Reload" for resetting the changes
- Program access during operation via PLC (see TwinCAT3 | PLC Library: Tc2_EtherCAT and Example program R/W CoE)

Data management and function "NoCoeStorage"

Some parameters, particularly the setting parameters of the slave, are configurable and writeable. This can be done in write or read mode

- via the System Manager (Fig. "CoE Online" tab) by clicking
  This is useful for commissioning of the system/slaves. Click on the row of the index to be parameterized and enter a value in the "SetValue" dialog.
- from the control system/PLC via ADS, e.g. through blocks from the TcEtherCAT.lib library
  This is recommended for modifications while the system is running or if no System Manager or operating staff are available.

Data management

If slave CoE parameters are modified online, Beckhoff devices store any changes in a fail-safe manner in the EEPROM, i.e. the modified CoE parameters are still available after a restart. The situation may be different with other manufacturers.

An EEPROM is subject to a limited lifetime with respect to write operations. From typically 100,000 write operations onwards it can no longer be guaranteed that new (changed) data are reliably saved or are still readable. This is irrelevant for normal commissioning. However, if CoE parameters are continuously changed via ADS at machine runtime, it is quite possible for the lifetime limit to be reached. Support for the NoCoeStorage function, which suppresses the saving of changed CoE values, depends on the firmware version.
Please refer to the technical data in this documentation as to whether this applies to the respective device.

- If the function is supported: the function is activated by entering the code word 0x12345678 once in CoE 0xF008 and remains active as long as the code word is not changed. After switching the device on it is then inactive. Changed CoE values are not saved in the EEPROM and can thus be changed any number of times.
- Function is not supported: continuous changing of CoE values is not permissible in view of the lifetime limit.

Startup list

Changes in the local CoE list of the terminal are lost if the terminal is replaced. If a terminal is replaced with a new Beckhoff terminal, it will have the default settings. It is therefore advisable to link all changes in the CoE list of an EtherCAT slave with the Startup list of the slave, which is processed whenever the EtherCAT fieldbus is started. In this way a replacement EtherCAT slave can automatically be parameterized with the specifications of the user.

If EtherCAT slaves are used which are unable to store local CoE values permanently, the Startup list must be used.

Recommended approach for manual modification of CoE parameters

- Make the required change in the System Manager
  The values are stored locally in the EtherCAT slave

BECKHOFF

- If the value is to be stored permanently, enter it in the Startup list.
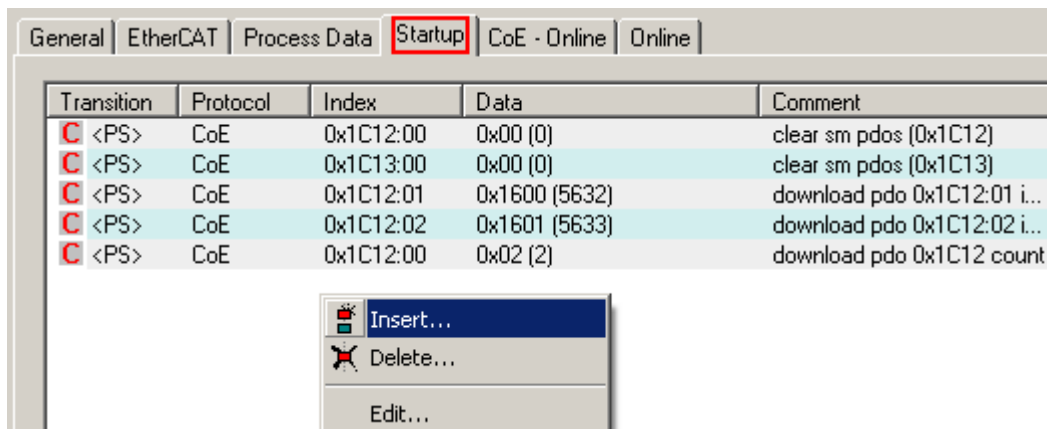 The order of the Startup entries is usually irrelevant.

Transition	Protocol	Index	Data	Comment
C <PS>	CoE	0x1C12:00	0x00 (0)	clear sm pdos (0x1C12)
C <PS>	CoE	0x1C13:00	0x00 (0)	clear sm pdos (0x1C13)
C <PS>	CoE	0x1C12:01	0x1600 (5632)	download pdo 0x1C12:01 i...
C <PS>	CoE	0x1C12:02	0x1601 (5633)	download pdo 0x1C12:02 i...
C <PS>	CoE	0x1C12:00	0x02 (2)	download pdo 0x1C12 count

Insert...
Delete...
Edit...

Fig. 7: Startup list in the TwinCAT System Manager

The Startup list may already contain values that were configured by the System Manager based on the ESI specifications. Additional application-specific entries can be created.

Online/offline list

While working with the TwinCAT System Manager, a distinction has to be made whether the EtherCAT device is "available", i.e. switched on and linked via EtherCAT and therefore **online**, or whether a configuration is created **offline** without connected slaves.

In both cases a CoE list as shown in Fig. "CoE online tab" is displayed. The connectivity is shown as offline/ online.

- If the slave is offline
  - The offline list from the ESI file is displayed. In this case modifications are not meaningful or possible.
  - The configured status is shown under Identity.
  - No firmware or hardware version is displayed, since these are features of the physical device.
  - **Offline** is shown in red.
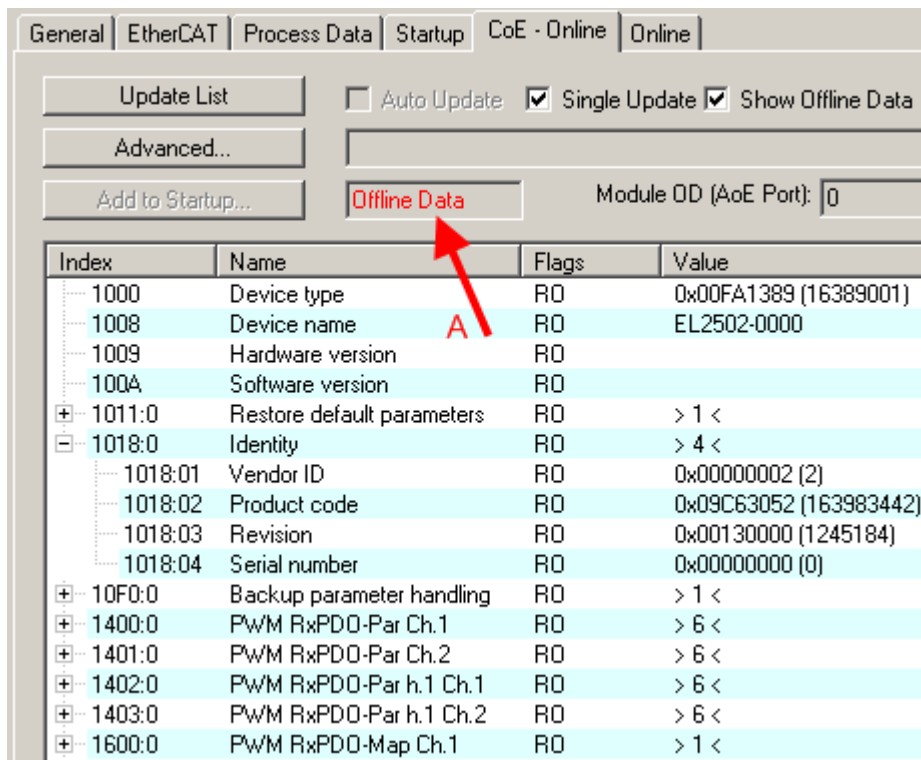
Fig. 8: Offline list

- If the slave is online
  - The actual current slave list is read. This may take several seconds, depending on the size and cycle time.
  - The actual identity is displayed
  - The firmware and hardware version of the equipment according to the electronic information is displayed
  - **Online** is shown in green.
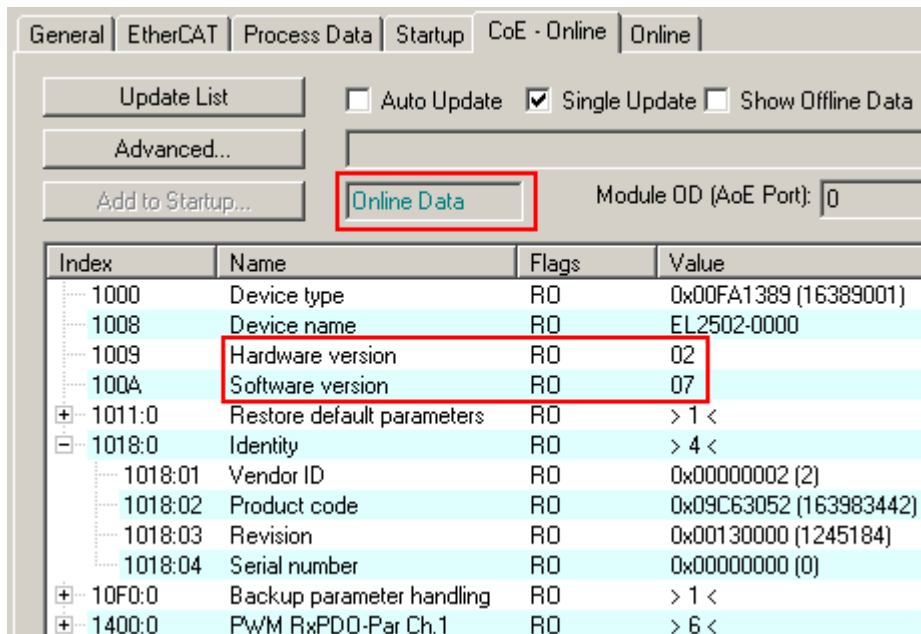
Fig. 9: Online list

Channel-based order

The CoE list is available in EtherCAT devices that usually feature several functionally equivalent channels. For example, a 4-channel analog 0...10 V input terminal also has four logical channels and therefore four identical sets of parameter data for the channels. In order to avoid having to list each channel in the documentation, the placeholder "n" tends to be used for the individual channel numbers.

In the CoE system 16 indices, each with 255 subindices, are generally sufficient for representing all channel parameters. The channel-based order is therefore arranged in $16_{dec}/10_{hex}$ steps. The parameter range 0x8000 exemplifies this:

- Channel 0: parameter range 0x8000:00 ... 0x800F:255
- Channel 1: parameter range 0x8010:00 ... 0x801F:255
- Channel 2: parameter range 0x8020:00 ... 0x802F:255
- ...

This is generally written as 0x80n0.

Detailed information on the CoE interface can be found in the EtherCAT system documentation on the Beckhoff website.

4   Mounting and wiring

4.1   Instructions for ESD protection

NOTICE
Destruction of the devices by electrostatic discharge possible!
The devices contain components at risk from electrostatic discharge caused by improper handling.
• Please ensure you are electrostatically discharged and avoid touching the contacts of the device directly.
• Avoid contact with highly insulating materials (synthetic fibers, plastic film etc.).
• Surroundings (working place, packaging and personnel) should by grounded probably, when handling with the devices.
• Each assembly must be terminated at the right hand end with an EL9011 or EL9012 bus end cap, to ensure the protection class and ESD protection.

Fig. 10: Spring contacts of the Beckhoff I/O components

4.2   Mounting and demounting - top front unlocking

The terminal modules are fastened to the assembly surface with the aid of a 35 mm mounting rail (e. g. mounting rail TH 35-15).

> **ⓘ** **Fixing of mounting rails**
>
> The locking mechanism of the terminals and couplers extends to the profile of the mounting rail. At the installation, the locking mechanism of the components must not come into conflict with the fixing bolts of the mounting rail. To mount the recommended mounting rails under the terminals and couplers, you should use flat mounting connections (e.g. countersunk screws or blind rivets).

⚠ WARNING
Risk of electric shock and damage of device!
Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the Bus Terminals!

Mounting

- Fit the mounting rail to the planned assembly location.
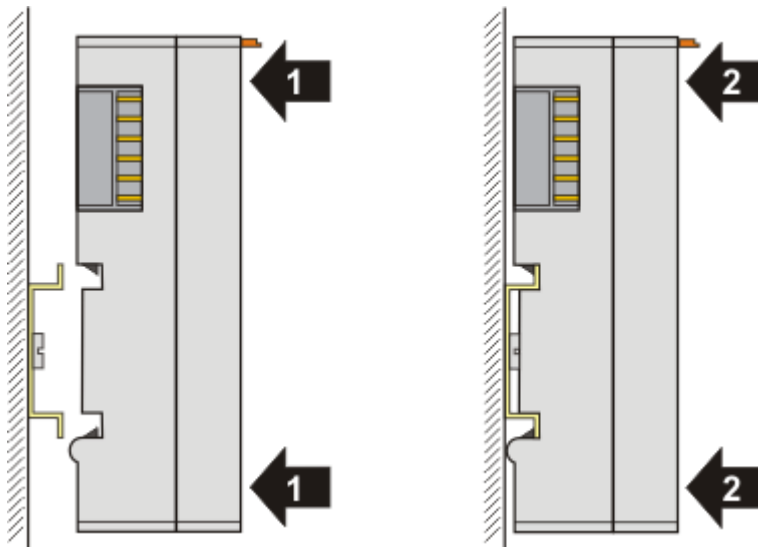
and press (1) the terminal module against the mounting rail until it latches in place on the mounting rail (2).

- Attach the cables.

Demounting

- Remove all the cables.
- Lever the unlatching hook back with thumb and forefinger (3). An internal mechanism pulls the two latching lugs (3a) from the top hat rail back into the terminal module.
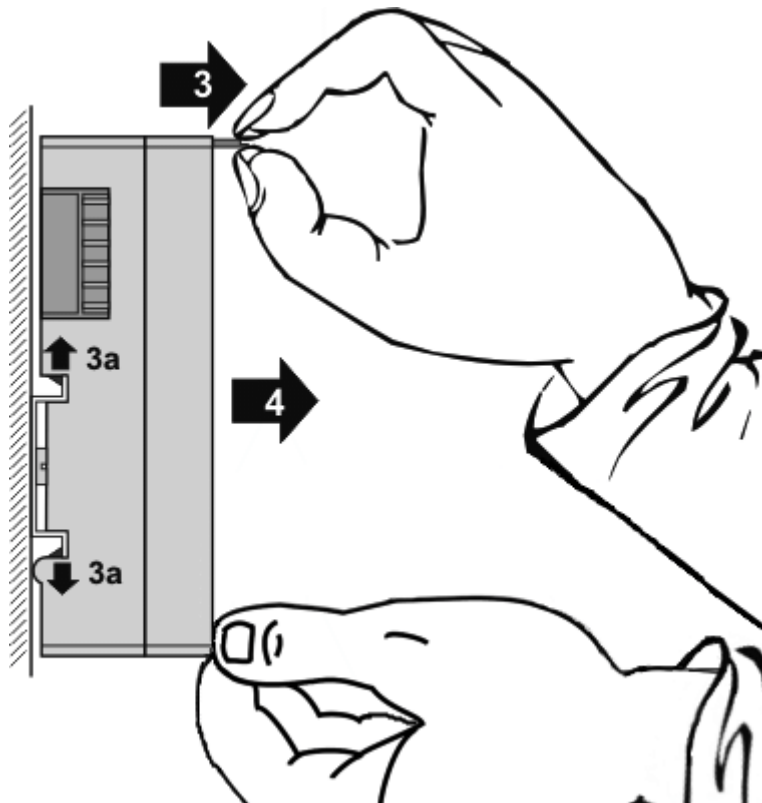
- Pull (4) the terminal module away from the mounting surface.
  Avoid canting of the module; you should stabilize the module with the other hand, if required.

4.3 Disposal

Products marked with a crossed-out wheeled bin shall not be discarded
with the normal waste stream. The device is considered as waste
electrical and electronic equipment. The national regulations for the
disposal of waste electrical and electronic equipment must be observed.

4.4 Positioning of passive Terminals

ℹ **Hint for positioning of passive terminals in the bus terminal block**

EtherCAT Terminals (ELxxxx / ESxxxx), which do not take an active part in data transfer within the bus terminal block are so called passive terminals. The passive terminals have no current consumption out of the E-Bus.
To ensure an optimal data transfer, you must not directly string together more than two passive terminals!

Examples for positioning of passive terminals (highlighted)
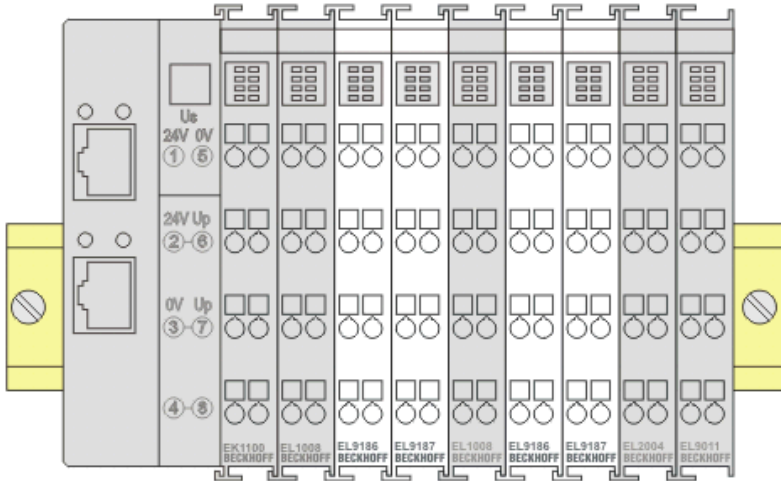
Fig. 11: Correct positioning
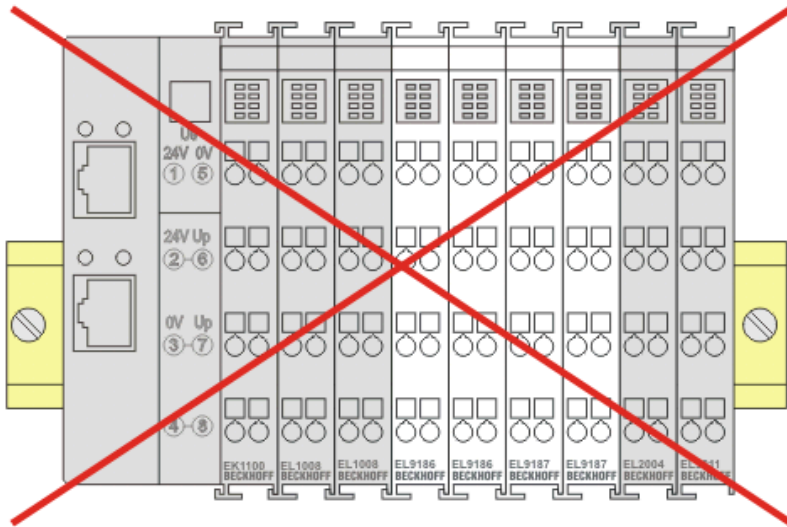
Fig. 12: Incorrect positioning

4.5   Installation positions

NOTICE
Constraints regarding installation position and operating temperature range
Please refer to the technical data for a terminal to ascertain whether any restrictions regarding the installation position and/or the operating temperature range have been specified. When installing high power dissipation terminals ensure that an adequate spacing is maintained between other components above and below the terminal in order to guarantee adequate ventilation!

Optimum installation position (standard)

The optimum installation position requires the mounting rail to be installed horizontally and the connection surfaces of the EL/KL terminals to face forward (see Fig. *Recommended distances for standard installation position*). The terminals are ventilated from below, which enables optimum cooling of the electronics through convection. "From below" is relative to the acceleration of gravity.
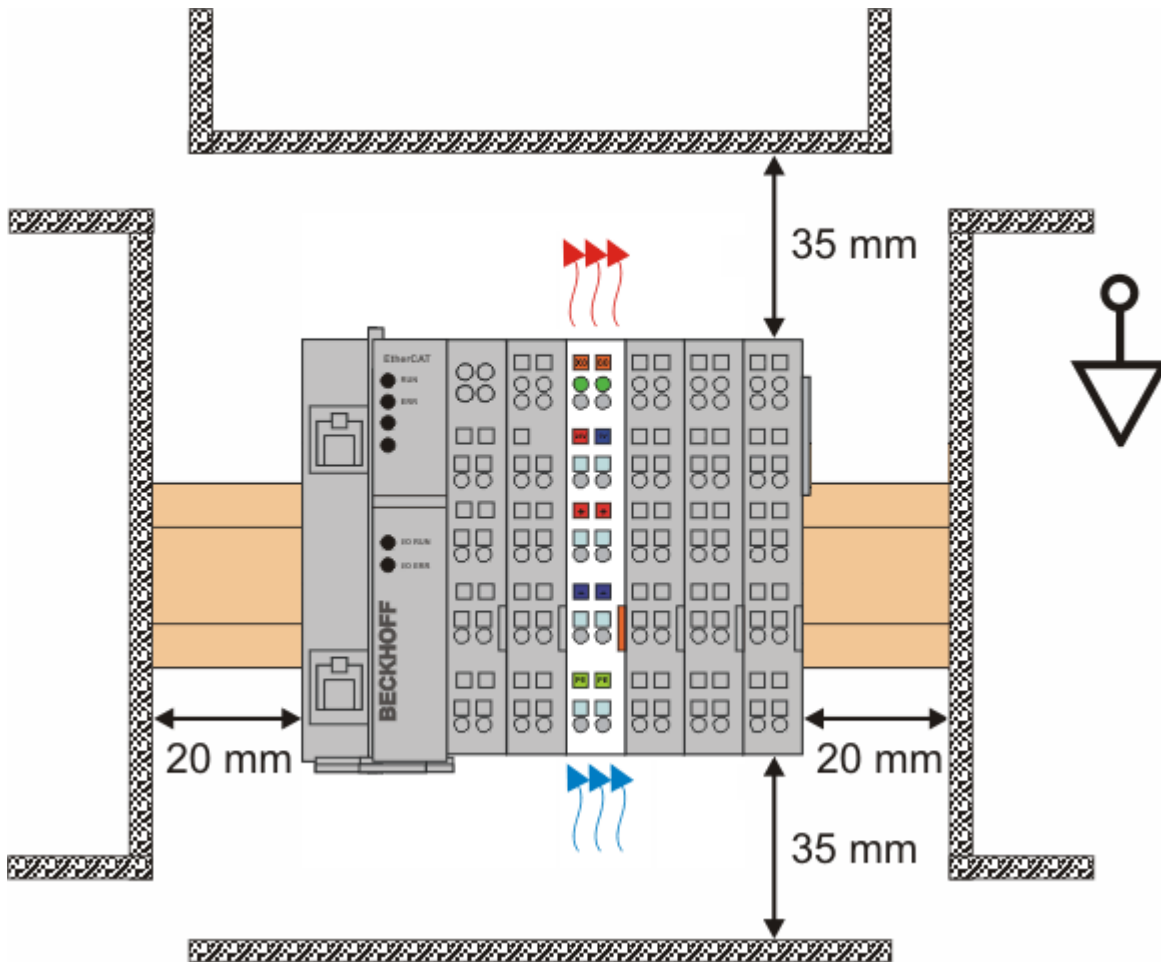
Fig. 13: Recommended distances for standard installation position

Compliance with the distances shown in Fig. *Recommended distances for standard installation position* is recommended.

Other installation positions

All other installation positions are characterized by different spatial arrangement of the mounting rail - see Fig *Other installation positions.*

The minimum distances to ambient specified above also apply to these installation positions.

BECKHOFF
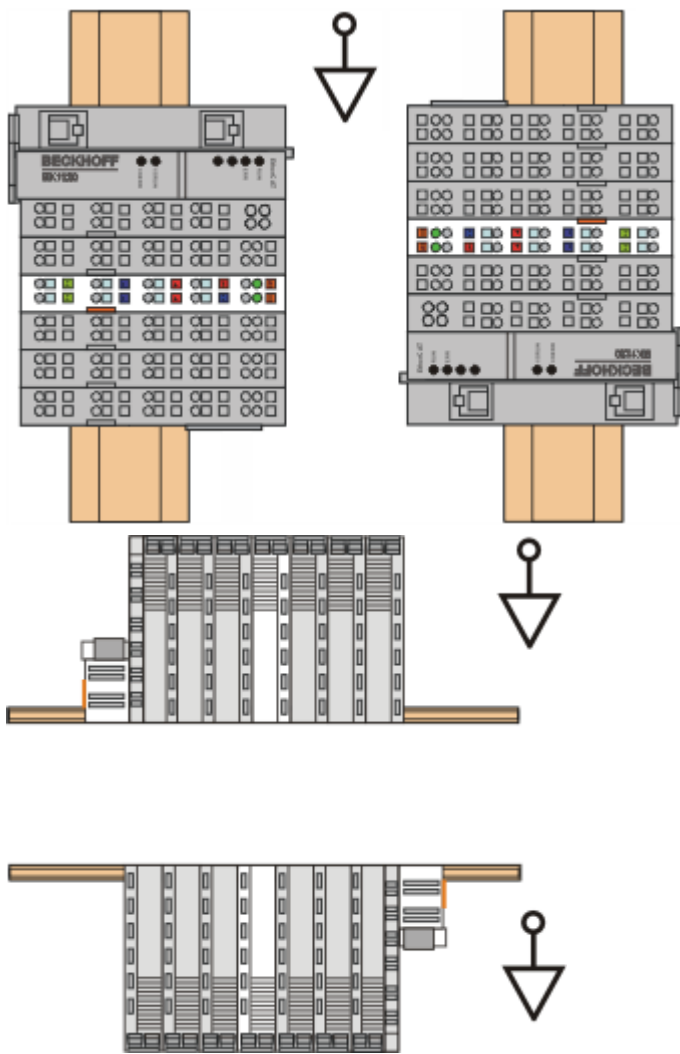
Fig. 14: Other installation positions

4.6   LEDs
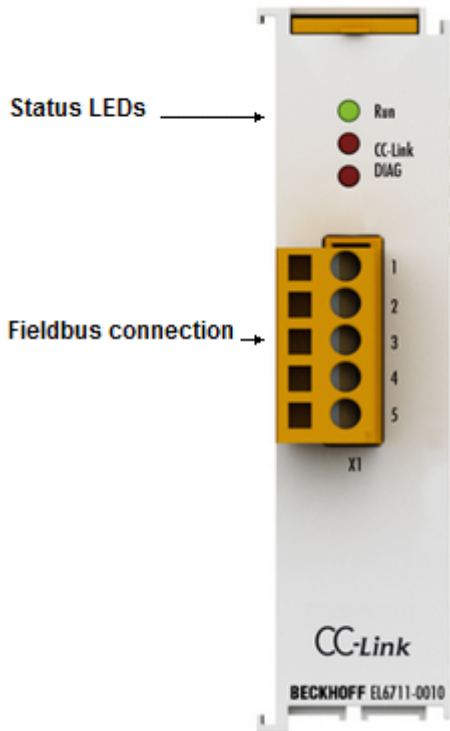
Fig. 15: LEDs

LED	Color	Meaning	
RUN	green	These LEDs indicate the terminal's operating state:	
		off	State of the EtherCAT State Machine [▶ 20]: **INIT** = initialization of the terminal
		flashing uniformly	State of the EtherCAT State Machine: **PREOP =** function for mailbox communication and different default settings set
		flashing slowly	State of the EtherCAT State Machine: **SAFEOP** = verification of the Sync Manager [▶ 88] channels
		on	State of the EtherCAT State Machine: **OP =** normal operating state; mailbox and process data communication is possible
		flashing rapidly	State of the EtherCAT State Machine: **BOOTSTRAP** = function for terminal firmware updates
CC-Link (RUN)	green	off	• before participating in the network • carriers cannot be detected • timeout • resetting the hardware
		on	normal reception of refresh and polling signal or only the refresh signal, after participation in the network
CC-Link (ERR)	red	off	• normal communication • resetting the hardware
		on	• CRC error • error in setting the station number (0 or >65) since the last restart / reset • error in setting the baud rate (>4) since last restart / reset
		flashing	The settings were changed after the last restart / reset
DIAG (CC-Link SD)	green	off	• no transmission • resetting the hardware
		on	during transmission
DIAG (CC-Link RD)	blue	off	• carriers cannot be detected • resetting the hardware
		on	Recognition of the carrier

4.7   Note - Power supply

⚠ WARNING
Power supply from SELV/PELV power supply unit!
SELV/PELV circuits (Safety Extra Low Voltage, Protective Extra Low Voltage) according to IEC 61010-2-201 must be used to supply this device.
Notes:
• SELV/PELV circuits may give rise to further requirements from standards such as IEC 60204-1 et al, for example with regard to cable spacing and insulation.
• A SELV (Safety Extra Low Voltage) supply provides safe electrical isolation and limitation of the voltage without a connection to the protective conductor, a PELV (Protective Extra Low Voltage) supply also requires a safe connection to the protective conductor.

4.8   Cabling

Specification					
Communication speed	156 kbps	625 kbps	2.5 Mbps	5 Mbps	10 Mbps
Cable length between stations	at least 20 cm				
Max. total extension cable length	1200 m	900 m	400 m	160 m	100 m
Connection cable (recommendation)	CC-Link cable version 1.10, (shielded, 3-wire twisted pair cable); mixing cables of different brands is only possible if all cables are CC-Link cable version 1.10-compatible. (110 Ω termination resistors are used)				
Bus interface	1 x open pluggable connector, 5-pin, included				
Termination resistor	110Ω,1/2W (connection between DA and DB at both line ends) 				
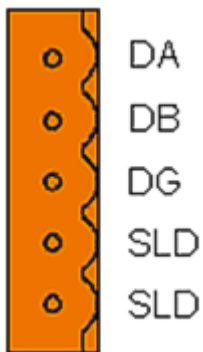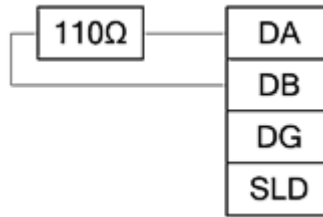
Fig. 16: CC-Link connector
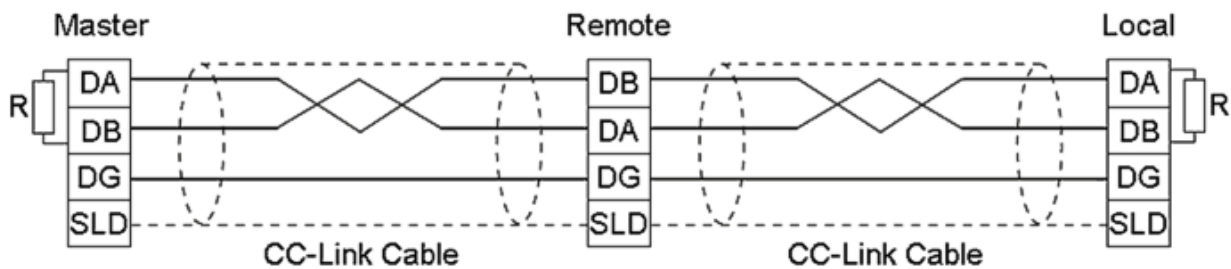
Fig. 17: CC-Link cabling

4.9   Shield connection in the terminal block
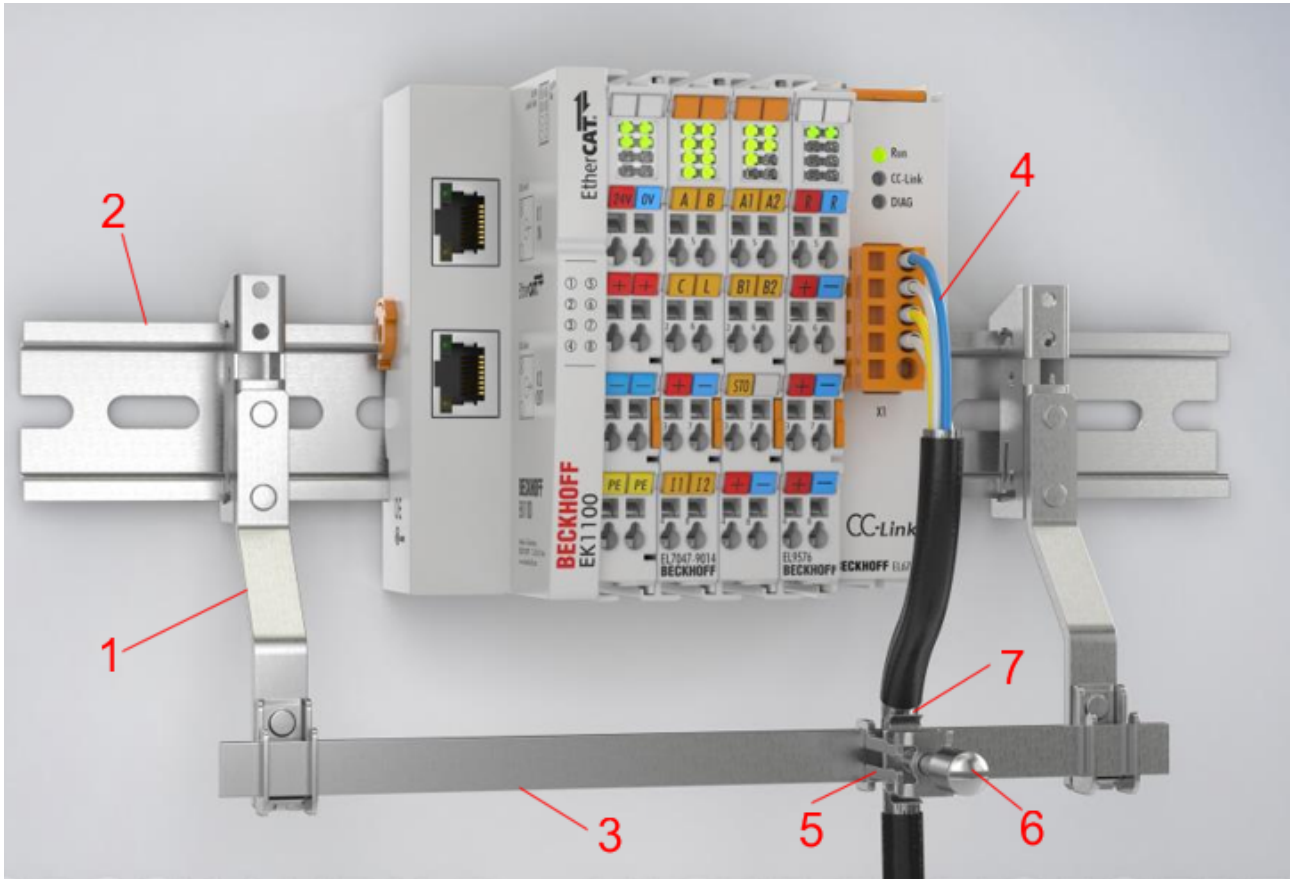
Fig. 18: Shield connection in the terminal block

Attach the shield rail carrier 1 to the mounting rail 2 as shown in the figure. The mounting rail 2 must be connected to the metallic rear panel of the control cabinet over a large area. Mount the shield busbar 3 also as shown.

Connect the plug of the connecting cable 4 to the terminal and mark the point on the sheathing of the connecting cable where the cable crosses the shield busbar 3.

Now disconnect the plug again and carefully remove the sheathing of the connecting cable at the marking with an appropriate stripping tool so that the shielding braid 7 of the cable is visible.

Remove enough of the sheathing so that the clamper of the shield clamp 5 has secure contact over its entire width on the shield braid. Be careful not to damage the shielding braid and the strands of the connection cable! Now tighten the screw 6 and thus fix the clamper of the shield clamp 5 to the shielding braid 7.

Suitable shield rails with shield bus carriers are available from Beckhoff as accessories (ZB8xxx).

5   Commissioning

5.1   TwinCAT Quick Start

TwinCAT is a development environment for real-time control including a multi PLC system, NC axis control, programming and operation. The whole system is mapped through this environment and enables access to a programming environment (including compilation) for the controller. Individual digital or analog inputs or outputs can also be read or written directly, in order to verify their functionality, for example.

For further information, please refer to http://infosys.beckhoff.com:

- **EtherCAT System Manual:**
  Fieldbus Components → EtherCAT Terminals → EtherCAT System Documentation → Setup in the TwinCAT System Manager
- **TwinCAT 2** → TwinCAT System Manager → I/O Configuration
- In particular, for TwinCAT – driver installation:
  Fieldbus components → Fieldbus Cards and Switches → FC900x – PCI Cards for Ethernet → Installation

Devices contain the relevant terminals for the actual configuration. All configuration data can be entered directly via editor functions (offline) or via the `scan function (online):

- **"offline"**: The configuration can be customized by adding and positioning individual components. These can be selected from a directory and configured.

  ◦ The procedure for the offline mode can be found under http://infosys.beckhoff.com:
    TwinCAT 2 → TwinCAT System Manager → IO Configuration → Add an I/O device

- **"online"**: The existing hardware configuration is read

  ◦ See also http://infosys.beckhoff.com:
    Fieldbus components → Fieldbus Cards and Switches → FC900x – PCI Cards for Ethernet → Installation → Searching for devices

The following relationship is envisaged between the user PC and individual control elements:
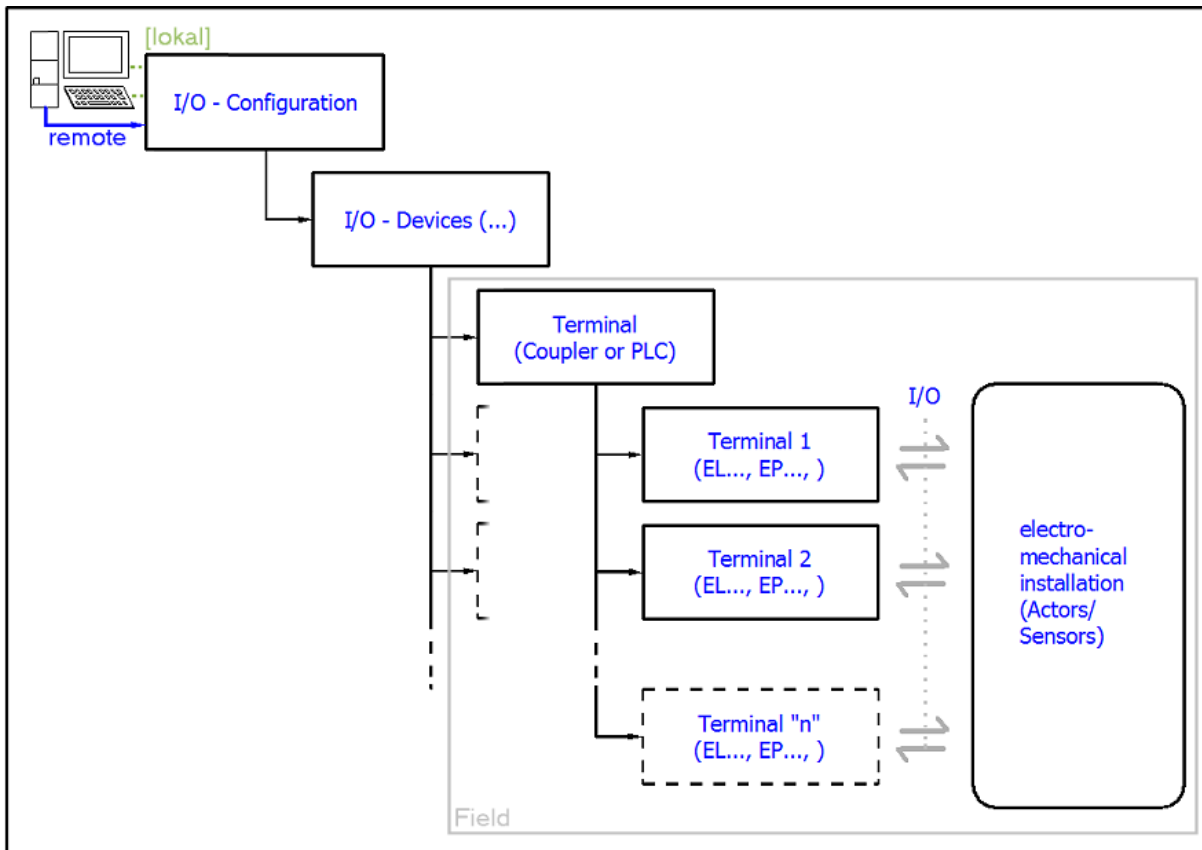
Fig. 19: Relationship between user side (commissioning) and installation

Insertion of certain components (I/O device, terminal, box...) by users functions the same way as in TwinCAT 2 and TwinCAT 3. The descriptions below relate solely to the online procedure.

Example configuration (actual configuration)

Based on the following example configuration, the subsequent subsections describe the procedure for TwinCAT 2 and TwinCAT 3:

- **CX2040** control system (PLC) including **CX2100-0004** power supply unit
- Connected to CX2040 on the right (E-bus):
  EL1004 (4-channel digital input terminal 24 V_{DC})
- Linked via the X001 port (RJ-45): **EK1100** EtherCAT Coupler
- Connected to the EK1100 EtherCAT Coupler on the right (E-bus):
  EL2008 (8-channel digital output terminal 24 V_{DC}; 0.5 A)
- (Optional via X000: a link to an external PC for the user interface)

Fig. 20: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

Note that all combinations of a configuration are possible; for example, the EL1004 terminal could also be connected after the coupler, or the EL2008 terminal could additionally be connected to the CX2040 on the right, in which case the EK1100 coupler wouldn't be necessary.

5.1.1   TwinCAT 2

Startup

TwinCAT 2 basically uses two user interfaces: the TwinCAT System Manager for communication with the electromechanical components and TwinCAT PLC Control for the development and compilation of a controller. The starting point is the TwinCAT System Manager.

After successful installation of the TwinCAT system on the PC to be used for development, the TwinCAT 2 System Manager displays the following user interface after startup:
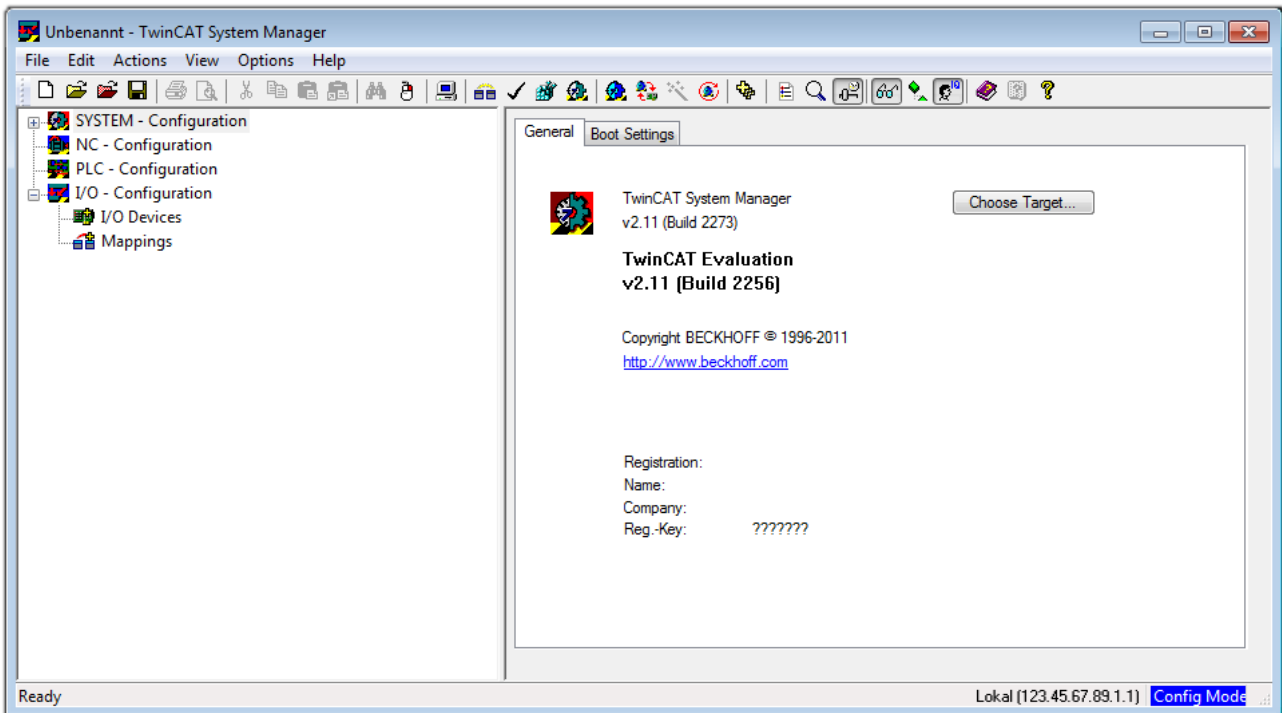
Fig. 21: Initial TwinCAT 2 user interface

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system, including the user interface (standard) is installed on the respective PLC, TwinCAT can be used in local mode and thus the next step is "Insert Device [▶ 42]".

If the intention is to address the TwinCAT runtime environment installed on a PLC remotely from another system used as a development environment, the target system must be made known first. In the menu under

"Actions" → "Choose Target System...", the following window is opened for this via the symbol " " or the "F8" key:
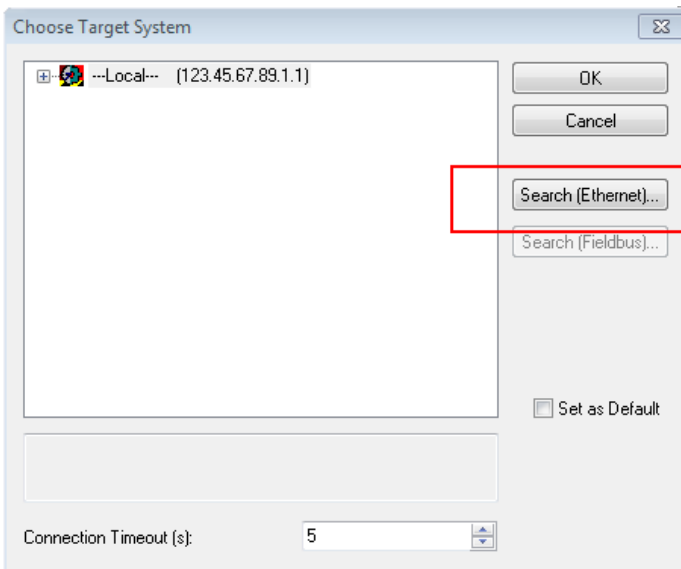
Fig. 22: Selection of the target system

Use "Search (Ethernet)..." to enter the target system. Thus another dialog opens to either:

- enter the known computer name after "Enter Host Name / IP:" (as shown in red)
- perform a "Broadcast Search" (if the exact computer name is not known)
- enter the known computer – IP or AmsNetID
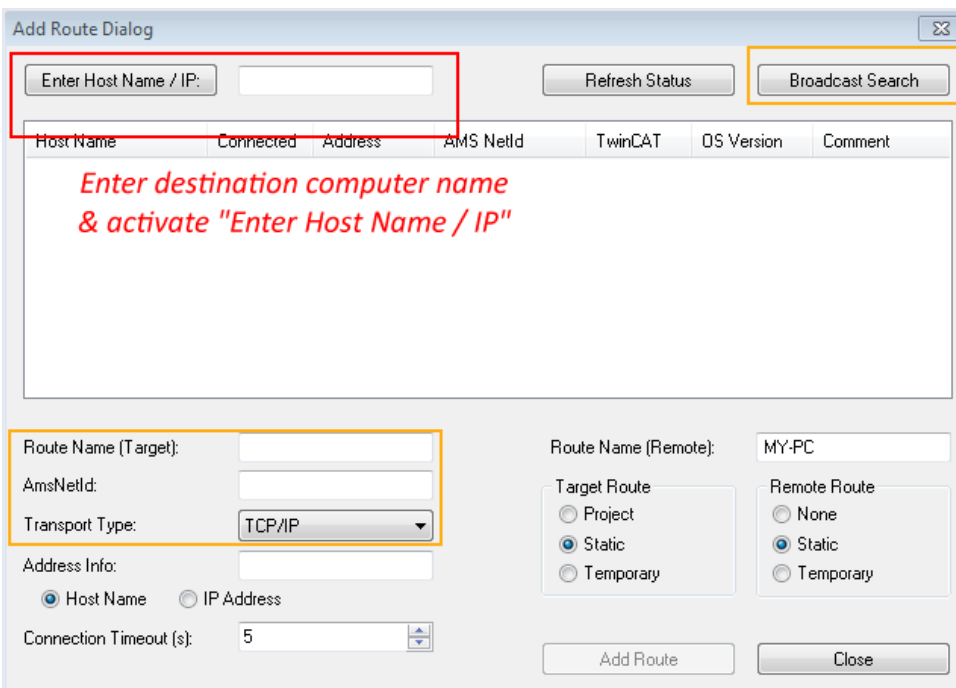
Fig. 23: specify the PLC for access by the TwinCAT System Manager: selection of the target system

Once the target system has been entered, it is available for selection as follows (a correct password may have to be entered before this):
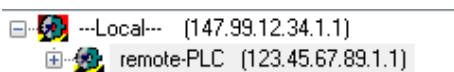
After confirmation with "OK", the target system can be accessed via the System Manager.

BECKHOFF

Adding devices

In the configuration tree of the TwinCAT 2 System Manager user interface on the left, select "I/O Devices" and then right-click to open a context menu and select "Scan Devices…", or start the action in the menu bar

via [icon] . The TwinCAT System Manager may first have to be set to "Config Mode" via [icon] or via the menu

"Actions" → "Set/Reset TwinCAT to Config Mode…" (Shift + F4).
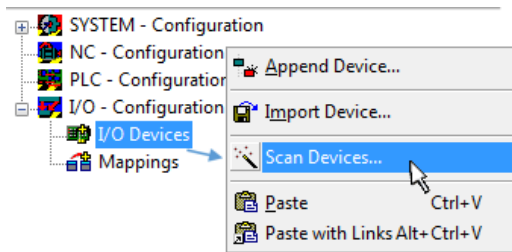
Fig. 24: Select "Scan Devices..."

Confirm the warning message, which follows, and select the "EtherCAT" devices in the dialog:
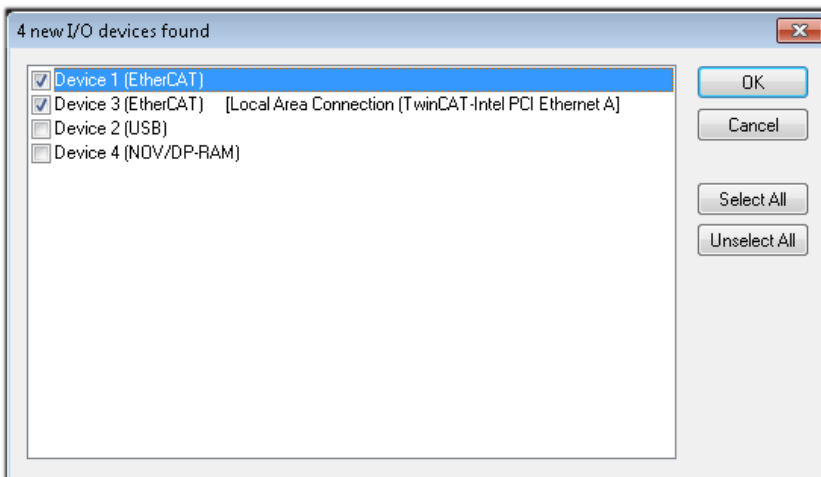
Fig. 25: Automatic detection of I/O devices: selection of the devices to be integrated

Confirm the message "Find new boxes", in order to determine the terminals connected to the devices. "Free Run" enables manipulation of input and output values in "Config Mode" and should also be acknowledged.

Based on the example configuration [▶ 38] described at the beginning of this section, the result is as follows:
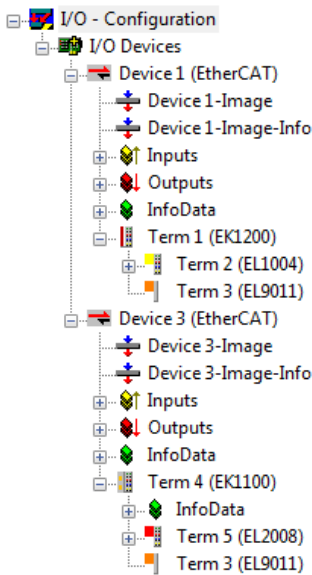
Fig. 26: Mapping of the configuration in the TwinCAT 2 System Manager

The whole process consists of two stages, which can also be performed separately (first determine the devices, then determine the connected elements such as boxes, terminals, etc.). A scan (search function) can also be initiated by selecting "Device ..." from the context menu, which then only reads the elements below which are present in the configuration:
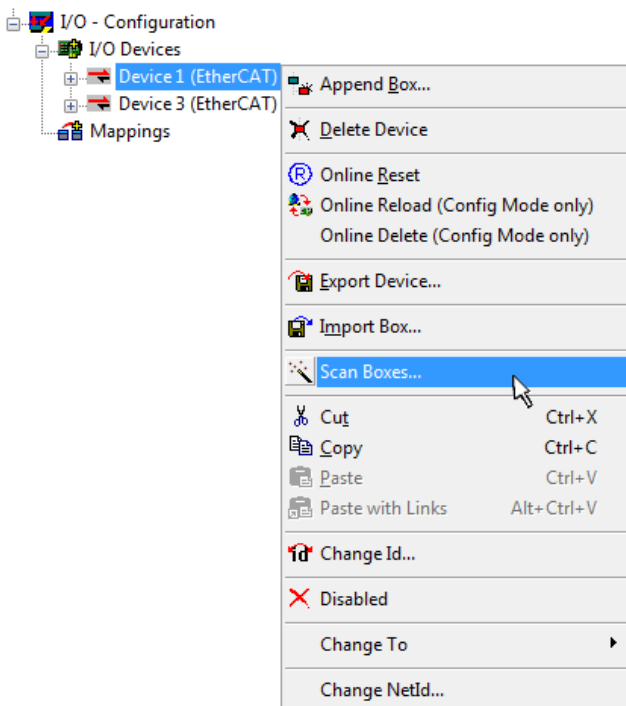
Fig. 27: Reading of individual terminals connected to a device

This functionality is useful if the actual configuration is modified at short notice.

Programming and integrating the PLC

TwinCAT PLC Control is the development environment for generating the controller in different program environments: TwinCAT PLC Control supports all languages described in IEC 61131-3. There are two text-based languages and three graphical languages.

- **Text-based languages**
  - Instruction List (IL)
  - Structured Text (ST)

BECKHOFF

- **Graphical languages**
  - ◦ Function Block Diagram (FBD)
  - ◦ Ladder Diagram (LD)
  - ◦ The Continuous Function Chart Editor (CFC)
  - ◦ Sequential Function Chart (SFC)

The following section refers solely to Structured Text (ST).

After starting TwinCAT PLC Control, the following user interface is shown for an initial project:
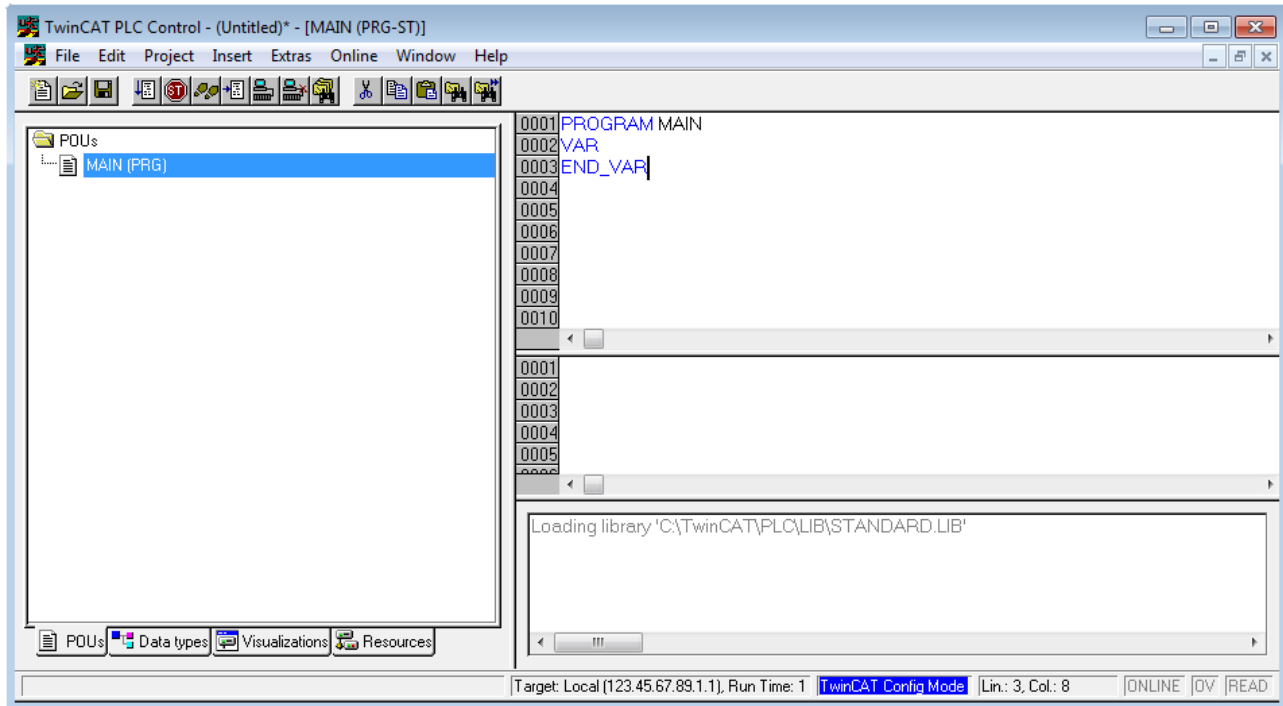
Fig. 28: TwinCAT PLC Control after startup

Example variables and an example program have been created and stored under the name "PLC_example.pro":
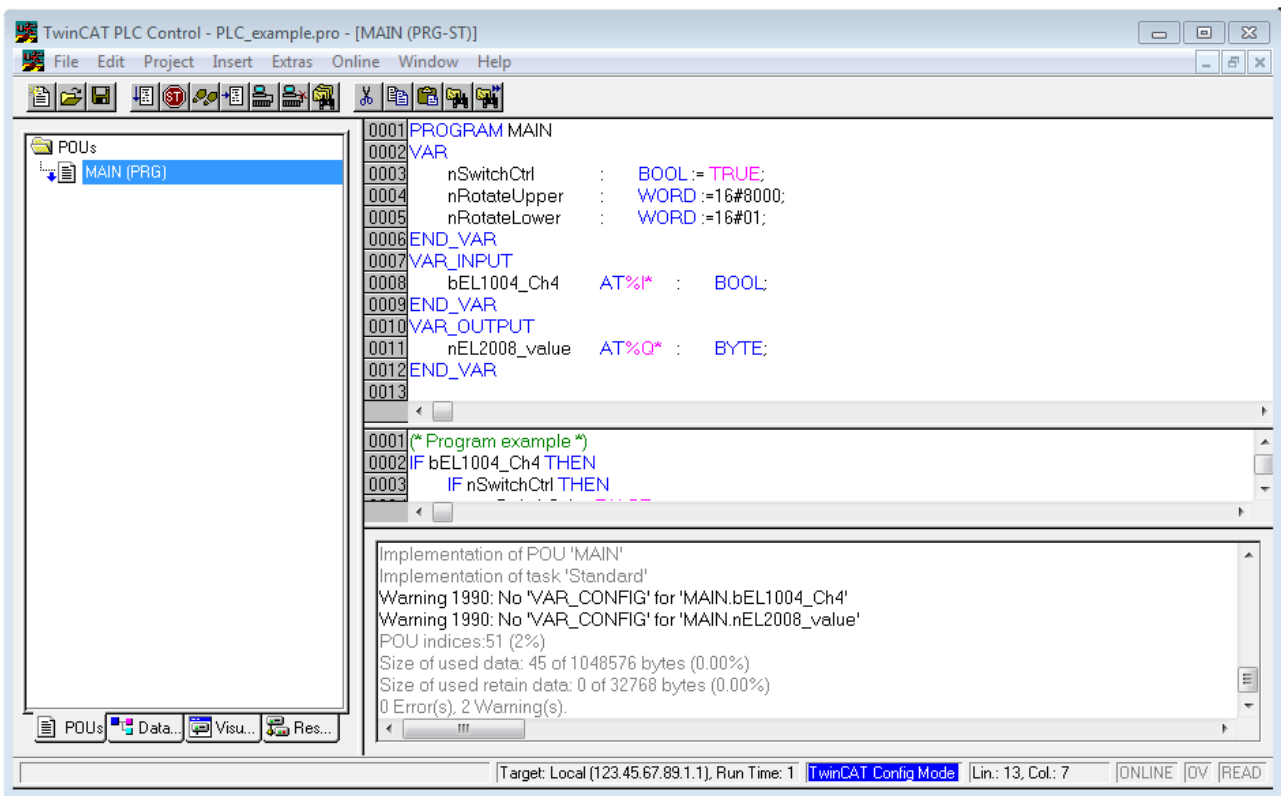
Fig. 29: Example program with variables after a compile process (without variable integration)

Warning 1990 (missing "VAR_CONFIG") after a compile process indicates that the variables defined as external (with the ID "AT%I*" or "AT%Q*") have not been assigned. After successful compilation, TwinCAT PLC Control creates a "*.tpy" file in the directory in which the project was stored. This file ("*.tpy") contains variable assignments and is not known to the System Manager, hence the warning. Once the System Manager has been notified, the warning no longer appears.

First, integrate the TwinCAT PLC Control project in the **System Manager**. This is performed via the context menu of the PLC configuration (right-click) and selecting "Append PLC Project…":
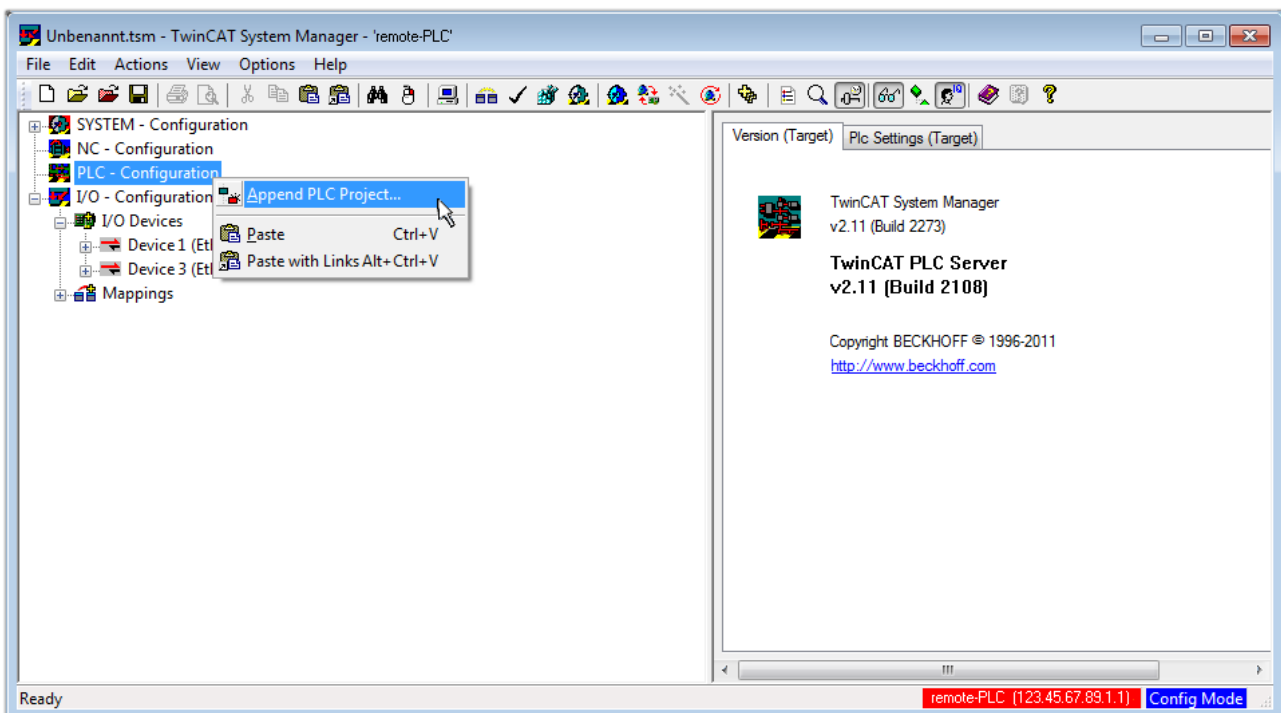
Fig. 30: Appending the TwinCAT PLC Control project

Select the PLC configuration "PLC_example.tpy" in the browser window that opens. The project including the two variables identified with "AT" are then integrated in the configuration tree of the System Manager:
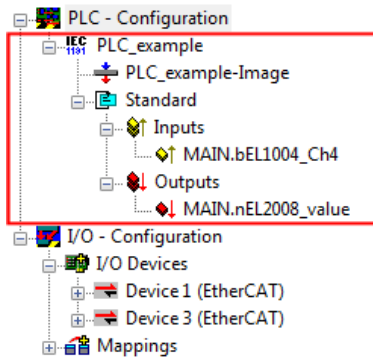
Fig. 31: PLC project integrated in the PLC configuration of the System Manager

The two variables "bEL1004_Ch4" and "nEL2008_value" can now be assigned to certain process objects of the I/O configuration.

Assigning variables

Open a window for selecting a suitable process object (PDO) via the context menu of a variable of the integrated project "PLC_example" and via "Modify Link..." "Standard":
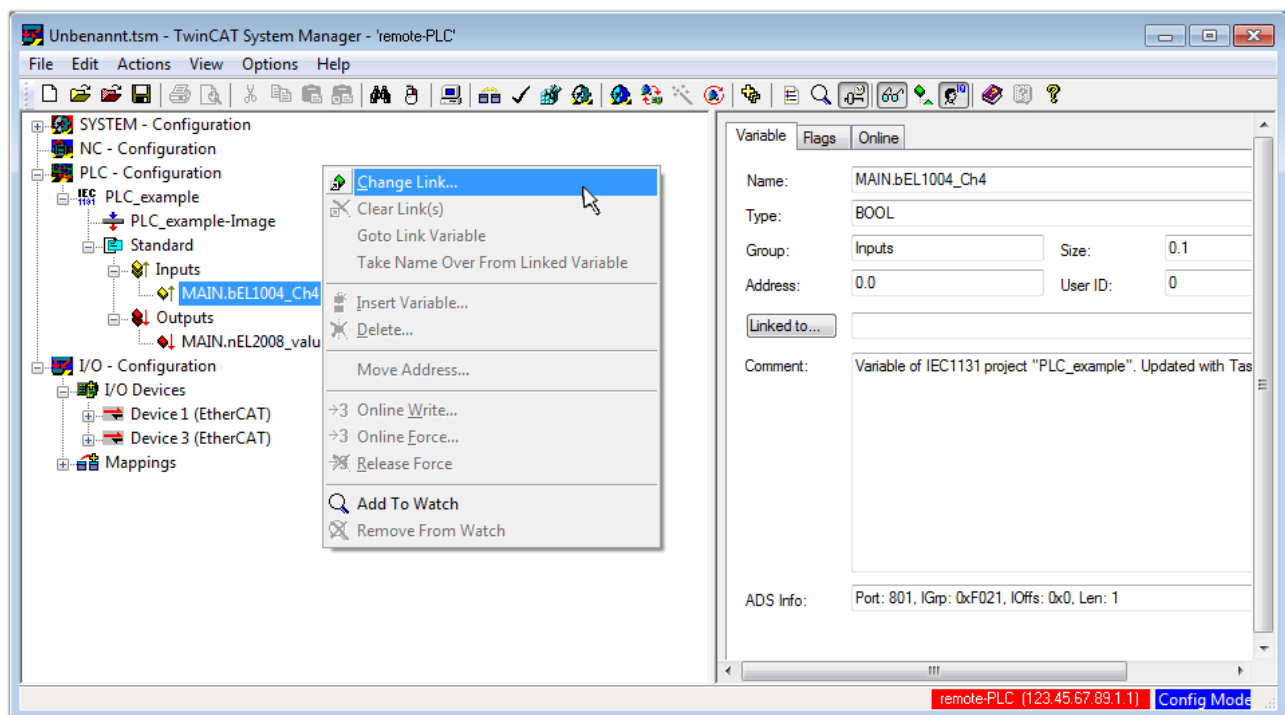
Fig. 32: Creating the links between PLC variables and process objects

In the window that opens, the process object for the "bEL1004_Ch4" BOOL-type variable can be selected from the PLC configuration tree:
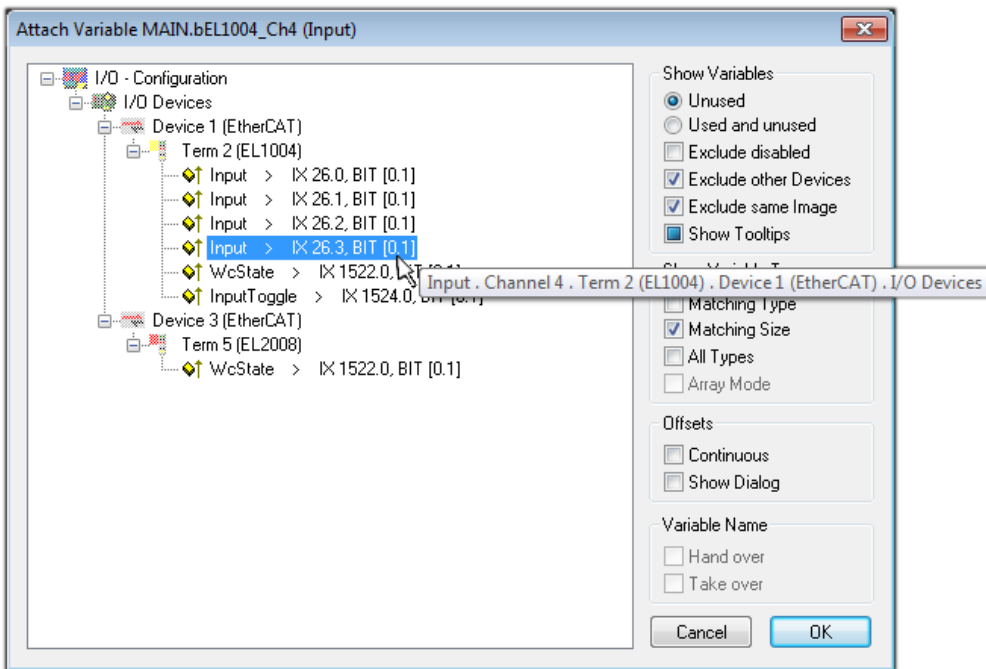
Fig. 33: Selecting BOOL-type PDO

According to the default setting, only certain PDO objects are now available for selection. In this example, the input of channel 4 of the EL1004 terminal is selected for linking. In contrast, the checkbox "All types" must be ticked to create the link for the output variables, in order to allocate a set of eight separate output bits to a byte variable in this case. The following diagram shows the whole process:

Fig. 34: Selecting several PDOs simultaneously: activate "Continuous" and "All types"

Note that the "Continuous" checkbox was also activated. This is designed to allocate the bits contained in the byte of the "nEL2008_value" variable sequentially to all eight selected output bits of the EL2008 Terminal. It is thus possible to subsequently address all eight outputs of the terminal in the program with a byte

corresponding to bit 0 for channel 1 to bit 7 for channel 8 of the PLC. A special symbol (▣) on the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting "Goto Link Variable" from the context menu of a variable. The opposite linked object, in this case the PDO, is automatically selected:

Fig. 35: Application of a "Goto Link Variable", using "MAIN.bEL1004_Ch4" as an example

The process of assigning variables to the PDO is completed via the menu option "Actions" → "Create

assignment", or via .

This can be visualized in the configuration:
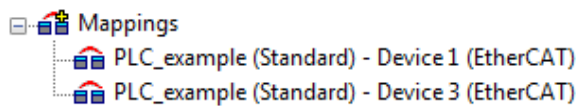
The process of creating links can also be performed in the opposite direction, i.e. starting with individual PDOs to a variable. However, in this example, it would not be possible to select all output bits for the EL2008, since the terminal only makes individual digital outputs available. If a terminal has a byte, word, integer or similar PDO, it is also possible to allocate this to a set of bit-standardized variables. Here, too, a "Goto Link Variable" can be executed in the other direction, so that the respective PLC instance can then be selected.

Activation of the configuration

The allocation of PDO to PLC variables has now established the connection from the controller to the inputs and outputs of the terminals. The configuration can now be activated. First, the configuration can be verified

via (or via "Actions" → "Check Configuration"). If no error is present, the configuration can be

activated via (or via "Actions" → "Activate Configuration…") to transfer the System Manager settings to the runtime system. Confirm the messages "Old configurations will be overwritten!" and "Restart TwinCAT system in Run mode" with "OK".

A few seconds later, the real-time status is displayed at the bottom right in the System Manager. The PLC system can then be started as described below.

Starting the controller

Starting from a remote system, the PLC control has to be linked with the embedded PC over the Ethernet via "Online" → "Choose Runtime System…":
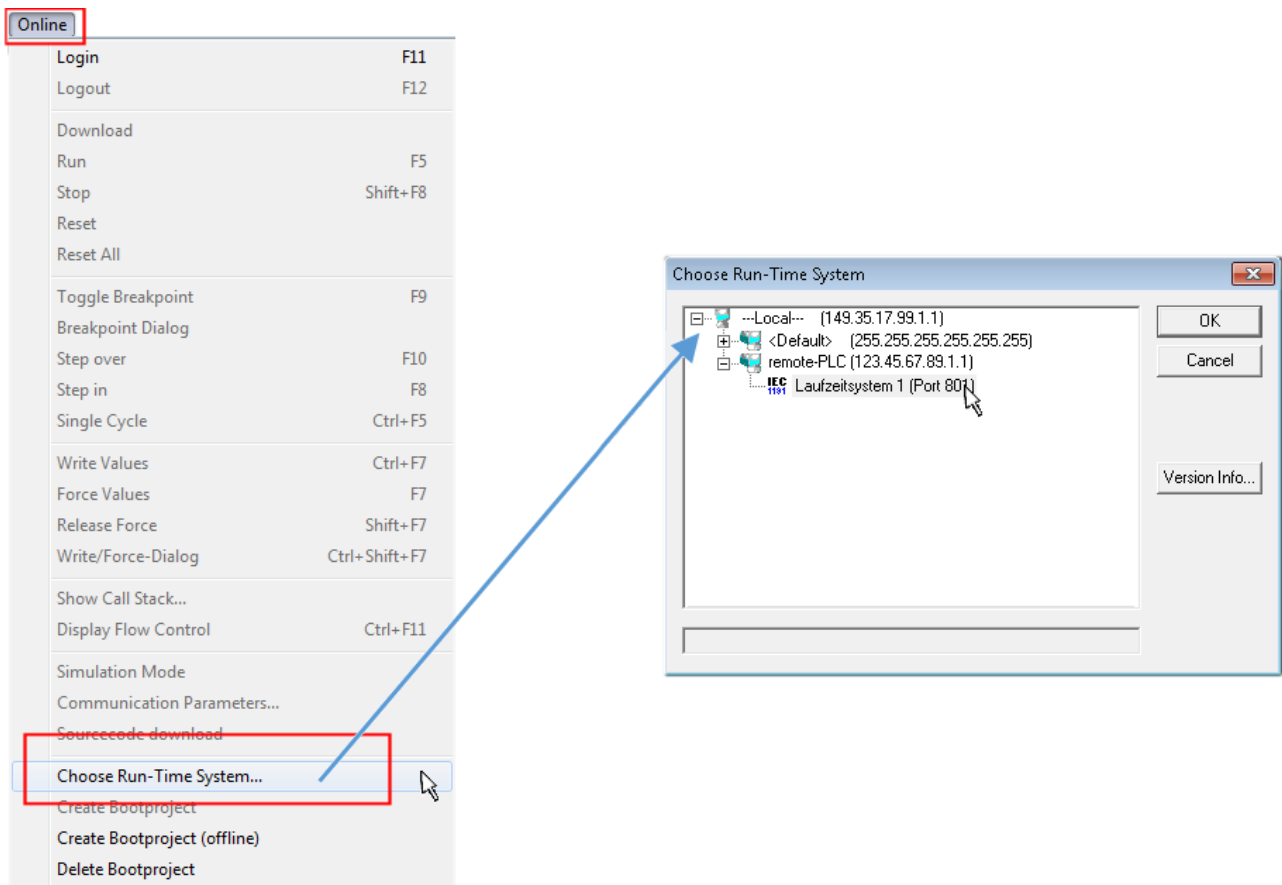
Fig. 36: Choose target system (remote)

In this example, "Runtime system 1 (port 801)" is selected and confirmed. Link the PLC with the real-time

system via the menu option "Online" → "Login", the F11 key or by clicking on the symbol . The control program can then be loaded for execution. This results in the message "No program on the controller! Should the new program be loaded?", which should be confirmed with "Yes". The runtime environment is ready for the program start:
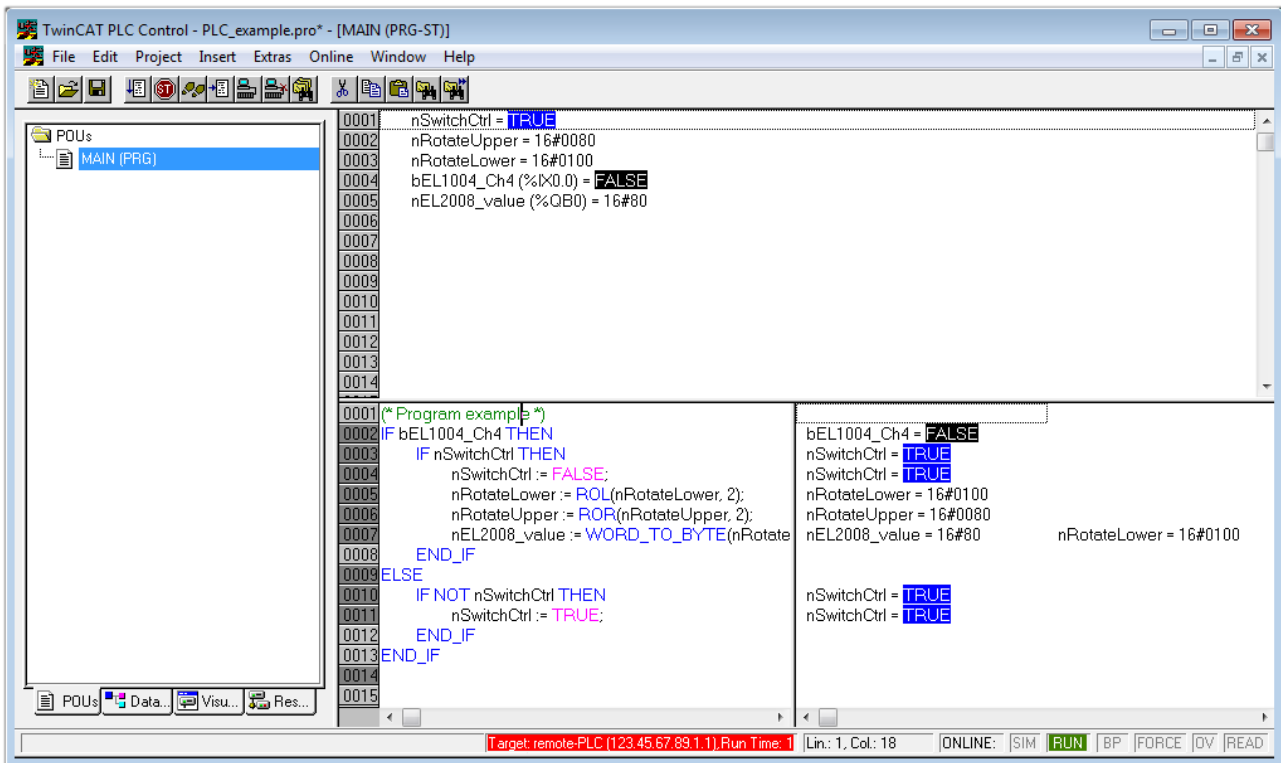
BECKHOFF

Fig. 37: PLC Control logged in, ready for program startup

The PLC can now be started via "Online" → "Run", F5 key or .

5.1.2 TwinCAT 3

Startup

TwinCAT 3 makes the development environment areas available all together, with Microsoft Visual Studio: after startup, the project folder explorer appears on the left in the general window area (see "TwinCAT System Manager" of TwinCAT 2) for communication with the electromechanical components.

After successful installation of the TwinCAT system on the PC to be used for development, TwinCAT 3 (shell) displays the following user interface after startup:

Fig. 38: Initial TwinCAT 3 user interface

First create a new project via (or under "File"→"New"→ "Project…"). In the following dialog, make the corresponding entries as required (as shown in the diagram):
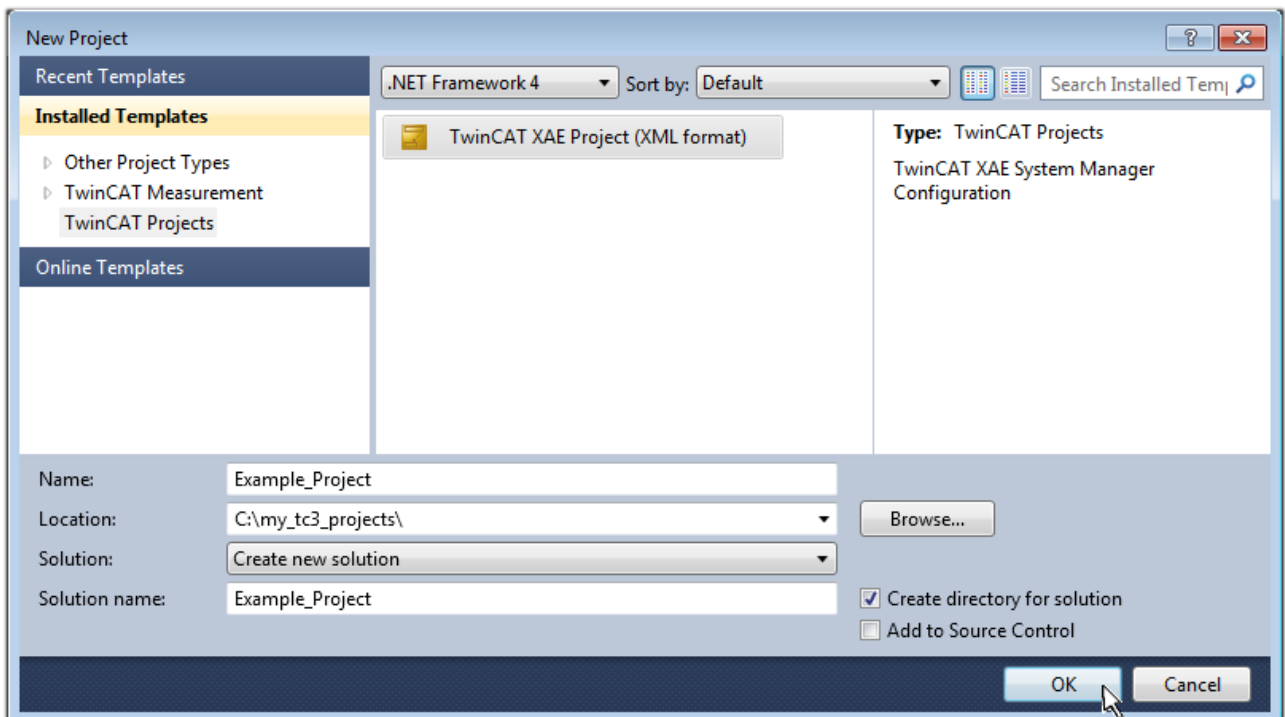
Fig. 39: Create new TwinCAT 3 project

The new project is then available in the project folder explorer:
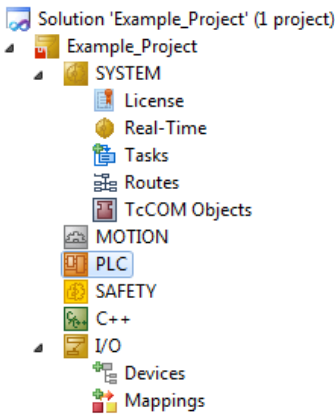
Fig. 40: New TwinCAT 3 project in the project folder explorer

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC (locally), TwinCAT can be used in local mode and the process can be continued with the next step, "Insert Device [▶ 53]".

If the intention is to address the TwinCAT runtime environment installed on a PLC remotely from another system used as a development environment, the target system must be made known first. Via the symbol in the menu bar:
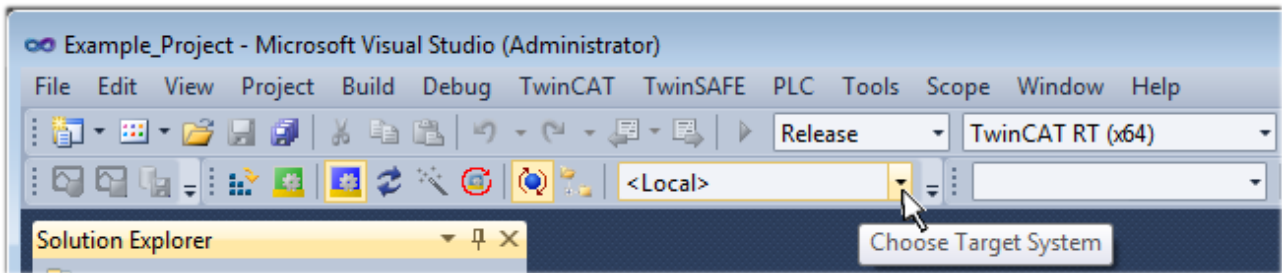
expand the pull-down menu:
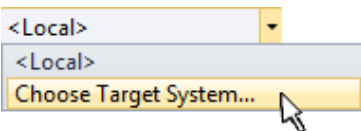
and open the following window:
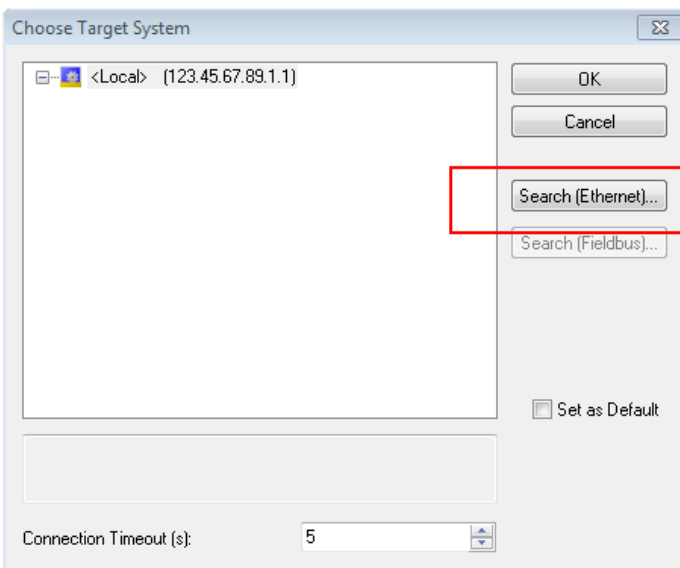
Fig. 41: Selection dialog: Choose the target system

Use "Search (Ethernet)..." to enter the target system. Thus another dialog opens to either:

- enter the known computer name after "Enter Host Name / IP:" (as shown in red)
- perform a "Broadcast Search" (if the exact computer name is not known)
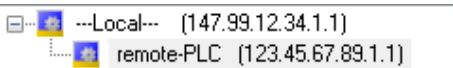- enter the known computer – IP or AmsNetID
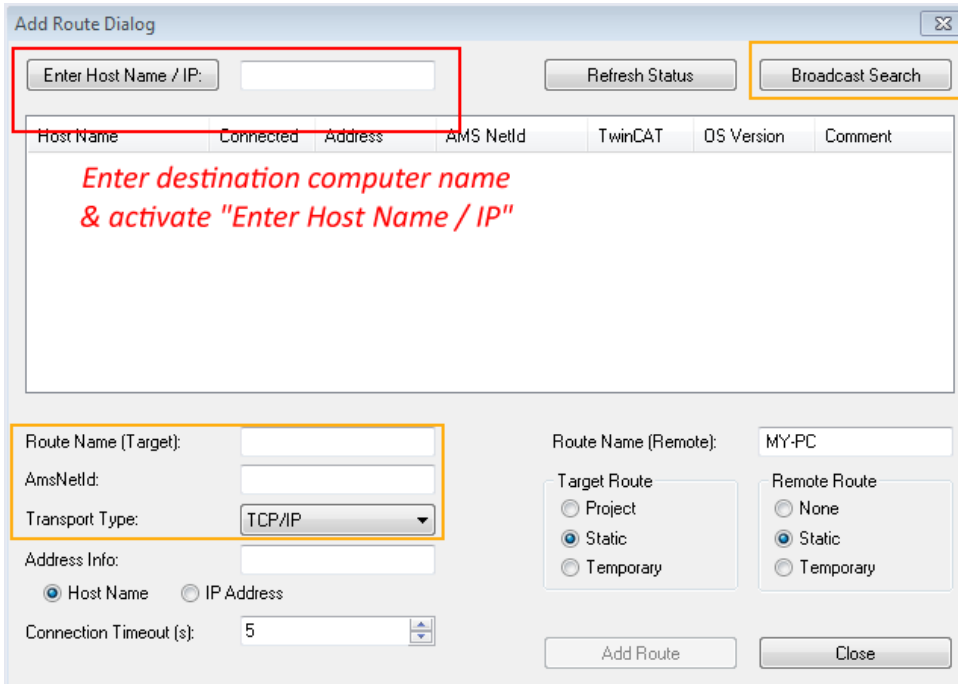
Fig. 42: specify the PLC for access by the TwinCAT System Manager: selection of the target system

Once the target system has been entered, it is available for selection as follows (the correct password may have to be entered beforehand):

After confirmation with "OK" the target system can be accessed via the Visual Studio shell.

Adding devices

In the project folder explorer on the left of the Visual Studio shell user interface, select "Devices" within the

element "I/O", then right-click to open a context menu and select "Scan" or start the action via [icon] in the

menu bar. The TwinCAT System Manager may first have to be set to "Config mode" via [icon] or via the menu "TwinCAT" → "Restart TwinCAT (Config Mode)".
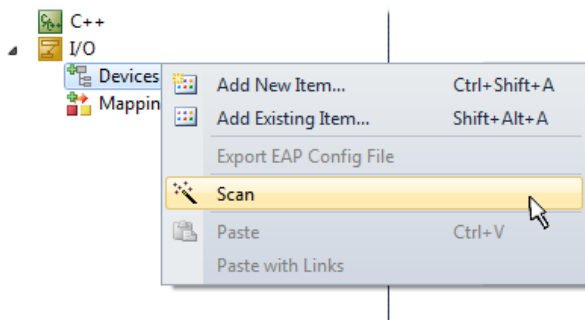

Fig. 43: Select "Scan"

Confirm the warning message, which follows, and select the "EtherCAT" devices in the dialog:
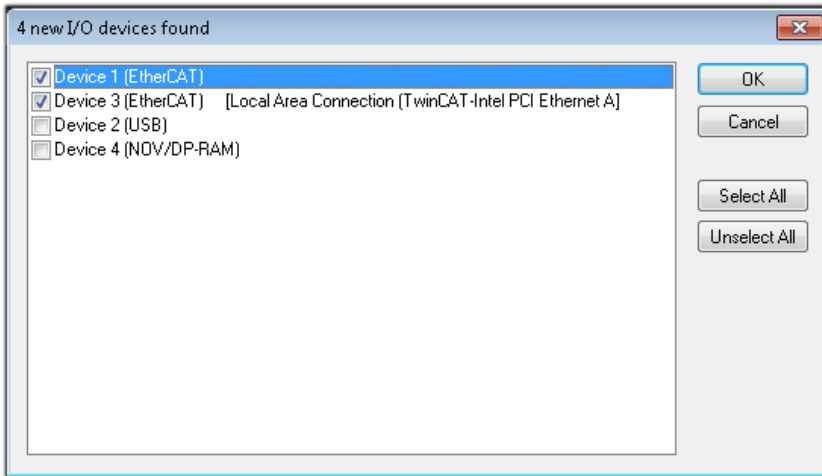
BECKHOFF

Fig. 44: Automatic detection of I/O devices: selection of the devices to be integrated

Confirm the message "Find new boxes", in order to determine the terminals connected to the devices. "Free Run" enables manipulation of input and output values in "Config Mode" and should also be acknowledged.

Based on the example configuration [▶ 38] described at the beginning of this section, the result is as follows:
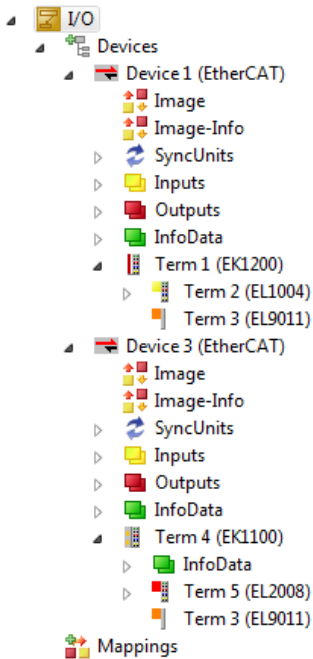

Fig. 45: Mapping of the configuration in VS shell of the TwinCAT 3 environment

The whole process consists of two stages, which can also be performed separately (first determine the devices, then determine the connected elements such as boxes, terminals, etc.). A scan (search function) can also be initiated by selecting "Device ..." from the context menu, which then only reads the elements below which are present in the configuration:
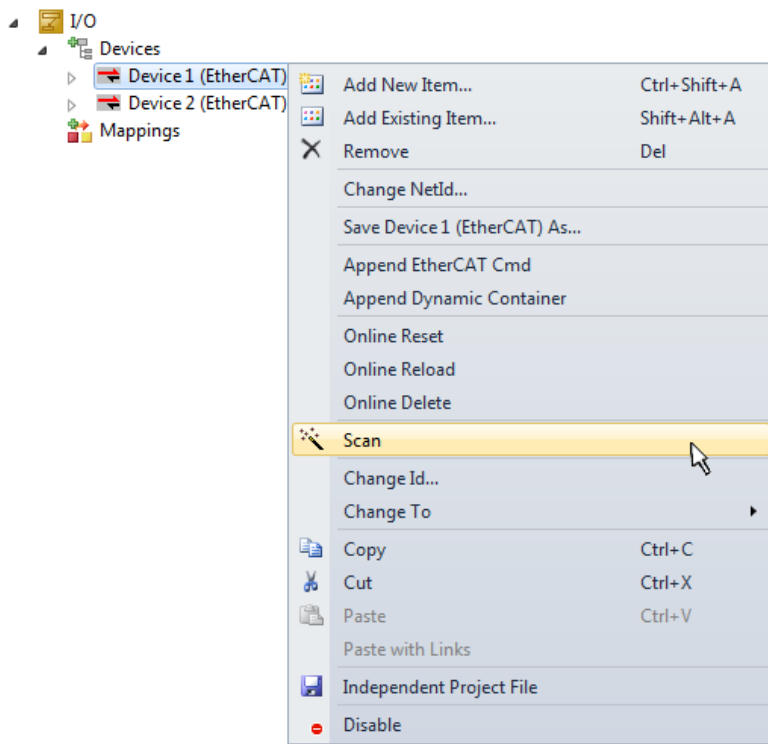
Version: 1.1

Fig. 46: Reading of individual terminals connected to a device

This functionality is useful if the actual configuration is modified at short notice.

Programming the PLC

TwinCAT PLC Control is the development environment for generating the controller in different program environments: TwinCAT PLC Control supports all languages described in IEC 61131-3. There are two text-based languages and three graphical languages.

- **Text-based languages**
  - Instruction List (IL)
  - Structured Text (ST)
- **Graphical languages**
  - Function Block Diagram (FBD)
  - Ladder Diagram (LD)
  - The Continuous Function Chart Editor (CFC)
  - Sequential Function Chart (SFC)

The following section refers solely to Structured Text (ST).

In order to create a programming environment, a PLC subproject is added to the example project via the context menu of the "PLC" in the project folder explorer by selecting "Add New Item….":
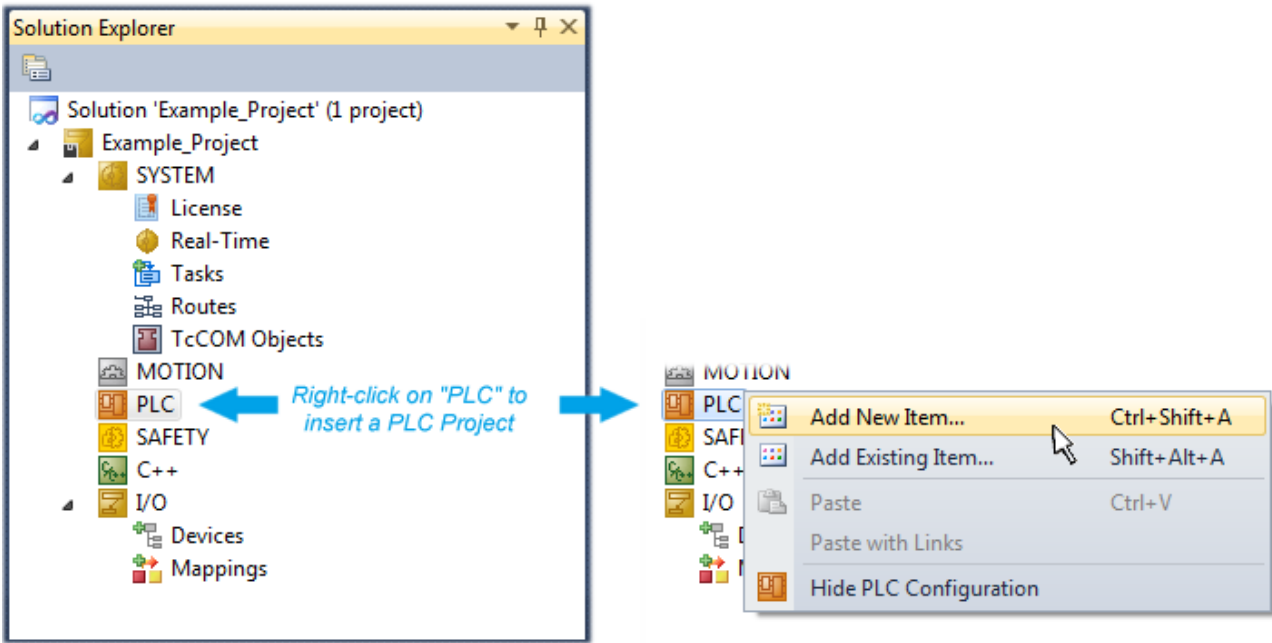
BECKHOFF

Fig. 47: Adding the programming environment in "PLC"

In the dialog that opens, select "Standard PLC project" and enter "PLC_example" as project name, for example, and select a corresponding directory:
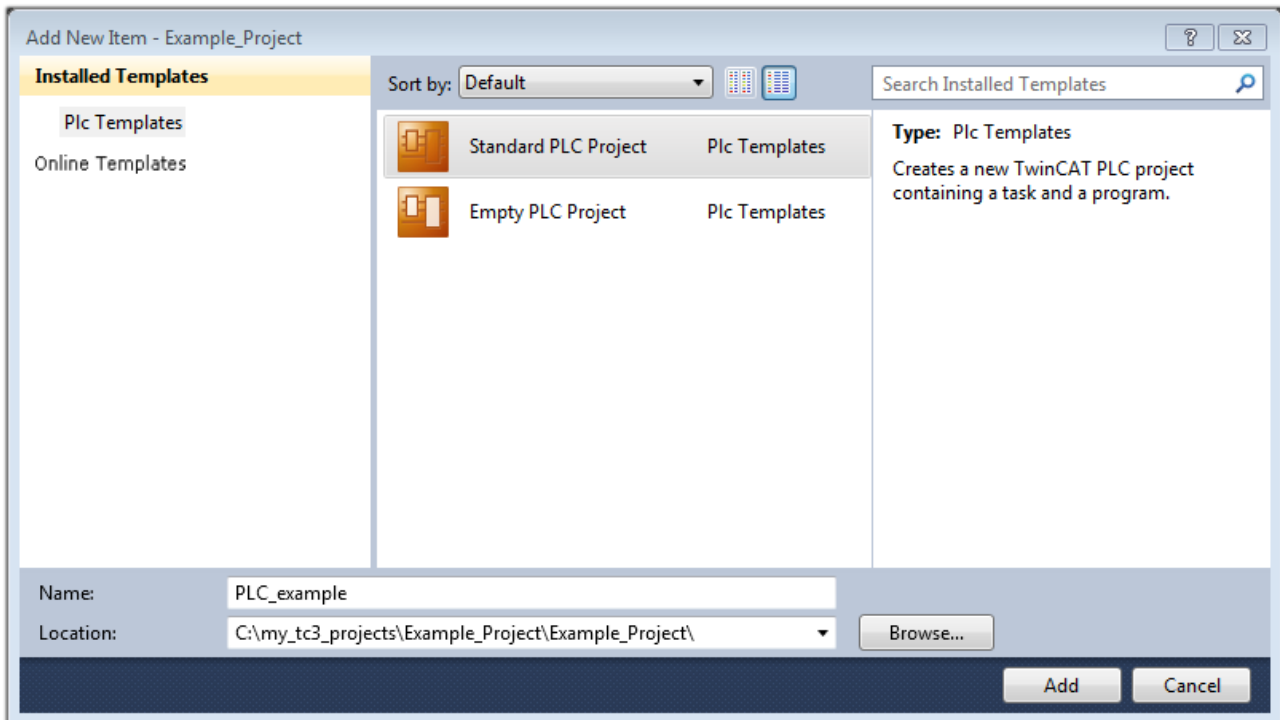

Fig. 48: Specifying the name and directory for the PLC programming environment

The "Main" program, which already exists due to selecting "Standard PLC project", can be opened by double-clicking on "PLC_example_project" in "POUs". The following user interface is shown for an initial project:

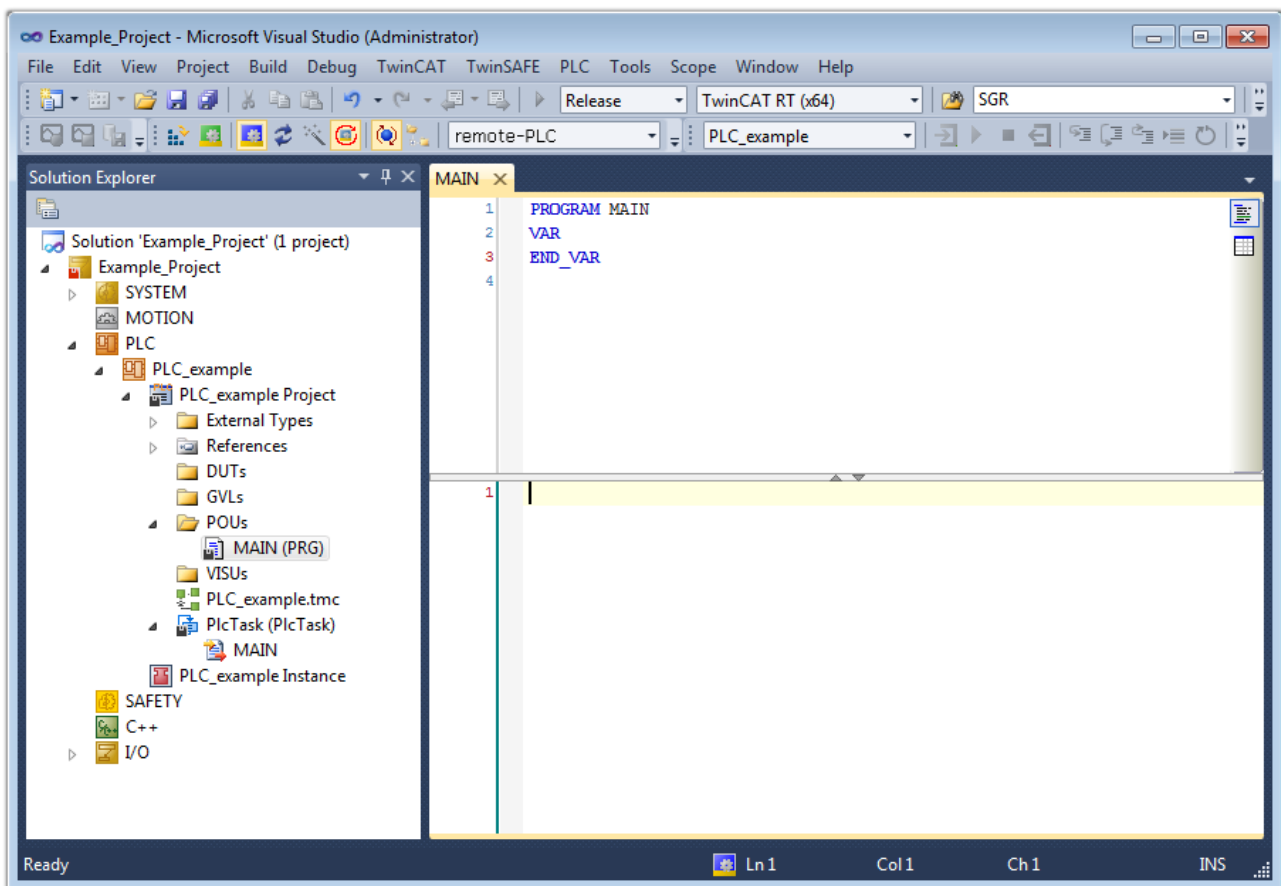Version: 1.1   EL6711-0010

Fig. 49: Initial "Main" program for the standard PLC project

Now example variables and an example program have been created for the next stage of the process:
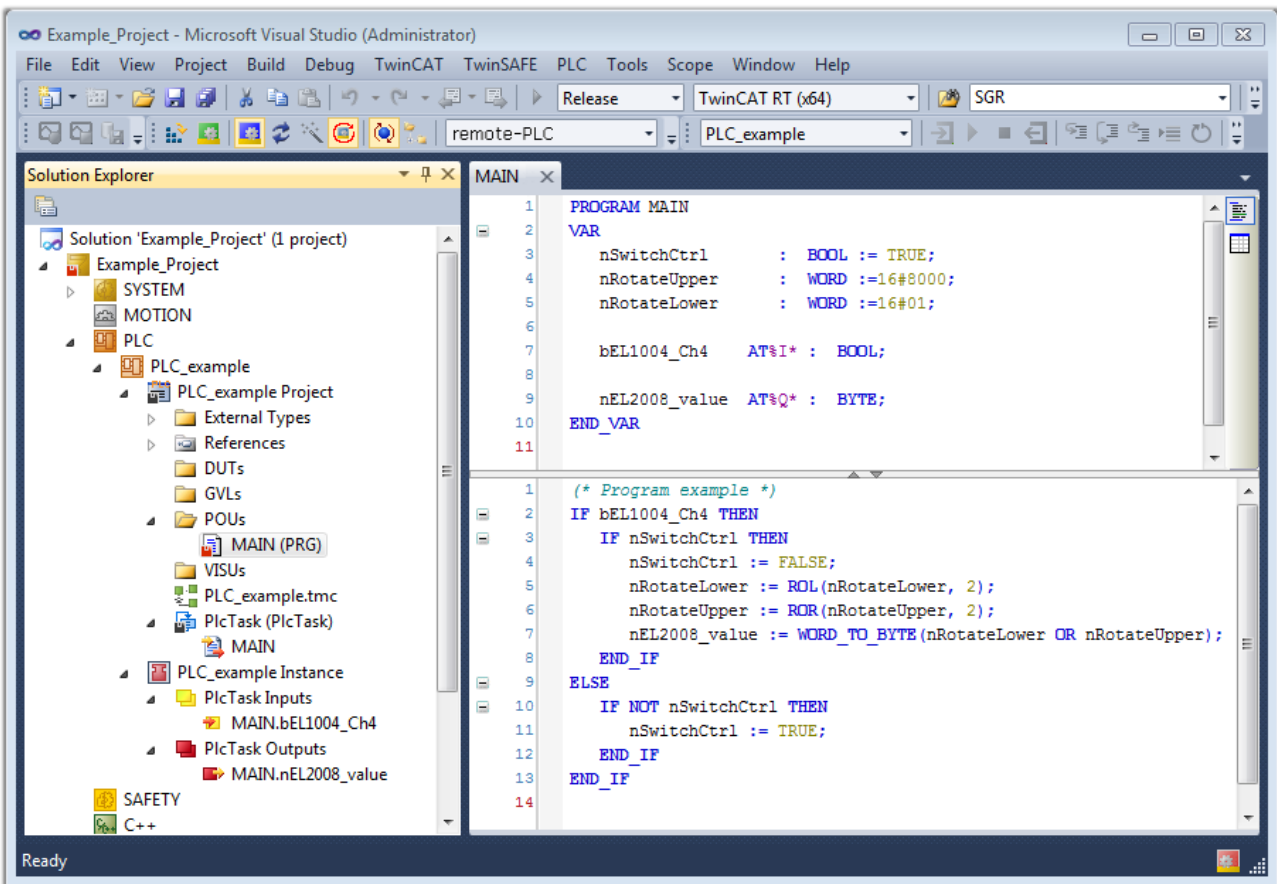
BECKHOFF

Fig. 50: Example program with variables after a compile process (without variable integration)

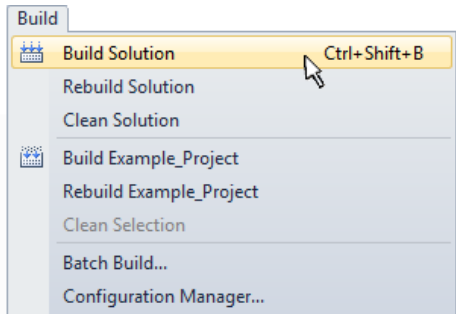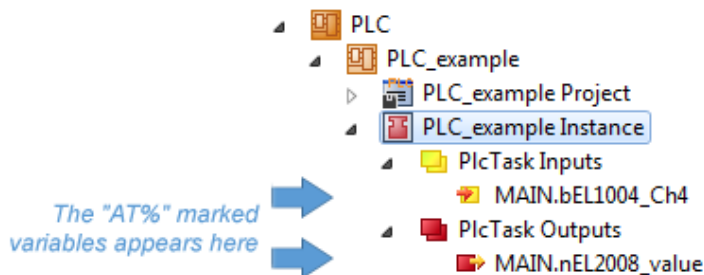The control program is now created as a project folder, followed by the compile process:

Fig. 51: Start program compilation

The following variables, identified in the ST/PLC program with "AT%", are then available under "Assignments" in the project folder explorer:

Assigning variables

Via the menu of an instance – variables in the "PLC" context, use the "Modify Link…" option to open a window to select a suitable process object (PDO) for linking:
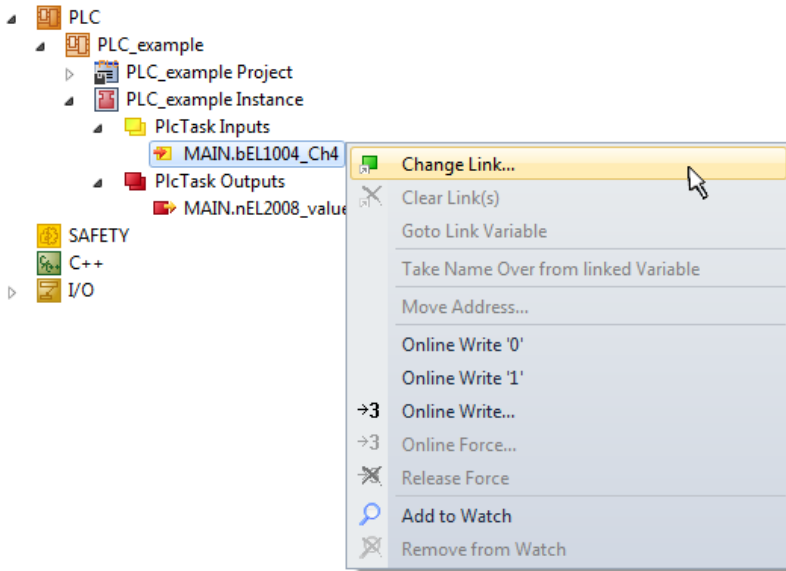
Fig. 52: Creating the links between PLC variables and process objects

In the window that opens, the process object for the "bEL1004_Ch4" BOOL-type variable can be selected from the PLC configuration tree:
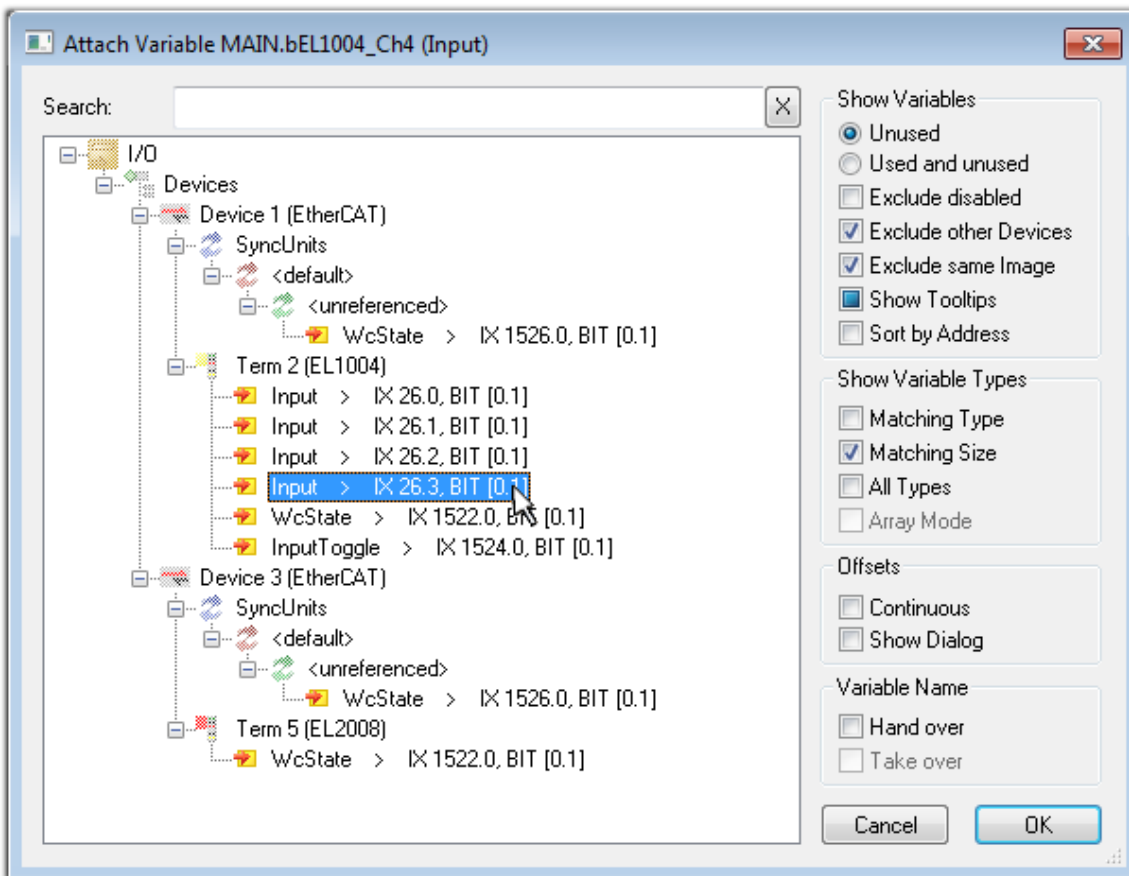
Fig. 53: Selecting BOOL-type PDO

According to the default setting, only certain PDO objects are now available for selection. In this example, the input of channel 4 of the EL1004 terminal is selected for linking. In contrast, the checkbox "All types" must be ticked to create the link for the output variables, in order to allocate a set of eight separate output bits to a byte variable in this case. The following diagram shows the whole process:
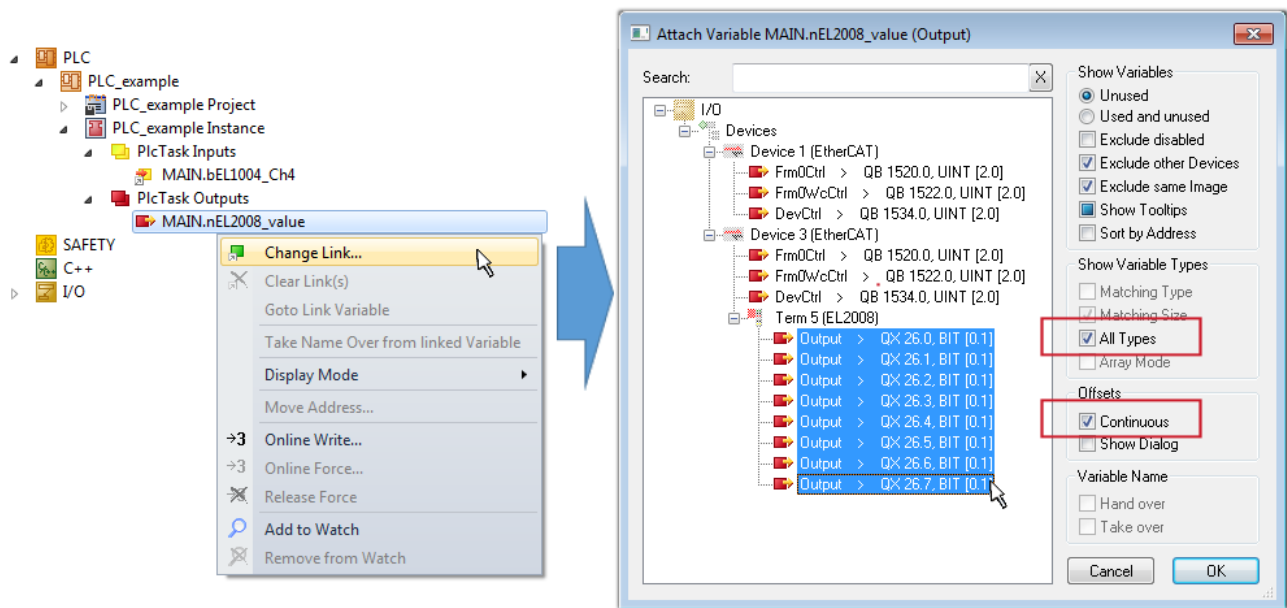
Fig. 54: Selecting several PDOs simultaneously: activate "Continuous" and "All types"

Note that the "Continuous" checkbox was also activated. This is designed to allocate the bits contained in the byte of the "nEL2008_value" variable sequentially to all eight selected output bits of the EL2008 Terminal. It is thus possible to subsequently address all eight outputs of the terminal in the program with a byte

corresponding to bit 0 for channel 1 to bit 7 for channel 8 of the PLC. A special symbol (⊡) on the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting "Goto Link Variable" from the context menu of a variable. The opposite linked object, in this case the PDO, is automatically selected:

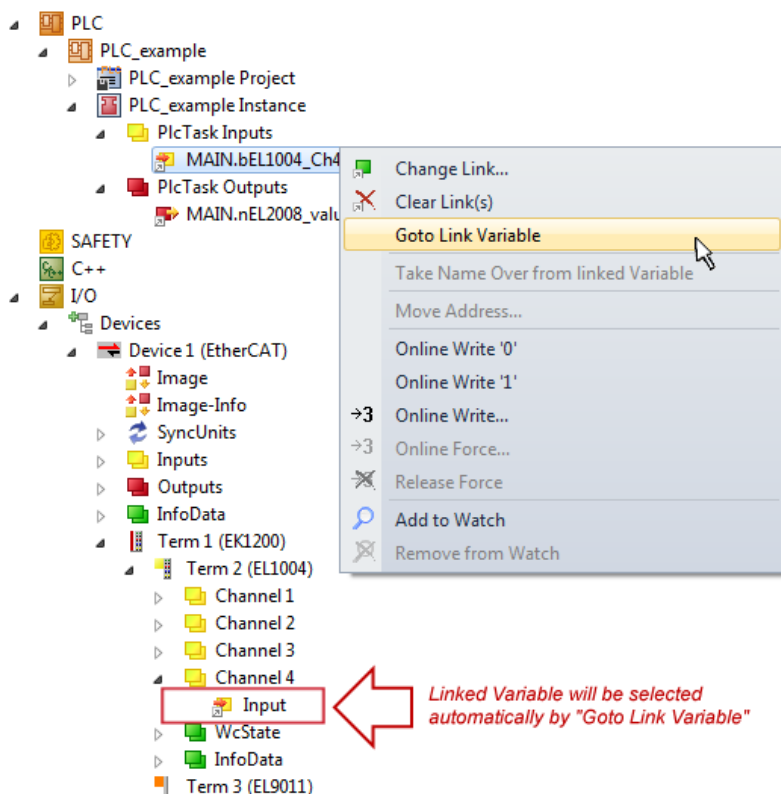Fig. 55: Application of a "Goto Link Variable", using "MAIN.bEL1004_Ch4" as an example

The process of creating links can also be performed in the opposite direction, i.e. starting with individual PDOs to a variable. However, in this example, it would not be possible to select all output bits for the EL2008, since the terminal only makes individual digital outputs available. If a terminal has a byte, word,

integer or similar PDO, it is also possible to allocate this to a set of bit-standardized variables. Here, too, a "Goto Link Variable" can be executed in the other direction, so that the respective PLC instance can then be selected.

ℹ Note on type of variable assignment

The following type of variable assignment can only be used from TwinCAT version V3.1.4024.4 onwards and is only available for terminals with a microcontroller.

In TwinCAT, a structure can be created from the mapped process data of a terminal. An instance of this structure can then be created in the PLC, so it is possible to access the process data directly from the PLC without having to declare own variables.

The procedure for the EL3001 1-channel analog input terminal -10...+10 V is shown as an example.

1. First, the required process data must be selected in the "Process data" tab in TwinCAT.
2. After that, the PLC data type must be generated in the "PLC" tab via the check box.
3. The data type in the "Data Type" field can then be copied using the "Copy" button.

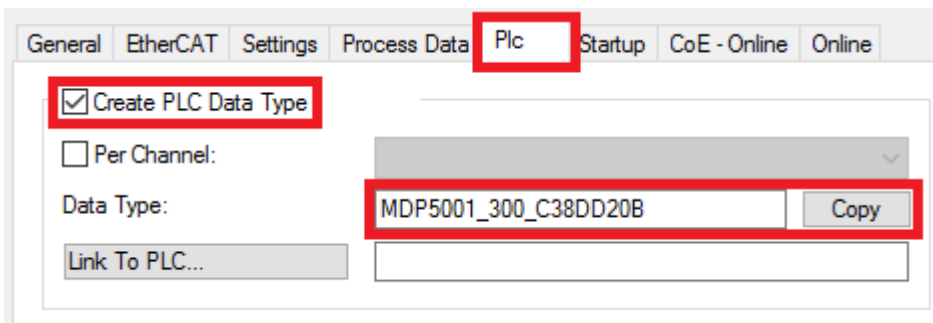Fig. 56: Creating a PLC data type

4. An instance of the data structure of the copied data type must then be created in the PLC.
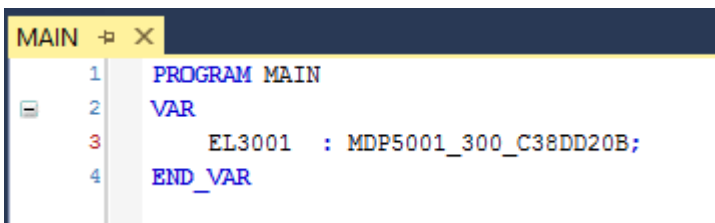

Fig. 57: Instance_of_struct

5. Then the project folder must be created. This can be done either via the key combination "CTRL + Shift + B" or via the "Build" tab in TwinCAT.
6. The structure in the "PLC" tab of the terminal must then be linked to the created instance.
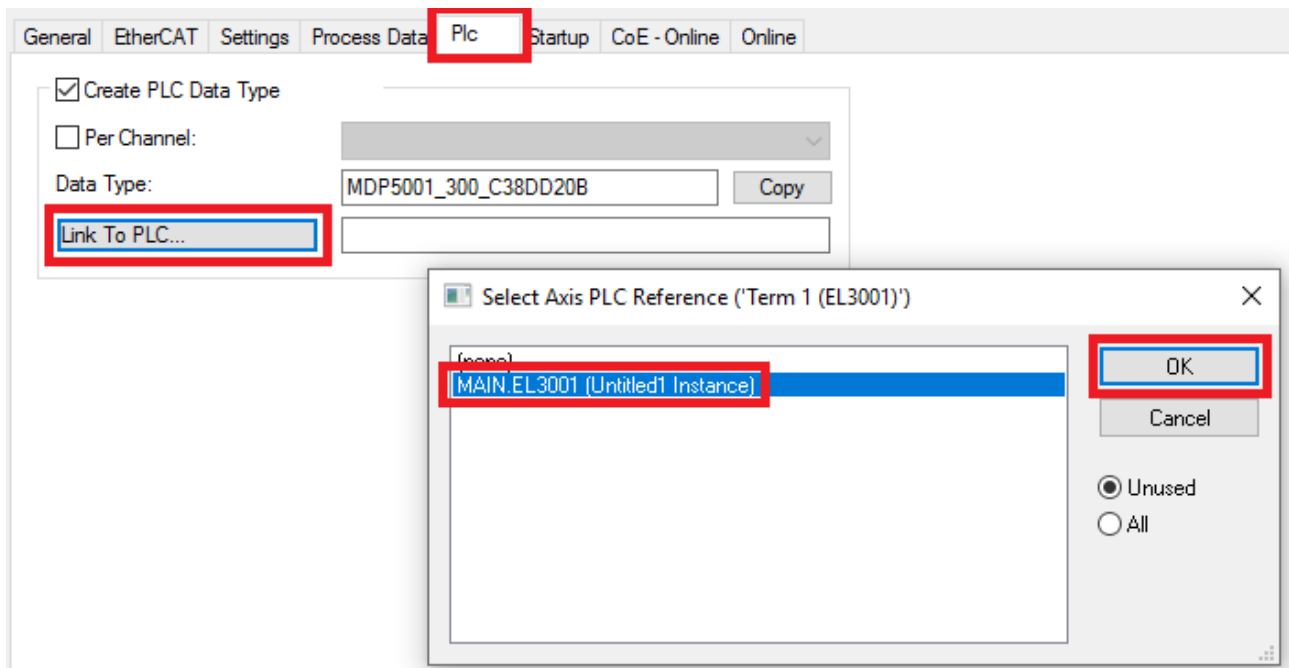
BECKHOFF

Fig. 58: Linking the structure

7. In the PLC, the process data can then be read or written via the structure in the program code.
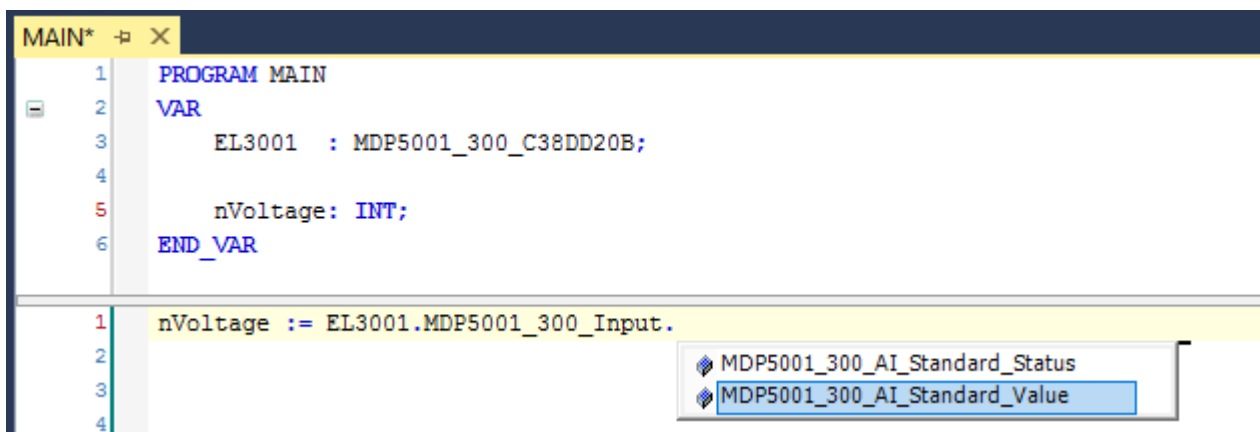
Fig. 59: Reading a variable from the structure of the process data

Activation of the configuration

The allocation of PDO to PLC variables has now established the connection from the controller to the inputs

and outputs of the terminals. The configuration can now be activated with or via the menu under "TwinCAT" in order to transfer the settings of the development environment to the runtime system. Confirm the messages "Old configurations will be overwritten!" and "Restart TwinCAT system in Run mode" with "OK". The corresponding assignments can be seen in the project folder explorer:
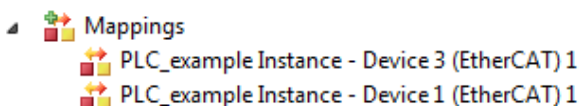
A few seconds later, the corresponding status of the Run mode is displayed in the form of a rotating symbol

 at the bottom right of the VS shell development environment. The PLC system can then be started as described below.

Starting the controller

Select the menu option "PLC" → "Login" or click on ⏭ to link the PLC with the real-time system and load the control program for execution. This results in the message "*No program on the controller! Should the new program be loaded?*", which should be acknowledged with "Yes". The runtime environment is ready for

the program to be started by clicking on symbol ▶, the "F5" key or via "PLC" in the menu, by selecting "Start". The started programming environment shows the runtime values of individual variables:
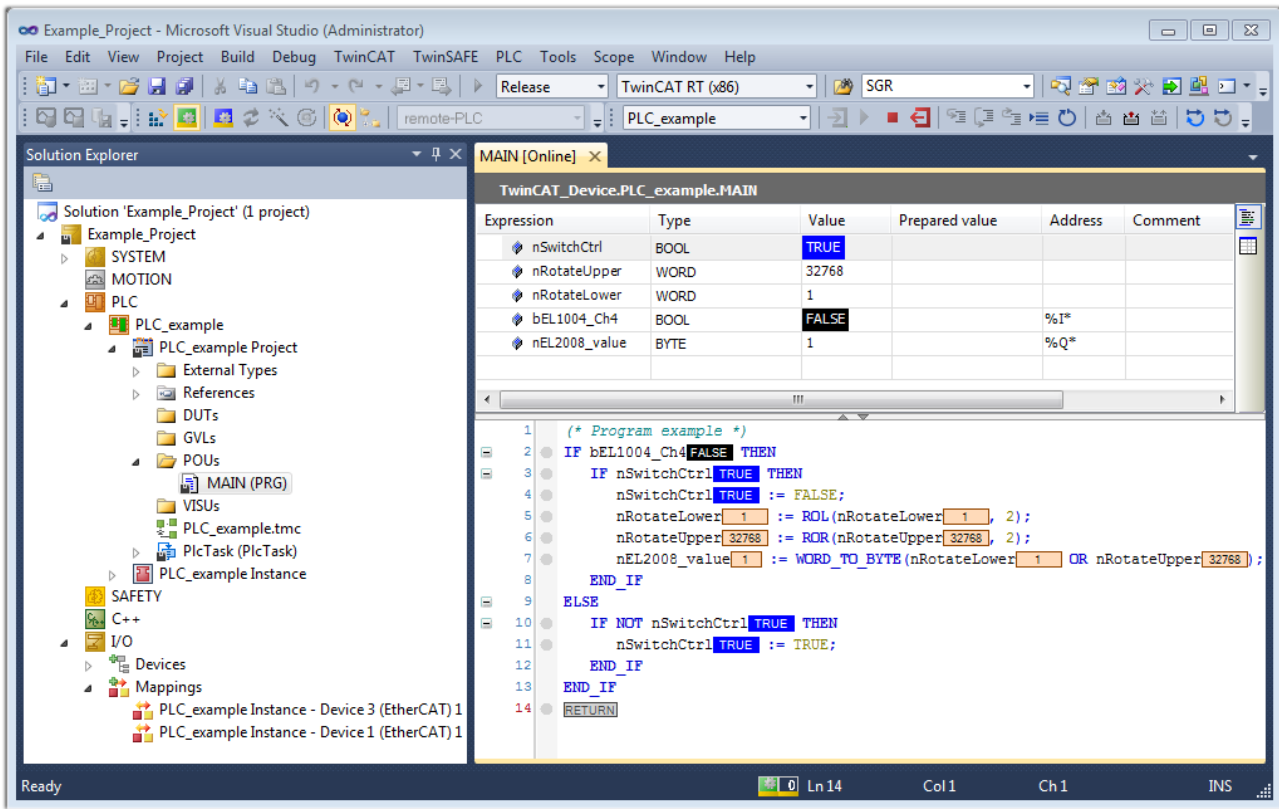
Fig. 60: TwinCAT 3 development environment (VS shell): logged-in, after program startup

The two operator control elements for stopping ■ and logout ⏮ result in the required action (also, "Shift + F5" can be used for stop, or both actions can be selected via the PLC menu).

5.2   TwinCAT Development Environment

The Software for automation TwinCAT (The Windows Control and Automation Technology) will be distinguished into:

- TwinCAT 2: System Manager (Configuration) & PLC Control (Programming)
- TwinCAT 3: Enhancement of TwinCAT 2 (Programming and Configuration takes place via a common Development Environment)

Details:

- **TwinCAT 2:**
  ◦ Connects I/O devices to tasks in a variable-oriented manner
  ◦ Connects tasks to tasks in a variable-oriented manner
  ◦ Supports units at the bit level
  ◦ Supports synchronous or asynchronous relationships
  ◦ Exchange of consistent data areas and process images

◦ Datalink on NT - Programs by open Microsoft Standards (OLE, OCX, ActiveX, DCOM+, etc.)

◦ Integration of IEC 61131-3-Software-SPS, Software- NC and Software-CNC within Windows NT/ 2000/XP/Vista, Windows 7, NT/XP Embedded, CE

◦ Interconnection to all common fieldbusses

◦ More...

Additional features:

- **TwinCAT 3** (eXtended Automation)**:**

  ◦ Visual Studio® integration

  ◦ Choice of the programming language

  ◦ Supports object orientated extension of IEC 61131-3

  ◦ Usage of C/C++ as programming language for real time applications

  ◦ Connection to MATLAB®/Simulink®

  ◦ Open interface for expandability

  ◦ Flexible run-time environment

  ◦ Active support of multi-core- and 64 bit operating system

  ◦ Automatic code generation and project creation with the TwinCAT Automation Interface

  ◦ More...

Within the following sections commissioning of the TwinCAT Development Environment on a PC System for the control and also the basically functions of unique control elements will be explained.

Please see further information to TwinCAT 2 and TwinCAT 3 at http://infosys.beckhoff.com.

5.2.1 Installation of the TwinCAT real-time driver

In order to assign real-time capability to a standard Ethernet port of an IPC controller, the Beckhoff real-time driver has to be installed on this port under Windows.

This can be done in several ways.

A: Via the TwinCAT Adapter dialog

In the System Manager call up the TwinCAT overview of the local network interfaces via Options → Show Real Time Ethernet Compatible Devices.
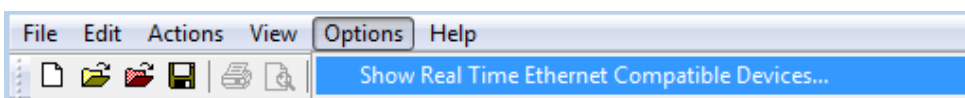
Fig. 61: System Manager "Options" (TwinCAT 2)

This have to be called up by the menu "TwinCAT" within the TwinCAT 3 environment:
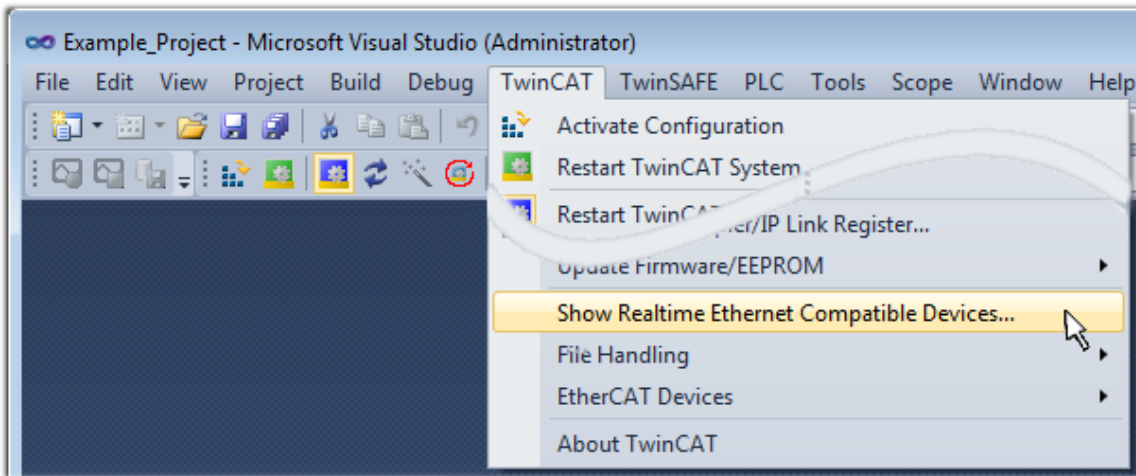
Fig. 62: Call up under VS Shell (TwinCAT 3)

B: Via TcRteInstall.exe in the TwinCAT directory
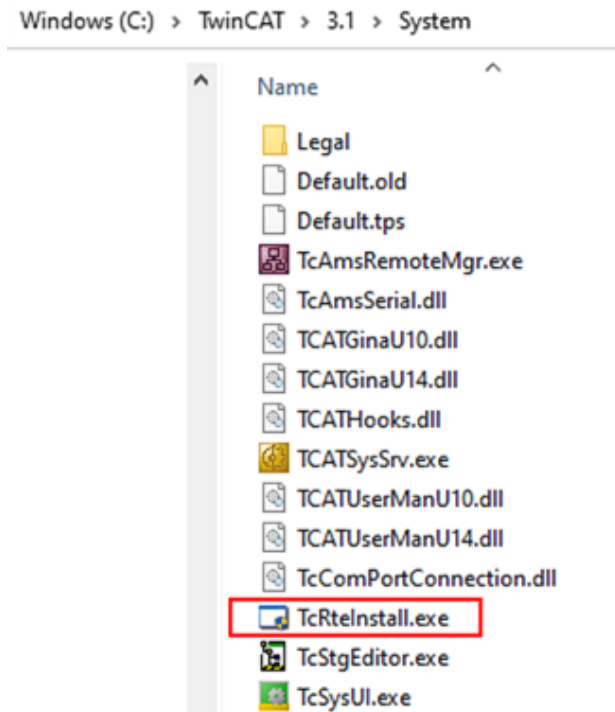
Fig. 63: TcRteInstall in the TwinCAT directory

In both cases, the following dialog appears:

BECKHOFF
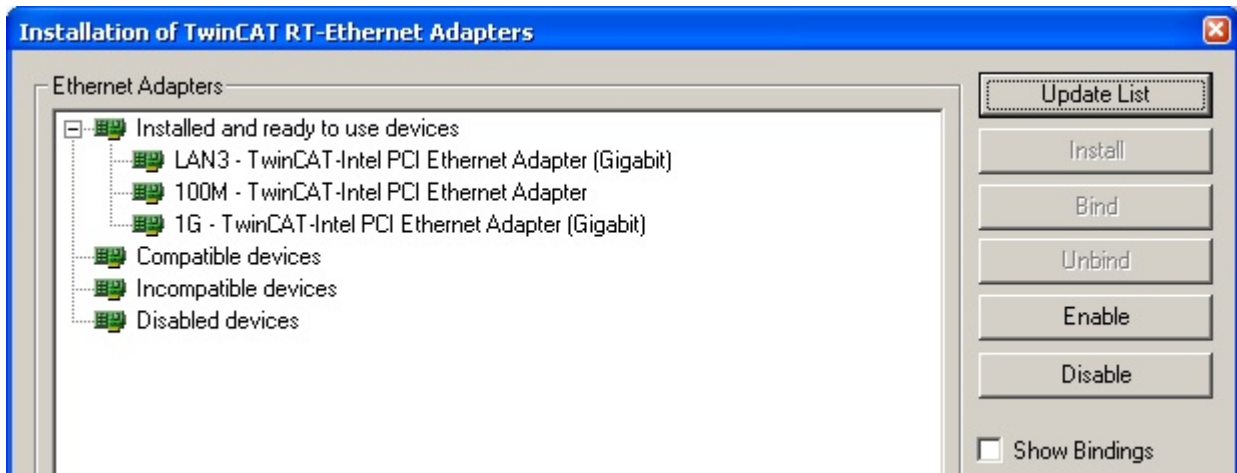
Fig. 64: Overview of network interfaces

Interfaces listed under "Compatible devices" can be assigned a driver via the "Install" button. A driver should only be installed on compatible devices.

A Windows warning regarding the unsigned driver can be ignored.

Alternatively an EtherCAT-device can be inserted first of all as described in chapter Offline configuration creation, section "Creating the EtherCAT device" [▶ 75] in order to view the compatible ethernet ports via its EtherCAT properties (tab "Adapter", button "Compatible Devices…"):

Fig. 65: EtherCAT device properties (TwinCAT 2): click on "Compatible Devices…" of tab "Adapter"

TwinCAT 3: the properties of the EtherCAT device can be opened by double click on "Device .. (EtherCAT)" within the Solution Explorer under "I/O":
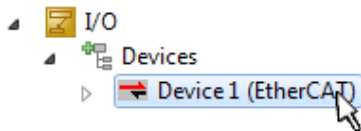
After the installation the driver appears activated in the Windows overview for the network interface (Windows Start → System Properties → Network)
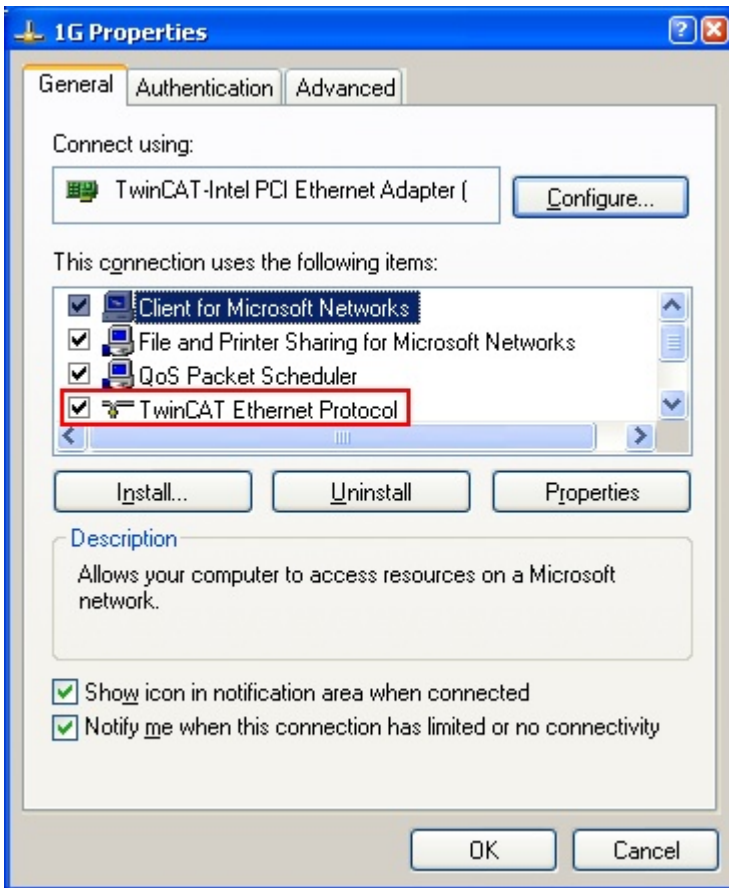
Fig. 66: Windows properties of the network interface

A correct setting of the driver could be:
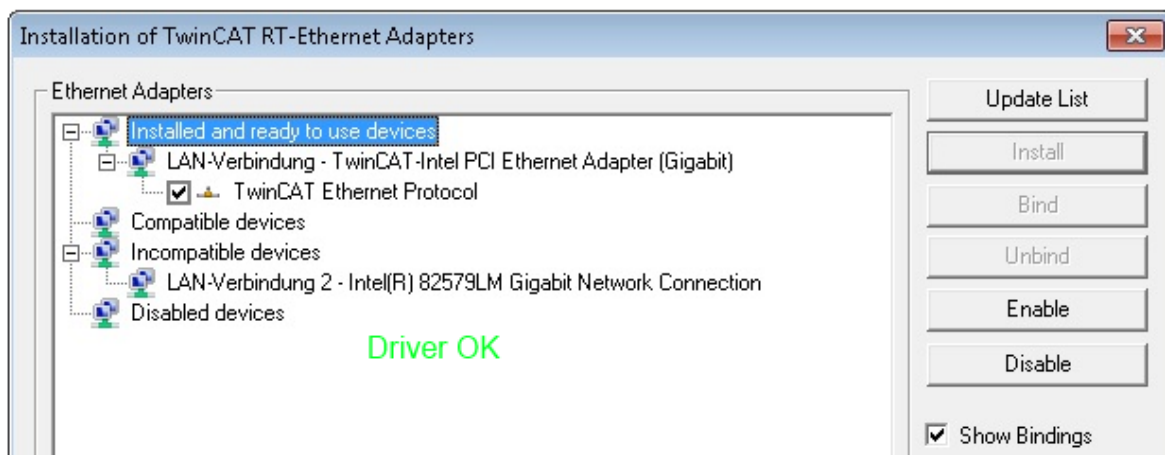
Fig. 67: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:
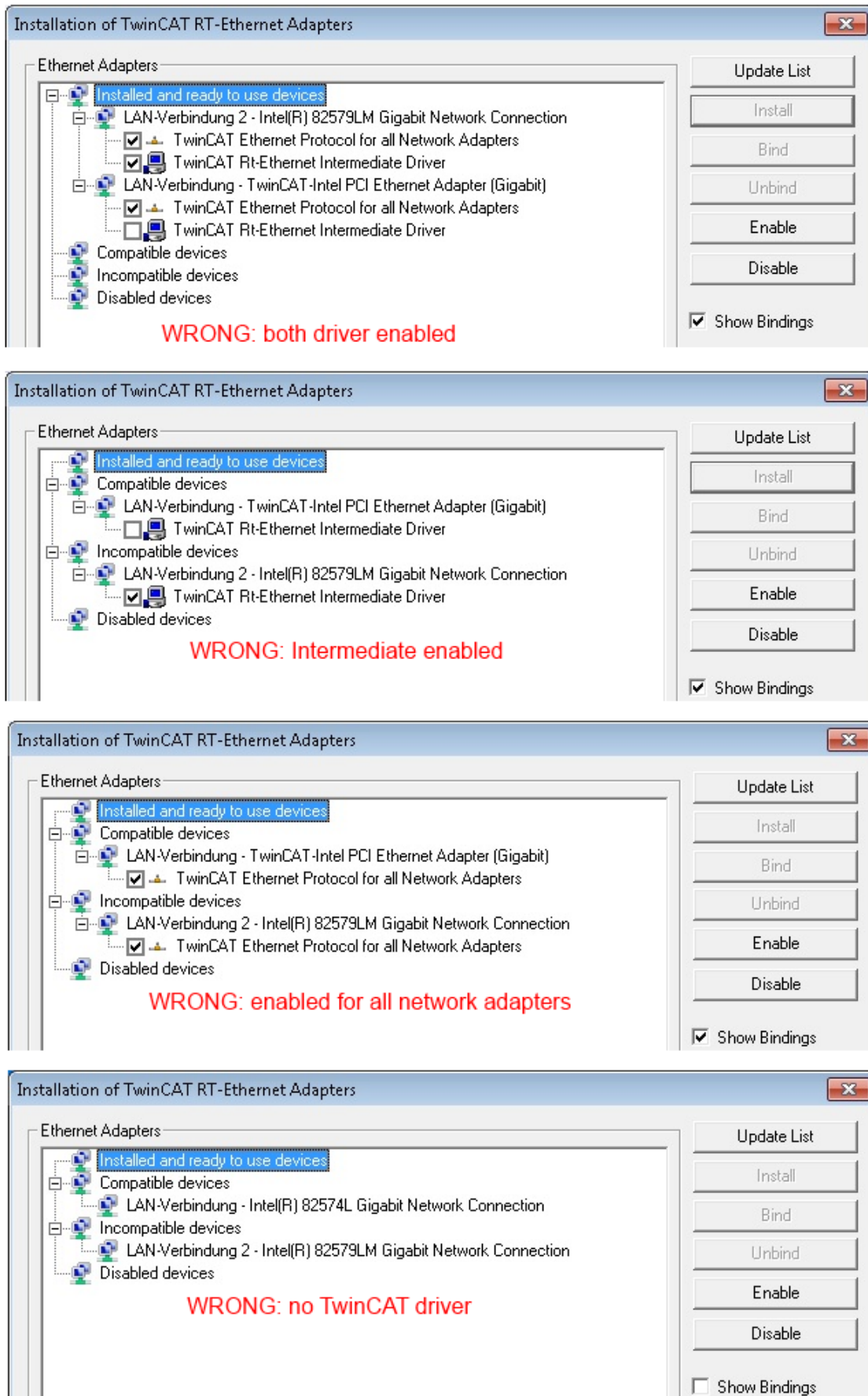
Fig. 68: Incorrect driver settings for the Ethernet port

IP address of the port used

ℹ️ ● **IP address/DHCP**

In most cases an Ethernet port that is configured as an EtherCAT device will not transport general IP packets. For this reason and in cases where an EL6601 or similar devices are used it is useful to specify a fixed IP address for this port via the "Internet Protocol TCP/IP" driver setting and to disable DHCP. In this way the delay associated with the DHCP client for the Ethernet port assigning itself a default IP address in the absence of a DHCP server is avoided. A suitable address space is 192.168.x.x, for example.
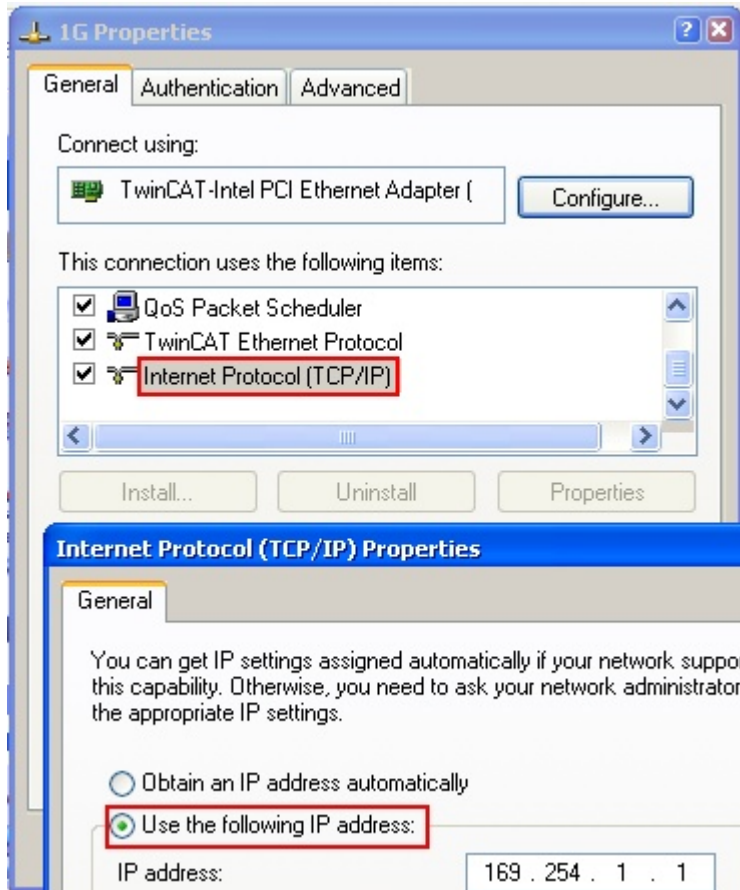
Fig. 69: TCP/IP setting for the Ethernet port

5.2.2 Notes regarding ESI device description

Installation of the latest ESI device description

The TwinCAT EtherCAT master/System Manager needs the device description files for the devices to be used in order to generate the configuration in online or offline mode. The device descriptions are contained in the so-called ESI files (EtherCAT Slave Information) in XML format. These files can be requested from the respective manufacturer and are made available for download. An *.xml file may contain several device descriptions.

The ESI files for Beckhoff EtherCAT devices are available on the Beckhoff website.

The ESI files should be stored in the TwinCAT installation directory.

Default settings:

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

The files are read (once) when a new System Manager window is opened, if they have changed since the last time the System Manager window was opened.

A TwinCAT installation includes the set of Beckhoff ESI files that was current at the time when the TwinCAT build was created.

For TwinCAT 2.11/TwinCAT 3 and higher, the ESI directory can be updated from the System Manager, if the programming PC is connected to the Internet; by

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

The TwinCAT ESI Updater [▶ 74] is available for this purpose.

> **ⓘ ESI**
>
> The *.xml files are associated with *.xsd files, which describe the structure of the ESI XML files. To update the ESI device descriptions, both file types should therefore be updated.

Device differentiation

EtherCAT devices/slaves are distinguished by four properties, which determine the full device identifier. For example, the device identifier EL2521-0025-1018 consists of:

- family key "EL"
- name "2521"
- type "0025"
- and revision "1018"
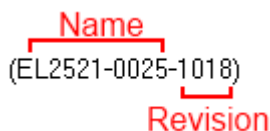
Fig. 70: Identifier structure

The order identifier consisting of name + type (here: EL2521-0025) describes the device function. The revision indicates the technical progress and is managed by Beckhoff. In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation. Each revision has its own ESI description. See further notes [▶ 8].

Online description

If the EtherCAT configuration is created online through scanning of real devices (see section Online setup) and no ESI descriptions are available for a slave (specified by name and revision) that was found, the System Manager asks whether the description stored in the device should be used. In any case, the System Manager needs this information for setting up the cyclic and acyclic communication with the slave correctly.
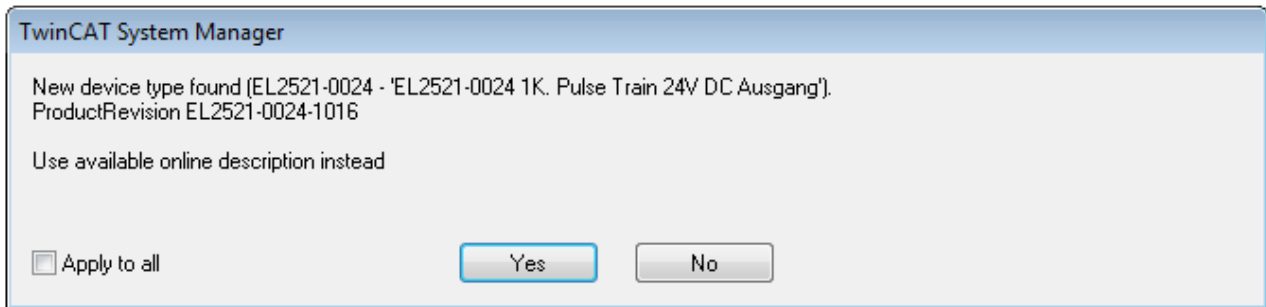

TwinCAT System Manager

New device type found (EL2521-0024 - 'EL2521-0024 1K. Pulse Train 24V DC Ausgang').
ProductRevision EL2521-0024-1016

Use available online description instead

☐ Apply to all   [Yes]   [No]

Fig. 71: OnlineDescription information window (TwinCAT 2)

In TwinCAT 3 a similar window appears, which also offers the Web update:

TwinCAT XAE

New device type found (EL2521-0024 - 'EL2521-0024 1K. Pulse Train 24V DC Ausgang').
ProductRevision EL2521-0024-1016

Use available online description instead (YES) or try to load appropriate descriptions from the web

☐ Apply to all   [Yes]   [No]   [Online ESI Update (Web access required)...]
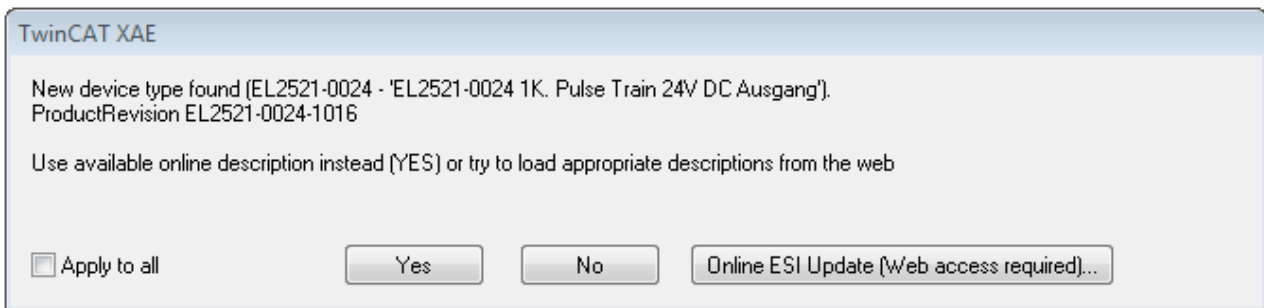
Fig. 72: Information window OnlineDescription (TwinCAT 3)

If possible, the *Yes* is to be rejected and the required ESI is to be requested from the device manufacturer. After installation of the XML/XSD file the configuration process should be repeated.

NOTICE
Changing the "usual" configuration through a scan
✓ If a scan discovers a device that is not yet known to TwinCAT, distinction has to be made between two cases. Taking the example here of the EL2521-0000 in the revision 1019
a) no ESI is present for the EL2521-0000 device at all, either for the revision 1019 or for an older revision. The ESI must then be requested from the manufacturer (in this case Beckhoff).
b) an ESI is present for the EL2521-0000 device, but only in an older revision, e.g. 1018 or 1017. In this case an in-house check should first be performed to determine whether the spare parts stock allows the integration of the increased revision into the configuration at all. A new/higher revision usually also brings along new features. If these are not to be used, work can continue without reservations with the previous revision 1018 in the configuration. This is also stated by the Beckhoff compatibility rule.

Refer in particular to the chapter "General notes on the use of Beckhoff EtherCAT IO components" and for manual configuration to the chapter "Offline configuration creation [▶ 75]".

If the OnlineDescription is used regardless, the System Manager reads a copy of the device description from the EEPROM in the EtherCAT slave. In complex slaves the size of the EEPROM may not be sufficient for the complete ESI, in which case the ESI would be *incomplete* in the configurator. Therefore it's recommended using an offline ESI file with priority in such a case.

The System Manager creates for online recorded device descriptions a new file "OnlineDescription0000...xml" in its ESI directory, which contains all ESI descriptions that were read online.

BECKHOFF

`OnlineDescriptionCache00000002.xml`

Fig. 73: File OnlineDescription.xml created by the System Manager

Is a slave desired to be added manually to the configuration at a later stage, online created slaves are indicated by a prepended symbol ">" in the selection list (see Figure *Indication of an online recorded ESI of EL2521 as an example*).
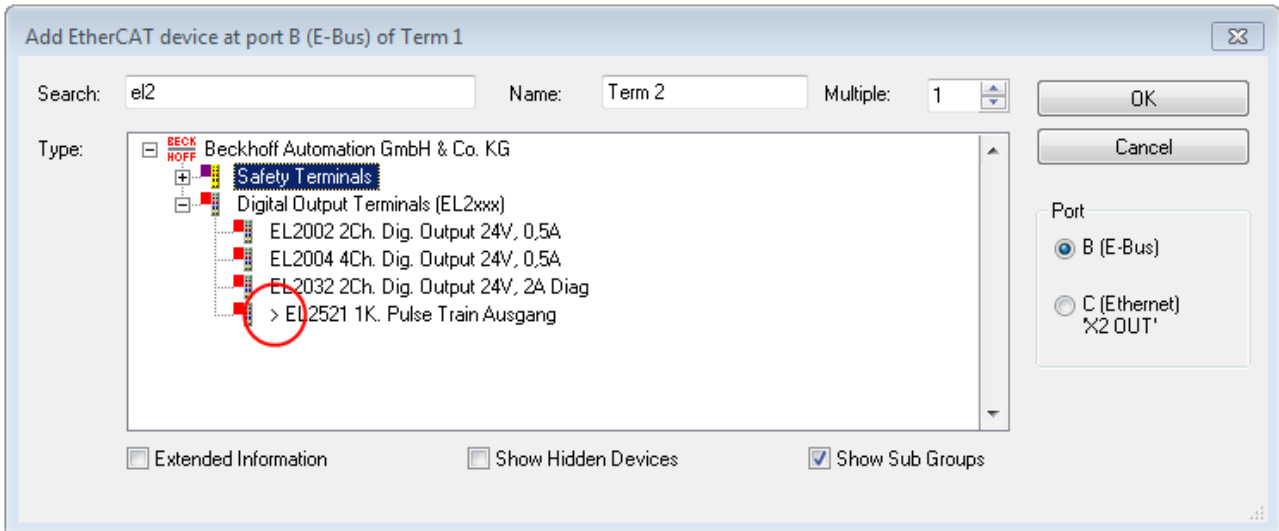
Fig. 74: Indication of an online recorded ESI of EL2521 as an example

If such ESI files are used and the manufacturer's files become available later, the file OnlineDescription.xml should be deleted as follows:

- close all System Manager windows
- restart TwinCAT in Config mode
- delete "OnlineDescription0000...xml"
- restart TwinCAT System Manager

This file should not be visible after this procedure, if necessary press <F5> to update

ⓘ **OnlineDescription for TwinCAT 3.x**

In addition to the file described above "OnlineDescription0000...xml", a so called EtherCAT cache with new discovered devices is created by TwinCAT 3.x, e.g. under Windows 7:

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

(Please note the language settings of the OS!)
You have to delete this file, too.

Faulty ESI file

If an ESI file is faulty and the System Manager is unable to read it, the System Manager brings up an information window.
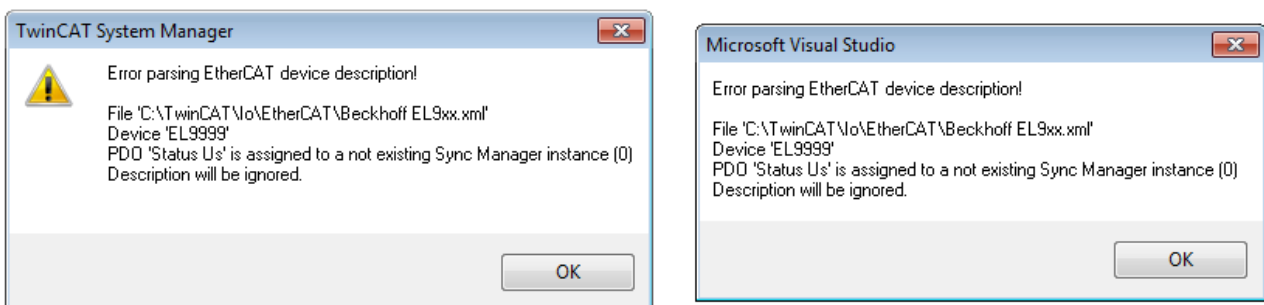
Fig. 75: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

Reasons may include:

- Structure of the *.xml does not correspond to the associated *.xsd file → check your schematics
- Contents cannot be translated into a device description → contact the file manufacturer

5.2.3 TwinCAT ESI Updater

For TwinCAT 2.11 and higher, the System Manager can search for current Beckhoff ESI files automatically, if an online connection is available:

Fig. 76: Using the ESI Updater (>= TwinCAT 2.11)

The call up takes place under:
"Options" → "Update EtherCAT Device Descriptions"

Selection under TwinCAT 3:
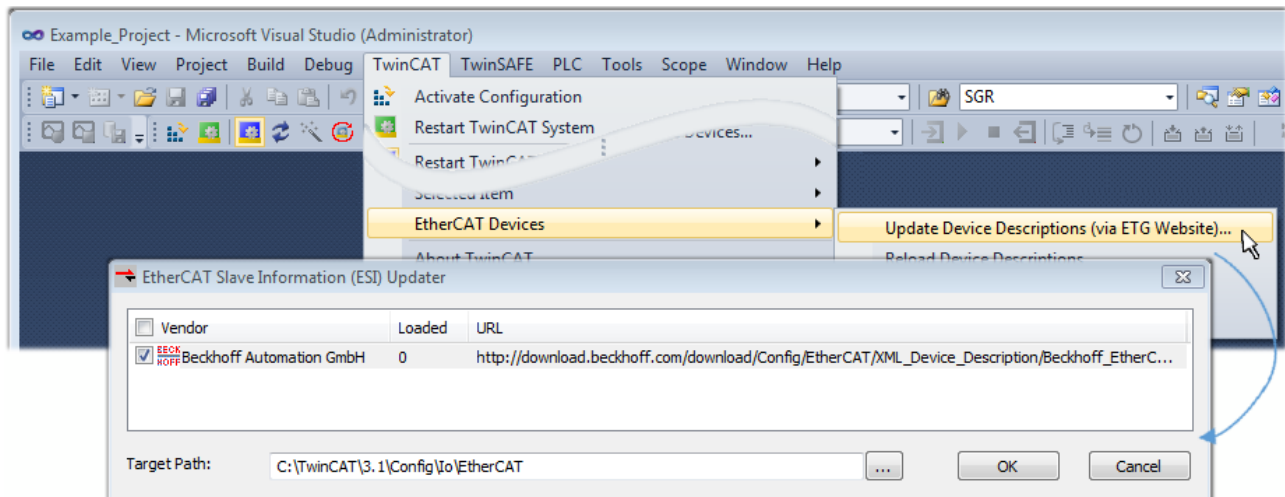
Fig. 77: Using the ESI Updater (TwinCAT 3)

The ESI Updater (TwinCAT 3) is a convenient option for automatic downloading of ESI data provided by EtherCAT manufacturers via the Internet into the TwinCAT direct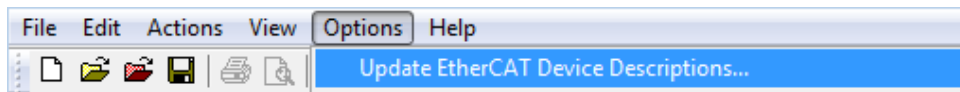ory (ESI = EtherCAT slave information). TwinCAT accesses the central ESI ULR directory list stored at ETG; the entries can then be viewed in the Updater dialog, although they cannot be changed there.

The call up takes place under:
"TwinCAT" → "EtherCAT Devices" → "Update Device Description (via ETG Website)…".

5.2.4 Distinction between Online and Offline

The distinction between online and offline refers to the presence of the actual I/O environment (drives, terminals, EJ-modules). If the configuration is to be prepared in advance of the system configuration as a programming system, e.g. on a laptop, this is only possible in "Offline configuration" mode. In this case all components have to be entered manually in the configuration, e.g. based on the electrical design.

If the designed control system is already connected to the EtherCAT system and all components are energised and the infrastructure is ready for operation, the TwinCAT configuration can simply be generated through "scanning" from the runtime system. This is referred to as online configuration.

In any case, during each startup the EtherCAT master checks whether the slaves it finds match the configuration. This test can be parameterised in the extended slave settings. Refer to note "Installation of the latest ESI-XML device description" [▶ 70].

For preparation of a configuration:

- the real EtherCAT hardware (devices, couplers, drives) must be present and installed
- the devices/modules must be connected via EtherCAT cables or in the terminal/ module strand in the same way as they are intended to be used later
- the devices/modules be connected to the power supply and ready for communication

- TwinCAT must be in CONFIG mode on the target system.

The online scan process consists of:

- detecting the EtherCAT device [▶ 80] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 81]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 84]

The scan with existing configuration [▶ 85] can also be carried out for comparison.

5.2.5   OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.
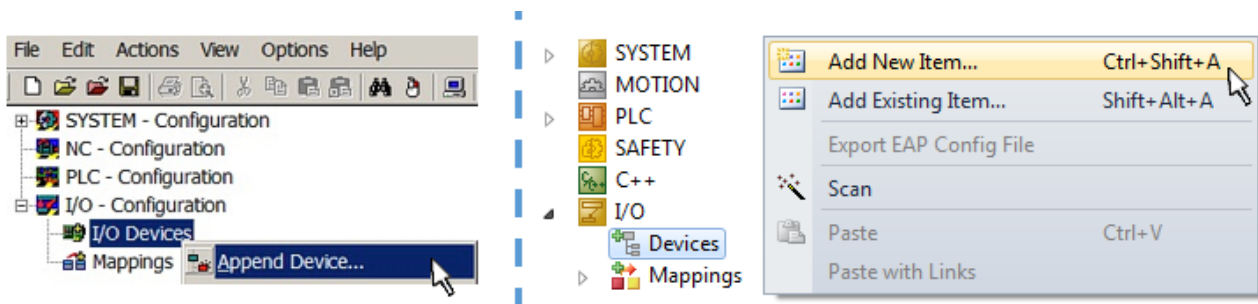
Fig. 78: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

Select type "EtherCAT" for an EtherCAT I/O application with EtherCAT slaves. For the present publisher/subscriber service in combination with an EL6601/EL6614 terminal select "EtherCAT Automation Protocol via EL6601".
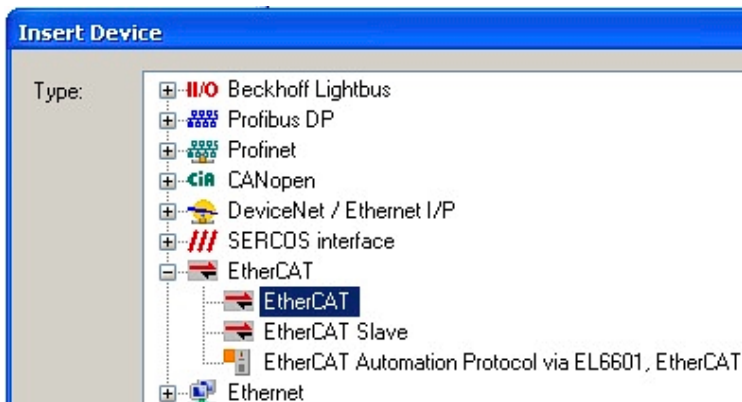
Fig. 79: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

Then assign a real Ethernet port to this virtual device in the runtime system.
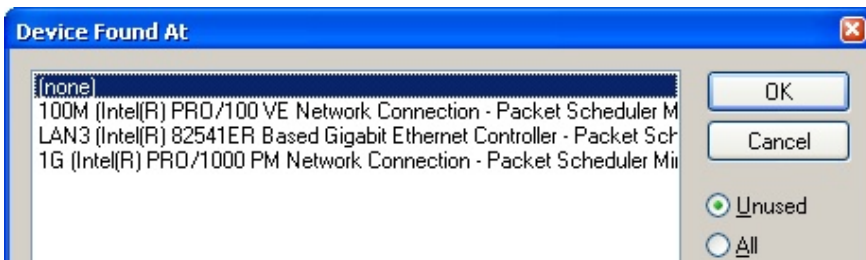
Fig. 80: Selecting the Ethernet port

BECKHOFF

This query may appear automatically when the EtherCAT device is created, or the assignment can be set/modified later in the properties dialog; see Fig. "EtherCAT device properties (TwinCAT 2)".
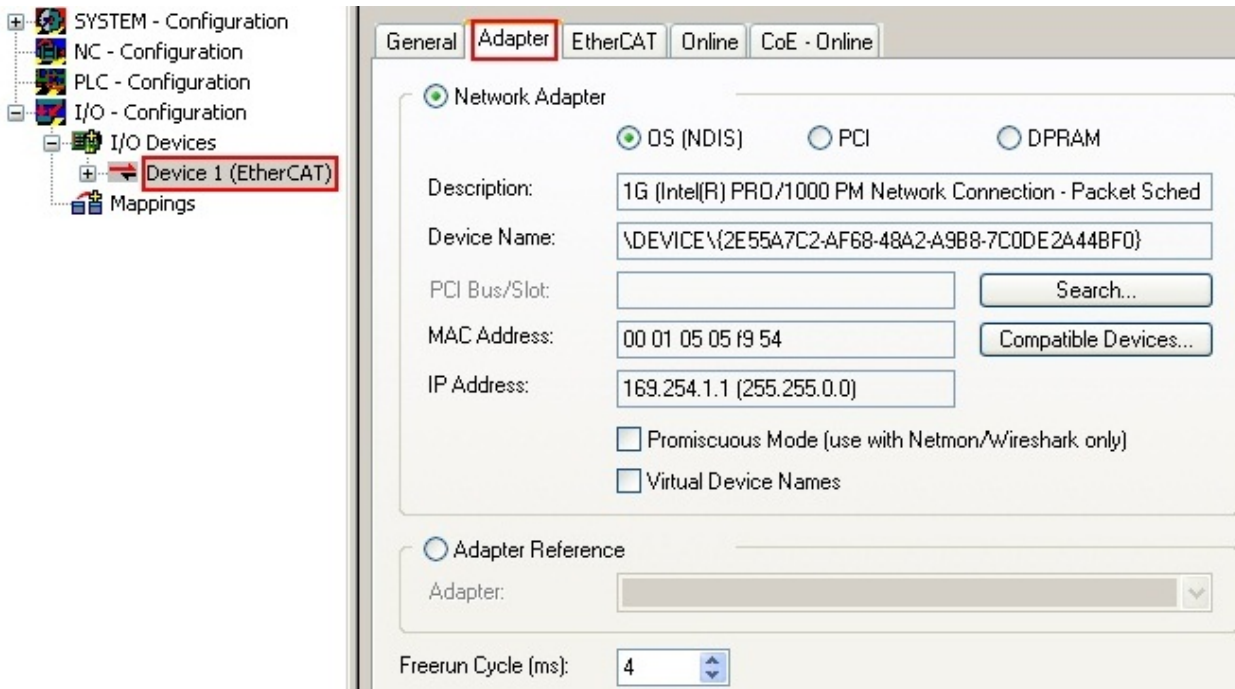
Fig. 81: EtherCAT device properties (TwinCAT 2)

TwinCAT 3: the properties of the EtherCAT device can be opened by double click on "Device .. (EtherCAT)" within the Solution Explorer under "I/O":
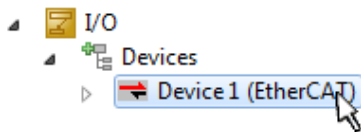
● **Selecting the Ethernet port**

ℹ Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective installation page [▶ 64].

Defining EtherCAT slaves

Further devices can be appended by right-clicking on a device in the configuration tree.
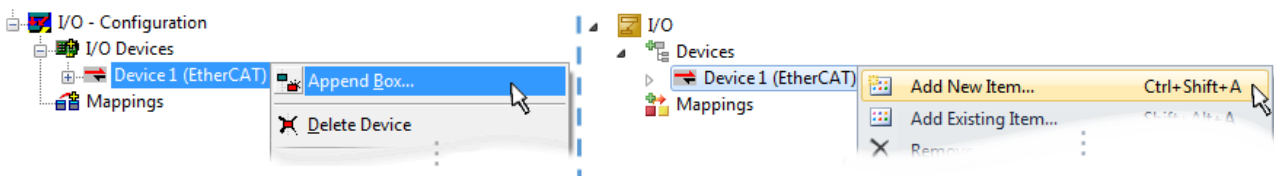
Fig. 82: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

The dialog for selecting a new device opens. Only devices for which ESI files are available are displayed.

Only devices are offered for selection that can be appended to the previously selected device. Therefore, the physical layer available for this port is also displayed (Fig. "Selection dialog for new EtherCAT device", A). In the case of cable-based Fast-Ethernet physical layer with PHY transfer, then also only cable-based devices are available, as shown in Fig. "Selection dialog for new EtherCAT device". If the preceding device has several free ports (e.g. EK1122 or EK1100), the required port can be selected on the right-hand side (A).

Overview of physical layer

• "Ethernet": cable-based 100BASE-TX: couplers, box modules, devices with RJ45/M8/M12 connector

- "E-Bus": LVDS "terminal bus", EtherCAT plug-in modules (EJ), EtherCAT terminals (EL/ES), various modular modules

The search field facilitates finding specific devices (since TwinCAT 2.11 or TwinCAT 3).
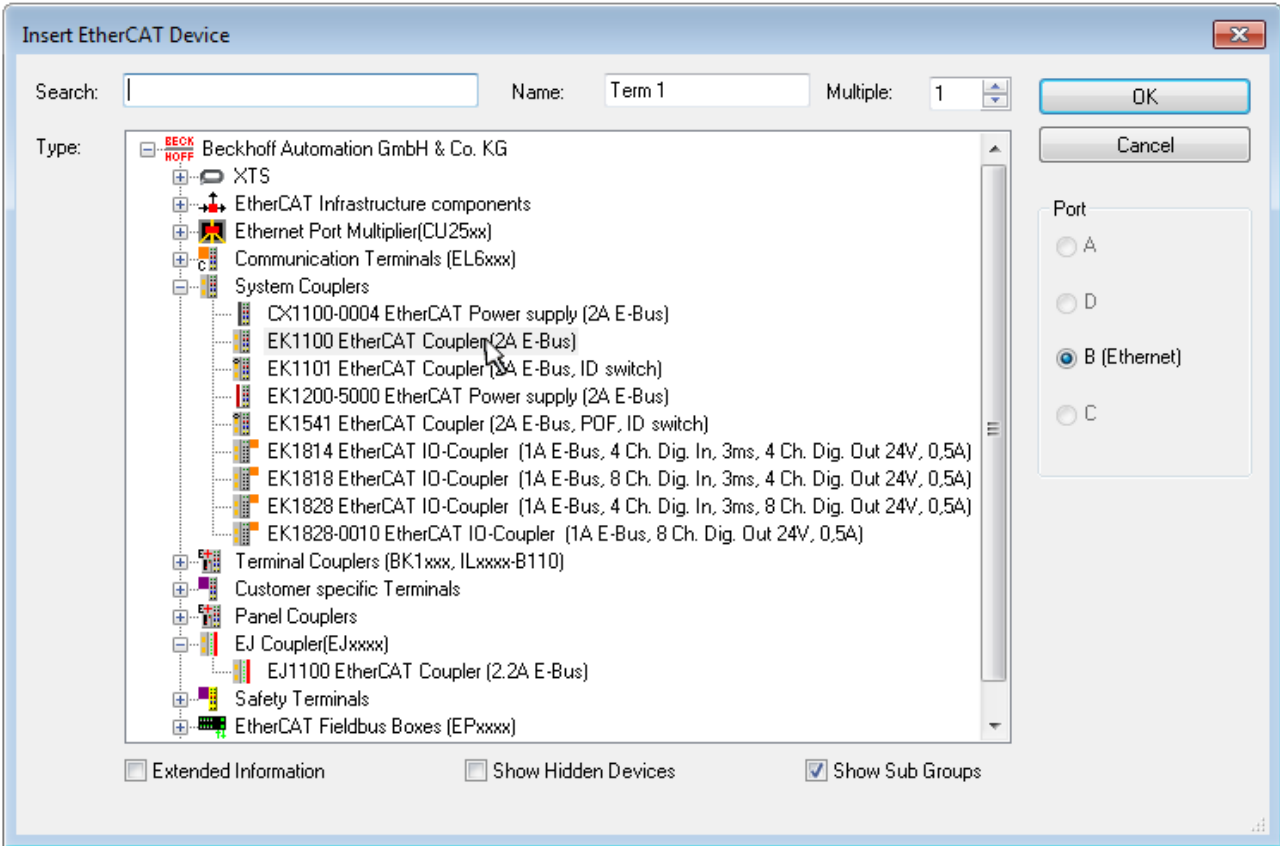
Fig. 83: Selection dialog for new EtherCAT device

By default, only the name/device type is used as selection criterion. For selecting a specific revision of the device, the revision can be displayed as "Extended Information".
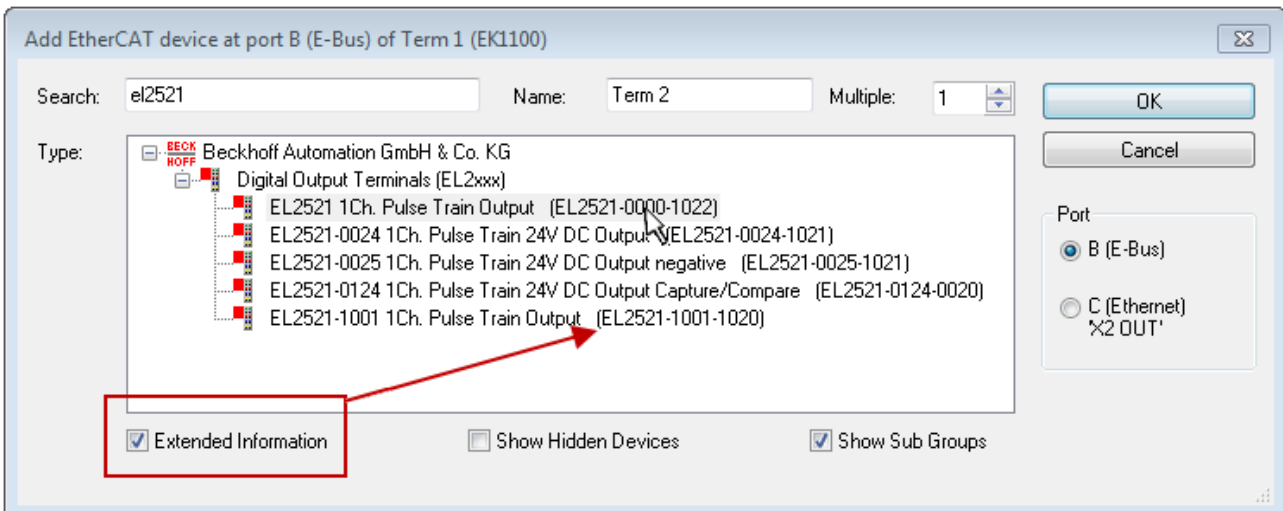
Fig. 84: Display of device revision

In many cases several device revisions were created for historic or functional reasons, e.g. through technological advancement. For simplification purposes (see Fig. "Selection dialog for new EtherCAT device") only the last (i.e. highest) revision and therefore the latest state of production is displayed in the selection dialog for Beckhoff devices. To show all device revisions available in the system as ESI descriptions tick the "Show Hidden Devices" check box, see Fig. "Display of previous revisions".

Fig. 85: Display of previous revisions

ℹ **Device selection based on revision, compatibility**

The ESI description also defines the process image, the communication type between master and slave/device and the device functions, if applicable. The physical device (firmware, if available) has to support the communication queries/settings of the master. This is backward compatible, i.e. newer devices (higher revision) should be supported if the EtherCAT master addresses them as an older revision. The following compatibility rule of thumb is to be assumed for Beckhoff EtherCAT Terminals/ Boxes/ EJ-modules:

device revision in the system >= device revision in the configuration

This also enables subsequent replacement of devices without changing the configuration (different specifications are possible for drives).

Example

If an EL2521-0025-**1018** is specified in the configuration, an EL2521-0025-**1018** or higher (-**1019**, -**1020**) can be used in practice.

Fig. 86: Name/revision of the terminal

If current ESI descriptions are available in the TwinCAT system, the last revision offered in the selection dialog matches the Beckhoff state of production. It is recommended to use the last device revision when creating a new configuration, if current Beckhoff devices are used in the real application. Older revisions should only be used if older devices from stock are to be used in the application.

In this case the process image of the device is shown in the configuration tree and can be parameterized as follows: linking with the task, CoE/DC settings, plug-in definition, startup settings, ...

Fig. 87: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)

BECKHOFF

5.2.6 ONLINE configuration creation

Detecting/scanning of the EtherCAT device

The online device search can be used if the TwinCAT system is in CONFIG mode. This can be indicated by a symbol right below in the information bar:

- on TwinCAT 2 by a blue display "Config Mode" within the System Manager window: Config Mode .

- on TwinCAT 3 within the user interface of the development environment by a symbol 🖼 .

TwinCAT can be set into this mode:

- TwinCAT 2: by selection of 🌐 in the Menubar or by "Actions" → "Set/Reset TwinCAT to Config Mode…"

- TwinCAT 3: by selection of 🖼 in the Menubar or by "TwinCAT" → "Restart TwinCAT (Config Mode)"

ℹ **Online scanning in Config mode**

The online search is not available in RUN mode (production operation). Note the differentiation between TwinCAT programming system and TwinCAT target system.

The TwinCAT 2 icon (🌐) or TwinCAT 3 icon (🖼) within the Windows-Taskbar always shows the TwinCAT mode of the local IPC. Compared to that, the System Manager window of TwinCAT 2 or the user interface of TwinCAT 3 indicates the state of the target system.
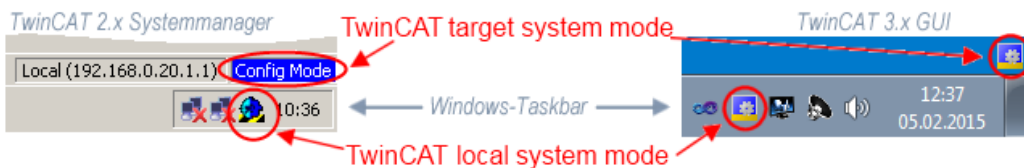
Fig. 88: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

Right-clicking on "I/O Devices" in the configuration tree opens the search dialog.
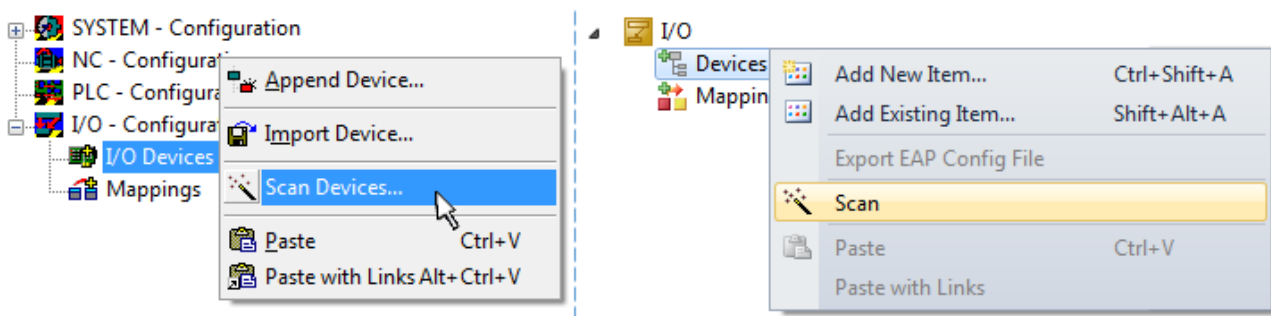
Fig. 89: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

This scan mode attempts to find not only EtherCAT devices (or Ethernet ports that are usable as such), but also NOVRAM, fieldbus cards, SMB etc. However, not all devices can be found automatically.
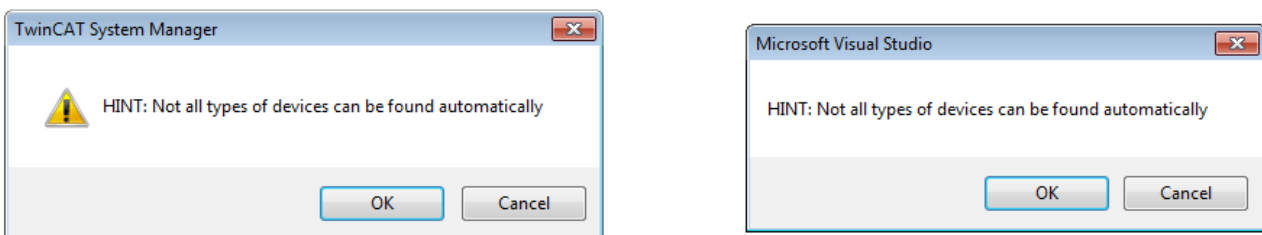
Fig. 90: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

Ethernet ports with installed TwinCAT real-time driver are shown as "RT Ethernet" devices. An EtherCAT frame is sent to these ports for testing purposes. If the scan agent detects from the response that an EtherCAT slave is connected, the port is immediately shown as an "EtherCAT Device" .
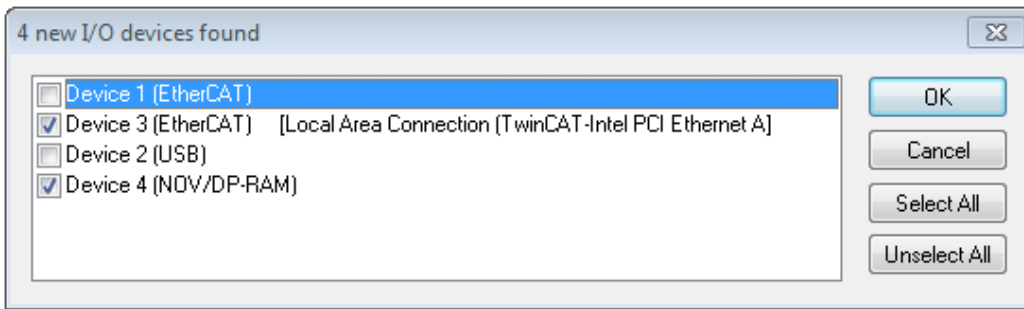
Fig. 91: Detected Ethernet devices

Via respective checkboxes devices can be selected (as illustrated in Fig. "Detected Ethernet devices" e.g. Device 3 and Device 4 were chosen). After confirmation with "OK" a device scan is suggested for all selected devices, see Fig.: "Scan query after automatic creation of an EtherCAT device".

ⓘ **Selecting the Ethernet port**

Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective installation page [▶ 64].

Detecting/Scanning the EtherCAT devices

ⓘ **Online scan functionality**

During a scan the master queries the identity information of the EtherCAT slaves from the slave EEPROM. The name and revision are used for determining the type. The respective devices are located in the stored ESI data and integrated in the configuration tree in the default state defined there.
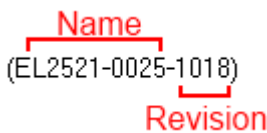
Fig. 92: Example default state

NOTICE
Slave scanning in practice in series machine production
The scanning function should be used with care. It is a practical and fast tool for creating an initial configuration as a basis for commissioning. In series machine production or reproduction of the plant, however, the function should no longer be used for the creation of the configuration, but if necessary for comparison [▶ 85] with the defined initial configuration.Background: since Beckhoff occasionally increases the revision version of the delivered products for product maintenance reasons, a configuration can be created by such a scan which (with an identical machine construction) is identical according to the device list; however, the respective device revision may differ from the initial configuration.

Example:

Company A builds the prototype of a machine B, which is to be produced in series later on. To do this the prototype is built, a scan of the IO devices is performed in TwinCAT and the initial configuration "B.tsm" is created. The EL2521-0025 EtherCAT terminal with the revision 1018 is located somewhere. It is thus built into the TwinCAT configuration in this way:
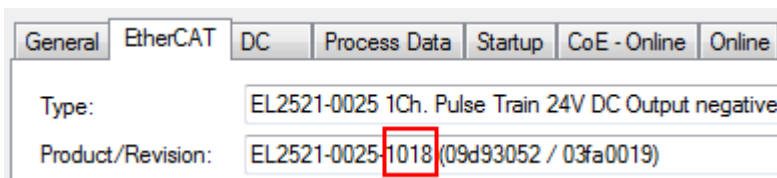
Fig. 93: Installing EthetCAT terminal with revision -1018

Likewise, during the prototype test phase, the functions and properties of this terminal are tested by the programmers/commissioning engineers and used if necessary, i.e. addressed from the PLC "B.pro" or the NC. (the same applies correspondingly to the TwinCAT 3 solution files).

The prototype development is now completed and series production of machine B starts, for which Beckhoff continues to supply the EL2521-0025-0018. If the commissioning engineers of the series machine production department always carry out a scan, a B configuration with the identical contents results again for each machine. Likewise, A might create spare parts stores worldwide for the coming series-produced machines with EL2521-0025-1018 terminals.

After some time Beckhoff extends the EL2521-0025 by a new feature C. Therefore the FW is changed, outwardly recognizable by a higher FW version and **a new revision -1019**. Nevertheless the new device naturally supports functions and interfaces of the predecessor version(s); an adaptation of "B.tsm" or even "B.pro" is therefore unnecessary. The series-produced machines can continue to be built with "B.tsm" and "B.pro"; it makes sense to perform a <u>comparative scan [▶ 85]</u> against the initial configuration "B.tsm" in order to check the built machine.

However, if the series machine production department now doesn't use "B.tsm", but instead carries out a scan to create the productive configuration, the revision **-1019** is automatically detected and built into the configuration:
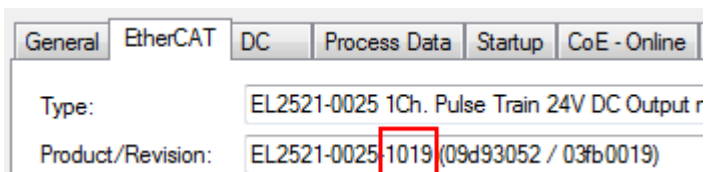
Fig. 94: Detection of EtherCAT terminal with revision -1019

This is usually not noticed by the commissioning engineers. TwinCAT cannot signal anything either, since a new configuration is essentially created. According to the compatibility rule, however, this means that no EL2521-0025-**1018** should be built into this machine as a spare part (even if this nevertheless works in the vast majority of cases).

In addition, it could be the case that, due to the development accompanying production in company A, the new feature C of the EL2521-0025-1019 (for example, an improved analog filter or an additional process data for the diagnosis) is discovered and used without in-house consultation. The previous stock of spare part devices are then no longer to be used for the new configuration "B2.tsm" created in this way. Þ if series machine production is established, the scan should only be performed for informative purposes for comparison with a defined initial configuration. Changes are to be made with care!

If an EtherCAT device was created in the configuration (manually or through a scan), the I/O field can be scanned for devices/slaves.

Fig. 95: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)
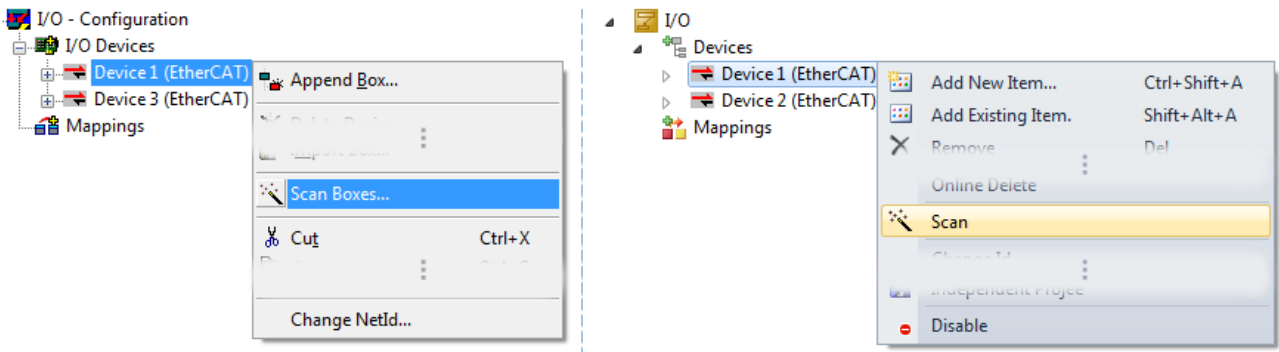
Fig. 96: Manual scanning for devices on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

In the System Manager (TwinCAT 2) or the User Interface (TwinCAT 3) the scan process can be monitored via the progress bar at the bottom in the status bar.

Fig. 97: *Scan progressexemplary by TwinCAT 2*

The configuration is established and can then be switched to online state (OPERATIONAL).

Fig. 98: Config/FreeRun query (left: TwinCAT 2; right: TwinCAT 3)

In Config/FreeRun mode the System Manager display alternates between blue and red, and the EtherCAT device continues to operate with the idling cycle time of 4 ms (default setting), even without active task (NC, PLC).

Fig. 99: Displaying of "Free Run" and "Config Mode" toggling right below in the status bar

Fig. 100: TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3)

The EtherCAT system should then be in a functional cyclic state, as shown in Fig. *Online display example*.
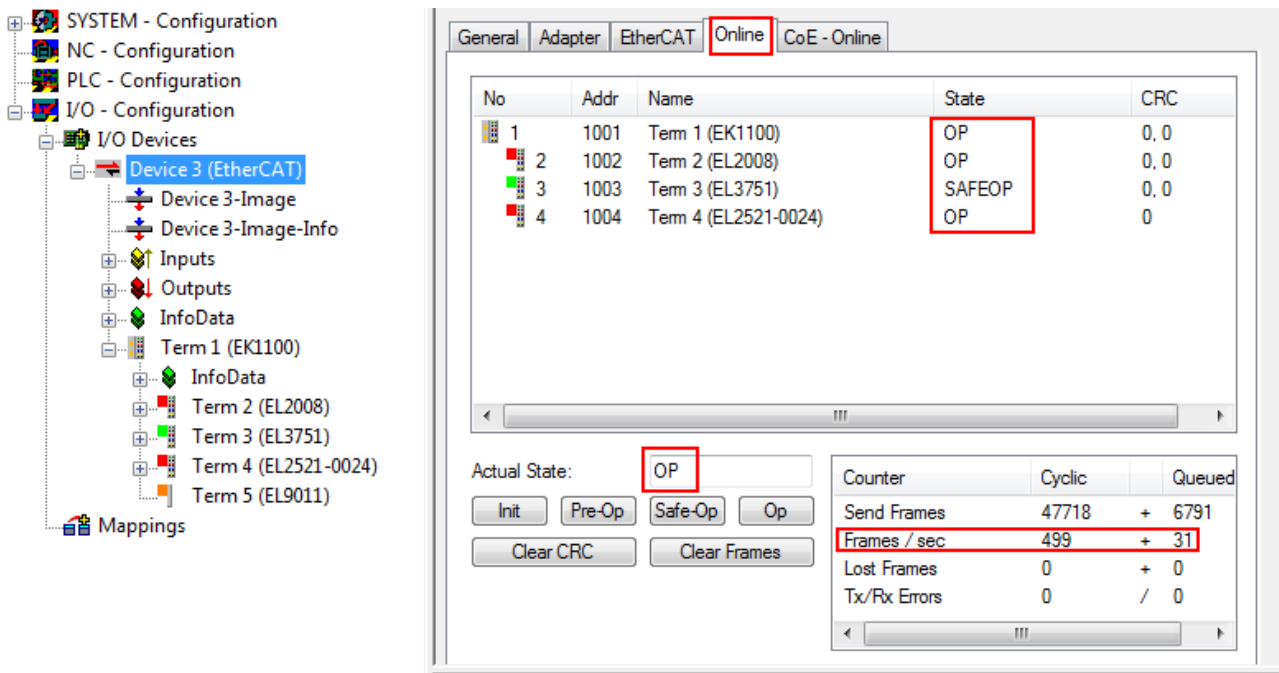
Fig. 101: Online display example

Please note:

- all slaves should be in OP state
- the EtherCAT master should be in "Actual State" OP
- "frames/sec" should match the cycle time taking into account the sent number of frames
- no excessive "LostFrames" or CRC errors should occur

The configuration is now complete. It can be modified as described under

Troubleshooting

Various effects may occur during scanning.

- An **unknown device** is detected, i.e. an EtherCAT slave for which no ESI XML description is available. In this case the System Manager offers to read any ESI that may be stored in the device. This case is described in the chapter "Notes regarding ESI device description".
- **Device are not detected properly**
  Possible reasons include:
  ◦ faulty data links, resulting in data loss during the scan
  ◦ slave has invalid device description

    The connections and devices should be checked in a targeted manner, e.g. via the emergency scan.
    Then re-run the scan.
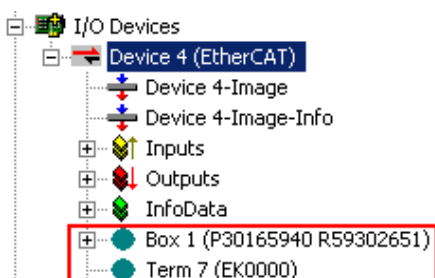
Fig. 102: Faulty identification

In the System Manager such devices may be set up as EK0000 or unknown devices. Operation is not possible or meaningful.

Scan over existing Configuration

NOTICE
Change of the configuration after comparison
With this scan (TwinCAT 2.11 or 3.1) only the device properties vendor (manufacturer), device name and revision are compared at present! A "ChangeTo" or "Copy" should only be carried out with care, taking into consideration the Beckhoff IO compatibility rule (see above). The device configuration is then replaced by the revision found; this can affect the supported process data and functions.

If a scan is initiated for an existing configuration, the actual I/O environment may match the configuration exactly or it may differ. This enables the configuration to be compared.

Fig. 103: Identical configuration (left: TwinCAT 2; right: TwinCAT 3)

If differences are detected, they are shown in the correction dialog, so that the user can modify the configuration as required.
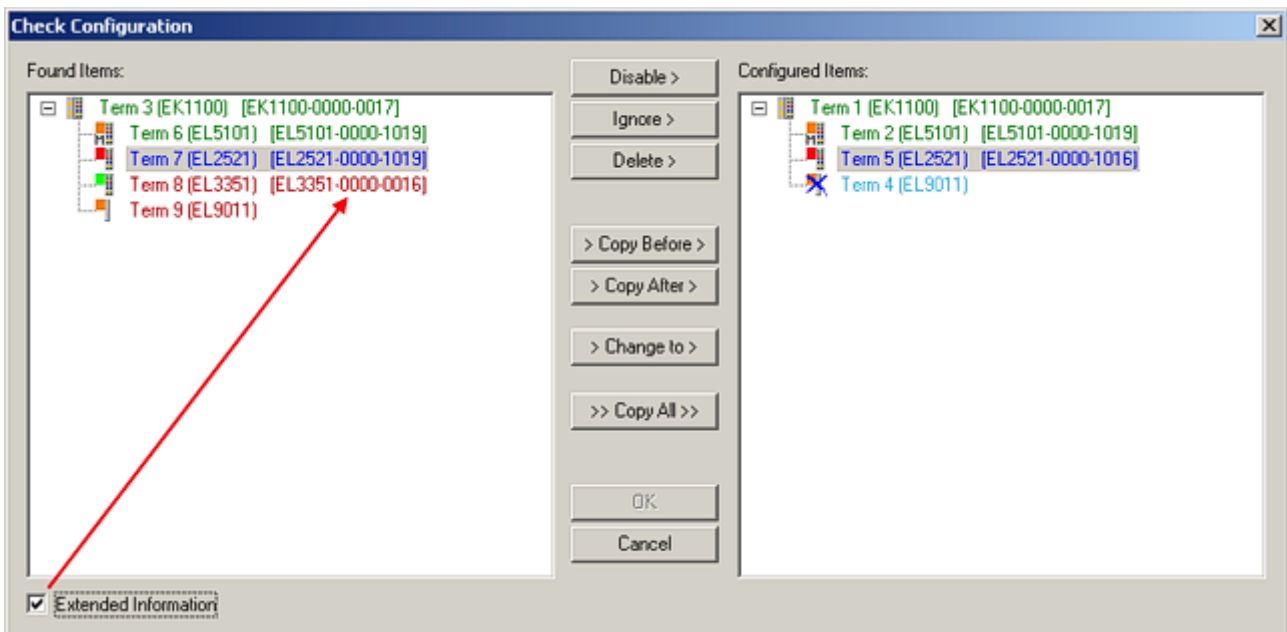
Fig. 104: Correction dialog

It is advisable to tick the "Extended Information" check box to reveal differences in the revision.

Color	Explanation
green	This EtherCAT slave matches the entry on the other side. Both type and revision match.
blue	This EtherCAT slave is present on the other side, but in a different revision. This other revision can have other default values for the process data as well as other/additional functions. If the found revision is higher than the configured revision, the slave may be used provided compatibility issues are taken into account. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.
light blue	This EtherCAT slave is ignored ("Ignore" button)
red	• This EtherCAT slave is not present on the other side. • It is present, but in a different revision, which also differs in its properties from the one specified. The compatibility principle then also applies here: if the found revision is higher than the configured revision, use is possible provided compatibility issues are taken into account, since the successor devices should support the functions of the predecessor devices. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.

Device selection based on revision, compatibility

The ESI description also defines the process image, the communication type between master and slave/device and the device functions, if applicable. The physical device (firmware, if available) has to support the communication queries/settings of the master. This is backward compatible, i.e. newer devices (higher revision) should be supported if the EtherCAT master addresses them as an older revision. The following compatibility rule of thumb is to be assumed for Beckhoff EtherCAT Terminals/ Boxes/ EJ-modules:

device revision in the system >= device revision in the configuration

This also enables subsequent replacement of devices without changing the configuration (different specifications are possible for drives).

Example

If an EL2521-0025-**1018** is specified in the configuration, an EL2521-0025-**1018** or higher (-**1019**, -**1020**) can be used in practice.

Fig. 105: Name/revision of the terminal

If current ESI descriptions are available in the TwinCAT system, the last revision offered in the selection dialog matches the Beckhoff state of production. It is recommended to use the last device revision when creating a new configuration, if current Beckhoff devices are used in the real application. Older revisions should only be used if older devices from stock are to be used in the application.

In this case the process image of the device is shown in the configuration tree and can be parameterized as follows: linking with the task, CoE/DC settings, plug-in definition, startup settings, ...
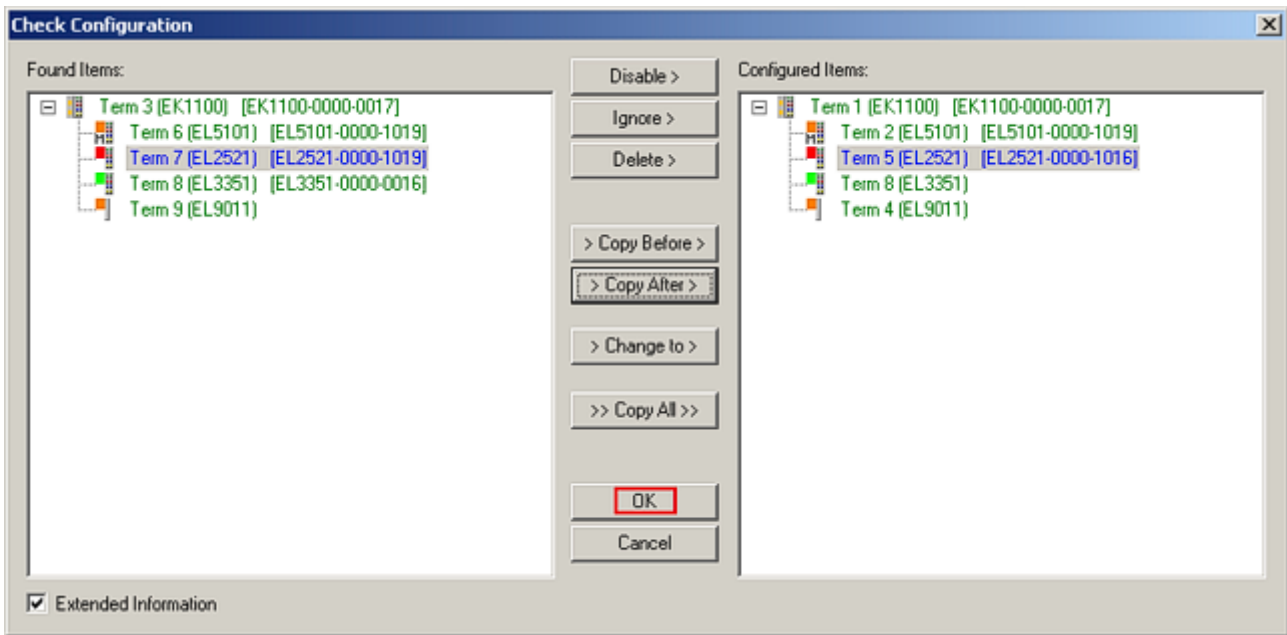
Fig. 106: Correction dialog with modifications

Once all modifications have been saved or accepted, click "OK" to transfer them to the real *.tsm configuration.

Change to Compatible Type

TwinCAT offers a function *Change to Compatible Type…* for the exchange of a device whilst retaining the links in the task.
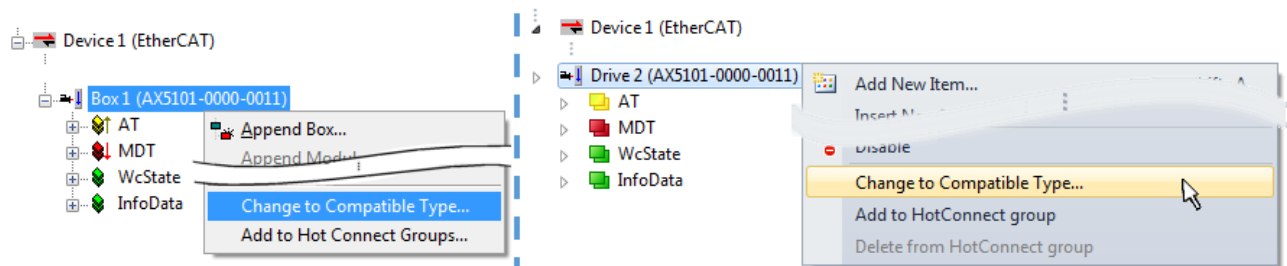
Fig. 107: Dialog "Change to Compatible Type…" (left: TwinCAT 2; right: TwinCAT 3)

The following elements in the ESI of an EtherCAT device are compared by TwinCAT and assumed to be the same in order to decide whether a device is indicated as "compatible":

- Physics (e.g. RJ45, Ebus...)

- FMMU (additional ones are allowed)

- SyncManager (SM, additional ones are allowed)

- EoE (attributes MAC, IP)

- CoE (attributes SdoInfo, PdoAssign, PdoConfig, PdoUpload, CompleteAccess)

- FoE

- PDO (process data: Sequence, SyncUnit SU, SyncManager SM, EntryCount, Ent-ry.Datatype)

This function is preferably to be used on AX5000 devices.

Change to Alternative Type

The TwinCAT System Manager offers a function for the exchange of a device: Change to Alternative Type

BECKHOFF
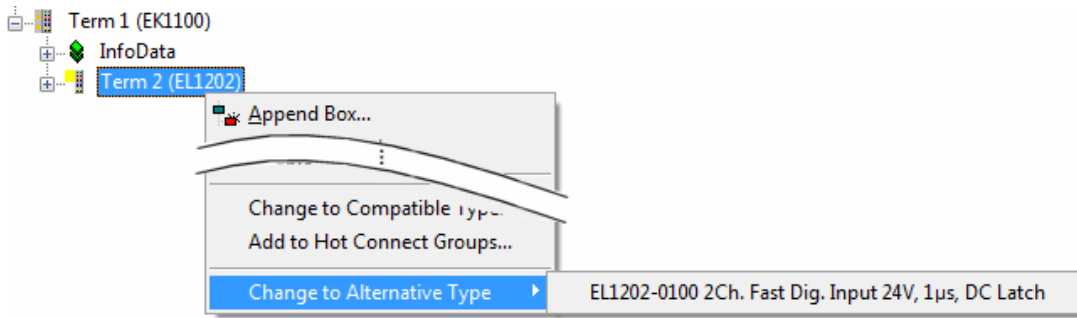
Fig. 108: TwinCAT 2 Dialog Change to Alternative Type

If called, the System Manager searches in the procured device ESI (in this example: EL1202-0000) for details of compatible devices contained there. The configuration is changed and the ESI-EEPROM is overwritten at the same time – therefore this process is possible only in the online state (ConfigMode).

5.2.7   EtherCAT subscriber configuration

In the left-hand window of the TwinCAT 2 System Manager or the Solution Explorer of the TwinCAT 3 Development Environment respectively, click on the element of the terminal within the tree you wish to configure (in the example: EL3751 Terminal 3).
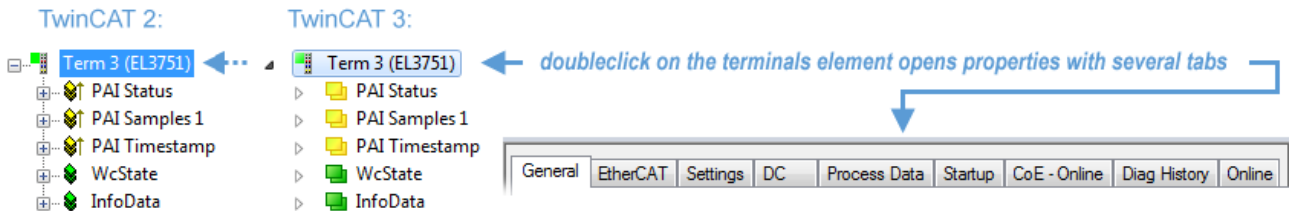
Fig. 109: Branch element as terminal EL3751

In the right-hand window of the TwinCAT System Manager (TwinCAT 2) or the Development Environment (TwinCAT 3), various tabs are now available for configuring the terminal. And yet the dimension of complexity of a subscriber determines which tabs are provided. Thus as illustrated in the example above the terminal EL3751 provides many setup options and also a respective number of tabs are available. On the contrary by the terminal EL1004 for example the tabs "General", "EtherCAT", "Process Data" and "Online" are available only. Several terminals, as for instance the EL6695 provide special functions by a tab with its own terminal name, so "EL6695" in this case. A specific tab "Settings" by terminals with a wide range of setup options will be provided also (e.g. EL3751).

"General" tab
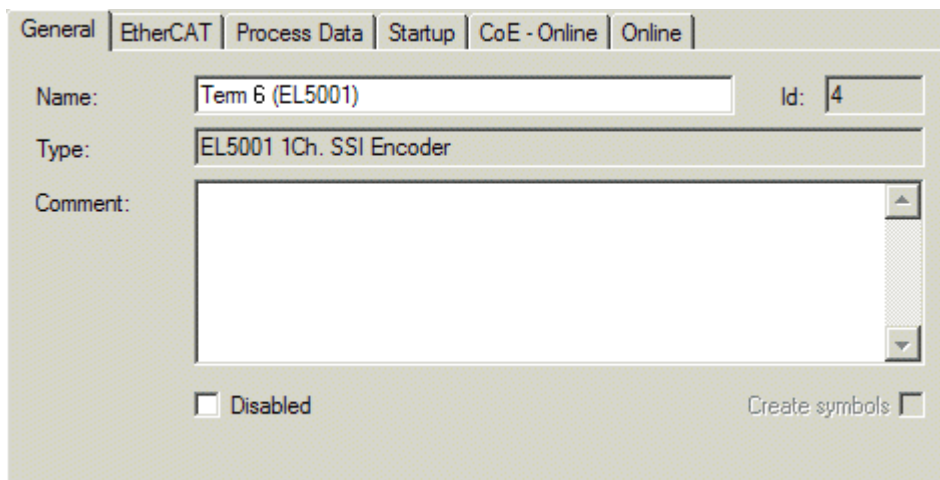
Fig. 110: "General" tab

Name	Name of the EtherCAT device
Id	Number of the EtherCAT device
Type	EtherCAT device type
Comment	Here you can add a comment (e.g. regarding the system).
Disabled	Here you can deactivate the EtherCAT device.
Create symbols	Access to this EtherCAT slave via ADS is only available if this control box is activated.

"EtherCAT" tab

Fig. 111: "EtherCAT" tab

Type	EtherCAT device type
Product/Revision	Product and revision number of the EtherCAT device
Auto Inc Addr.	Auto increment address of the EtherCAT device. The auto increment address can be used for addressing each EtherCAT device in the communication ring through its physical position. Auto increment addressing is used during the start-up phase when the EtherCAT master allocates addresses to the EtherCAT devices. With auto increment addressing the first EtherCAT slave in the ring has the address 0000_{hex}. For each further slave the address is decremented by 1 ($FFFF_{hex}$, $FFFE_{hex}$ etc.).
EtherCAT Addr.	Fixed address of an EtherCAT slave. This address is allocated by the EtherCAT master during the start-up phase. Tick the control box to the left of the input field in order to modify the default value.
Previous Port	Name and port of the EtherCAT device to which this device is connected. If it is possible to connect this device with another one without changing the order of the EtherCAT devices in the communication ring, then this combination field is activated and the EtherCAT device to which this device is to be connected can be selected.
Advanced Settings	This button opens the dialogs for advanced settings.

The link at the bottom of the tab points to the product page for this EtherCAT device on the web.

"Process Data" tab

Indicates the configuration of the process data. The input and output data of the EtherCAT slave are represented as CANopen process data objects (**P**rocess **D**ata **O**bjects, PDOs). The user can select a PDO via PDO assignment and modify the content of the individual PDO via this dialog, if the EtherCAT slave supports this function.
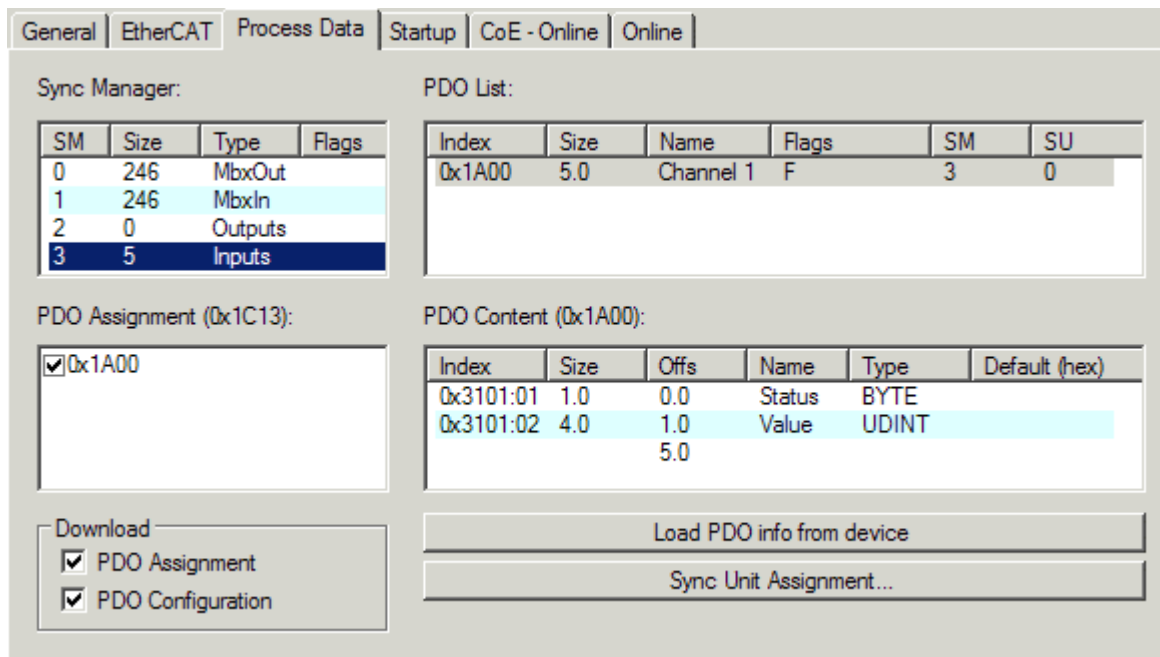
BECKHOFF

Fig. 112: "Process Data" tab

The process data (PDOs) transferred by an EtherCAT slave during each cycle are user data which the application expects to be updated cyclically or which are sent to the slave. To this end the EtherCAT master (Beckhoff TwinCAT) parameterizes each EtherCAT slave during the start-up phase to define which process data (size in bits/bytes, source location, transmission type) it wants to transfer to or from this slave. Incorrect configuration can prevent successful start-up of the slave.

For Beckhoff EtherCAT EL, ES, EM, EJ and EP slaves the following applies in general:

- The input/output process data supported by the device are defined by the manufacturer in the ESI/XML description. The TwinCAT EtherCAT Master uses the ESI description to configure the slave correctly.

- The process data can be modified in the System Manager. See the device documentation. Examples of modifications include: mask out a channel, displaying additional cyclic information, 16-bit display instead of 8-bit data size, etc.

- In so-called "intelligent" EtherCAT devices the process data information is also stored in the CoE directory. Any changes in the CoE directory that lead to different PDO settings prevent successful startup of the slave. It is not advisable to deviate from the designated process data, because the device firmware (if available) is adapted to these PDO combinations.

If the device documentation allows modification of process data, proceed as follows (see Figure *Configuring the process data*).

- A: select the device to configure
- B: in the "Process Data" tab select Input or Output under SyncManager (C)
- D: the PDOs can be selected or deselected
- H: the new process data are visible as linkable variables in the System Manager
  The new process data are active once the configuration has been activated and TwinCAT has been restarted (or the EtherCAT master has been restarted)
- E: if a slave supports this, Input and Output PDO can be modified simultaneously by selecting a so-called PDO record ("predefined PDO settings").
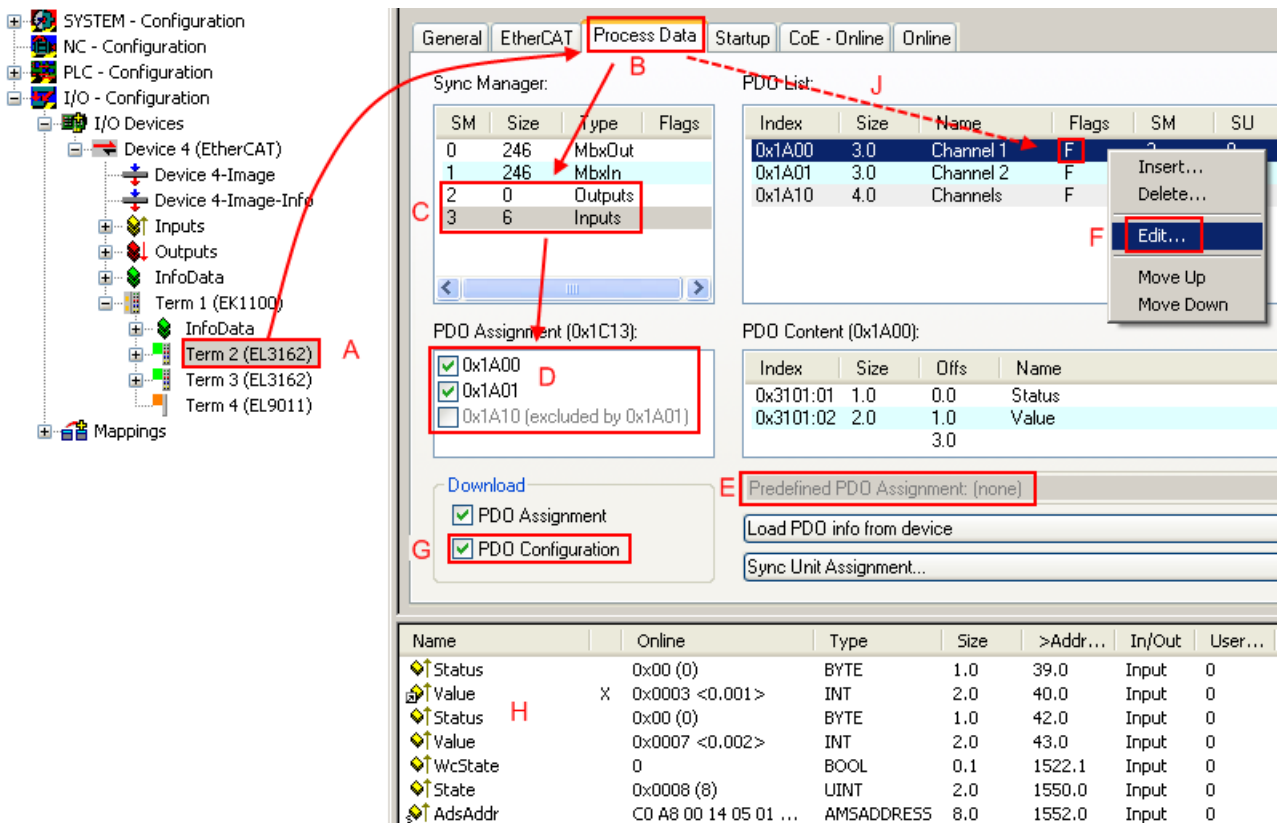
Fig. 113: Configuring the process data

ⓘ **Manual modification of the process data**

According to the ESI description, a PDO can be identified as "fixed" with the flag "F" in the PDO overview (Fig. *Configuring the process data*, J). The configuration of such PDOs cannot be changed, even if TwinCAT offers the associated dialog ("Edit"). In particular, CoE content cannot be displayed as cyclic process data. This generally also applies in cases where a device supports download of the PDO configuration, "G". In case of incorrect configuration the EtherCAT slave usually refuses to start and change to OP state. The System Manager displays an "invalid SM cfg" logger message: This error message ("invalid SM IN cfg" or "invalid SM OUT cfg") also indicates the reason for the failed start.

"Startup" tab

The *Startup* tab is displayed if the EtherCAT slave has a mailbox and supports the *CANopen over EtherCAT* (CoE) or *Servo drive over EtherCAT* protocol. This tab indicates which download requests are sent to the mailbox during startup. It is also possible to add new mailbox requests to the list display. The download requests are sent to the slave in the same order as they are shown in the list.

BECKHOFF
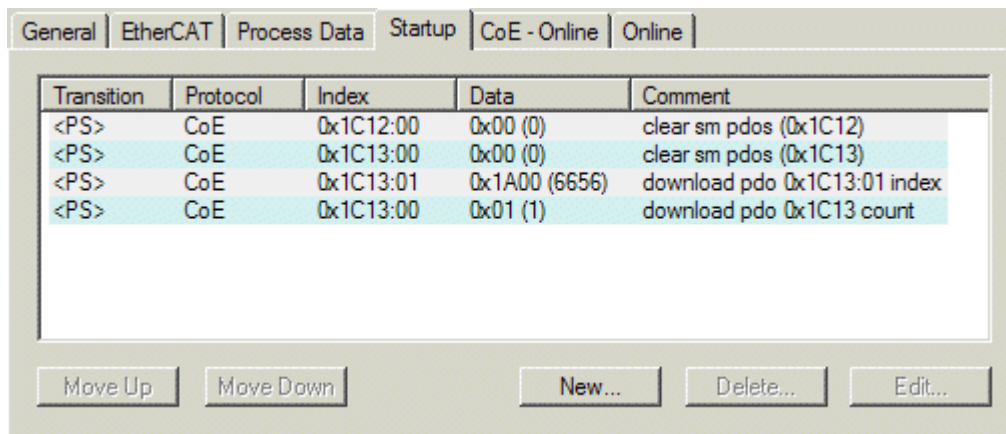
Fig. 114: "Startup" tab

Column	Description
Transition	Transition to which the request is sent. This can either be • the transition from pre-operational to safe-operational (PS), or • the transition from safe-operational to operational (SO). If the transition is enclosed in "<>" (e.g. <PS>), the mailbox request is fixed and cannot be modified or deleted by the user.
Protocol	Type of mailbox protocol
Index	Index of the object
Data	Date on which this object is to be downloaded.
Comment	Description of the request to be sent to the mailbox

Move Up	This button moves the selected request up by one position in the list.
Move Down	This button moves the selected request down by one position in the list.
New	This button adds a new mailbox download request to be sent during startup.
Delete	This button deletes the selected entry.
Edit	This button edits an existing request.

"CoE - Online" tab

The additional *CoE - Online* tab is displayed if the EtherCAT slave supports the *CANopen over EtherCAT* (CoE) protocol. This dialog lists the content of the object list of the slave (SDO upload) and enables the user to modify the content of an object from this list. Details for the objects of the individual EtherCAT devices can be found in the device-specific object descriptions.

Version: 1.1 EL6711-0010
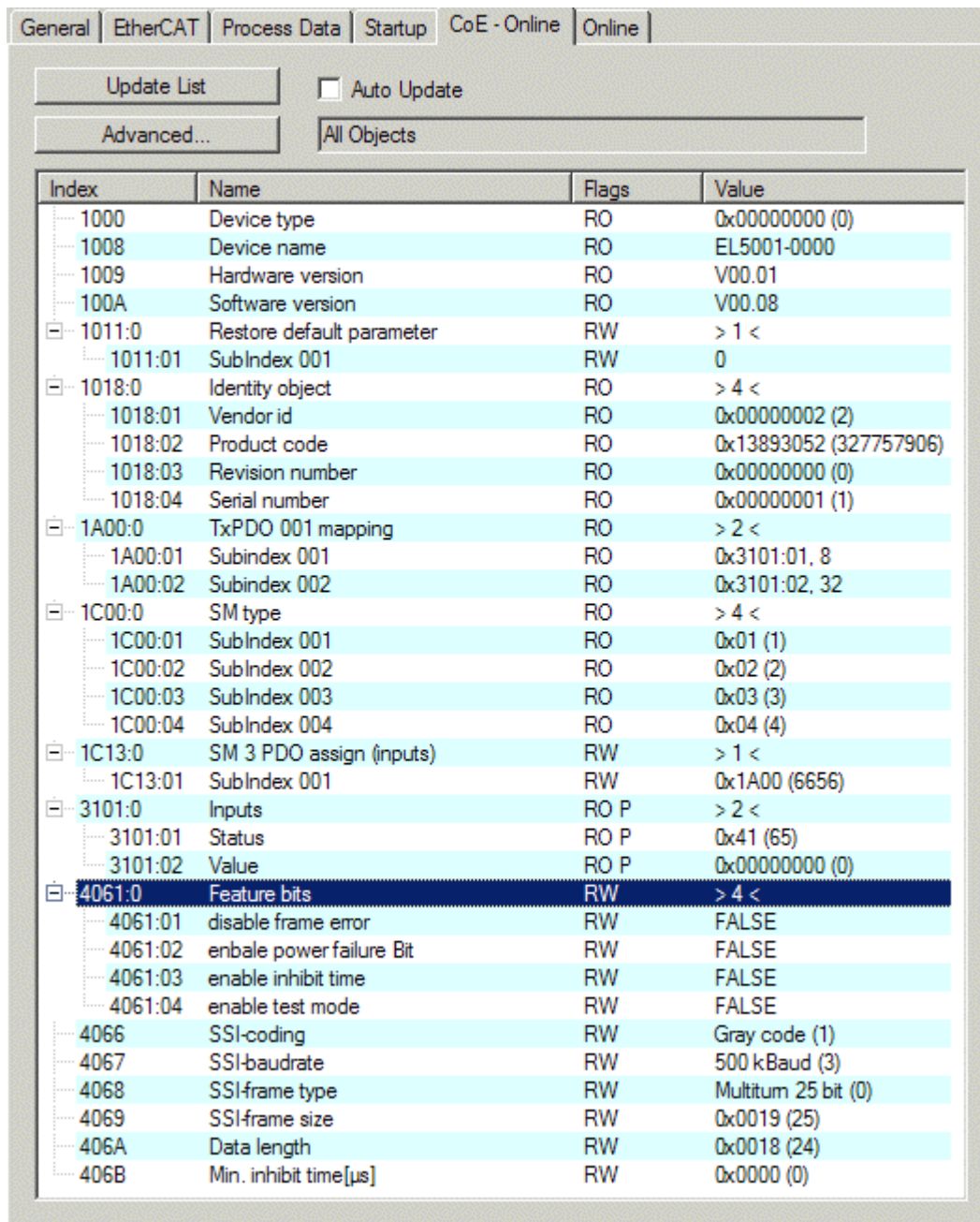
Fig. 115: "CoE - Online" tab

Object list display

Column	Description		
Index	Index and sub-index of the object		
Name	Name of the object		
Flags	RW	The object can be read, and data can be written to the object (read/write)
	RO	The object can be read, but no data can be written to the object (read only)
	P	An additional P identifies the object as a process data object.
Value	Value of the object		

Update List	The *Update list* button updates all objects in the displayed list
Auto Update	If this check box is selected, the content of the objects is updated automatically.
Advanced	The *Advanced* button opens the *Advanced Settings* dialog. Here you can specify which objects are displayed in the list.

BECKHOFF
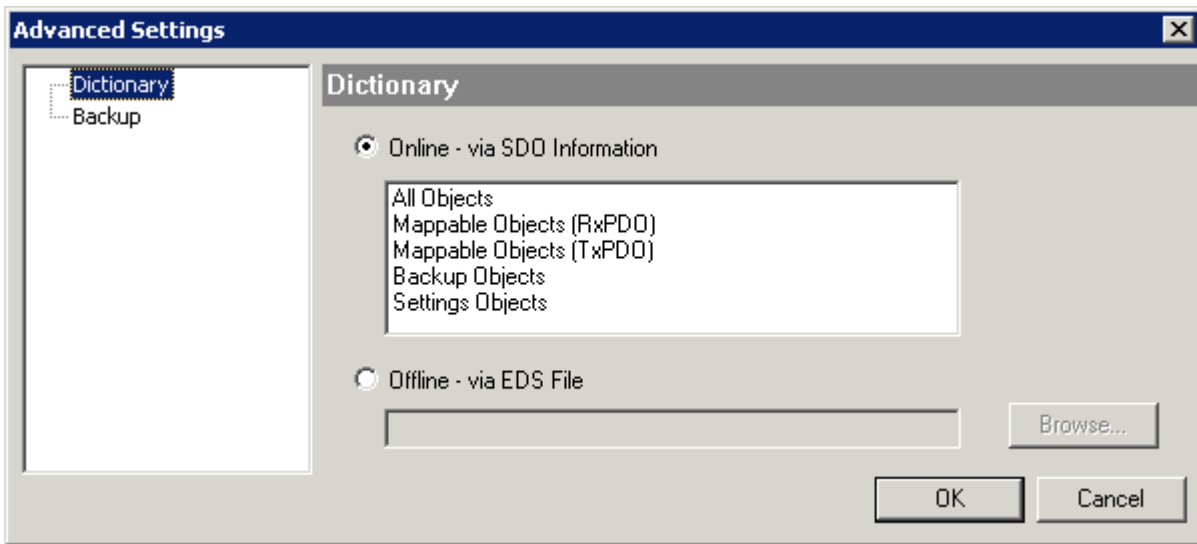
Fig. 116: Dialog "Advanced settings"

Online - via SDO Information	If this option button is selected, the list of the objects included in the object list of the slave is uploaded from the slave via SDO information. The list below can be used to specify which object types are to be uploaded.
Offline - via EDS File	If this option button is selected, the list of the objects included in the object list is read from an EDS file provided by the user.

"Online" tab
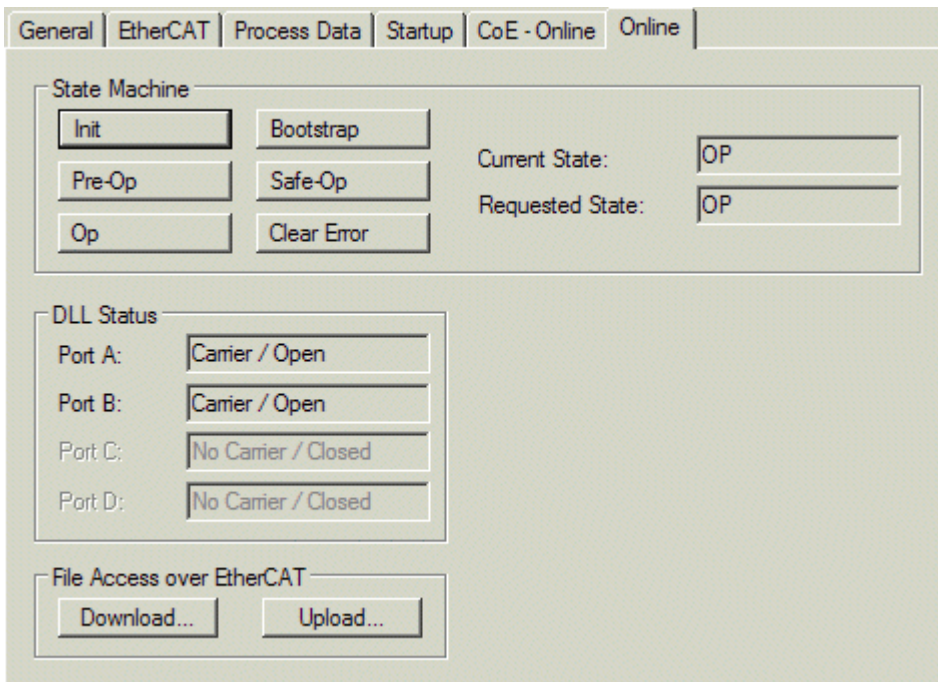
Fig. 117: "Online" tab

State Machine

Init	This button attempts to set the EtherCAT device to the *Init* state.
Pre-Op	This button attempts to set the EtherCAT device to the *pre-operational* state.
Op	This button attempts to set the EtherCAT device to the *operational* state.
Bootstrap	This button attempts to set the EtherCAT device to the *Bootstrap* state.
Safe-Op	This button attempts to set the EtherCAT device to the *safe-operational* state.
Clear Error	This button attempts to delete the fault display. If an EtherCAT slave fails during change of state it sets an error flag.
	Example: An EtherCAT slave is in PREOP state (pre-operational). The master now requests the SAFEOP state (safe-operational). If the slave fails during change of state it sets the error flag. The current state is now displayed as ERR PREOP. When the *Clear Error* button is pressed the error flag is cleared, and the current state is displayed as PREOP again.
Current State	Indicates the current state of the EtherCAT device.
Requested State	Indicates the state requested for the EtherCAT device.

DLL Status

Indicates the DLL status (data link layer status) of the individual ports of the EtherCAT slave. The DLL status can have four different states:

Status	Description
No Carrier / Open	No carrier signal is available at the port, but the port is open.
No Carrier / Closed	No carrier signal is available at the port, and the port is closed.
Carrier / Open	A carrier signal is available at the port, and the port is open.
Carrier / Closed	A carrier signal is available at the port, but the port is closed.

File Access over EtherCAT

Download	With this button a file can be written to the EtherCAT device.
Upload	With this button a file can be read from the EtherCAT device.

"DC" tab (Distributed Clocks)
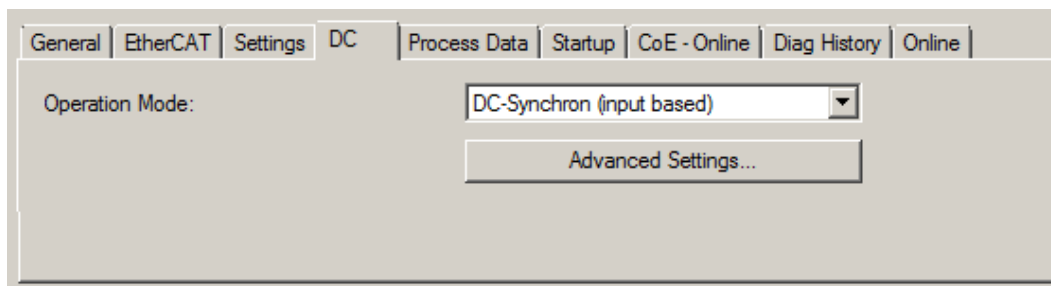
Fig. 118: "DC" tab (Distributed Clocks)

Operation Mode	Options (optional): • FreeRun • SM-Synchron • DC-Synchron (Input based) • DC-Synchron
Advanced Settings…	Advanced settings for readjustment of the real time determinant TwinCAT-clock

Detailed information to Distributed Clocks is specified on http://infosys.beckhoff.com:

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

BECKHOFF

5.2.7.1 Detailed description of Process Data tab

Sync Manager

Lists the configuration of the Sync Manager (SM).
If the EtherCAT device has a mailbox, SM0 is used for the mailbox output (MbxOut) and SM1 for the mailbox input (MbxIn).
SM2 is used for the output process data (outputs) and SM3 (inputs) for the input process data.

If an input is selected, the corresponding PDO assignment is displayed in the *PDO Assignment* list below.

PDO Assignment

PDO assignment of the selected Sync Manager. All PDOs defined for this Sync Manager type are listed here:

- If the output Sync Manager (outputs) is selected in the Sync Manager list, all RxPDOs are displayed.
- If the input Sync Manager (inputs) is selected in the Sync Manager list, all TxPDOs are displayed.

The selected entries are the PDOs involved in the process data transfer. In the tree diagram of the System Manager these PDOs are displayed as variables of the EtherCAT device. The name of the variable is identical to the *Name* parameter of the PDO, as displayed in the PDO list. If an entry in the PDO assignment list is deactivated (not selected and greyed out), this indicates that the input is excluded from the PDO assignment. In order to be able to select a greyed out PDO, the currently selected PDO has to be deselected first.

> **ⓘ** **Activation of PDO assignment**
>
> ✓ If you have changed the PDO assignment, in order to activate the new PDO assignment,
>
> a) the EtherCAT slave has to run through the PS status transition cycle (from pre-operational to safe-operational) once (see <u>Online tab</u> [▶ 94]),
>
> b) and the System Manager has to reload the EtherCAT slaves
>
> (button for TwinCAT 2 or button for TwinCAT 3)

PDO list

List of all PDOs supported by this EtherCAT device. The content of the selected PDOs is displayed in the *PDO Content* list. The PDO configuration can be modified by double-clicking on an entry.

Column	Description	
Index	PDO index.	
Size	Size of the PDO in bytes.	
Name	Name of the PDO. If this PDO is assigned to a Sync Manager, it appears as a variable of the slave with this parameter as the name.	
Flags	F	Fixed content: The content of this PDO is fixed and cannot be changed by the System Manager.
	M	Mandatory PDO. This PDO is mandatory and must therefore be assigned to a Sync Manager! Consequently, this PDO cannot be deleted from the *PDO Assignment* list
SM	Sync Manager to which this PDO is assigned. If this entry is empty, this PDO does not take part in the process data traffic.	
SU	Sync unit to which this PDO is assigned.	

PDO Content

Indicates the content of the PDO. If flag F (fixed content) of the PDO is not set the content can be modified.

Download

If the device is intelligent and has a mailbox, the configuration of the PDO and the PDO assignments can be downloaded to the device. This is an optional feature that is not supported by all EtherCAT slaves.

PDO Assignment

If this check box is selected, the PDO assignment that is configured in the PDO Assignment list is downloaded to the device on startup. The required commands to be sent to the device can be viewed in the tab.

PDO Configuration

If this check box is selected, the configuration of the respective PDOs (as shown in the PDO list and the PDO Content display) is downloaded to the EtherCAT slave.

5.2.8   Import/Export of EtherCAT devices with SCI and XTI

SCI and XTI Export/Import – Handling of user-defined modified EtherCAT slaves

5.2.8.1   Basic principles

An EtherCAT slave is basically parameterized through the following elements:

- Cyclic process data (PDO)
- Synchronization (Distributed Clocks, FreeRun, SM-Synchron)
- CoE parameters (acyclic object dictionary)

Note: Not all three elements may be present, depending on the slave.

For a better understanding of the export/import function, let's consider the usual procedure for IO configuration:

- The user/programmer processes the IO configuration in the TwinCAT system environment. This involves all input/output devices such as drives that are connected to the fieldbuses used.
  Note: In the following sections, only EtherCAT configurations in the TwinCAT system environment are considered.
- For example, the user manually adds devices to a configuration or performs a scan on the online system.
- This results in the IO system configuration.
- On insertion, the slave appears in the system configuration in the default configuration provided by the vendor, consisting of default PDO, default synchronization method and CoE StartUp parameter as defined in the ESI (XML device description).
- If necessary, elements of the slave configuration can be changed, e.g. the PDO configuration or the synchronization method, based on the respective device documentation.

It may become necessary to reuse the modified slave in other projects in this way, without having to make equivalent configuration changes to the slave again. To accomplish this, proceed as follows:

- Export the slave configuration from the project,
- Store and transport as a file,
- Import into another EtherCAT project.

TwinCAT offers two methods for this purpose:

- within the TwinCAT environment: Export/Import as **xti** file or
- outside, i.e. beyond the TwinCAT limits: Export/Import as **sci** file.

An example is provided below for illustration purposes: an EL3702 terminal with standard setting is switched to 2-fold oversampling (blue) and the optional PDO "StartTimeNextLatch" is added (red):
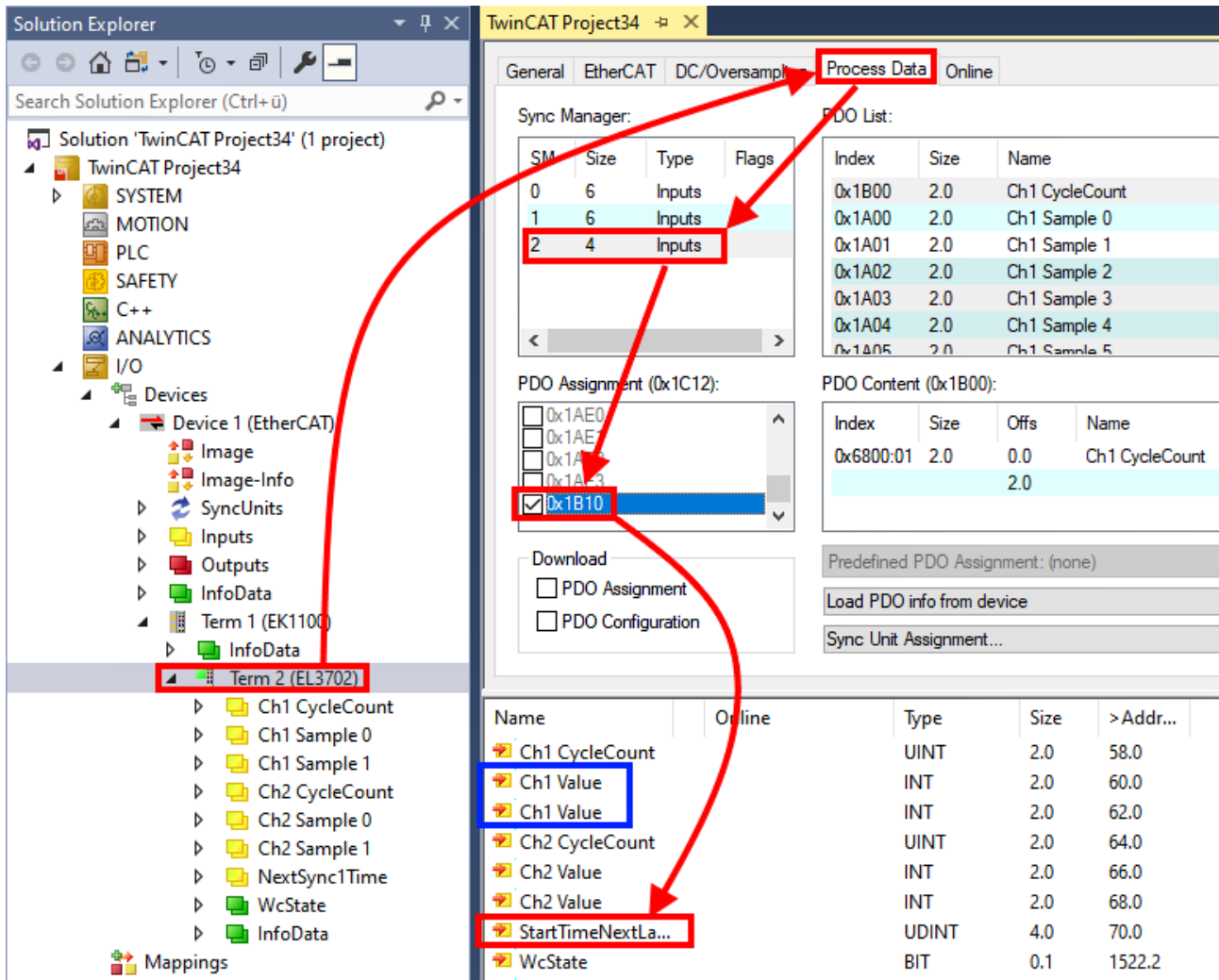
The two methods for exporting and importing the modified terminal referred to above are demonstrated below.

5.2.8.2 Procedure within TwinCAT with xti files

Each IO device can be exported/saved individually:
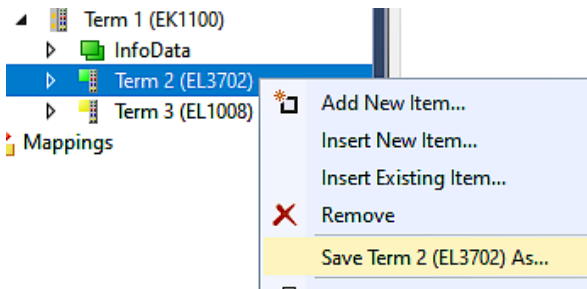
The xti file can be stored:
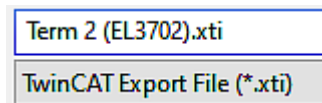
and imported again in another TwinCAT system via "Insert Existing item":
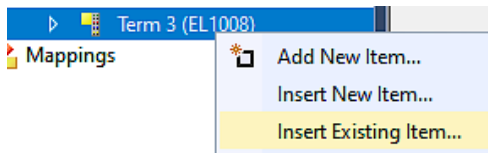
5.2.8.3 Procedure within and outside TwinCAT with sci file

Note regarding availability (2021/01)

The SCI method is available from TwinCAT 3.1 build 4024.14.

The Slave Configuration Information (SCI) describes a specific complete configuration for an EtherCAT slave (terminal, box, drive...) based on the setting options of the device description file (ESI, EtherCAT Slave Information). That is, it includes PDO, CoE, synchronization.

Export:

- select a single device via the menu (multiple selection is also possible):
  TwinCAT → EtherCAT Devices → Export SCI.
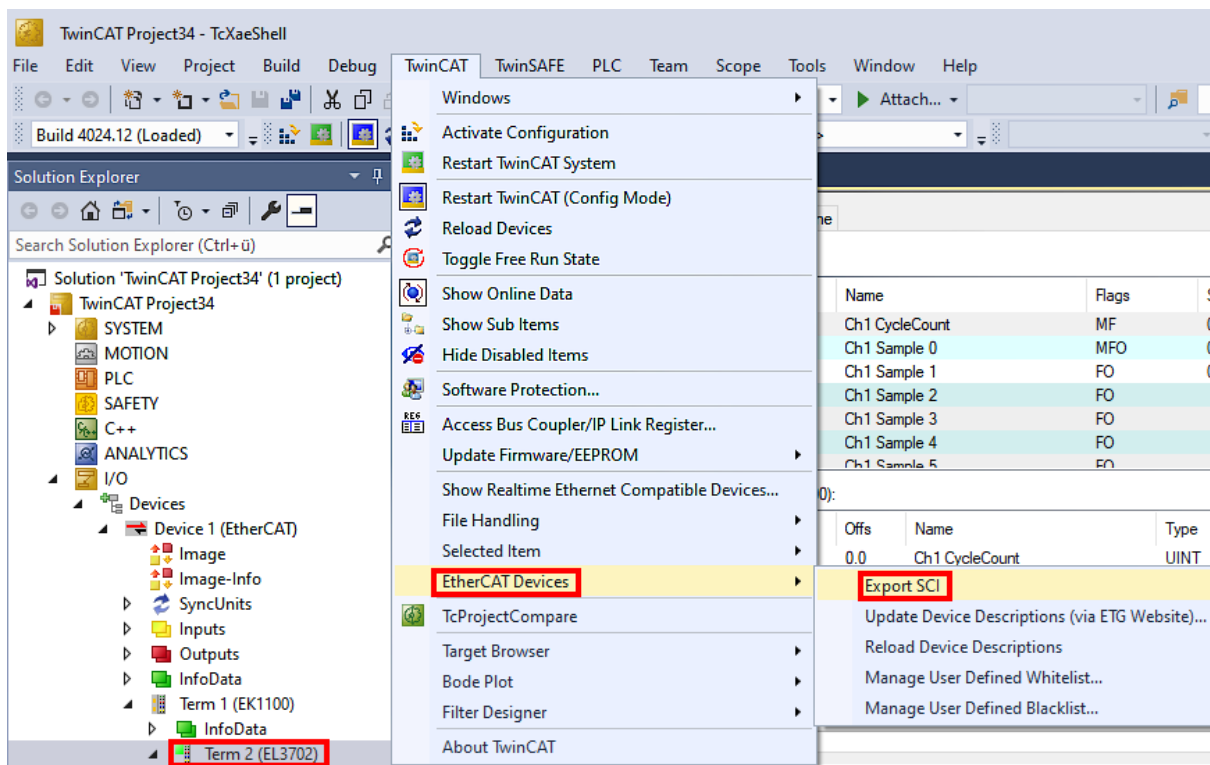
- If TwinCAT is offline (i.e. if there is no connection to an actual running controller) a warning message may appear, because after executing the function the system attempts to reload the EtherCAT segment. However, in this case this is not relevant for the result and can be acknowledged by clicking OK:
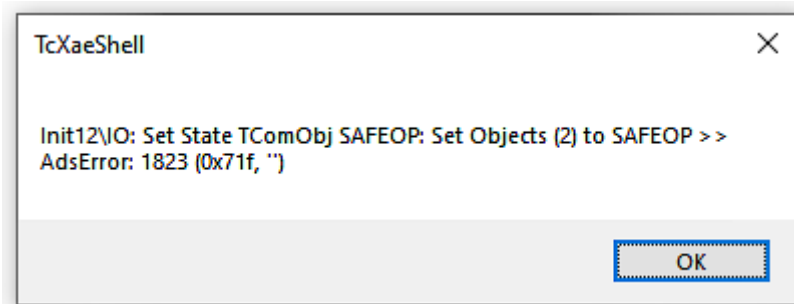
- A description may also be provided:
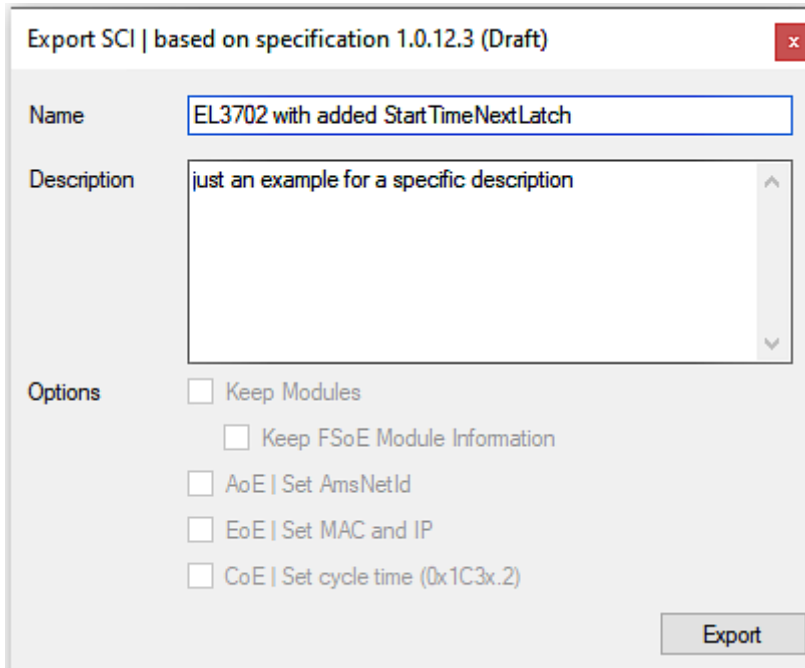
- Explanation of the dialog box:

Name		Name of the SCI, assigned by the user.
Description		Description of the slave configuration for the use case, assigned by the user.
Options	Keep modules	If a slave supports modules/slots, the user can decide whether these are to be exported or whether the module and device data are to be combined during export.
	AoE | Set AmsNetId	The configured AmsNetId is exported. Usually this is network-dependent and cannot always be determined in advance.
	EoE | Set MAC and IP	The configured virtual MAC and IP addresses are stored in the SCI. Usually these are network-dependent and cannot always be determined in advance.
	CoE | Set cycle time(0x1C3x.2)	The configured cycle time is exported. Usually this is network-dependent and cannot always be determined in advance.
ESI		Reference to the original ESI file.
Export		Save SCI file.

- A list view is available for multiple selections (*Export multiple SCI files)*:
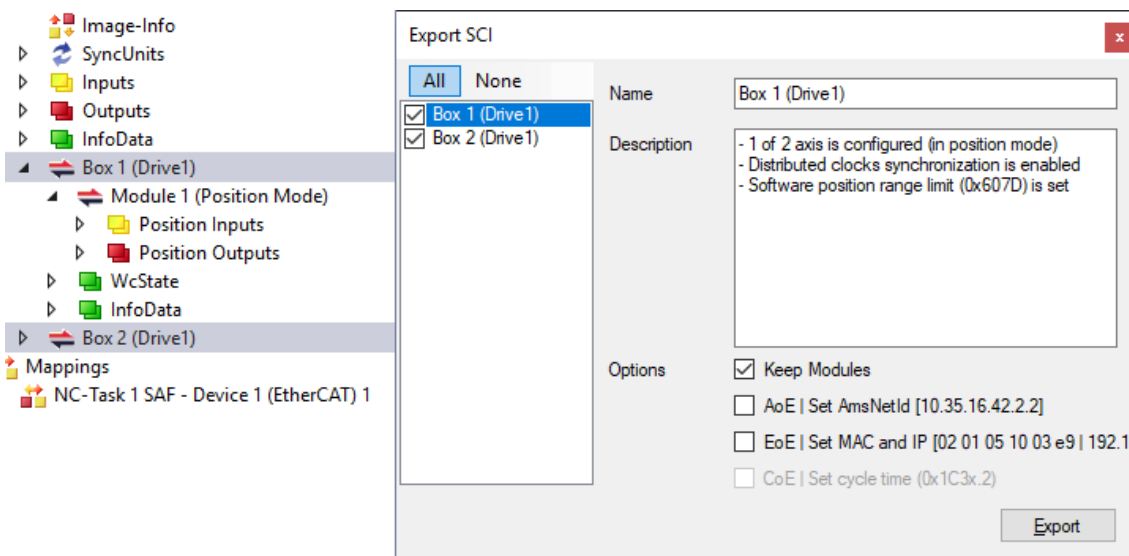
- Selection of the slaves to be exported:
  - All:
    All slaves are selected for export.

- ◦ None:
  All slaves are deselected.
- The sci file can be saved locally:

- The export takes place:
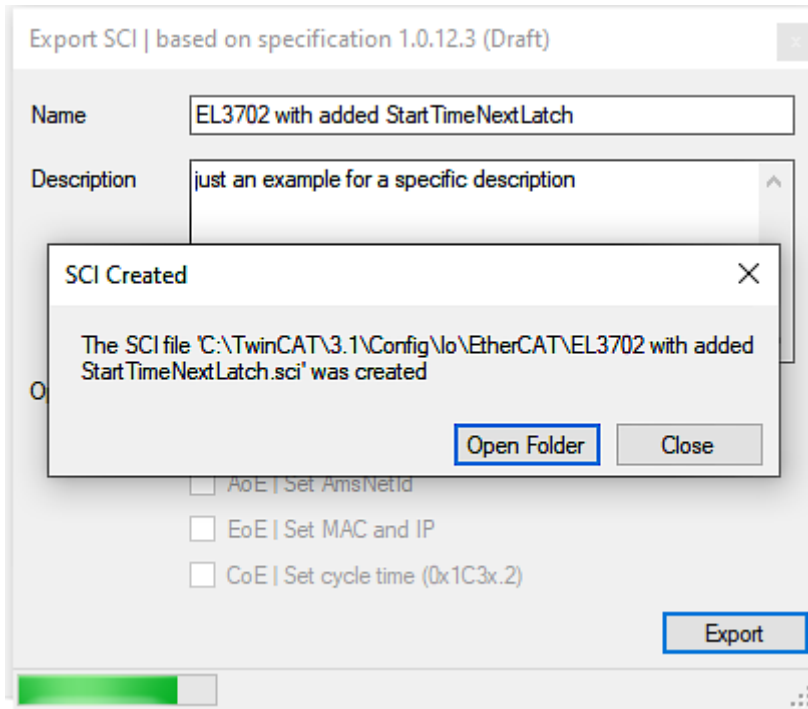
Import

- An sci description can be inserted manually into the TwinCAT configuration like any normal Beckhoff device description.
- The sci file must be located in the TwinCAT ESI path, usually under: C:\TwinCAT\3.1\Config\Io\EtherCAT

- Open the selection dialog:

BECKHOFF
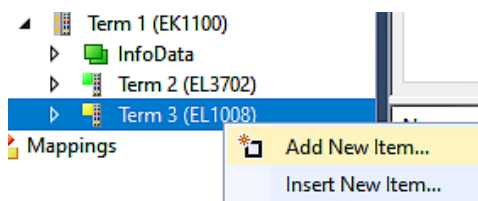
- Display SCI devices and select and insert the desired device:
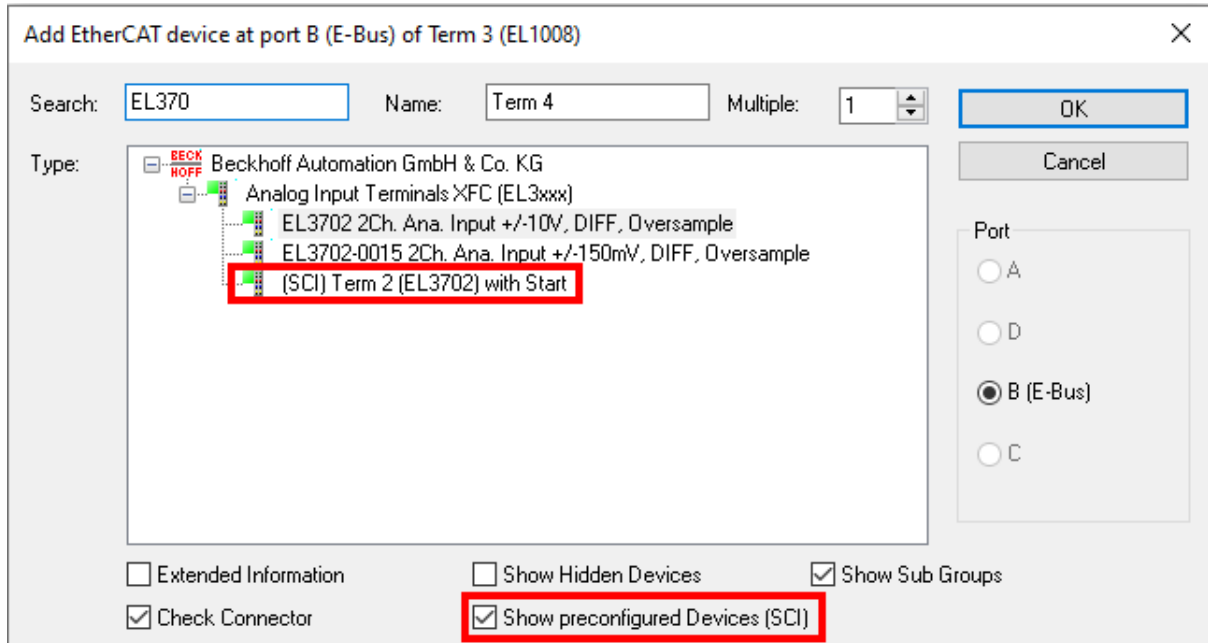
Additional Notes

- Settings for the SCI function can be made via the general Options dialog
  (Tools → Options → TwinCAT → Export SCI):
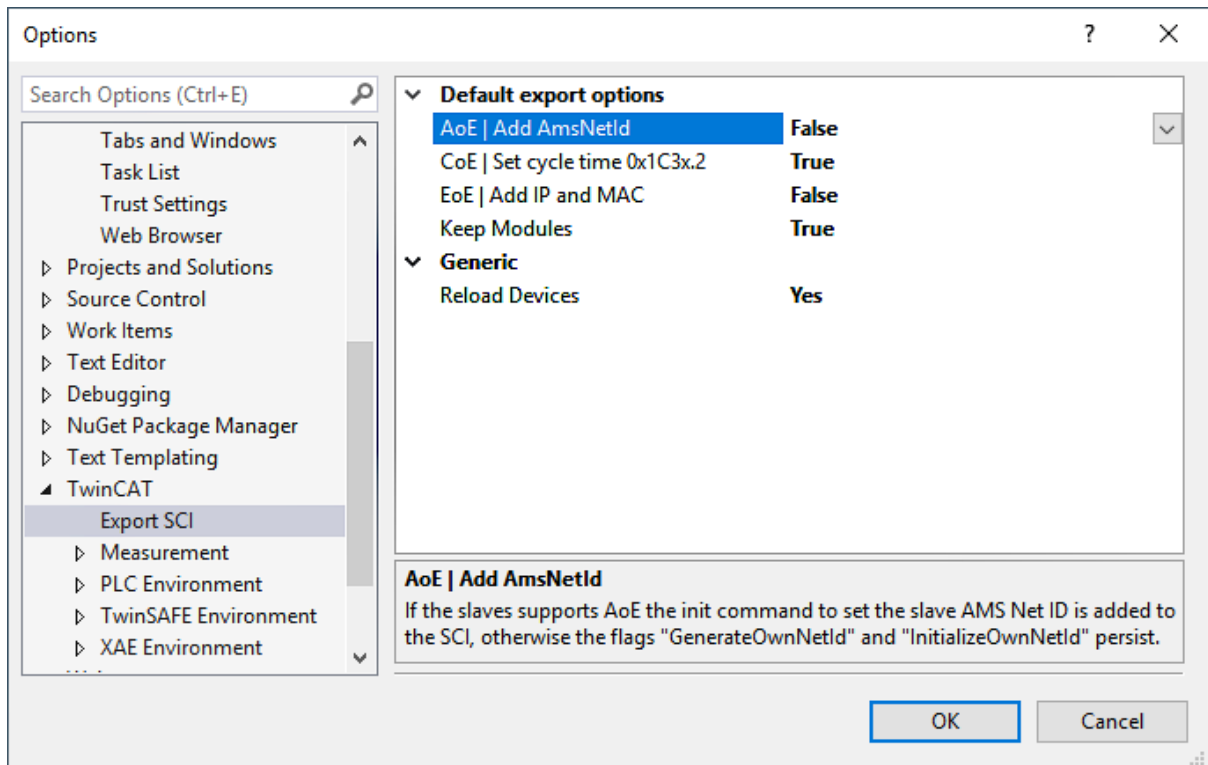
Explanation of the settings:

Default export options	AoE | Set AmsNetId	Default setting whether the configured AmsNetId is exported.
	CoE | Set cycle time(0x1C3x.2)	Default setting whether the configured cycle time is exported.
	EoE | Set MAC and IP	Default setting whether the configured MAC and IP addresses are exported.
	Keep modules	Default setting whether the modules persist.
Generic	Reload Devices	Setting whether the Reload Devices command is executed before the SCI export. This is strongly recommended to ensure a consistent slave configuration.

SCI error messages are displayed in the TwinCAT logger output window if required:

```
Output
Show output from: Export SCI                            ▼  | ⚲ | ⇐ ⇒ | ⊠ | ⁂
 02/07/2020 14:09:17 Reload Devices
 02/07/2020 14:09:18 | Box 1 (Drive1) No EtherCAT Slave Information (ESI) available for 'Box 1 (Drive1)
```
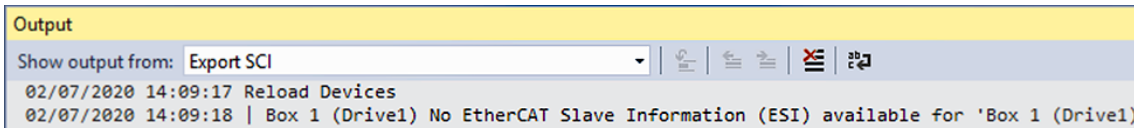
5.3   General Commissioning Instructions for an EtherCAT Slave

This summary briefly deals with a number of aspects of EtherCAT Slave operation under TwinCAT. More detailed information on this may be found in the corresponding sections of, for instance, the EtherCAT System Documentation.

Diagnosis in real time: WorkingCounter, EtherCAT State and Status

Generally speaking an EtherCAT Slave provides a variety of diagnostic information that can be used by the controlling task.

This diagnostic information relates to differing levels of communication. It therefore has a variety of sources, and is also updated at various times.

Any application that relies on I/O data from a fieldbus being correct and up to date must make diagnostic access to the corresponding underlying layers. EtherCAT and the TwinCAT System Manager offer comprehensive diagnostic elements of this kind. Those diagnostic elements that are helpful to the controlling task for diagnosis that is accurate for the current cycle when in operation (not during commissioning) are discussed below.
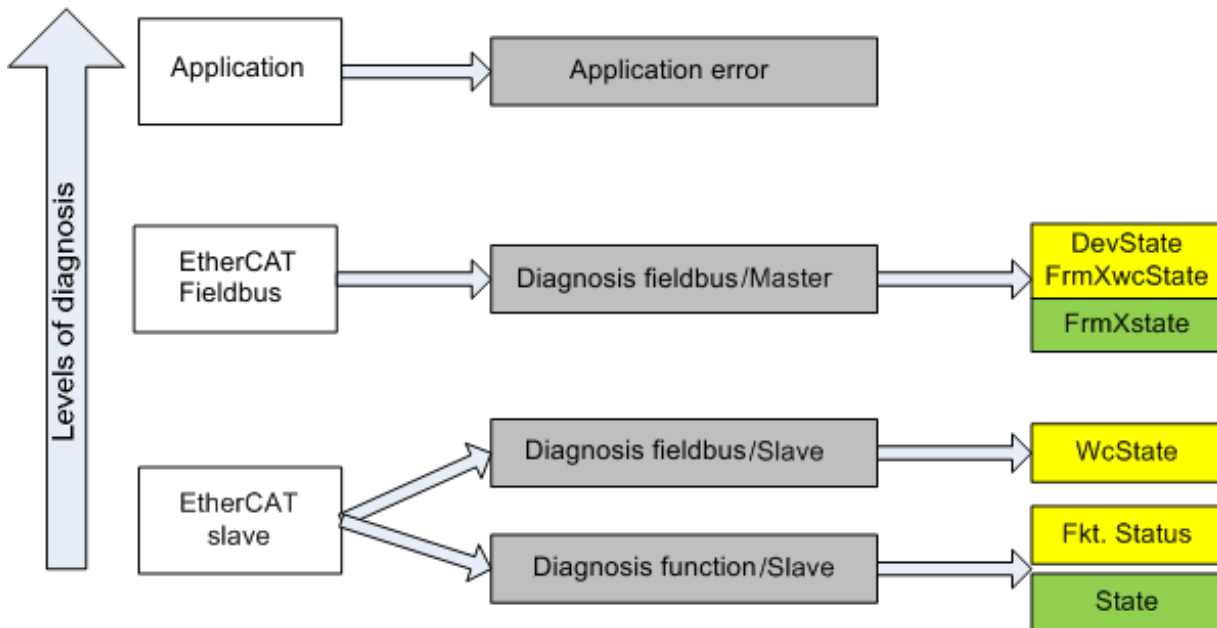
Fig. 119: Selection of the diagnostic information of an EtherCAT Slave

In general, an EtherCAT Slave offers

- communication diagnosis typical for a slave (diagnosis of successful participation in the exchange of process data, and correct operating mode)
  This diagnosis is the same for all slaves.

  as well as

- function diagnosis typical for a channel (device-dependent)
  See the corresponding device documentation

The colors in Fig. *Selection of the diagnostic information of an EtherCAT Slave* also correspond to the variable colors in the System Manager, see Fig. *Basic EtherCAT Slave Diagnosis in the PLC*.

Colour	Meaning
yellow	Input variables from the Slave to the EtherCAT Master, updated in every cycle
red	Output variables from the Slave to the EtherCAT Master, updated in every cycle
green	Information variables for the EtherCAT Master that are updated acyclically. This means that it is possible that in any particular cycle they do not represent the latest possible status. It is therefore useful to read such variables through ADS.

Fig. *Basic EtherCAT Slave Diagnosis in the PLC* shows an example of an implementation of basic EtherCAT Slave Diagnosis. A Beckhoff EL3102 (2-channel analogue input terminal) is used here, as it offers both the communication diagnosis typical of a slave and the functional diagnosis that is specific to a channel. Structures are created as input variables in the PLC, each corresponding to the process image.

Fig. 120: Basic EtherCAT Slave Diagnosis in the PLC

The following aspects are covered here:

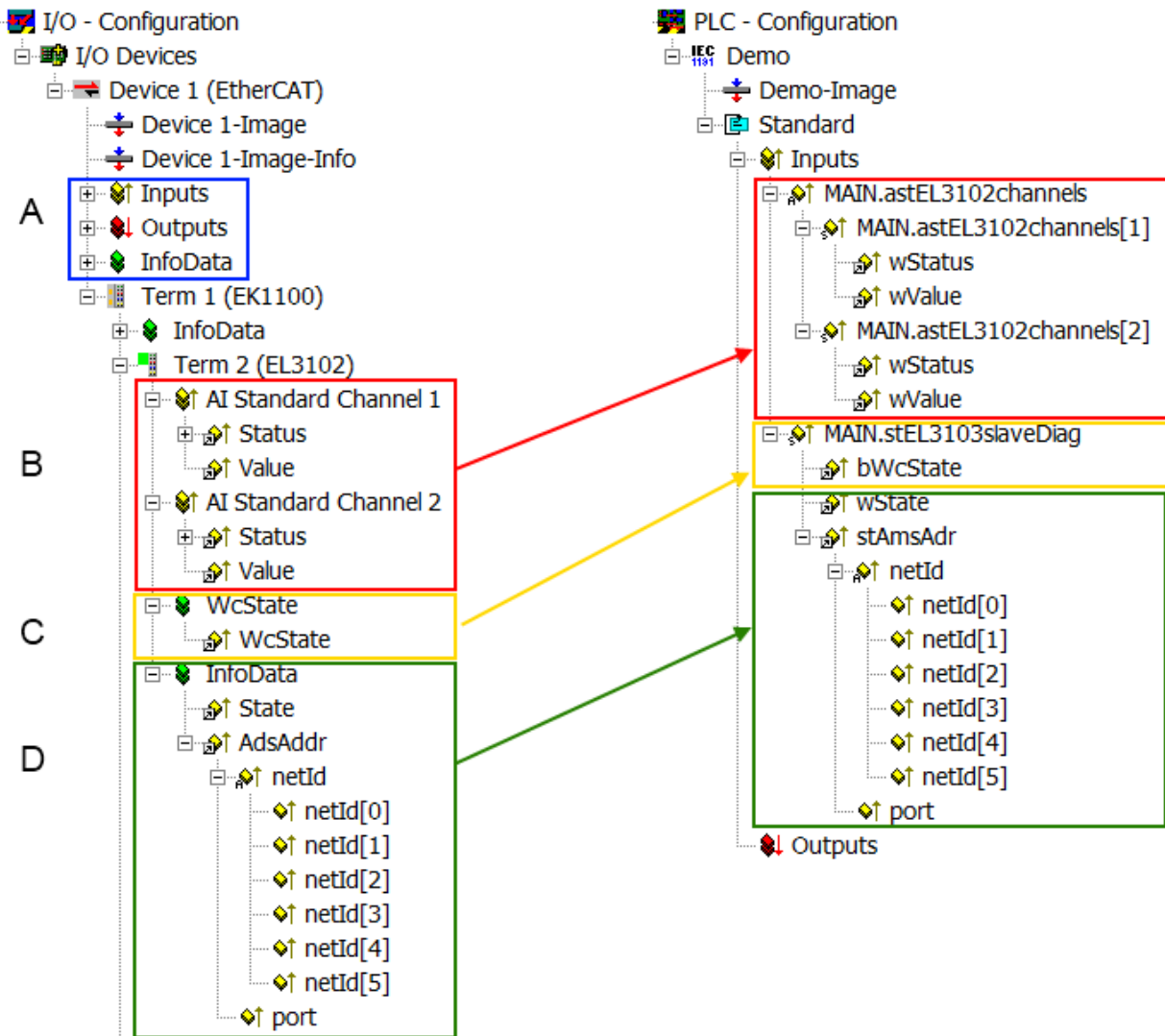| Code	Function	Implementation	Application/evaluation
A	The EtherCAT Master's diagnostic information updated acyclically (yellow) or provided acyclically (green).		At least the DevState is to be evaluated for the most recent cycle in the PLC. The EtherCAT Master's diagnostic information offers many more possibilities than are treated in the EtherCAT System Documentation. A few keywords: • CoE in the Master for communication with/through the Slaves • Functions from *TcEtherCAT.lib* • Perform an OnlineScan
B	In the example chosen (EL3102) the EL3102 comprises two analogue input channels that transmit a single function status for the most recent cycle.	Status • the bit significations may be found in the device documentation • other devices may supply more information, or none that is typical of a slave	In order for the higher-level PLC task (or corresponding control applications) to be able to rely on correct data, the function status must be evaluated there. Such information is therefore provided with the process data for the most recent cycle.
C	For every EtherCAT Slave that has cyclic process data, the Master displays, using what is known as a WorkingCounter, whether the slave is participating successfully and without error in the cyclic exchange of process data. This important, elementary information is therefore provided for the most recent cycle in the System Manager 1. at the EtherCAT Slave, and, with identical contents 2. as a collective variable at the EtherCAT Master (see Point A) for linking.	WcState (Working Counter) 0: valid real-time communication in the last cycle 1: invalid real-time communication This may possibly have effects on the process data of other Slaves that are located in the same SyncUnit	In order for the higher-level PLC task (or corresponding control applications) to be able to rely on correct data, the communication status of the EtherCAT Slave must be evaluated there. Such information is therefore provided with the process data for the most recent cycle.
D	Diagnostic information of the EtherCAT Master which, while it is represented at the slave for linking, is actually determined by the Master for the Slave concerned and represented there. This information cannot be characterized as real-time, because it • is only rarely/never changed, except when the system starts up • is itself determined acyclically (e.g. EtherCAT Status)	State current Status (INIT..OP) of the Slave. The Slave must be in OP (=8) when operating normally. *AdsAddr* The ADS address is useful for communicating from the PLC/task via ADS with the EtherCAT Slave, e.g. for reading/writing to the CoE. The AMS-NetID of a slave corresponds to the AMS-NetID of the EtherCAT Master; communication with the individual Slave is possible via the *port* (= EtherCAT address).	Information variables for the EtherCAT Master that are updated acyclically. This means that it is possible that in any particular cycle they do not represent the latest possible status. It is therefore possible to read such variables through ADS.

NOTICE

Diagnostic information

It is strongly recommended that the diagnostic information made available is evaluated so that the application can react accordingly.

CoE Parameter Directory

The CoE parameter directory (CanOpen-over-EtherCAT) is used to manage the set values for the slave concerned. Changes may, in some circumstances, have to be made here when commissioning a relatively complex EtherCAT Slave. It can be accessed through the TwinCAT System Manager, see Fig. *EL3102, CoE directory*:

BECKHOFF
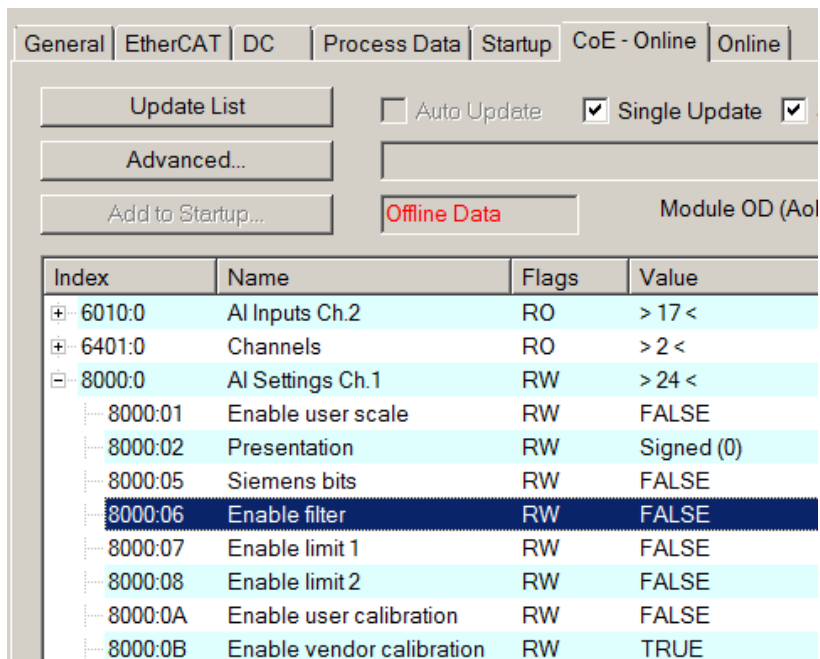
Fig. 121: EL3102, CoE directory

ⓘ EtherCAT System Documentation

The comprehensive description in the EtherCAT System Documentation (EtherCAT Basics --> CoE Interface) must be observed!

A few brief extracts:

- Whether changes in the online directory are saved locally in the slave depends on the device. EL terminals (except the EL66xx) are able to save in this way.
- The user must manage the changes to the StartUp list.

Commissioning aid in the TwinCAT System Manager

Commissioning interfaces are being introduced as part of an ongoing process for EL/EP EtherCAT devices. These are available in TwinCAT System Managers from TwinCAT 2.11R2 and above. They are integrated into the System Manager through appropriately extended ESI configuration files.
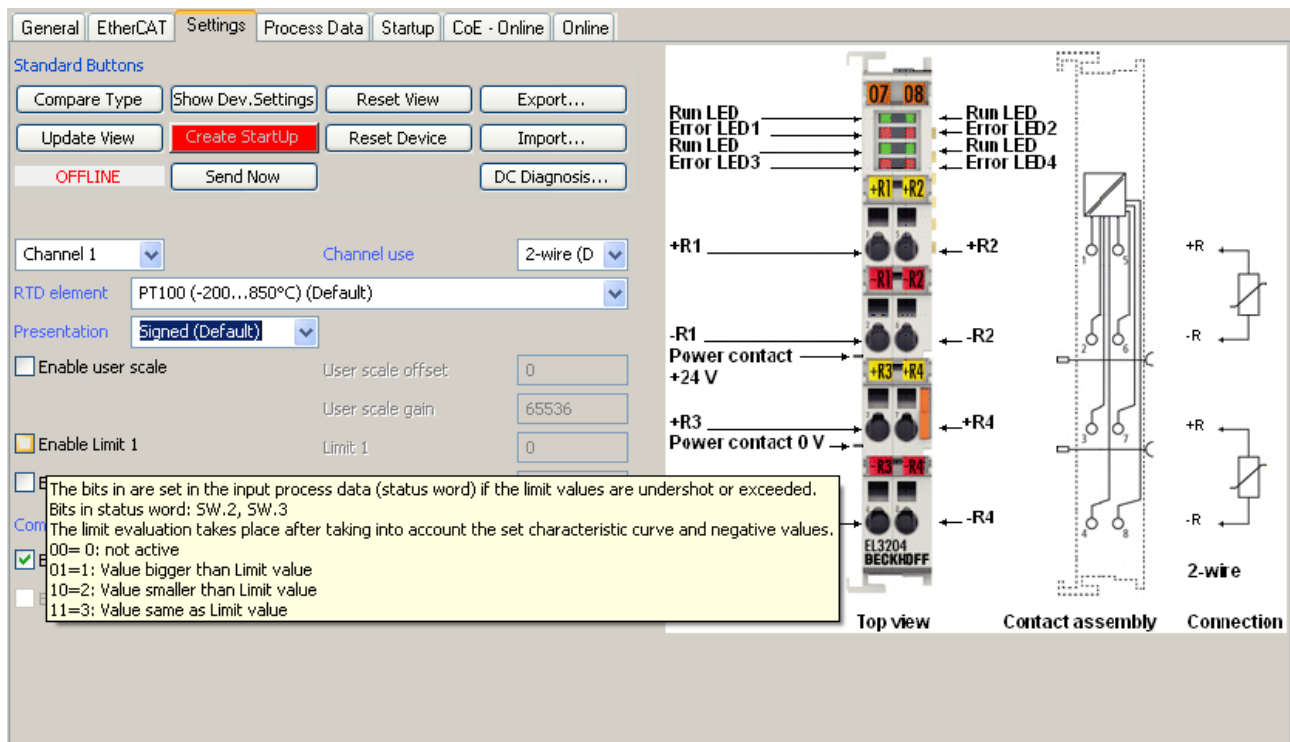
Fig. 122: Example of commissioning aid for a EL3204

This commissioning process simultaneously manages

- CoE Parameter Directory
- DC/FreeRun mode
- the available process data records (PDO)

Although the "Process Data", "DC", "Startup" and "CoE-Online" that used to be necessary for this are still displayed, it is recommended that, if the commissioning aid is used, the automatically generated settings are not changed by it.

The commissioning tool does not cover every possible application of an EL/EP device. If the available setting options are not adequate, the user can make the DC, PDO and CoE settings manually, as in the past.

EtherCAT State: automatic default behaviour of the TwinCAT System Manager and manual operation

After the operating power is switched on, an EtherCAT Slave must go through the following statuses

- INIT
- PREOP
- SAFEOP
- OP

to ensure sound operation. The EtherCAT Master directs these statuses in accordance with the initialization routines that are defined for commissioning the device by the ES/XML and user settings (Distributed Clocks (DC), PDO, CoE). See also the section on "Principles of Communication, EtherCAT State Machine [▶ 20]" in this connection. Depending how much configuration has to be done, and on the overall communication, booting can take up to a few seconds.

The EtherCAT Master itself must go through these routines when starting, until it has reached at least the OP target state.

The target state wanted by the user, and which is brought about automatically at start-up by TwinCAT, can be set in the System Manager. As soon as TwinCAT reaches the status RUN, the TwinCAT EtherCAT Master will approach the target states.

Standard setting

The advanced settings of the EtherCAT Master are set as standard:

- EtherCAT Master: OP
- Slaves: OP
  This setting applies equally to all Slaves.
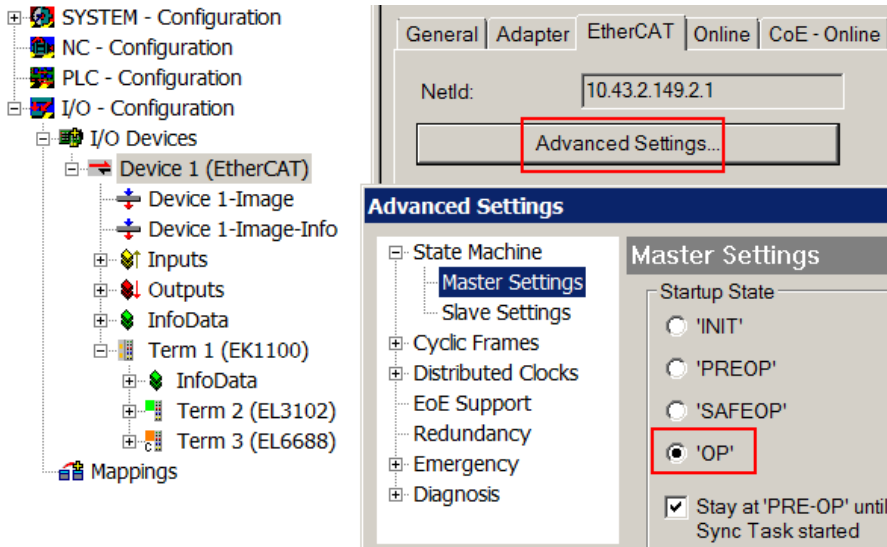
Fig. 123: Default behaviour of the System Manager

In addition, the target state of any particular Slave can be set in the "Advanced Settings" dialogue; the standard setting is again OP.
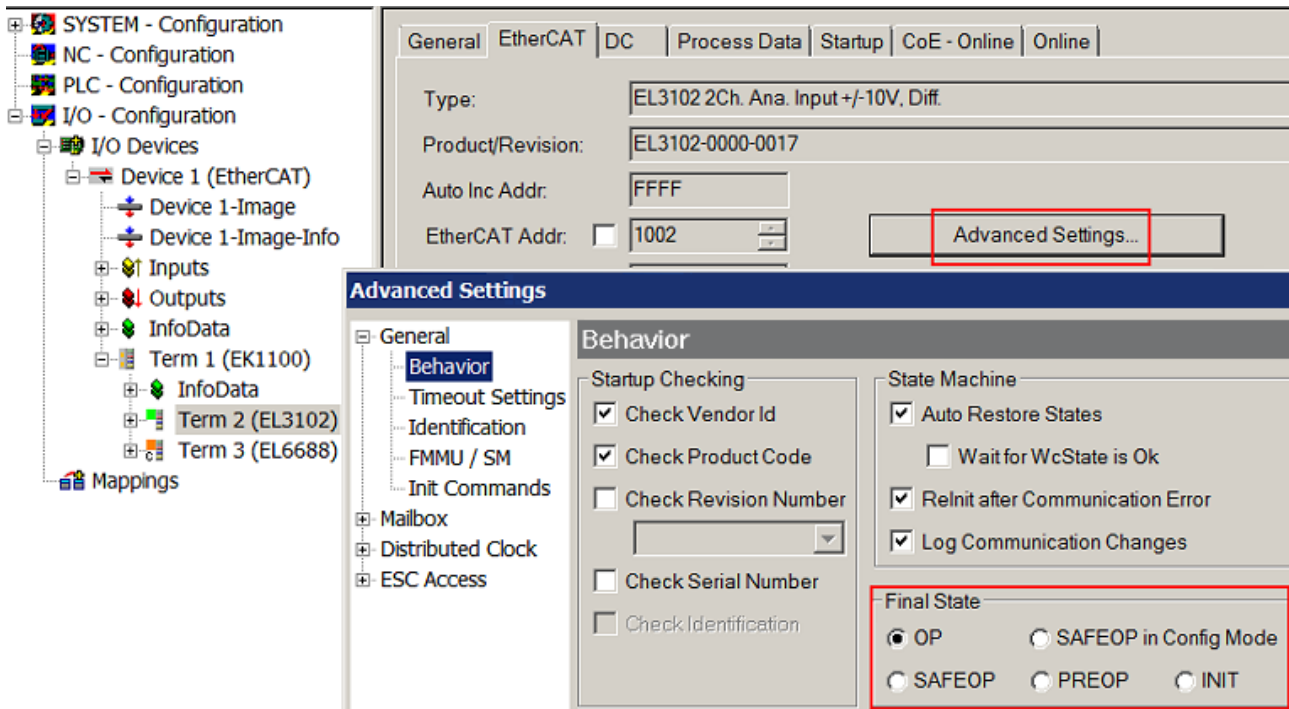
Fig. 124: Default target state in the Slave

Manual Control

There are particular reasons why it may be appropriate to control the states from the application/task/PLC. For instance:

- for diagnostic reasons
- to induce a controlled restart of axes

- because a change in the times involved in starting is desirable

In that case it is appropriate in the PLC application to use the PLC function blocks from the *TcEtherCAT.lib*, which is available as standard, and to work through the states in a controlled manner using, for instance, *FB_EcSetMasterState*.

It is then useful to put the settings in the EtherCAT Master to INIT for master and slave.
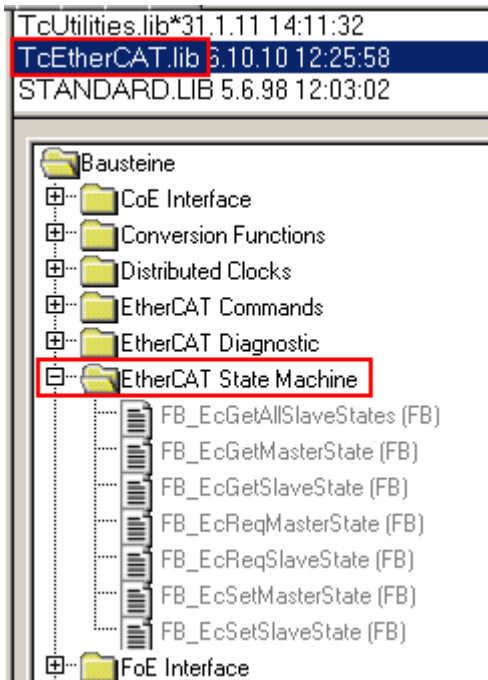
Fig. 125: PLC function blocks

Note regarding E-Bus current

EL/ES terminals are placed on the DIN rail at a coupler on the terminal strand. A Bus Coupler can supply the EL terminals added to it with the E-bus system voltage of 5 V; a coupler is thereby loadable up to 2 A as a rule. Information on how much current each EL terminal requires from the E-bus supply is available online and in the catalogue. If the added terminals require more current than the coupler can supply, then power feed terminals (e.g. EL9410) must be inserted at appropriate places in the terminal strand.

The pre-calculated theoretical maximum E-Bus current is displayed in the TwinCAT System Manager as a column value. A shortfall is marked by a negative total amount and an exclamation mark; a power feed terminal is to be placed before such a position.

BECKHOFF

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 !

NetId: 10.43.2.149.2.1   Advanced Settings...

Tabs: General | Adapter | EtherCAT | Online | CoE - Online

Fig. 126: Illegally exceeding the E-Bus current

From TwinCAT 2.11 and above, a warning message "E-Bus Power of Terminal..." is output in the logger window when such a configuration is activated:

Message

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

Fig. 127: Warning message for exceeding E-Bus current

NOTICE
Caution! Malfunction possible!
The same ground potential must be used for the E-Bus supply of all EtherCAT terminals in a terminal block!

5.4 Basic function principles

The EL6711-0010 CC-Link slave terminal enables data exchange between EtherCAT master and CC-Link master. It is therefore a slave from perspective of both bus systems. The process data are created and configured via the EtherCAT master, where bus-specific settings such as baud rate, station address etc. are also implemented.

In the delivery state the EL6711-0010 automatically applies the appropriate settings such as CC-Link version, number of occupied stations and number of extended cycles, based on the created process data. These automatically created settings can be adjusted so that the combination of the number of occupied stations or the extended cycles matches the user requirements (Index F800 [▶ 124], SubIndex 05: Minimize). Furthermore, it is possible to make these settings manually or to adjust and change them afterwards. This means that no settings, parameterization or configuration are required on the part of the CC-Link master.

During data exchange, predefined process data are copied to the memory area of the terminal, where they are available to the respective opposite side for further processing.

The context menu (right-click, then select "Add New Item...") can be used to create a configuration or integrate the CC-Link slave terminal in a TwinCAT project. Alternatively, select the "Scan" function from the context menu of device n (EtherCAT), which automatically reads all connected devices and boxes and integrates them into the structure.
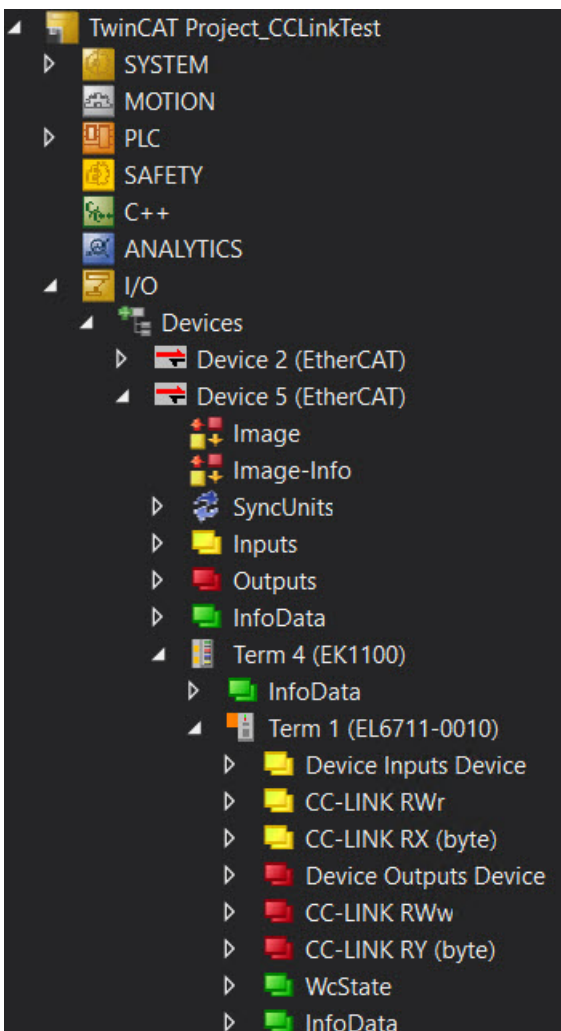

Fig. 128: Display in TwinCAT

5.4.1   Process data

This section describes the individual PDOs and their content. A PDO (Process Data Object) is a unit on cyclically transmitted process values. The individual PDOs can be activated or deactivated separately in the TwinCAT System Manager. The "Process data" tab is used for this (visible only if the terminal is selected on the left). A change in the process data composition in TwinCAT only becomes effective after restarting the EtherCAT system.

The EL6711-0010 can be operated in two CC-Link versions, which differ in the amount of process data they contain

- CC-Link version 1.00 or 1.10 is the first release of CC-Link. In contrast to version 1.00, version 1.10 allows cable lengths of 20 cm between the stations. This version change therefore affects network devices and network cables.
- CC-Link version 2.00 supports 8 times the amount of process data in the system and per station.

The amount of process data created determines the version in which the EL6711-0010 is operated. The version setting can also be fixed to CC-Link version 2, even if the sum of the created process data corresponds to CC-Link version 1. This means that the EL6711-0010 is always available to the master as CC-Link slave version 2 (Index F800 [▶ 124], SubIndex 06: Protocol). In addition to the self-configurable process data, the EL6711-0010 also transmits the status PDOI. This status information is transmitted cyclically from the terminal to the EtherCAT master as process data (PDO).

The EL6711-0010 transmits status information in the form of process data:

- **TxPDO State:** validity of the data of the associated TxPDO (0=valid, 1=invalid).
- **Input cycle counter:** is incremented with each process data cycle and switches to 0 after its maximum value of 3.
- **Status:** shows that the configured process data (RWr, RWw, RX, RY) are more, less or equal to the configurable process data. It should be noted that this is not an error message but is for information purposes only.
  - **0x0001:** more RWw are configured than can be transmitted via CC-Link
  - **0x0002:** fewer RWw are configured than can be transmitted via CC-Link
  - **0x0004:** more RWr are configured than can be transmitted via CC-Link
  - **0x0008:** fewer RWr are configured than can be transmitted via CC-Link
  - **0x0010:** more RY are configured than can be transmitted via CC-Link
  - **0x0020:** fewer RY are configured than can be transmitted via CC-Link
  - **0x0040:** more RX are configured than can be transmitted via CC-Link
  - **0x0080:** fewer RX are configured than can be transmitted via CC-Link
  - **0x0100:** invalid CC-Link station address
  - **0x0200**: CC-Link protocol conflict

5.4.1.1   Selection of process data

The process data of the EL6711-0010 are set up in TwinCAT. The PDOs can be activated or deactivated separately. The "Process data" tab is used for this (visible only if the terminal is selected on the left).
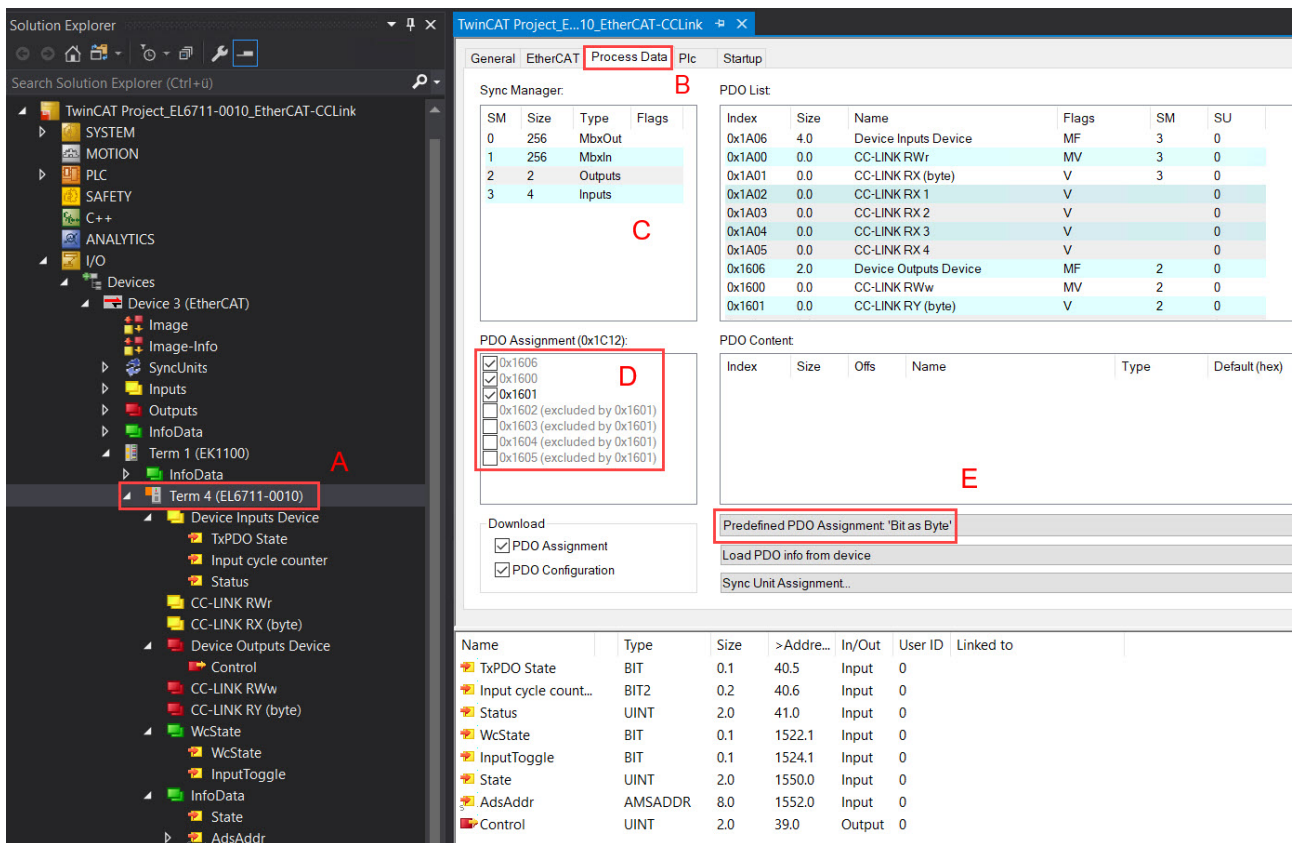
Fig. 129: Process data

If the terminal is selected in the project tree (A), the Process Data tab (B) shows the PDO selection. The two SyncManagers of the inputs (SM3) and outputs (SM2) can be changed (C). If one of the two is clicked on, the PDO possible for this SyncManager appears underneath it (D). PDOs that are already activated have an activated checkbox in front of them; this is activated by clicking on it. The process data which then belong to the device are listed underneath it.

Predefined PDO Assignment

In order to simplify the configuration, typical configuration combinations of process data are stored in the device description. The predefined configurations can be selected in the process data overview. Therefore, the function is only available if the ESI/XML files are stored on the system. The predefined PDO assignments only affect the remote I/Os, i.e. the bit data (RX and RY) but not the remote registers (RWw and RWr).

The following predefined PDO assignments are available for selection:
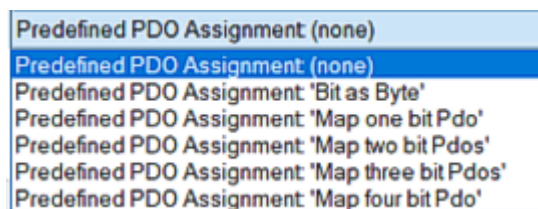
Fig. 130: Predefined PDO Assignments

- Bit as byte
    - Process data of type byte instead of bit can be appended.
- Map one, two, three or four bit Pdo
    - Depending on the selection, one, two, three or four PDO assignments are created in the process image
    - Only process data of type bit may be appended
    - The number of created process data must be a multiple of 8

◦ The sum of all applied bit process data must not exceed 896 bits (max. CC-Link Version 2). Therefore

◦ the maximum number of permitted entries for "Predefined PDO Assignment; 'Map one bit Pdo', 'Map two bit Pdo' and 'Map three bit Pdo'" is 255;

◦ the maximum number of permitted entries for "Predefined PDO Assignment. ,Map four bit Pdo'" is 131

5.4.2   Configuration

In order to exchange process data cyclically between the EtherCAT master and the CC-Link master, these must be created. The entire configuration and all settings are carried out via the EtherCAT master on the EL6711-0010 CC-Link slave terminal.

With the inclusion of the EL6711-0010 in a TwinCAT project, the way in which the process data are to be added must be defined. See the chapter Selection of process data [▶ 112], especially the subchapters.

Key points:

- Only process data of type UINT16, UINT8 and BIT are permitted
- The CC-Link specification [▶ 17] and its limits must be observed
- The points under "The following Predefined PDO Assignments are available for selection" from chapter Predefined PDO Assignments [▶ 113] must be taken into account.

The procedure for adding process data is always the same; the PDO assignment is irrelevant.

  1. Right-click on one of the PDO assignments and select **Add New Item…**.
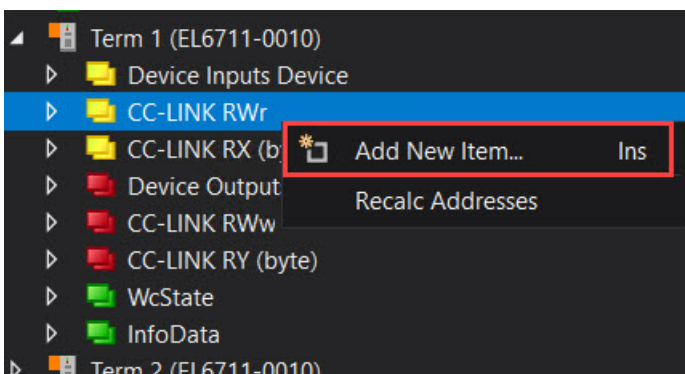
Fig. 131: Adding process data

  2. In the **Insert Device** window that opens, select the required data type, name, number etc.
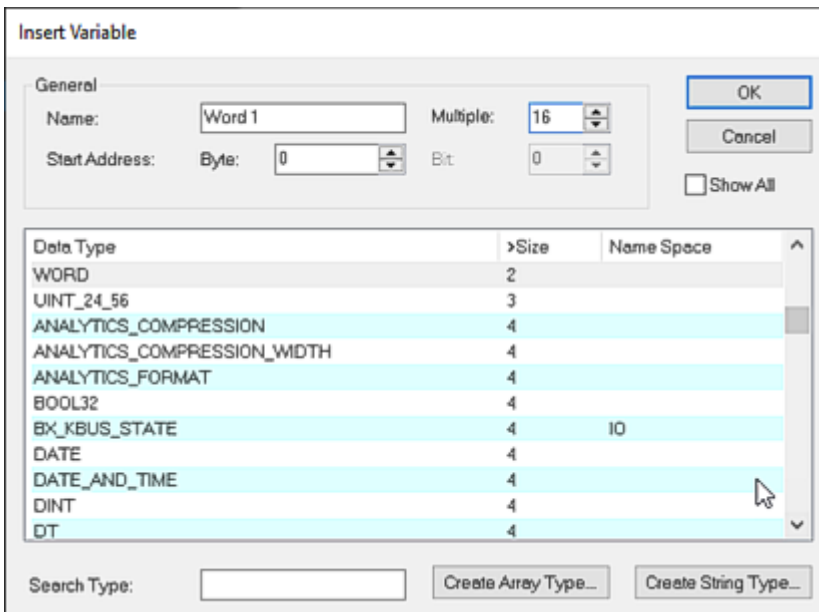
Fig. 132: Inserting variables

  3. Several variables of the same type can be declared simultaneously via the Multiple selection box, and a defined start address can be set. Once the above dialog is confirmed, the process image shown in the next step appears

BECKHOFF

4. The variable names are incremented automatically, as shown here. Output variables are created accordingly.
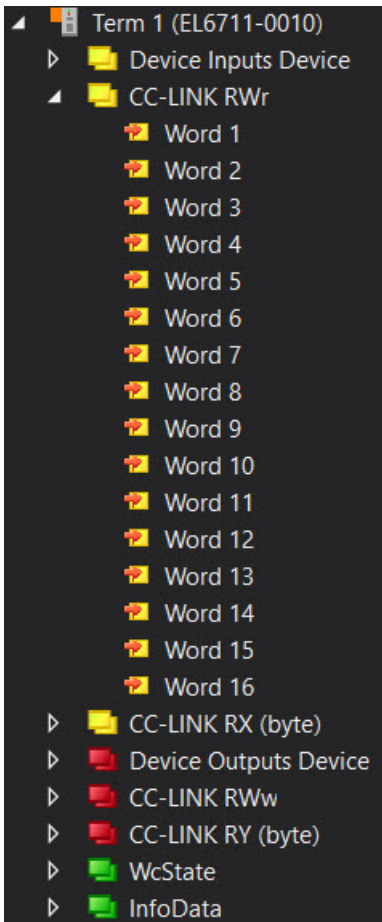
Fig. 133: Creating the input variables

Apply step 2) analogously to the other PDO assignments.

5.5   Application example 1 - Transferring floating-point numbers

This application example shows how a floating-point number of data type "real"(TwinCAT) or "float"(Mitsubishi) is transferred from a Beckhoff to a Mitsubishi controller and vice versa via CC-Link.

Because a floating-point variable of data type "float" or "real" occupies four bytes, we need two UINT16 PDO entries in the configuration for sending and receiving.

Fig. 134: PDO entries for sending/receiving

To transfer a floating-point number, a UNION with the structure as shown in the next diagram is used.

Fig. 135: UNION structure

This is instantiated in Main and linked with the applied PDOs of the terminal, once for sending and once for receiving a floating-point number.

BECKHOFF

Fig. 136: Linking with PDOs

If both controllers are logged in and operational, the data exchange looks as follows:
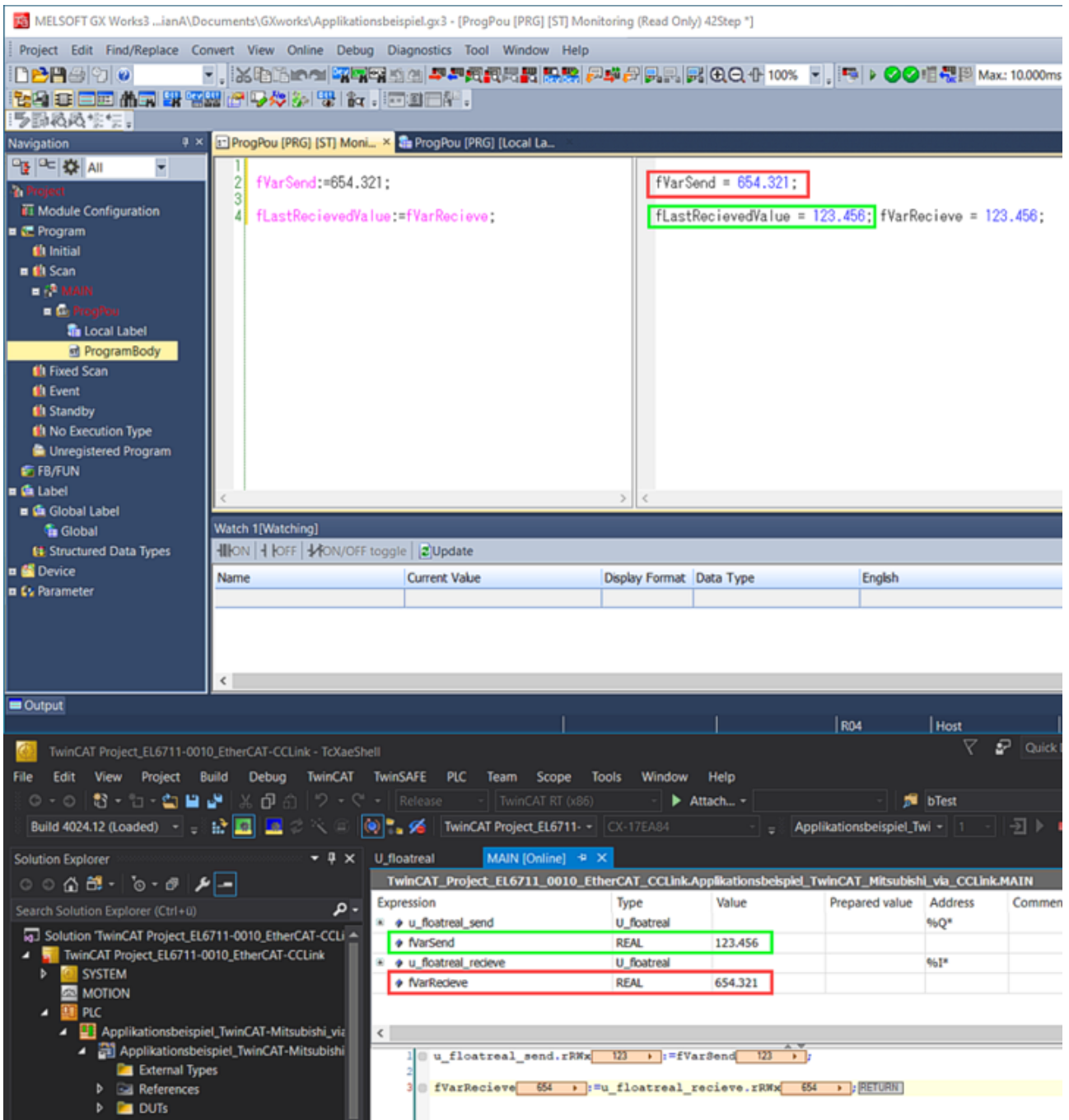
Fig. 137: Data exchange

BECKHOFF

5.6   Application example 2 - Transferring character strings

This application example shows how a character string of data type "String" is transferred from a Beckhoff controller to a Mitsubishi controller and vice versa via CC-Link.

Because a string variable of data type string requires a whole byte for each character and therefore usually more than two bytes are used for an instance, several UINT16 PDO entries are required in the configuration for sending and receiving.
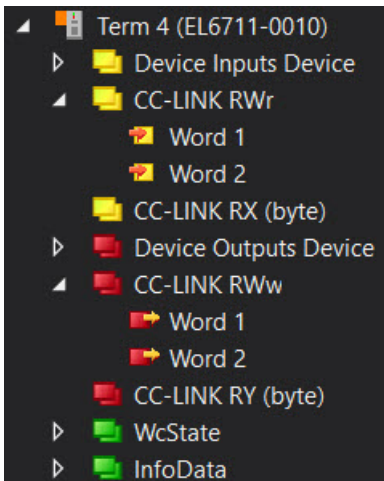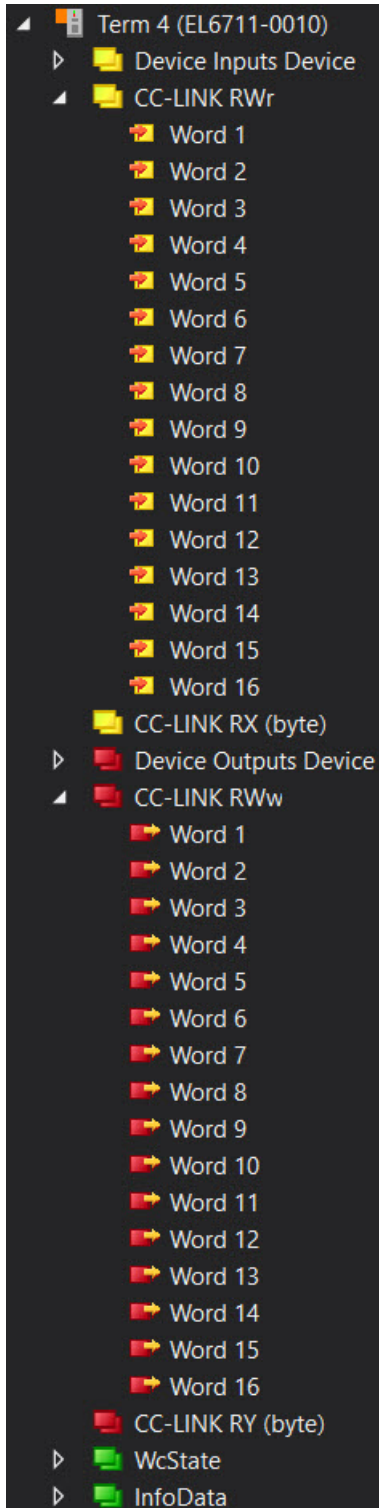

Fig. 138: PDO entries for sending/receiving

To transfer a string, a UNION with the structure as shown in the next diagram is used.

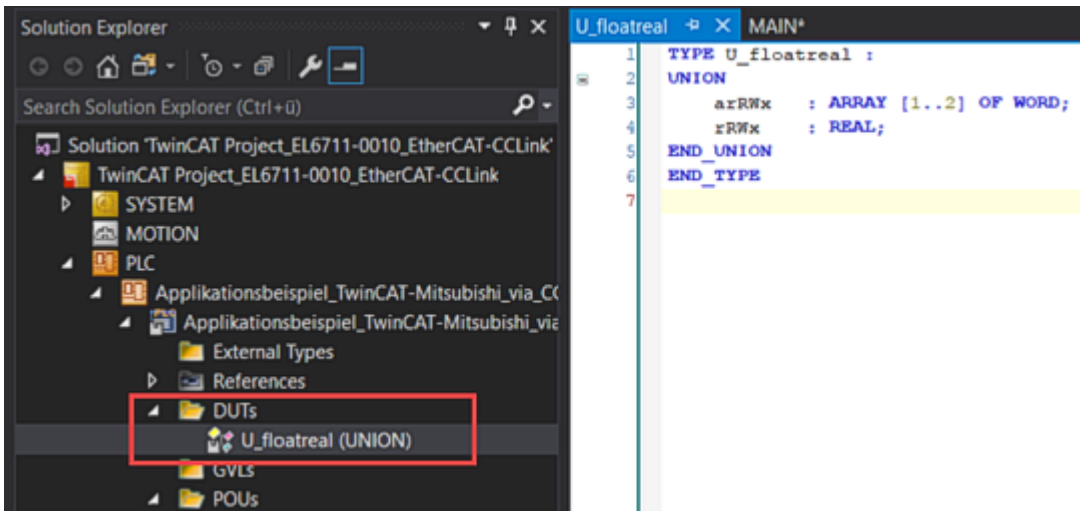
Fig. 139: UNION structure

This is instantiated in Main and linked with the applied PDOs of the terminal, once for sending and once for receiving a string.
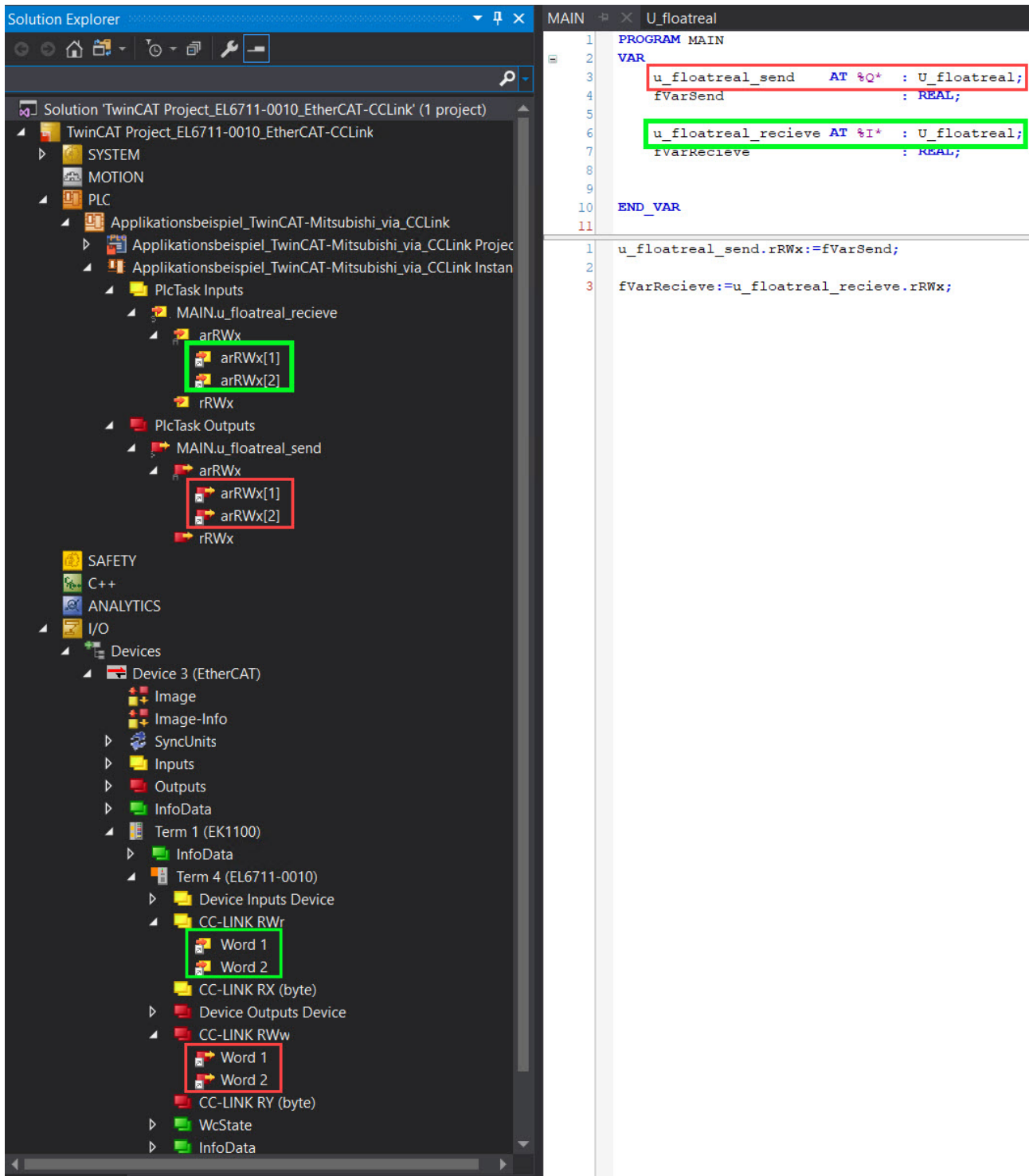
BECKHOFF

Fig. 140: Linking with PDOs

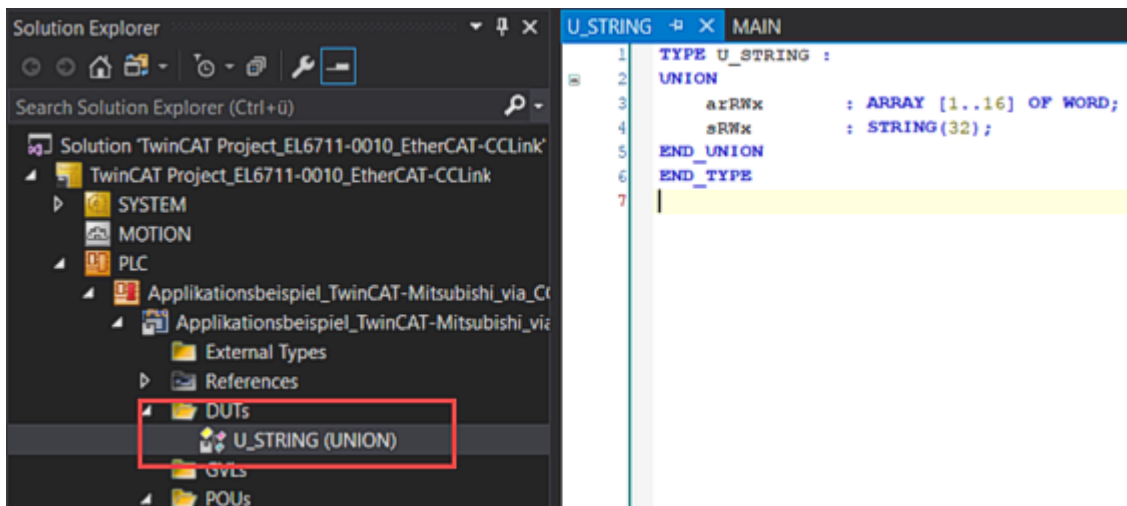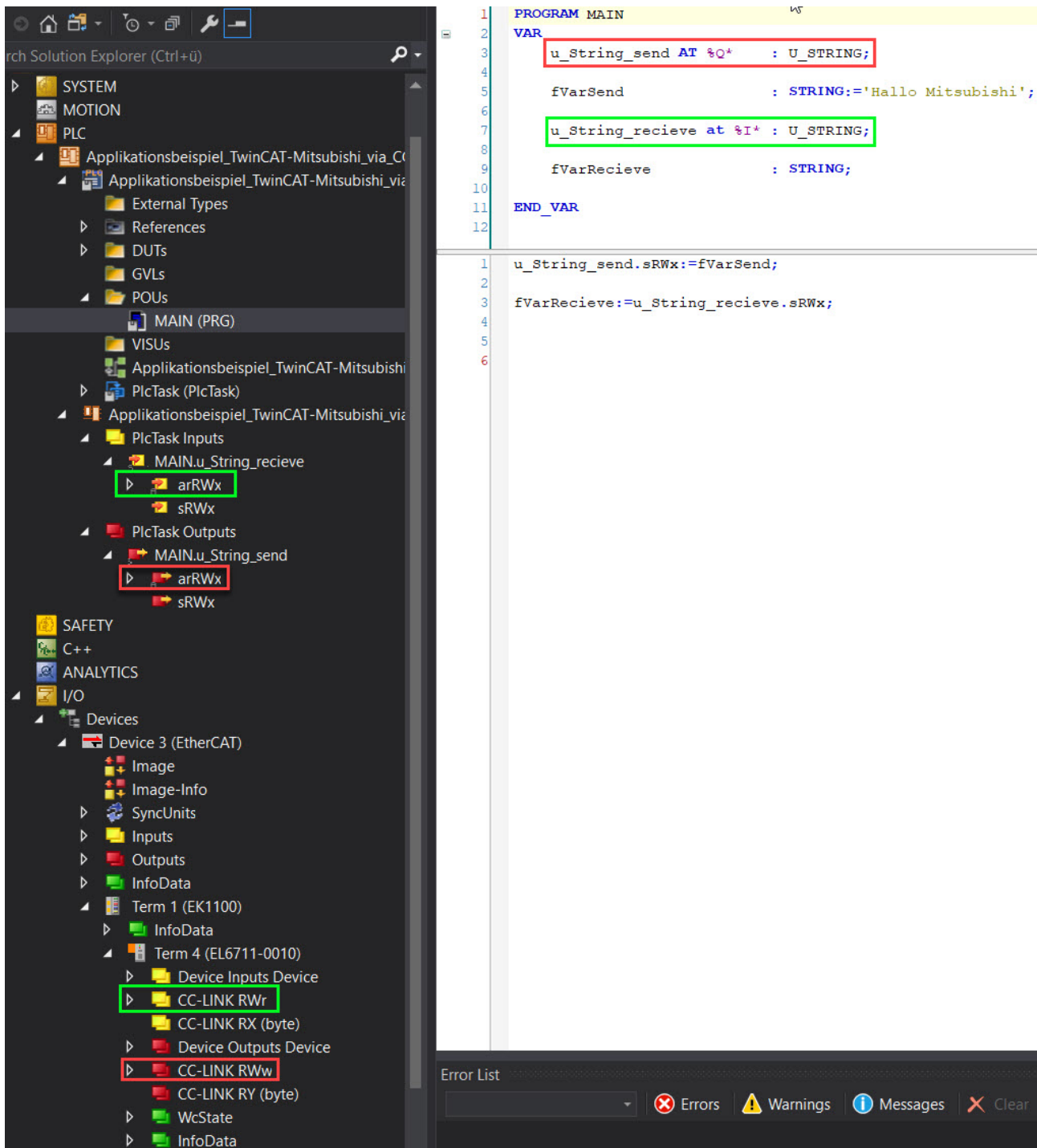If both controllers are logged in and operational, the data exchange looks as follows.
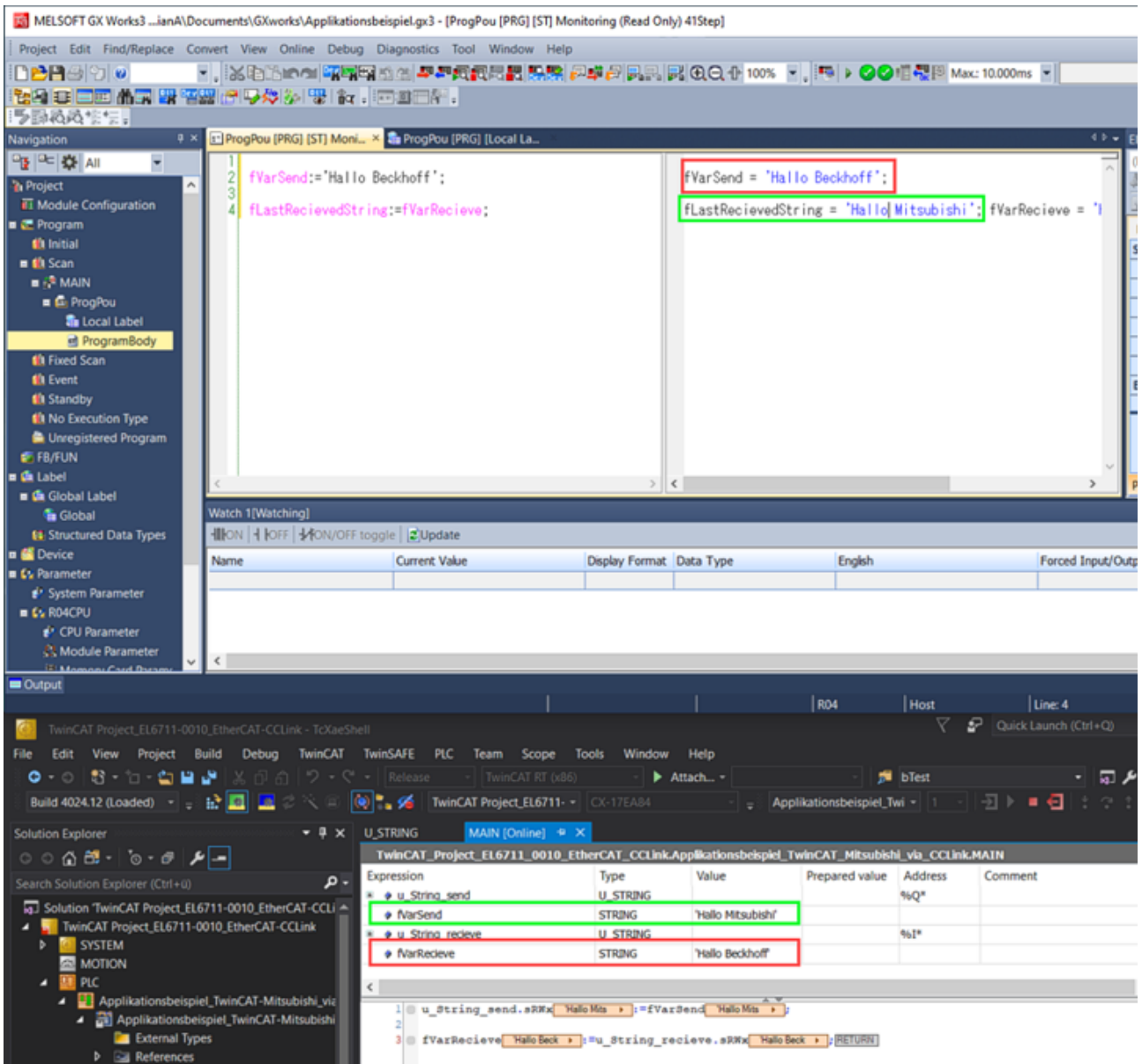
Fig. 141: Data exchange

5.7 Object description and parameterization

ℹ **EtherCAT XML Device Description**

The display matches that of the CoE objects from the EtherCAT XML Device Description. We recommend downloading the latest XML file from the download area of the Beckhoff website and installing it according to installation instructions.

ℹ **Parameterization via the CoE list (CAN over EtherCAT)**

The EtherCAT device is parameterized via the CoE-Online tab [▶ 92] (double-click on the respective object) or via the Process Data tab [▶ 89](allocation of PDOs). Please note the following general CoE notes [▶ 21] when using/manipulating the CoE parameters:

- Keep a startup list if components have to be replaced
- Differentiation between online/offline dictionary, existence of current XML description
- use "CoE reload" for resetting changes

5.7.1 Restore object

Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters [▶ 140]	Restore default parameters	UINT8	RO	0x01 (1_{dec})
1011:01	SubIndex 001	If this object is set to **"0x64616F6C"** in the set value dialog, all backup objects are reset to their delivery state.	UINT32	RW	0x00000000 (0_{dec})

5.7.2 Configuration and diagnostic data

Index F800

Index (hex)	Name	Meaning	Data type	Flags	Default
F800:0	CC-Link settings	Max. Subindex	USINT8	RO	0x6 (6_{dec})
F800:01	Station number	The station address of the CC-Link slave. Values between 1 (default) and 64 are allowed. The selected address must only exist once in the network. The station number "0" is reserved for the CC-Link master	UINT16	RW	0x0 (0_{dec})
F800:02	Baud rate	Possible baud rates 0: 10M (default) 1: 5M 2: 2.5M 3: 625K 4: 156K	UINT16	RW	0x0 (0_{dec})
F800:03	No. of occupied stations	Number of occupied stations (max. 4). Auto (default), One, Two, Three or Four.	UINT16	RW	0x0 (0_{dec})
F800:04	Extended cycle setting	Number of extended cycles. Auto (default), Single, Double, Quadruple, Octuple.	UINT16	RW	0x0 (0_{dec})
F800:05	Minimize	Specification which type of process data transmission should be minimized: "Extended cycles (default) or "No. of occupied station"	UINT16	RW	0x0 (0_{dec})
F800:06	Protocol	If required, CC-Link version 2.0 can be enforced, even if the configured amount of process data corresponds to CC-Link version 1.0	UINT16	RW	0x0 (0_{dec})

Index F900

Index (hex)	Name	Meaning	Data type	Flags	Default
F900:0	CC-Link info data	Max. Subindex	USINT8	RO	0x0B (11$_{dec}$)
F900:01	Current protocol	Currently set CC-Link version. Depending on the process data created or the settings made	UINT16	RO	0x0000 (0$_{dec}$)
F900:02	Occupied stations	Number of occupied stations. Between 1-4 stations can be occupied; this depends on the process data created or the settings made	UINT16	RO	0x0001 (1$_{dec}$)
F900:03	Extended cycles	Number of extended cycles used. "Single, Double, Quadruple or Octuple" can be set; this depends on the applied process data or the settings made	UINT16	RO	0x0001 (1$_{dec}$)
F900:04	Remote I/O size	Selected number of link addresses of the decentralized I/O, depending on the number of process data created	UINT16	RO	0x0000 (0$_{dec}$)
F900:05	Remote register size	Selected number of link addresses of the decentralized registers, depending on the number of process data created	UINT16	RO	0x0000 (0$_{dec}$)
F900:06	Flags	Empty, unused	UINT16	RO	0x0000 (0$_{dec}$)
F900:07	Station info	Low-Byte: station number of the CC-Link slave, High-Byte, Low-Nibble: set baud rate of the CC-Link slave, 0:156 Kbps 1: 625 Kbps 2: 2.5 Mbps 3: 5 Mbps 4: 10 Mbps High-Byte, High-Nibble: number of occupied stations, which are set in the CC-Link interface controller of the terminal	UINT16	RO	0x0000 (0$_{dec}$)
F900:08	CC-Link info 1	Low-Nibble = 1: error in the set station number (station number = 0 or >= 65) Low-Nibble = 2: error in the set baud rate High-Nibble = 1: station number was changed after last startup High-Nibble = 2: baud rate was changed after last startup	UINT16	RO	0x0000 (0$_{dec}$)
F900:09	CC-Link info 2	Version information	UINT16	RO	0x0000 (0$_{dec}$)
F900:0A	Master info	PLC status information 0.0 PLC CPU (0: Stop, 1: Run) 0.1 PLC CPU (0: Normal, 1: Abnormal) 0.2 Refresh (0: Stop, 1: Start) 0.3 Transient (0: No, 1: Yes) 0.4 Transient reception (0: Enable, 1: Disable) 0.5 Transfer (0: Enable, 1: Disable) 0.6: transfer to the master station (0: No, 1: Yes) 0.7: main master station, 1: standby master station	UINT16	RO	0x0000 (0$_{dec}$)
F900:0B	Sequence error counter	This value is incremented in the event of synchronicity problems with the CC-Link master	UDINT32	RO	0x0000 (0$_{dec}$)
F900:0C	System area control	CC-Link System Area	UINT16	RO	0x0000 (0$_{dec}$)
F900:0D	System area status	CC-Link System Area	UINT16	RO	0x0000 (0$_{dec}$)

5.7.3 Standard objects

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default
1000:0	Device type	Device type of the EtherCAT slave: The Lo-Word contains the CoE profile used (5001). The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	0x01681389 (23597961$_{dec}$)

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EL6720

Index 1009 Hardware version

Index (hex)	Name	Meaning	Data type	Flags	Default
1009:0	Hardware version	Hardware version of the EtherCAT slave	STRING	RO	00

Index 100A Software version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware version of the EtherCAT slave	STRING	RO	01

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Information for identifying the slave	UINT8	RO	0x04 (4_{dec})
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2_{dec})
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x0D1C3052 (219951186_{dec})
1018:03	Revision	Revision number of the EtherCAT slave; the low word (bit 0-15) indicates the special terminal number, the high word (bit 16-31) refers to the device description	UINT32	RO	0x00100000 (1048576_{dec})
1018:04	Serial number	Serial number of the EtherCAT slave; the low byte (bit 0-7) of the low word contains the year of production, the high byte (bit 8-15) of the low word contains the week of production, the high word (bit 16-31) is 0	UINT32	RO	0x00000000 (0_{dec})

6   Appendix

6.1   Firmware compatibility

Beckhoff EtherCAT devices are delivered with the latest available firmware version. Compatibility of firmware and hardware is mandatory; not every combination ensures compatibility. The overview below shows the hardware versions on which a firmware can be operated.

Note

- It is recommended to use the newest possible firmware for the respective hardware.
- Beckhoff is not under any obligation to provide customers with free firmware updates for delivered products.

NOTICE

Risk of damage to the device!

Pay attention to the instructions for firmware updates on the separate page [▶ 128]. If a device is placed in BOOTSTRAP mode for a firmware update, it does not check when downloading whether the new firmware is suitable. This can result in damage to the device! Therefore, always make sure that the firmware is suitable for the hardware version!

EL6711-0010			
Hardware (HW)	Firmware (FW)	Revision no.	Release date
00 - 01*	01	EL6711-0010-0017	2021/06
02 - 03	02	EL6711-0010-0018	2021/06
	03	EL6711-0010-0019	2022/03
03	04		2023/04
01 - 04	05		2023/04

*) This is the current compatible firmware/hardware version at the time of the preparing this documentation. Check on the Beckhoff web page whether more up-to-date documentation is available.

6.2 Firmware Update EL/ES/EM/ELM/EPxxxx

This section describes the device update for Beckhoff EtherCAT slaves from the EL/ES, ELM, EM, EK and EP series. A firmware update should only be carried out after consultation with Beckhoff support.

NOTICE

Only use TwinCAT 3 software!

A firmware update of Beckhoff IO devices must only be performed with a TwinCAT 3 installation. It is recommended to build as up-to-date as possible, available for free download on the Beckhoff website.

To update the firmware, TwinCAT can be operated in the so-called FreeRun mode, a paid license is not required.

The device to be updated can usually remain in the installation location, but TwinCAT has to be operated in the FreeRun. Please make sure that EtherCAT communication is trouble-free (no LostFrames etc.).

Other EtherCAT master software, such as the EtherCAT Configurator, should not be used, as they may not support the complexities of updating firmware, EEPROM and other device components.

Storage locations

An EtherCAT slave stores operating data in up to three locations:

* Each EtherCAT slave has a device description, consisting of identity (name, product code), timing specifications, communication settings, etc.
  This device description (ESI; EtherCAT Slave Information) can be downloaded from the Beckhoff website in the download area as a zip file and used in EtherCAT masters for offline configuration, e.g. in TwinCAT.
  Above all, each EtherCAT slave carries its device description (ESI) electronically readable in a local memory chip, the so-called **ESI EEPROM**. When the slave is switched on, this description is loaded locally in the slave and informs it of its communication configuration; on the other hand, the EtherCAT master can identify the slave in this way and, among other things, set up the EtherCAT communication accordingly.

NOTICE

Application-specific writing of the ESI-EEPROM

The ESI is developed by the device manufacturer according to ETG standard and released for the corresponding product.
- Meaning for the ESI file: Modification on the application side (i.e. by the user) is not permitted.
- Meaning for the ESI EEPROM: Even if a writeability is technically given, the ESI parts in the EEPROM and possibly still existing free memory areas must not be changed beyond the normal update process. Especially for cyclic memory processes (operating hours counter etc.), dedicated memory products such as EL6080 or IPC's own NOVRAM must be used.

---

* Depending on functionality and performance EtherCAT slaves have one or several local controllers for processing I/O data. The corresponding program is the so-called **firmware** in *.efw format.

* In some EtherCAT slaves the EtherCAT communication may also be integrated in these controllers. In this case the controller is usually a so-called **FPGA** chip with *.rbf firmware.

Customers can access the data via the EtherCAT fieldbus and its communication mechanisms. Acyclic mailbox communication or register access to the ESC is used for updating or reading of these data.

The TwinCAT System Manager offers mechanisms for programming all three parts with new data, if the slave is set up for this purpose. Generally the slave does not check whether the new data are suitable, i.e. it may no longer be able to operate if the data are unsuitable.

Simplified update by bundle firmware

The update using so-called **bundle firmware** is more convenient: in this case the controller firmware and the ESI description are combined in a *.efw file; during the update both the firmware and the ESI are changed in the terminal. For this to happen it is necessary

* for the firmware to be in a packed format: recognizable by the file name, which also contains the revision number, e.g. ELxxxx-xxxx_REV0016_SW01.efw

---

- for password=1 to be entered in the download dialog. If password=0 (default setting) only the firmware update is carried out, without an ESI update.
- for the device to support this function. The function usually cannot be retrofitted; it is a component of many new developments from year of manufacture 2016.

Following the update, its success should be verified

- ESI/Revision: e.g. by means of an online scan in TwinCAT ConfigMode/FreeRun – this is a convenient way to determine the revision
- Firmware: e.g. by looking in the online CoE of the device

NOTICE
Risk of damage to the device!
✓ Note the following when downloading new device files
a) Firmware downloads to an EtherCAT device must not be interrupted
b) Flawless EtherCAT communication must be ensured. CRC errors or LostFrames must be avoided.
c) The power supply must adequately dimensioned. The signal level must meet the specification.
⇨ In the event of malfunctions during the update process the EtherCAT device may become unusable and require re-commissioning by the manufacturer.

6.2.1   Device description ESI file/XML

NOTICE
Attention regarding update of the ESI description/EEPROM
Some slaves have stored calibration and configuration data from the production in the EEPROM. These are irretrievably overwritten during an update.

The ESI device description is stored locally on the slave and loaded on start-up. Each device description has a unique identifier consisting of slave name (9 characters/digits) and a revision number (4 digits). Each slave configured in the System Manager shows its identifier in the EtherCAT tab:
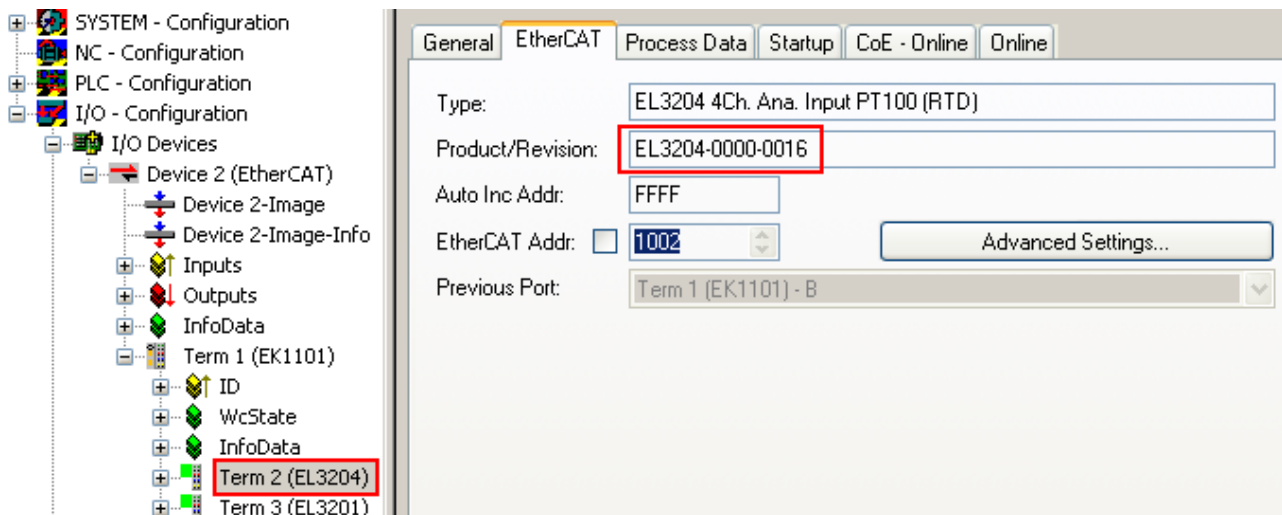

Fig. 142: Device identifier consisting of name EL3204-0000 and revision -0016

The configured identifier must be compatible with the actual device description used as hardware, i.e. the description which the slave has loaded on start-up (in this case EL3204). Normally the configured revision must be the same or lower than that actually present in the terminal network.

For further information on this, please refer to the EtherCAT system documentation.

BECKHOFF

- ● **Update of XML/ESI description**

ⓘ The device revision is closely linked to the firmware and hardware used. Incompatible combinations lead to malfunctions or even final shutdown of the device. Corresponding updates should only be carried out in consultation with Beckhoff support.

Display of ESI slave identifier

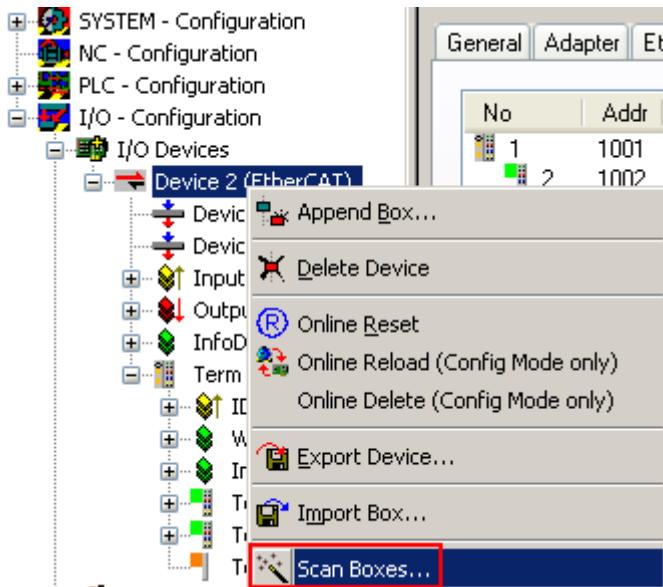The simplest way to ascertain compliance of configured and actual device description is to scan the EtherCAT boxes in TwinCAT mode Config/FreeRun:

Fig. 143: Scan the subordinate field by right-clicking on the EtherCAT device

If the found field matches the configured field, the display shows

Fig. 144: Configuration is identical

otherwise a change dialog appears for entering the actual data in the configuration.
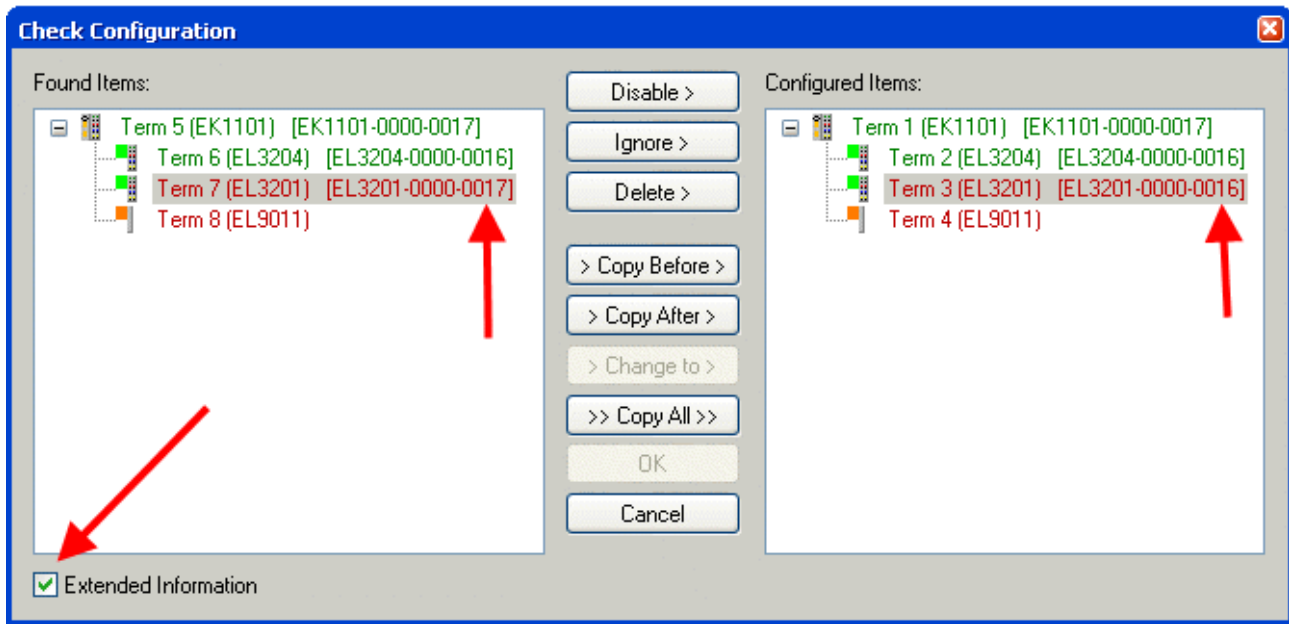
Fig. 145: Change dialog

In this example in Fig. *Change dialog*, an EL3201-0000-**0017** was found, while an EL3201-0000-**0016** was configured. In this case the configuration can be adapted with the *Copy Before* button. The *Extended Information* checkbox must be set in order to display the revision.

Changing the ESI slave identifier

The ESI/EEPROM identifier can be updated as follows under TwinCAT:

- Trouble-free EtherCAT communication must be established with the slave.
- The state of the slave is irrelevant.
- Right-clicking on the slave in the online display opens the *EEPROM Update* dialog, Fig. *EEPROM Update*
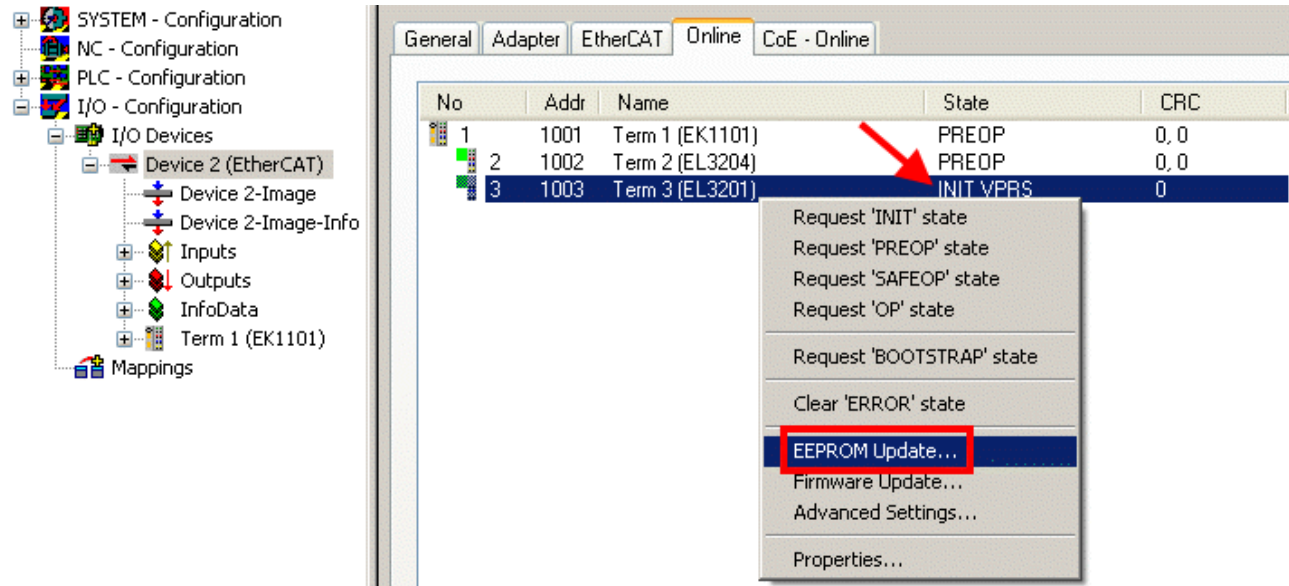
Fig. 146: EEPROM Update

The new ESI description is selected in the following dialog, see Fig. *Selecting the new ESI.* The checkbox *Show Hidden Devices* also displays older, normally hidden versions of a slave.
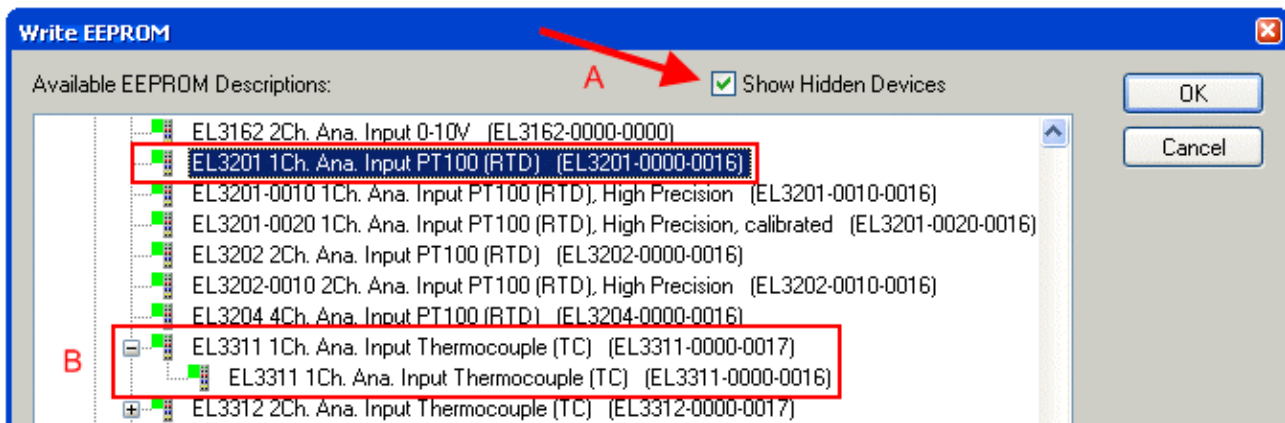
BECKHOFF

Fig. 147: Selecting the new ESI

A progress bar in the System Manager shows the progress. Data are first written, then verified.

> **ⓘ** **The change only takes effect after a restart.**
>
> Most EtherCAT devices read a modified ESI description immediately or after startup from the INIT. Some communication settings such as distributed clocks are only read during power-on. The EtherCAT slave therefore has to be switched off briefly in order for the change to take effect.

6.2.2 Firmware explanation

Determining the firmware version

Determining the version via the System Manager

The TwinCAT System Manager shows the version of the controller firmware if the master can access the slave online. Click on the E-Bus Terminal whose controller firmware you want to check (in the example terminal 2 (EL3204)) and select the tab *CoE Online* (CAN over EtherCAT).

> **ⓘ** **CoE Online and Offline CoE**
>
> Two CoE directories are available:
> • **online**: This is offered in the EtherCAT slave by the controller, if the EtherCAT slave supports this. This CoE directory can only be displayed if a slave is connected and operational.
> • **offline**: The EtherCAT Slave Information ESI/XML may contain the default content of the CoE. This CoE directory can only be displayed if it is included in the ESI (e.g. "Beckhoff EL5xxx.xml").
>
> The Advanced button must be used for switching between the two views.

In Fig. *Display of EL3204 firmware version* the firmware version of the selected EL3204 is shown as 03 in CoE entry 0x100A.
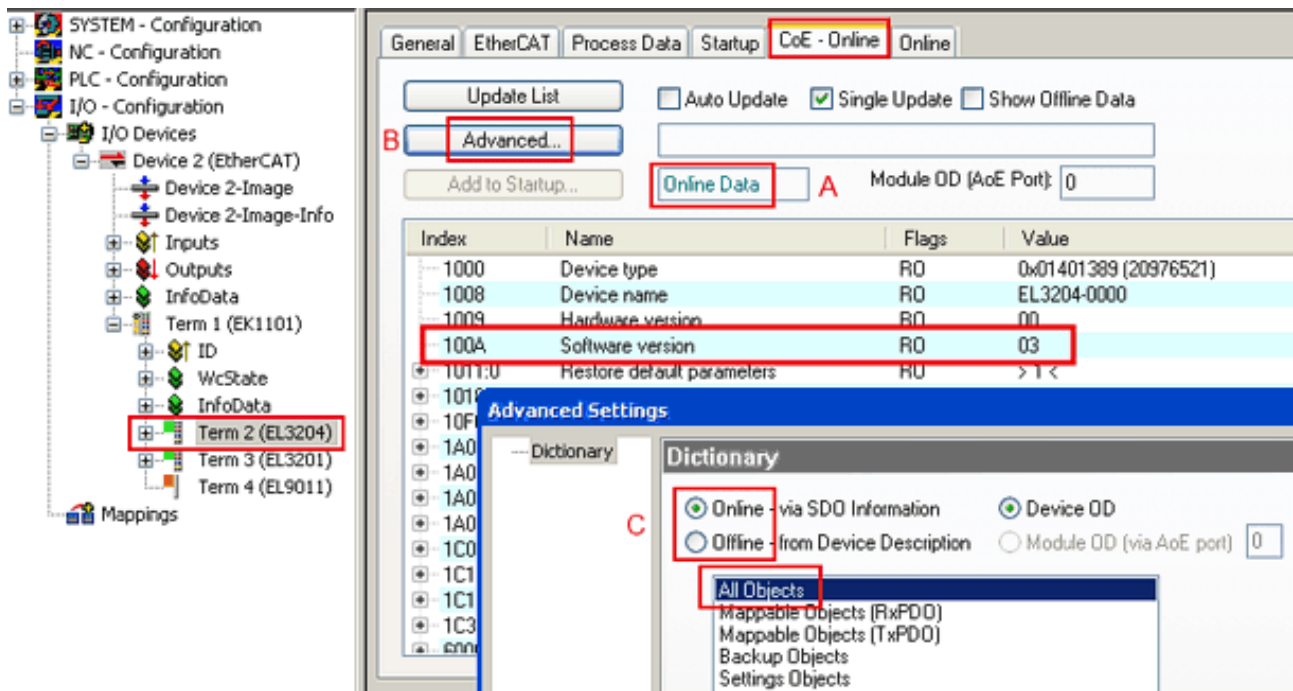
Fig. 148: Display of EL3204 firmware version

In (A) TwinCAT 2.11 shows that the Online CoE directory is currently displayed. If this is not the case, the Online directory can be loaded via the *Online* option in Advanced Settings (B) and double-clicking on *AllObjects*.

6.2.3   Updating controller firmware *.efw

ⓘ **CoE directory**

The Online CoE directory is managed by the controller and stored in a dedicated EEPROM, which is generally not changed during a firmware update.

Switch to the *Online* tab to update the controller firmware of a slave, see Fig. *Firmware Update*.
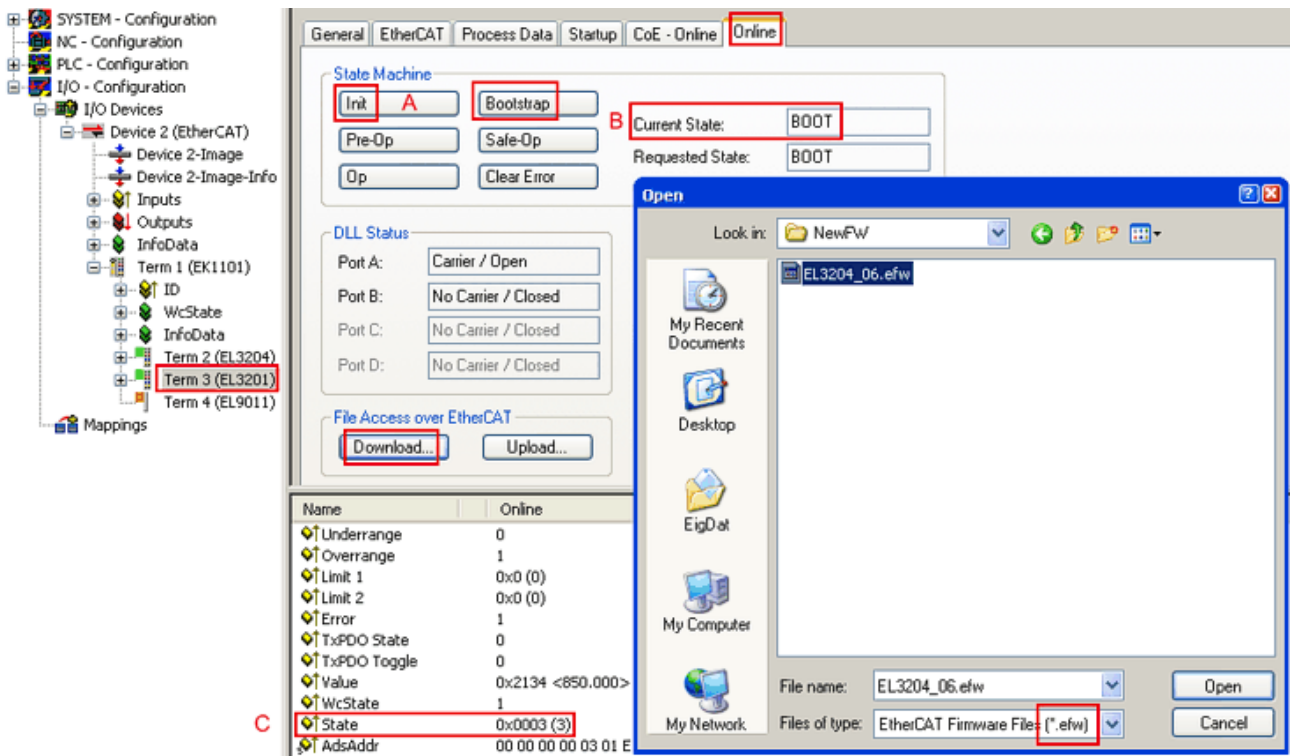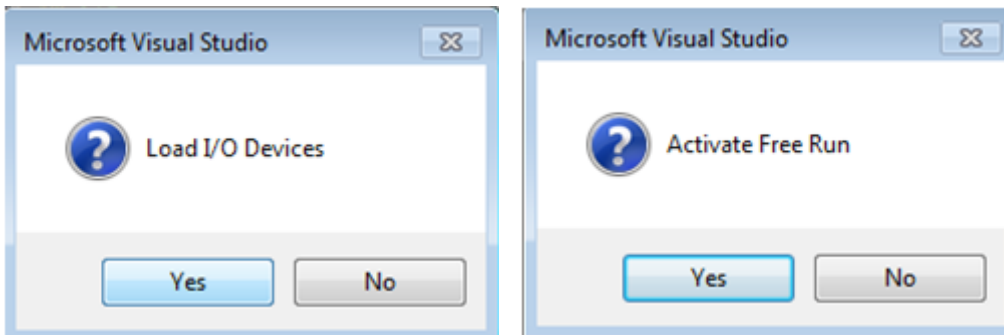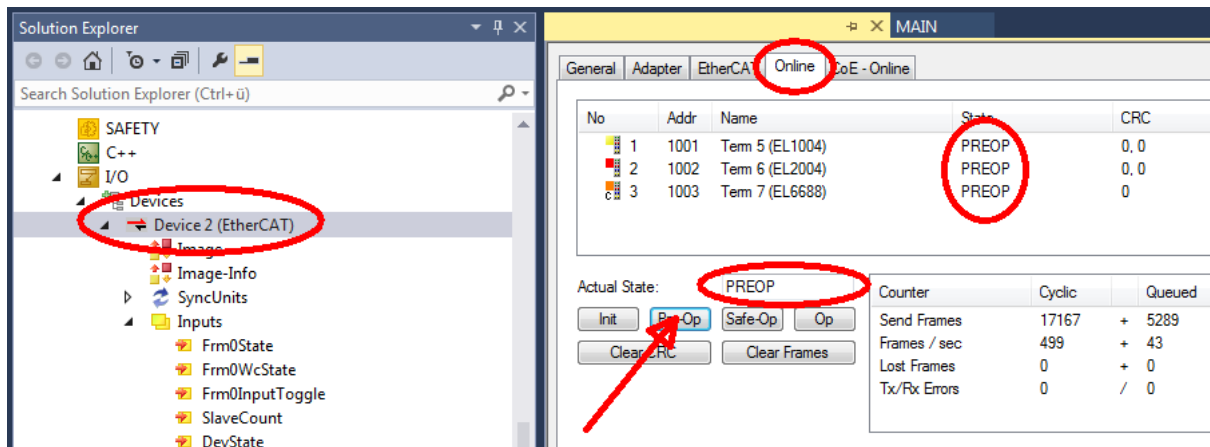
BECKHOFF

Fig. 149: Firmware Update

Proceed as follows, unless instructed otherwise by Beckhoff support. Valid for TwinCAT 2 and 3 as EtherCAT master.

- Switch TwinCAT system to ConfigMode/FreeRun with cycle time >= 1 ms (default in ConfigMode is 4 ms). A FW-Update during real time operation is not recommended.
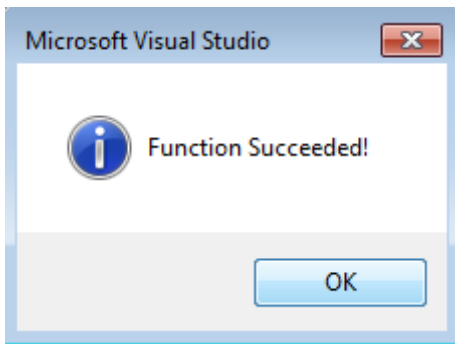

- Switch EtherCAT Master to PreOP

- Switch slave to INIT (A)
- Switch slave to BOOTSTRAP

Version: 1.1

EL6711-0010

- Check the current status (B, C)
- Download the new *efw file (wait until it ends). A password will not be necessary usually.

- After the download switch to INIT, then PreOP
- Switch off the slave briefly (don't pull under voltage!)
- Check within CoE 0x100A, if the FW status was correctly overtaken.

6.2.4   FPGA firmware *.rbf

If an FPGA chip deals with the EtherCAT communication an update may be accomplished via an *.rbf file.

- Controller firmware for processing I/O signals
- FPGA firmware for EtherCAT communication (only for terminals with FPGA)

The firmware version number included in the terminal serial number contains both firmware components. If one of these firmware components is modified this version number is updated.

Determining the version via the System Manager

The TwinCAT System Manager indicates the FPGA firmware version. Click on the Ethernet card of your EtherCAT strand (Device 2 in the example) and select the *Online* tab.

The *Reg:0002* column indicates the firmware version of the individual EtherCAT devices in hexadecimal and decimal representation.
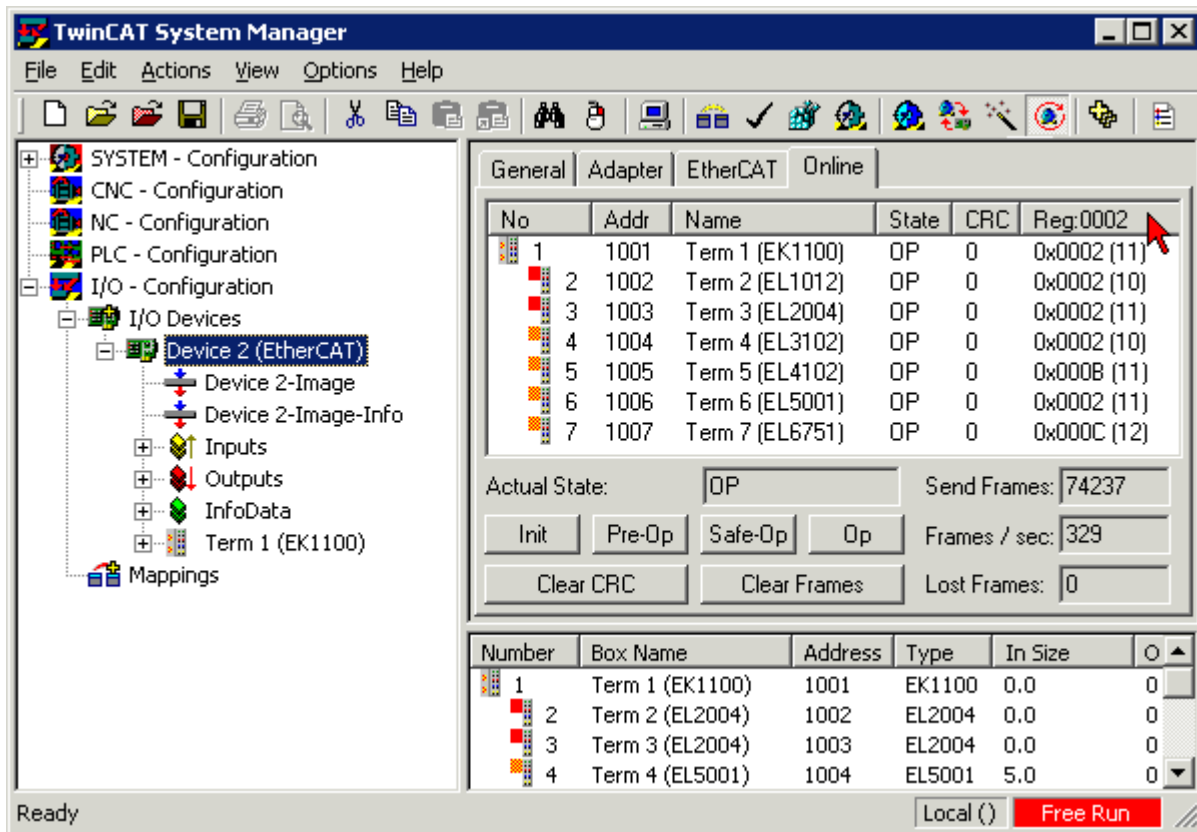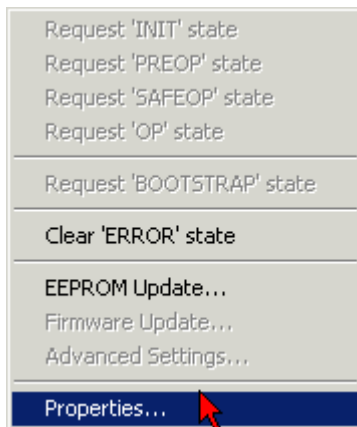
BECKHOFF

Fig. 150: FPGA firmware version definition

If the column *Reg:0002* is not displayed, right-click the table header and select *Properties* in the context menu.

Fig. 151: Context menu *Properties*

The *Advanced Settings* dialog appears where the columns to be displayed can be selected. Under *Diagnosis/*Online View select the *'0002 ETxxxx Build'* check box in order to activate the FPGA firmware version display.
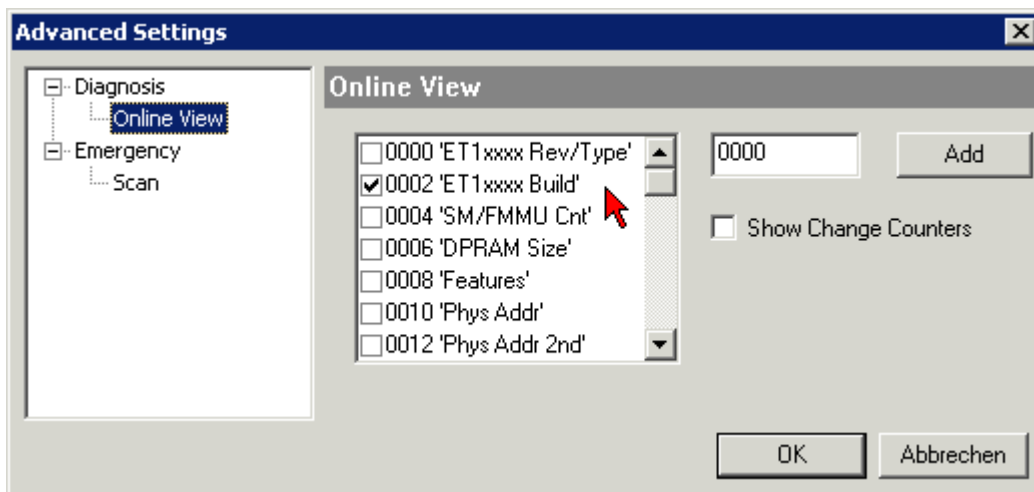
Fig. 152: Dialog *Advanced Settings*

Update

For updating the FPGA firmware

- of an EtherCAT coupler the coupler must have FPGA firmware version 11 or higher;
- of an E-Bus Terminal the terminal must have FPGA firmware version 10 or higher.

Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

- Switch TwinCAT system to ConfigMode/FreeRun with cycle time >= 1 ms (default in ConfigMode is 4 ms). A FW-Update during real time operation is not recommended.

BECKHOFF

- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and
click the *Advanced Settings* button in the *EtherCAT* tab:
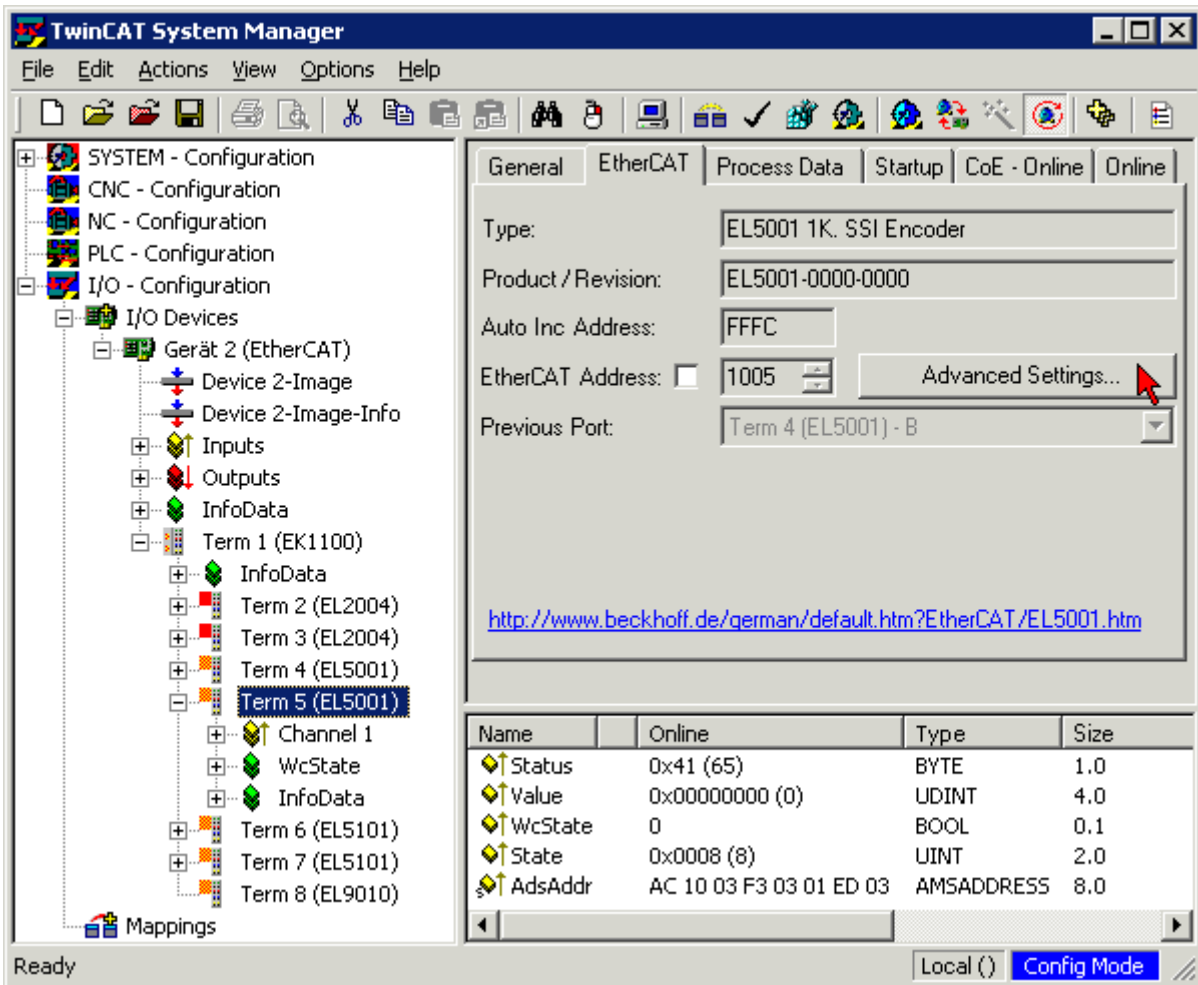
- The *Advanced Settings* dialog appears. Under *ESC Access/E²PROM*/FPGA click on *Write FPGA* button:
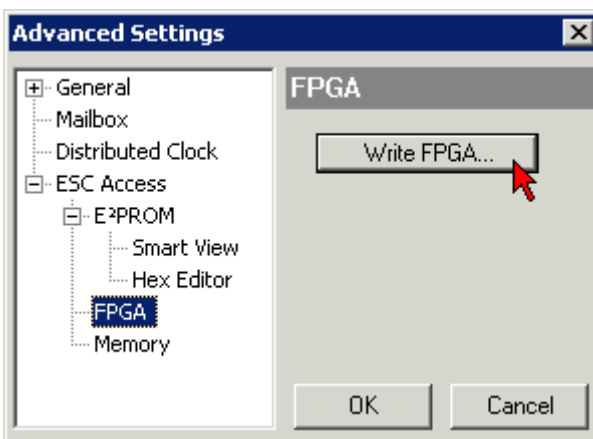
- Select the file (*.rbf) with the new FPGA firmware, and transfer it to the EtherCAT device:
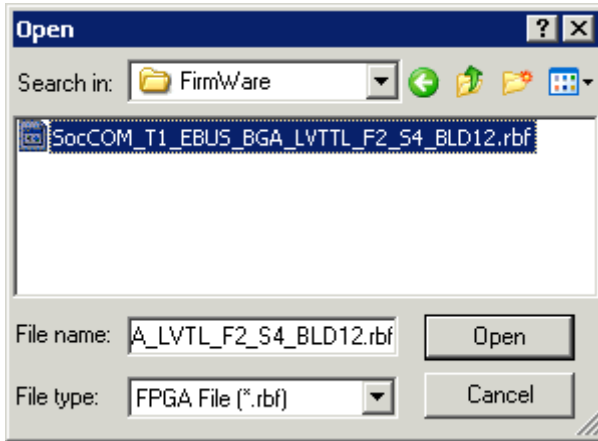
- Wait until download ends
- Switch slave current less for a short time (don't pull under voltage!). In order to activate the new FPGA firmware a restart (switching the power supply off and on again) of the EtherCAT device is required.
- Check the new FPGA status

NOTICE
Risk of damage to the device!
A download of firmware to an EtherCAT device must not be interrupted in any case! If you interrupt this process by switching off power supply or disconnecting the Ethernet link, the EtherCAT device can only be recommissioned by the manufacturer!

6.2.5   Simultaneous updating of several EtherCAT devices

The firmware and ESI descriptions of several devices can be updated simultaneously, provided the devices have the same firmware file/ESI.
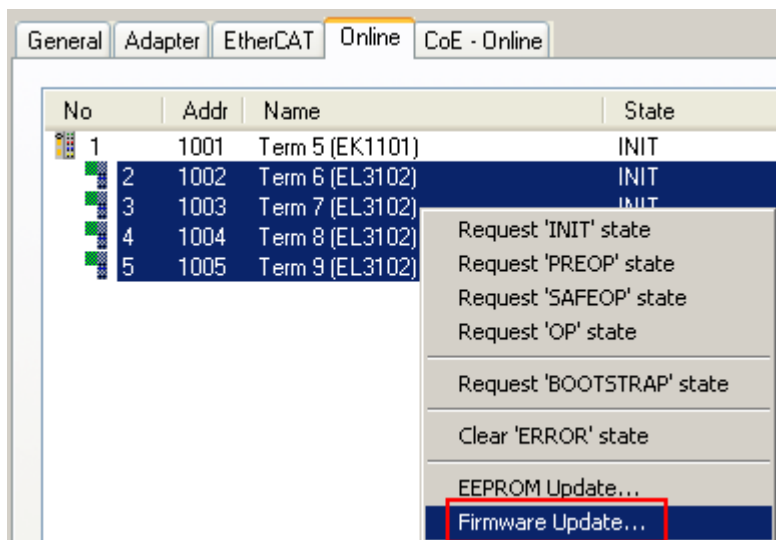

Fig. 153: Multiple selection and firmware update

Select the required slaves and carry out the firmware update in BOOTSTRAP mode as described above.

BECKHOFF

6.3   Restoring the delivery state

To restore the delivery state (factory settings) of CoE objects for EtherCAT devices ("slaves"), the CoE object *Restore default parameters*, SubIndex 001 can be used via EtherCAT master (e.g. TwinCAT) (see Fig. *Selecting the Restore default parameters PDO*).
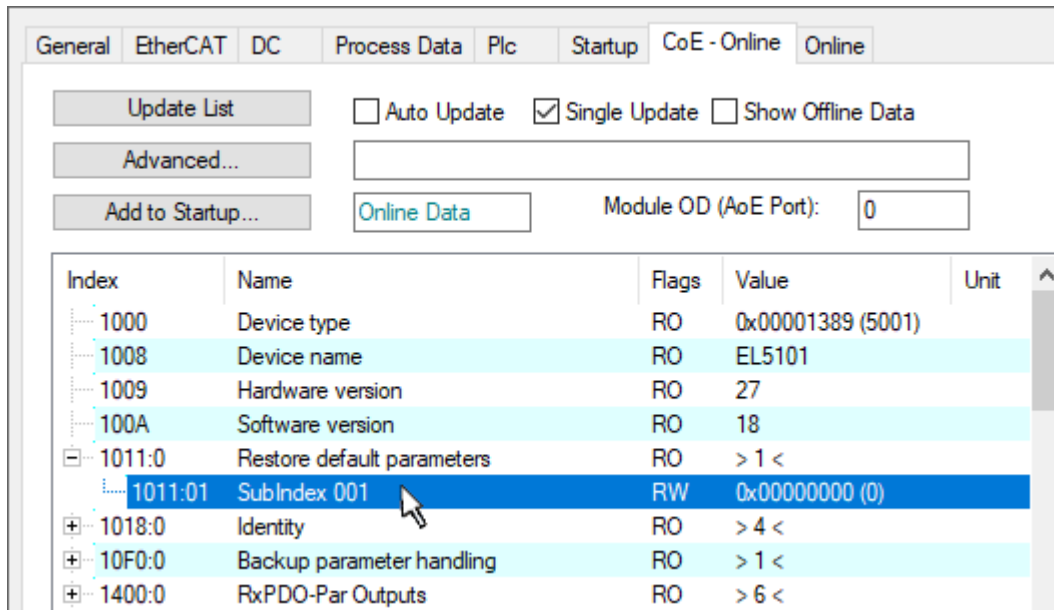

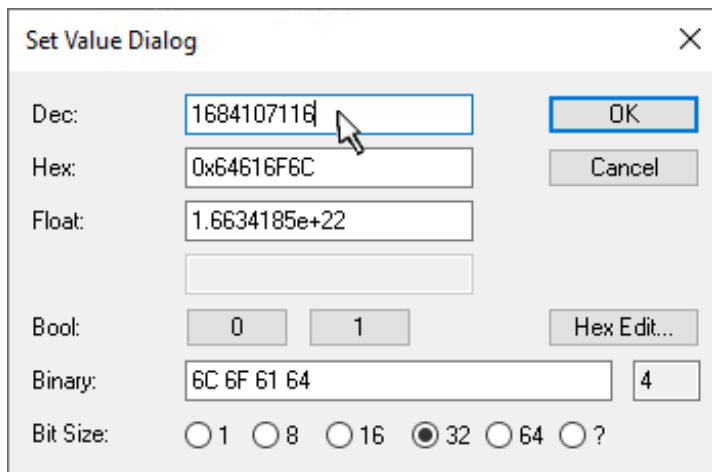Fig. 154: Selecting the *Restore default parameters* PDO

Fig. 155: Entering a restore value in the Set Value dialog

Double-click on *SubIndex 001* to enter the Set Value dialog. Enter the reset value **1684107116** in field *Dec* or the value **0x64616F6C** in field *Hex* (ASCII: "load") and confirm with *OK* (Fig. *Entering a restore value in the Set Value dialog*).

- All changeable entries in the slave are reset to the default values.
- The values can only be successfully restored if the reset is directly applied to the online CoE, i.e. to the slave. No values can be changed in the offline CoE.
- TwinCAT must be in the RUN or CONFIG/Freerun state for this; that means EtherCAT data exchange takes place. Ensure error-free EtherCAT transmission.
- No separate confirmation takes place due to the reset. A changeable object can be manipulated beforehand for the purposes of checking.
- This reset procedure can also be adopted as the first entry in the startup list of the slave, e.g. in the state transition PREOP->SAFEOP or, as in Fig. *CoE reset as a startup entry*, in SAFEOP->OP.

All backup objects are reset to the delivery state.

● **Alternative restore value**

i In some older terminals (FW creation approx. before 2007) the backup objects can be switched with
an alternative restore value: Decimal value: 1819238756, Hexadecimal value: 0x6C6F6164.

An incorrect entry for the restore value has no effect.

6.4 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Support

The Beckhoff Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963 157
e-mail:	support@beckhoff.com
web:	www.beckhoff.com/support

Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963 460
e-mail:	service@beckhoff.com
web:	www.beckhoff.com/service

Headquarters Germany

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963 0
e-mail:	info@beckhoff.com
web:	www.beckhoff.com

More Information:
www.beckhoff.com/EL6711-0010