



**Documentation**

**EL6720**

**Lightbus Master Terminal**

**Version: 2.2**  
**Date: 2017-05-17**

**BECKHOFF**



# Table of contents

<b>1</b>	<b>Foreword</b> .....	<b>5</b>
1.1	Notes on the documentation .....	5
1.2	Safety instructions .....	6
1.3	Documentation issue status .....	7
1.4	Version identification of EtherCAT devices .....	7
<b>2</b>	<b>Product overview</b> .....	<b>12</b>
2.1	Introduction .....	12
2.2	Technical data .....	13
2.3	Lightbus System Description .....	14
<b>3</b>	<b>Basics communication</b> .....	<b>16</b>
3.1	EtherCAT basics .....	16
3.2	EtherCAT State Machine .....	16
3.3	CoE Interface .....	17
3.4	Distributed Clock .....	22
<b>4</b>	<b>Mounting and wiring</b> .....	<b>23</b>
4.1	Instructions for ESD protection .....	23
4.2	Mounting and demounting - terminals with front unlocking .....	23
4.3	Recommended mounting rails .....	25
4.4	Mounting of Passive Terminals .....	26
4.5	ATEX - Special conditions (standard temperature range) .....	27
4.6	ATEX Documentation .....	28
4.7	Installation positions .....	28
4.8	LEDs .....	30
<b>5</b>	<b>Commissioning</b> .....	<b>31</b>
5.1	TwinCAT Quick Start .....	31
5.1.1	TwinCAT 2 .....	33
5.1.2	TwinCAT 3 .....	43
5.2	TwinCAT Development Environment .....	55
5.2.1	Installation of the TwinCAT real-time driver .....	55
5.2.2	Notes regarding ESI device description .....	61
5.2.3	TwinCAT ESI Updater .....	65
5.2.4	Distinction between Online and Offline .....	65
5.2.5	OFFLINE configuration creation .....	66
5.2.6	ONLINE configuration creation .....	71
5.2.7	EtherCAT subscriber configuration .....	79
5.3	General Notes - EtherCAT Slave Application .....	89
5.4	Basic function principles .....	97
5.5	Preparing a terminal configuration with EL6720 .....	101
5.6	Object description and parameterization .....	105
5.6.1	Restore object .....	105
5.6.2	Configuration and diagnostic data .....	106
5.6.3	Standard objects .....	107
<b>6</b>	<b>Appendix</b> .....	<b>109</b>
6.1	UL notice .....	109
6.2	Firmware compatibility .....	110

---

6.3	Firmware Update EL/ES/EM/EPxxxx.....	110
6.4	Support and Service .....	120

# 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.

It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

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®, EtherCAT®, Safety over EtherCAT®, TwinSAFE®, XFC® and XTS® 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, DE102004044764, DE102007017835 with corresponding applications or registrations in various other countries.

The TwinCAT Technology is covered, including but not limited to the following patent applications and patents: EP0851348, US6167425 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.

## 1.2 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.

### Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

 <b>DANGER</b>	<p><b>Serious risk of injury!</b> Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.</p>
 <b>WARNING</b>	<p><b>Risk of injury!</b> Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.</p>
 <b>CAUTION</b>	<p><b>Personal injuries!</b> Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.</p>
 <b>Attention</b>	<p><b>Damage to the environment or devices</b> Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.</p>
 <b>Note</b>	<p><b>Tip or pointer</b> This symbol indicates information that contributes to better understanding.</p>

### 1.3 Documentation issue status

Version	Comment
2.2	<ul style="list-style-type: none"> <li>• Update chapter "Technical data"</li> <li>• Addenda chapter "Instructions for ESD protection"</li> <li>• Chapter "ATEX - Special conditions" replaced with chapter "ATEX - Special conditions (standard temperature range)"</li> </ul>
2.1	<ul style="list-style-type: none"> <li>• Update chapter "Notes on the documentation"</li> <li>• Correction of Technical data</li> <li>• Update chapter "TwinCAT 2.1x" -&gt; "TwinCAT Development Environment" and "TwinCAT Quick Start"</li> <li>• Update revision status</li> </ul>
2.0	<ul style="list-style-type: none"> <li>• Migration</li> </ul>
1.0	<ul style="list-style-type: none"> <li>• 1. Publication</li> </ul>
0.3	<ul style="list-style-type: none"> <li>• Amendments</li> </ul>
0.2	<ul style="list-style-type: none"> <li>• Amendments</li> </ul>
0.1	<ul style="list-style-type: none"> <li>• provisional documentation for EL6720</li> </ul>

### 1.4 Version identification of EtherCAT devices

#### 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.

### Identification number

Beckhoff EtherCAT devices from the different lines have different kinds of identification numbers:

#### Production lot/batch number/serial number/date code/D number

The serial number 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

Ser. no.: 12063A02: 12 - production week 12 06 - production year 2006 3A - firmware version 3A 02 - hardware version 02

Exceptions can occur in the **IP67 area**, where the following syntax can be used (see respective device documentation):

Syntax: D ww yy x y z u

D - prefix designation  
ww - calendar week  
yy - year  
x - firmware version of the bus PCB  
y - hardware version of the bus PCB  
z - firmware version of the I/O PCB  
u - hardware version of the I/O PCB

Example: D.22081501 calendar week 22 of the year 2008 firmware version of bus PCB: 1 hardware version of bus PCB: 5 firmware version of I/O PCB: 0 (no firmware necessary for this PCB) hardware version of I/O PCB: 1

#### Unique serial number/ID, ID number

In addition, in some series each individual module has its own unique serial number.

See also the further documentation in the area

- IP67: [EtherCAT Box](#)
- Safety: [TwinSafe](#)
- Terminals with factory calibration certificate and other measuring terminals

**Examples of markings**

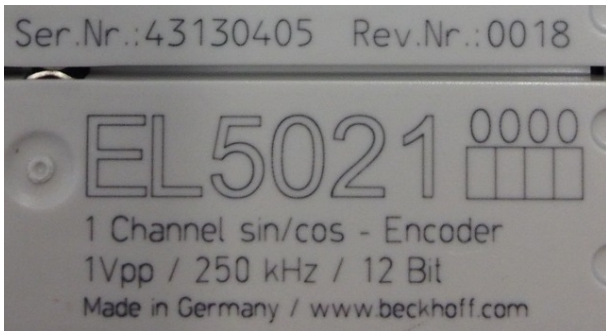


Fig. 1: EL5021 EL terminal, standard IP20 IO device with batch number and revision ID (since 2014/01)



Fig. 2: EK1100 EtherCAT coupler, standard IP20 IO device with batch number



Fig. 3: CU2016 switch with batch number



Fig. 4: EL3202-0020 with batch numbers 26131006 and unique ID-number 204418

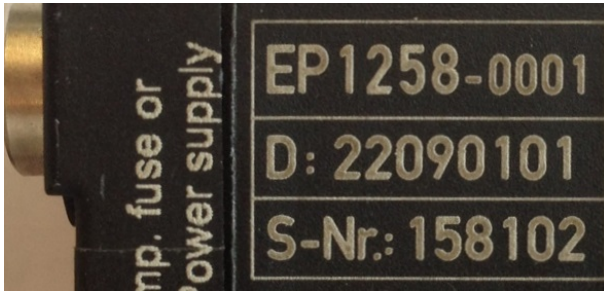


Fig. 5: EP1258-00001 IP67 EtherCAT Box with batch number 22090101 and unique serial number 158102



Fig. 6: EP1908-0002 IP67 EtherCAT Safety Box with batch number 071201FF and unique serial number 00346070



Fig. 7: EL2904 IP20 safety terminal with batch number/date code 50110302 and unique serial number 00331701



Fig. 8: ELM3604-0002 terminal with ID number (QR code) 100001051 and unique serial number 44160201

## 2 Product overview

### 2.1 Introduction

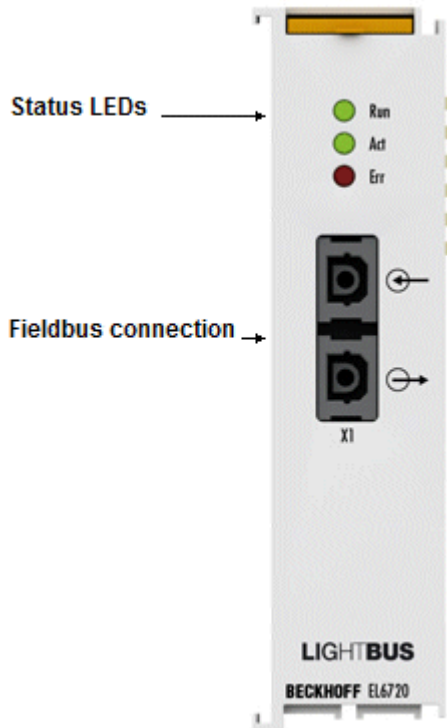


Fig. 9: EL6720

#### Lightbus master terminal

The EL6720 Lightbus master terminal is used, like the Beckhoff FC2001 Lightbus PCI card, for connecting Lightbus devices to a Lightbus circuit. If the connection is made via EtherCAT instead of a PC-internal card, no PCI slots are required in the PC.

The terminal can run the Lightbus protocol with all features and in the EtherCAT Terminal network enables the integration of any Lightbus slaves. Any number of EL6720 devices can be used in an EtherCAT system.

The terminal has a powerful protocol implementation with many features:

- fast cycle times up to 100  $\mu$ s are possible
- process data communication optionally free running or synchronous
- high-performance parameter and diagnostics interface (ADS)
- max. 3 CDLs
- same diagnostic features as FC200x (accessible via the CoE in the EL6720)

#### Quick links

- [EtherCAT basics](#)
- [Basic function principles \[► 97\]](#)
- [Creation of the TwinCAT EL6720 configuration \[► 101\]](#)
- [CoE object description and parameterization \[► 105\]](#)

## 2.2 Technical data

Technical data	EL6720
Fieldbus	LIGHTBUS
Function	Master
Number of fieldbus channels	1
Data transfer rates	2,5 Mbaud
Bus interface	2 x standard fiber optic connector Z1000 (plastic fiber), Z1010 (HCS fiber)
Communication	3 priority controlled logical communications channels (CDL)
Bus nodes	Max. 254 bus devices with a max. of 65,280 I/O points per fieldbus connection
Hardware diagnostics	Status LEDs
Supply voltage for electronics	via the E-bus
Current consumption via E-bus	typ. 240 mA
Electrical isolation	500 V (E-bus/field voltage)
Configuration	via TwinCAT System Manager
Weight	approx. 70 g
Permissible ambient temperature range during operation	0°C ... + 55°C
Permissible ambient temperature range during storage	-25°C ... + 85°C
Permissible relative humidity	95 %, no condensation
Dimensions (W x H x D)	approx. 26 mm x 100 mm x 52 mm (width aligned: 23 mm)
<u>Mounting</u> [ <a href="#">▶ 23</a> ]	on 35 mm mounting rail according to EN 60715, see <u>note</u> [ <a href="#">▶ 25</a> ]!
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
Protection class	IP20
Installation position	variable
Approval	CE cULus [ <a href="#">▶ 109</a> ] ATEX [ <a href="#">▶ 27</a> ]

## 2.3 Lightbus System Description



**Note**

### System description for FC200x / EL6720

The following system description is applicable both for the Beckhoff Lightbus PCI interface card FC200x and for the Lightbus master terminal EL6720.

The Lightbus consists of the Lightbus PCI interface card (in short Lightbus card) and an optical fiber-based fieldbus.

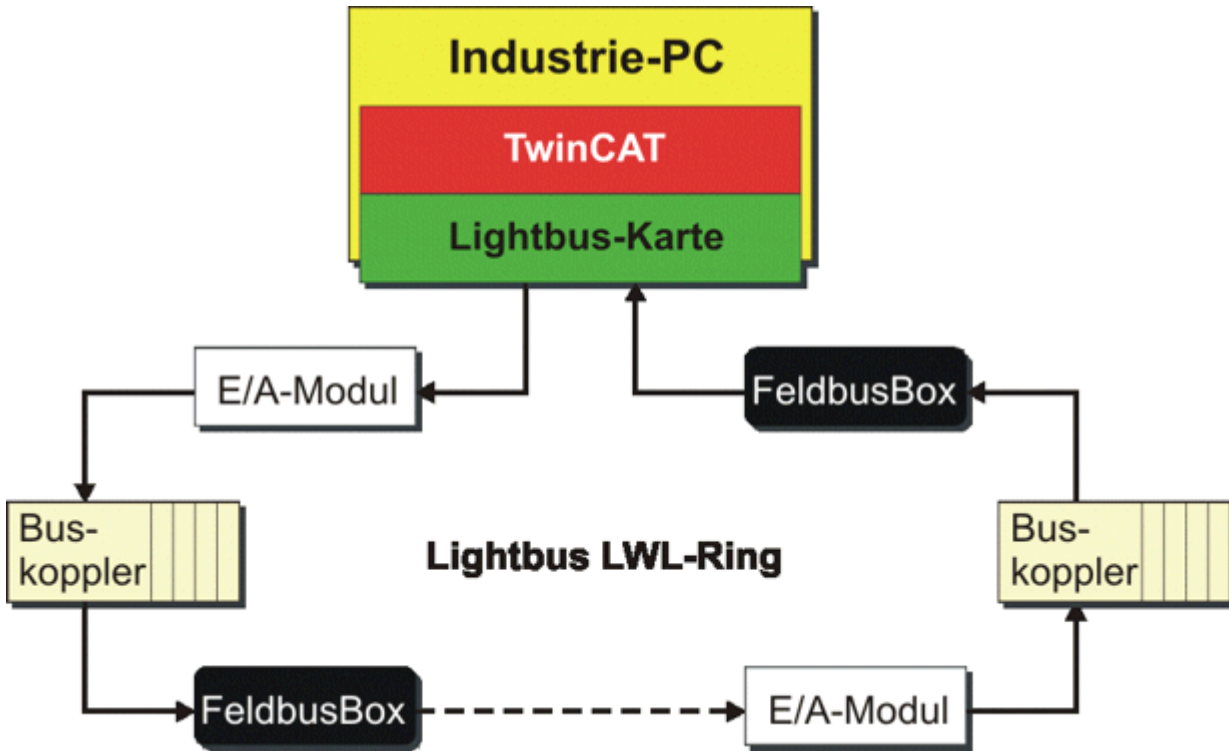


Fig. 10: Lightbus LWL-Ring

The coupling of the Lightbus with the PC is realized on the Lightbus card via a Dual Ported RAM (DPRAM).

Various I/O modules (Bus Coupler, Bus Terminals, Fieldbus Boxes etc.) are available for input and output of the process images, which are connected through the Lightbus in a ring structure. The use of optical fibers results in low susceptibility to interference and a high transfer rate of 2.5 MBaud. Any errors occurring in the optical fiber ring are detected by the Lightbus card and reported to the PC. TwinCAT provides comprehensive diagnostic functions that support fast detection and correction of any errors.

A communication protocol that is optimized for speed and simplicity is specified for the data transfer between the Lightbus card and the I/O modules. The Lightbus card controls the communication on the optical fiber ring. It sends telegrams that pass through the I/O modules in the optical fiber ring and are then received and checked back at the Lightbus card.

A telegram consists of a telegram frame and the telegram content.

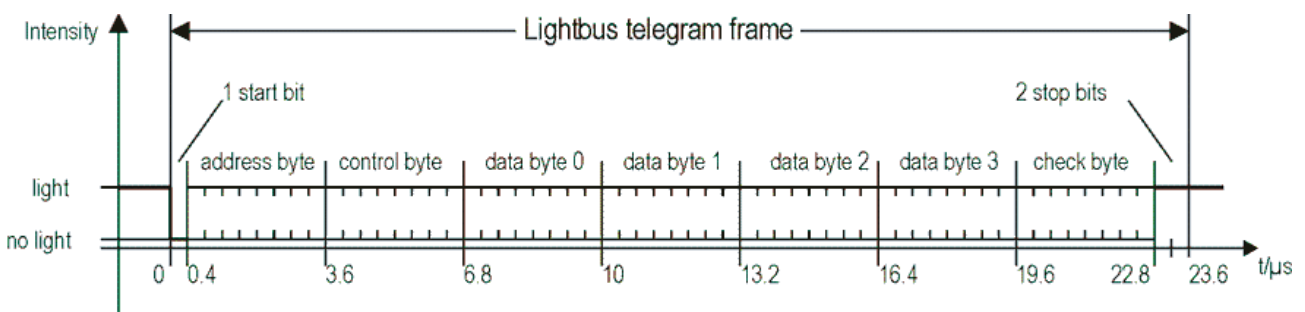


Fig. 11: Lightbus telegram structure

The telegram frame is required for serial data communication and consists of 1 start bit, 6 CRC check bits and 2 stop bits. It is generated and checked independently by the hardware of the Lightbus card. No software support is required.

The telegram content is essentially organized byte-wise. AD0 to AD7 form the address field, through which up to 254 I/O modules can be addressed (the addresses 0x00 and 0x0FF are reserved). The bits CR0 to CR3 specify the telegram type. The following functions can be specified in the telegram:

**Functions**

CR3	CR2	CR1	CR0	Function	Description
0	0	0	0	Read	The addressed module inserts the input information into data fields D0 to D3.
0	0	0	1	Read and write	The addressed module inserts the input information into data fields D0 to D3 and receives the output information.
0	0	1	0	Address initialization	The addressed module receives the content of D0 as the module address and sets D0 to 0.
0	0	1	1	RAM	Special telegram type for the BK2000 Bus Coupler.
0	1	0	0	Address check and count	Each passing module increases the content of D0 by 1. The addressed module moves the content of D0 to D3.
1	0	0	1	Low intensity	The addressed module reduces the transmission intensity by 20%.
1	0	1	1	Broadcast	Special telegram type for the BK2000 Bus Coupler.

The bytes D0 to D3 contain the actual user data. The *control field* specifies the processing of these user data. The last byte in the telegram contains 6 bits for forming a CRC checksum and 2 reserved bits. A user data length of 50 bits thus achieves a hamming distance of d=3.

The Lightbus consists of a physical ring, which can be split into up to 8 logical rings for processing the process image. A logical ring only operates for selected I/O modules, which are specified via communication description lists (CDLs).

The Lightbus card makes the process image available to the PC via the DPRAM. The DPRAM is divided into three areas:

- Data:  
Input, output and flags
- Communication:  
Initialization, testing, analysis and configuration of the Beckhoff Lightbus
- Process control:  
Updating the Process Image

The DPRAM of the Lightbus cards requires 4 kB per channel in the address space of the PC.

## 3 Basics communication

### 3.1 EtherCAT basics

Please refer to the chapter [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.

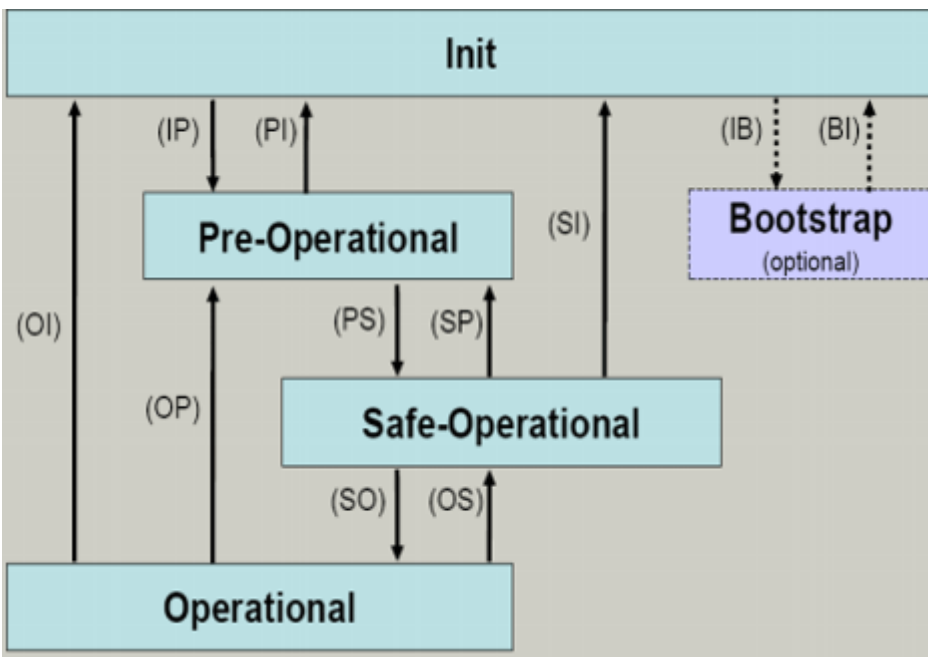


Fig. 12: States of the EtherCAT State Machine

#### 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.



#### Note

#### 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 (CANopen 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 2 levels via hexadecimal numbering: (main)index, followed by subindex. The value ranges are

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


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

The relevant ranges for EtherCAT fieldbus users are:

- 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: In some EtherCAT devices the channel parameters are stored here (as an alternative to the 0x8000 range).
- 0x6000: Input PDOs ("input" from the perspective of the EtherCAT master)
- 0x7000: Output PDOs ("output" from the perspective of the EtherCAT master)

 <b>Note</b>	<p><b>Availability</b></p> <p>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.</p>
--	---

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:

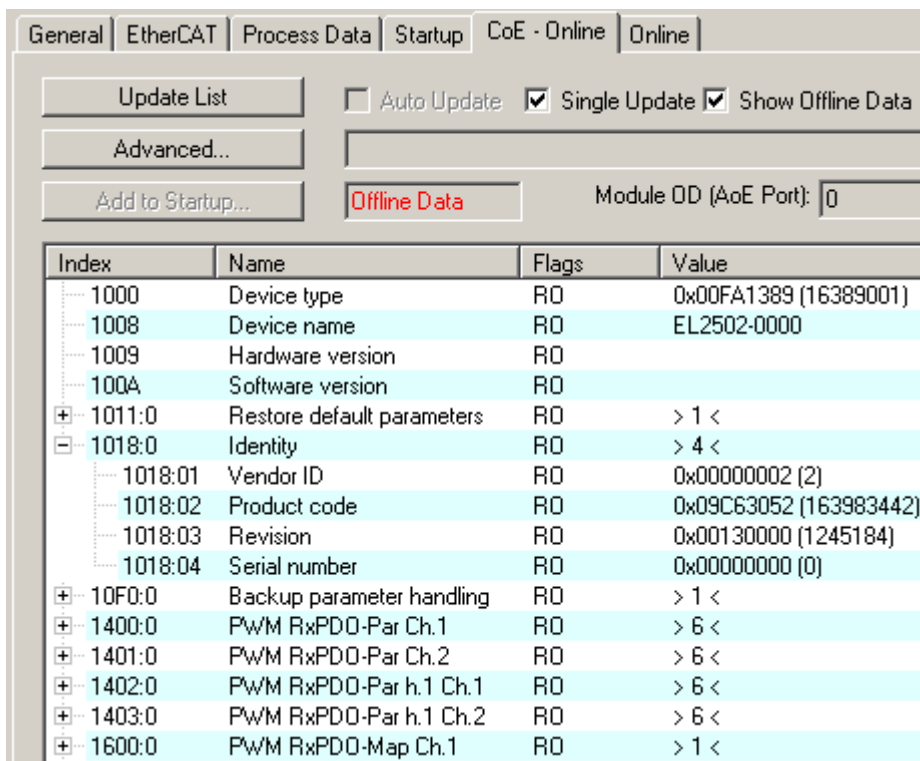


Fig. 13: "CoE Online " tab

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

**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 parameterised 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.

i  
**Note**

### 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.

i  
**Note**

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

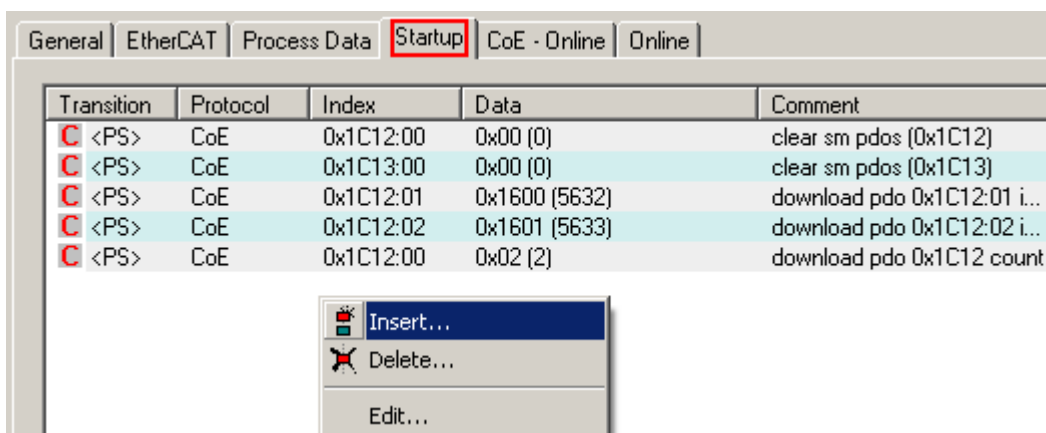


Fig. 14: Startup list in the TwinCAT System Manager

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.

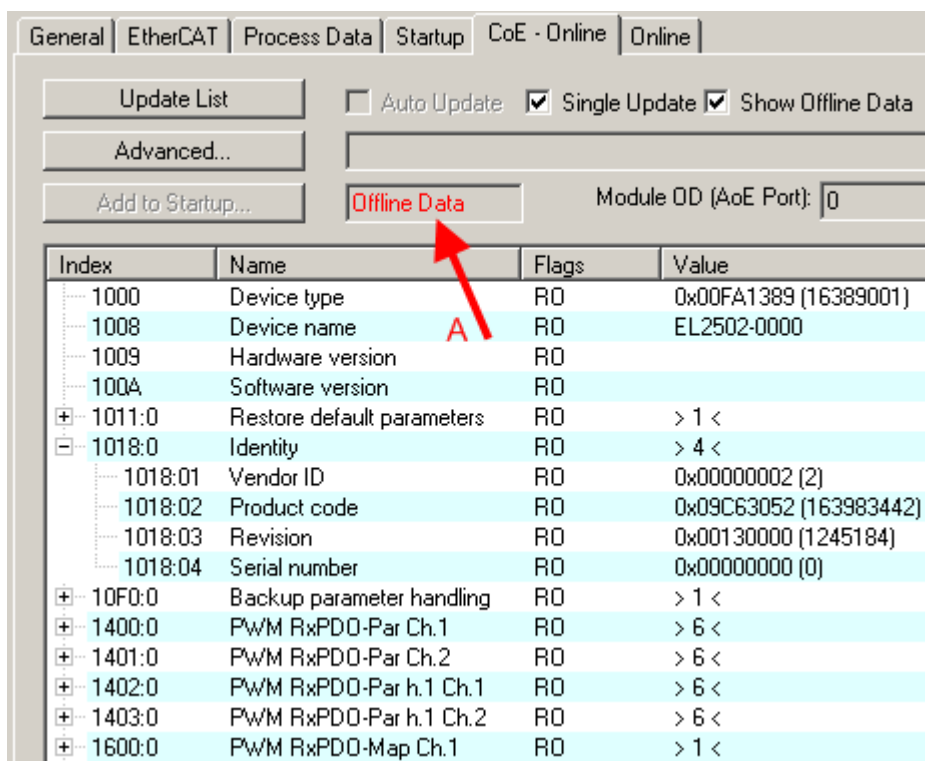


Fig. 15: Offline list

- 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.

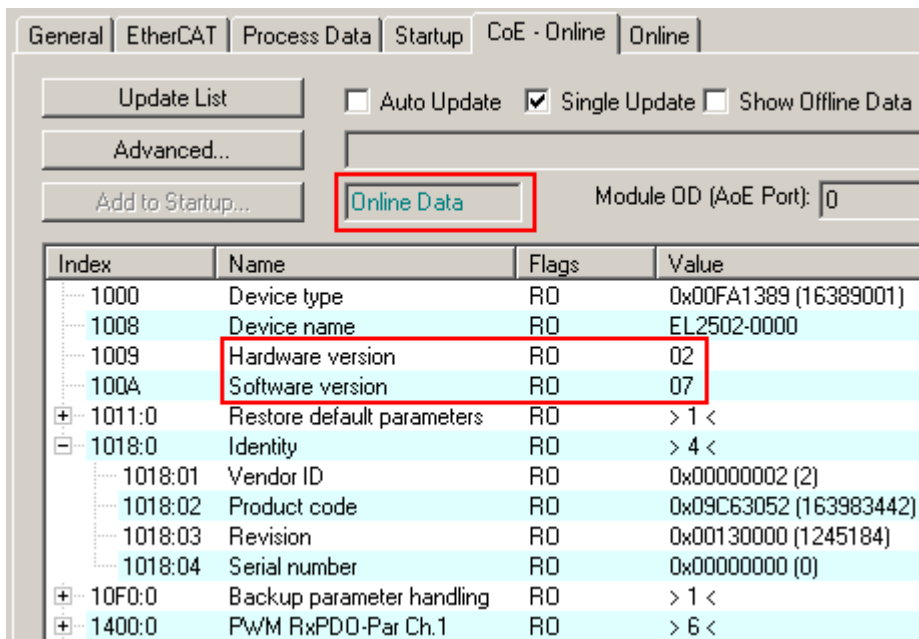


Fig. 16: Online list

**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 4 logical channels and therefore 4 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.

## 3.4 Distributed Clock


The distributed clock represents a local clock in the EtherCAT slave controller (ESC) with the following characteristics:

- Unit *1 ns*
- Zero point *1.1.2000 00:00*
- Size *64 bit* (sufficient for the next 584 years; however, some EtherCAT slaves only offer 32-bit support, i.e. the variable overflows after approx. 4.2 seconds)
- The EtherCAT master automatically synchronizes the local clock with the master clock in the EtherCAT bus with a precision of < 100 ns.

For detailed information please refer to the [EtherCAT system description](#).

## 4 Mounting and wiring

### 4.1 Instructions for ESD protection

	<p><b>Destruction of the devices by electrostatic discharge possible!</b></p> <p>The devices contain components at risk from electrostatic discharge caused by improper handling.</p> <ul style="list-style-type: none"> <li>✓ Please ensure you are electrostatically discharged and avoid touching the contacts of the device directly.</li> <li>a) Avoid contact with highly insulating materials (synthetic fibers, plastic film etc.).</li> <li>b) Surroundings (working place, packaging and personnel) should be grounded probably, when handling with the devices.</li> <li>c) Each assembly must be terminated at the right hand end with an <a href="#">EL9011</a> bus end cap, to ensure the protection class and ESD protection.</li> </ul>
<p><b>Attention</b></p>	

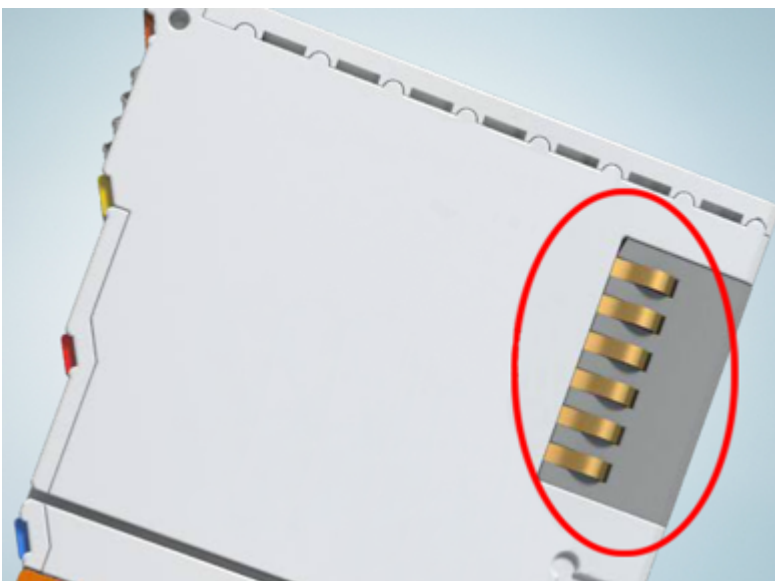




Fig. 17: Spring contacts of the Beckhoff I/O components

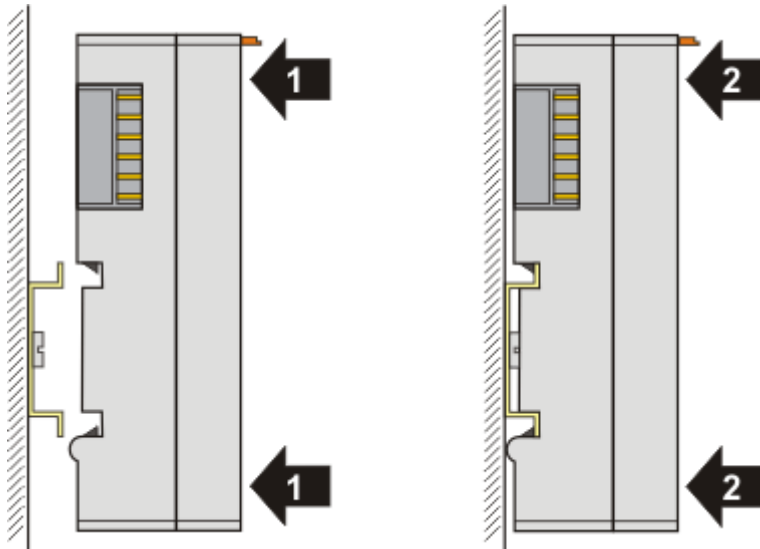
### 4.2 Mounting and demounting - terminals with front unlocking

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).

	<p><b>Fixing of mounting rails</b></p> <p>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).</p>
<p><b>Note</b></p>	
	<p><b>Risk of electric shock and damage of device!</b></p> <p>Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the Bus Terminals!</p>
<p><b>WARNING</b></p>	

## Mounting

- Fit the mounting rail to the planned assembly location.

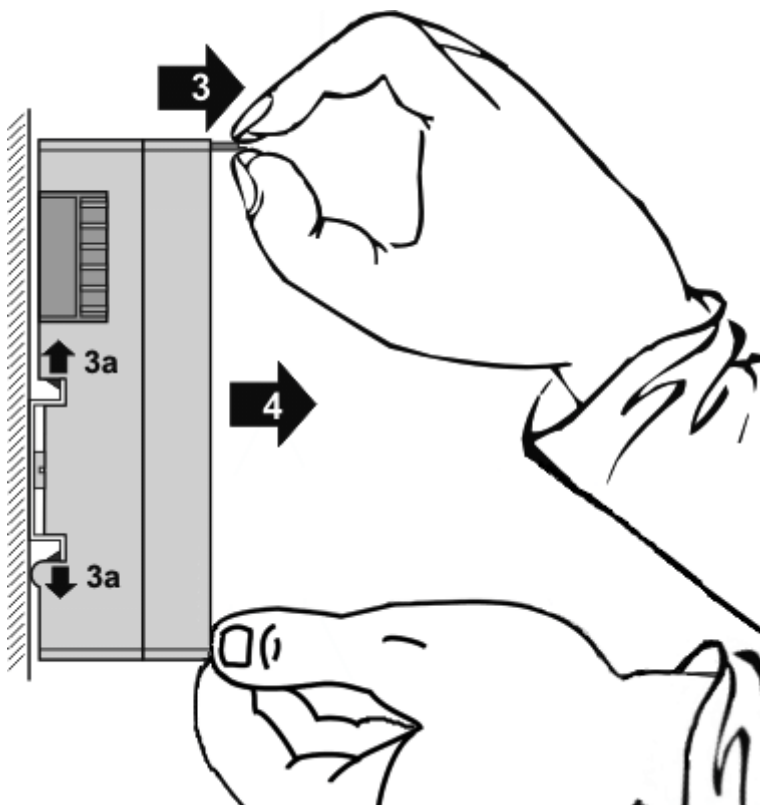


and press (1) the terminal module against the mounting rail until it latches in place on the mounting rail (2).

- 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 Recommended mounting rails

Terminal Modules und EtherCAT Modules of KMxxxx and EMxxxx series, same as the terminals of the EL66xx and EL67xx series can be snapped onto the following recommended mounting rails:

DIN Rail TH 35-7.5 with 1 mm material thickness (according to EN 60715)

DIN Rail TH 35-15 with 1,5 mm material thickness



**Note**

**Pay attention to the material thickness of the DIN Rail**

Terminal Modules und EtherCAT Modules of KMxxxx and EMxxxx series, same as the terminals of the EL66xx and EL67xx series does not fit to the DIN Rail TH 35-15 with 2,2 to 2,5 mm material thickness (according to EN 60715)!

## 4.4 Mounting of Passive Terminals



**Note**

**Hint for mounting passive terminals**

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 2 Passive Terminals!

**Examples for mounting passive terminals (highlighted)**

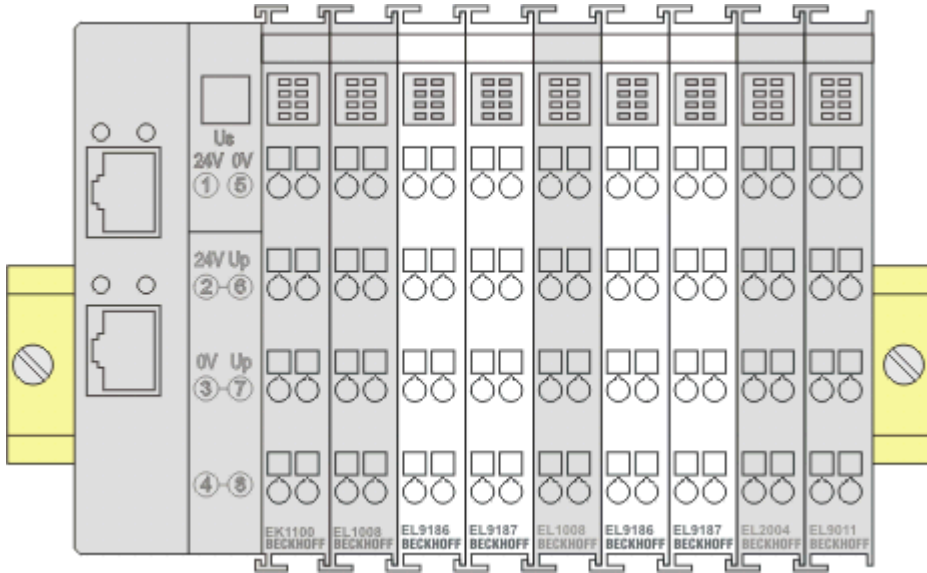


Fig. 18: *Correct configuration*

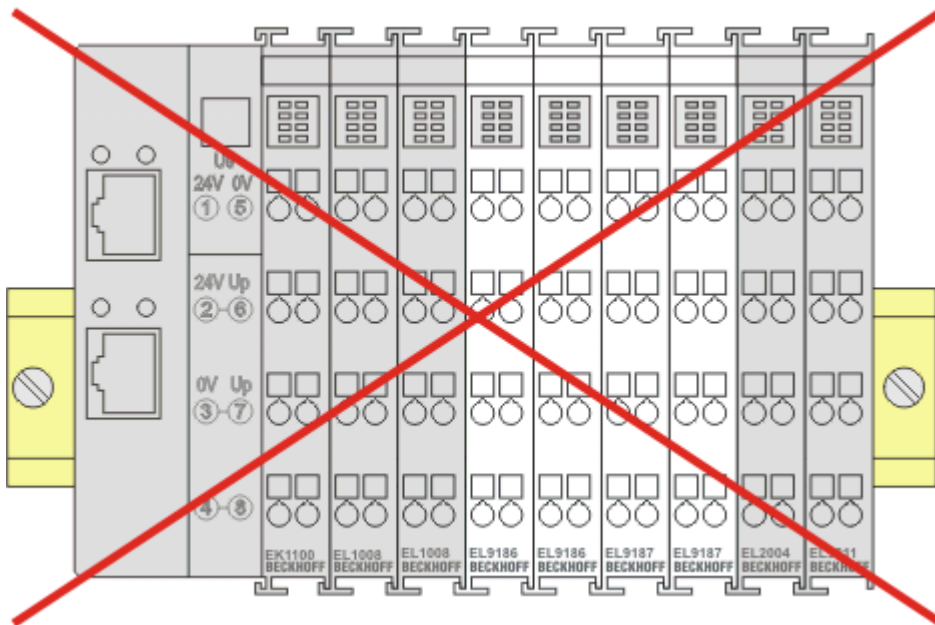


Fig. 19: *Incorrect configuration*

## 4.5 ATEX - Special conditions (standard temperature range)



**WARNING**

**Observe the special conditions for the intended use of Beckhoff fieldbus components with standard temperature range in potentially explosive areas (directive 94/9/EU)!**

- The certified components are to be installed in a suitable housing that guarantees a protection class of at least IP54 in accordance with EN 60529! The environmental conditions during use are thereby to be taken into account!
- If the temperatures during rated operation are higher than 70°C at the feed-in points of cables, lines or pipes, or higher than 80°C at the wire branching points, then cables must be selected whose temperature data correspond to the actual measured temperature values!
- Observe the permissible ambient temperature range of 0 to 55°C for the use of Beckhoff fieldbus components standard temperature range in potentially explosive areas!
- Measures must be taken to protect against the rated operating voltage being exceeded by more than 40% due to short-term interference voltages!
- The individual terminals may only be unplugged or removed from the Bus Terminal system if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The connections of the certified components may only be connected or disconnected if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The fuses of the KL92xx/EL92xx power feed terminals may only be exchanged if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- Address selectors and ID switches may only be adjusted if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!

### Standards

The fundamental health and safety requirements are fulfilled by compliance with the following standards:

- EN 60079-0:2012+A11:2013
- EN 60079-15:2010

### Marking

The Beckhoff fieldbus components with standard temperature range certified for potentially explosive areas bear one of the following markings:




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

or




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

## 4.6 ATEX Documentation

 <b>Note</b>	<p><b>Notes about operation of the Beckhoff terminal systems in potentially explosive areas (ATEX)</b></p> <p>Pay also attention to the continuative documentation</p> <p>Notes about operation of the Beckhoff terminal systems in potentially explosive areas (ATEX)</p> <p>that is available in the download area of the Beckhoff homepage <a href="http://www.beckhoff.com">http://www.beckhoff.com</a>!</p>
--	--

## 4.7 Installation positions

 <b>Attention</b>	<p><b>Constraints regarding installation position and operating temperature range</b></p> <p>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!</p>
---	---

### 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.

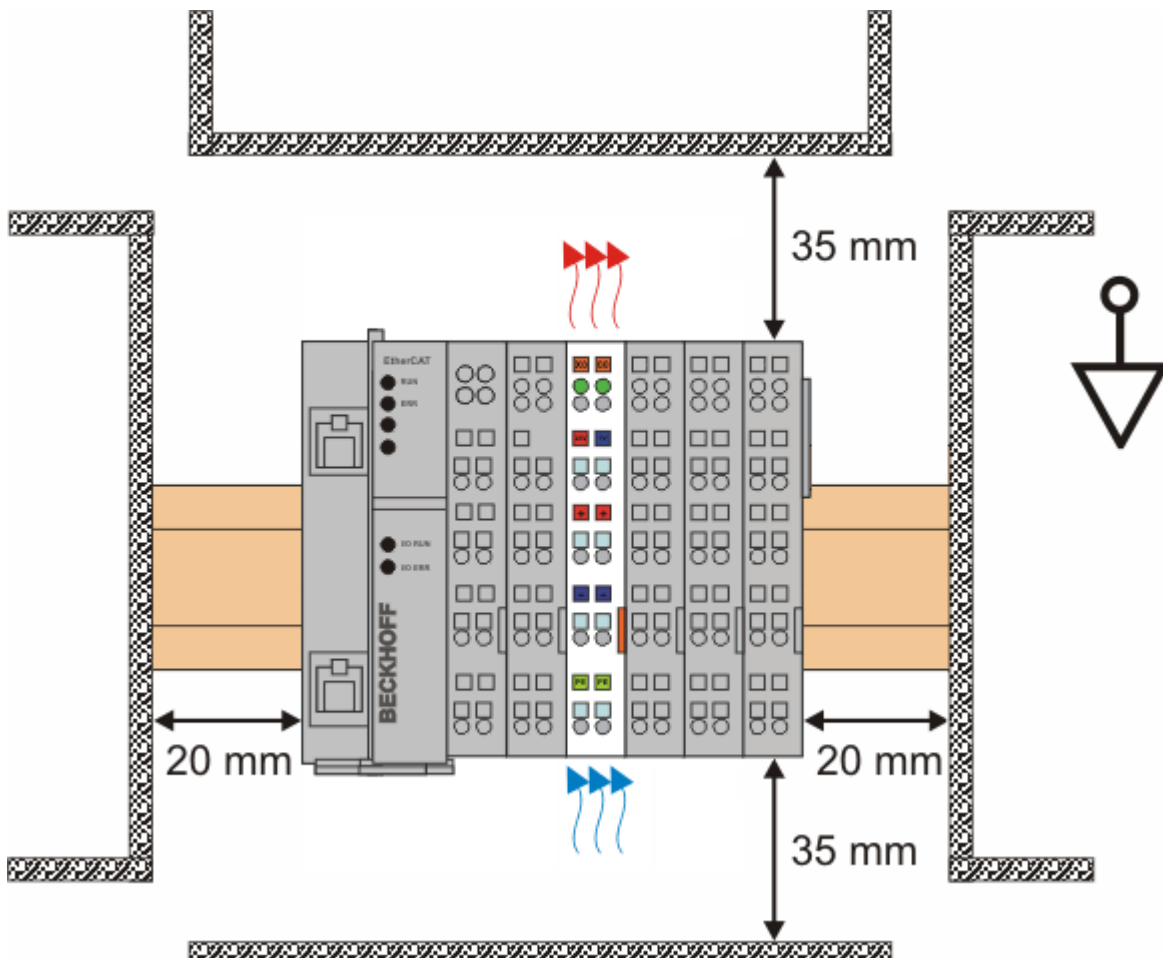


Fig. 20: Recommended distances for standard installation position

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.

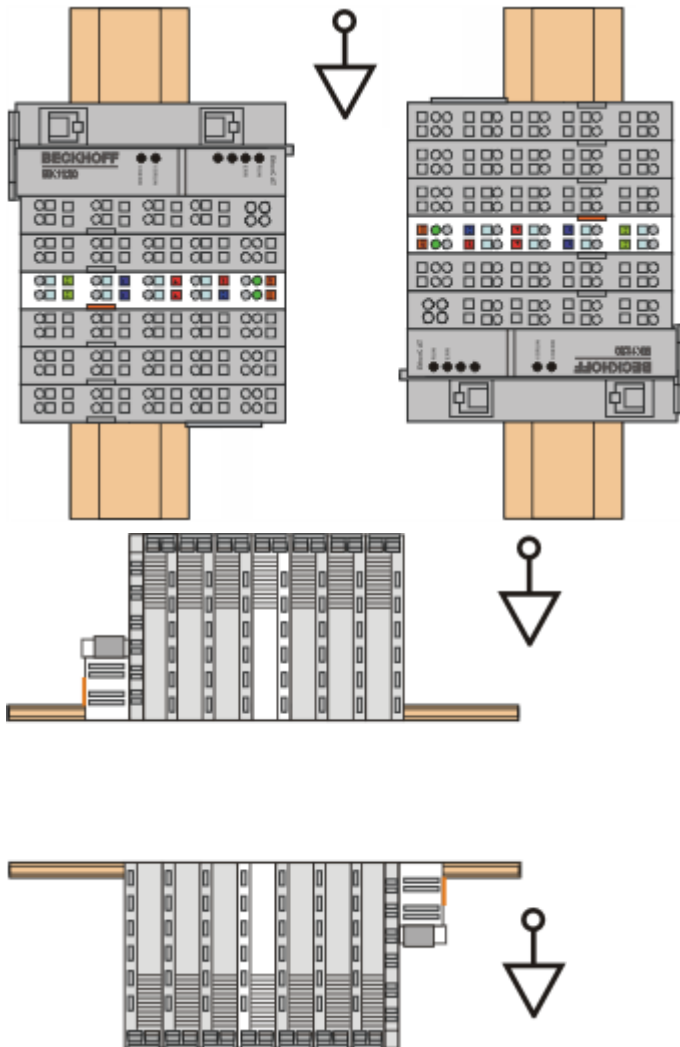


Fig. 21: Other installation positions

## 4.8 LEDs

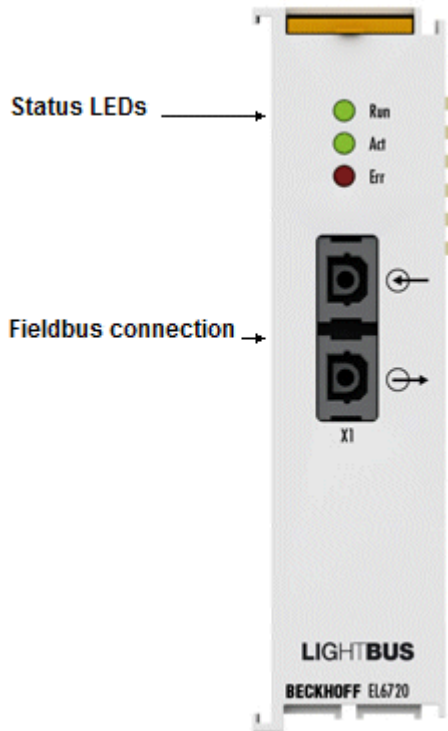


Fig. 22: LEDs

LED	Color	Meaning	
RUN	green	These LEDs indicate the terminal's operating state:	
		off	State of the <u>EtherCAT State Machine</u> [▶ 16]: <b>INIT</b> = initialization of the terminal
		flashing uniformly	State of the EtherCAT State Machine: <b>PREOP</b> = function for mailbox communication and different standard-settings set
		flashing slowly	State of the EtherCAT State Machine: <b>SAFEOP</b> = verification of the <u>Sync Manager</u> [▶ 79] channels
		on	State of the EtherCAT State Machine: <b>OP</b> = normal operating state; mailbox and process data communication is possible
		flashing rapidly	State of the EtherCAT State Machine: <b>BOOTSTRAP</b> = function for terminal firmware updates
ACT	green	Signal exchange LIGHTBUS	
Err	red	LIGHTBUS error	

## 5 Commissioning

### 5.1 TwinCAT Quick Start

TwinCAT is a development environment for real-time control including 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 Systemmanual:**  
Fieldbus Components → EtherCAT Terminals → EtherCAT System Documentation → Setup in the TwinCAT System Manager
- **TwinCAT 2** → TwinCAT System Manager → I/O - Configuration
- In particular, TwinCAT driver installation:  
**Fieldbus components** → Fieldbus Cards and Switches → FC900x – PCI Cards for Ethernet → Installation

Devices contain the 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 offline mode can be found under <http://infosys.beckhoff.com>:  
**TwinCAT 2** → TwinCAT System Manager → IO - Configuration → Adding 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 from user PC to the individual control elements:

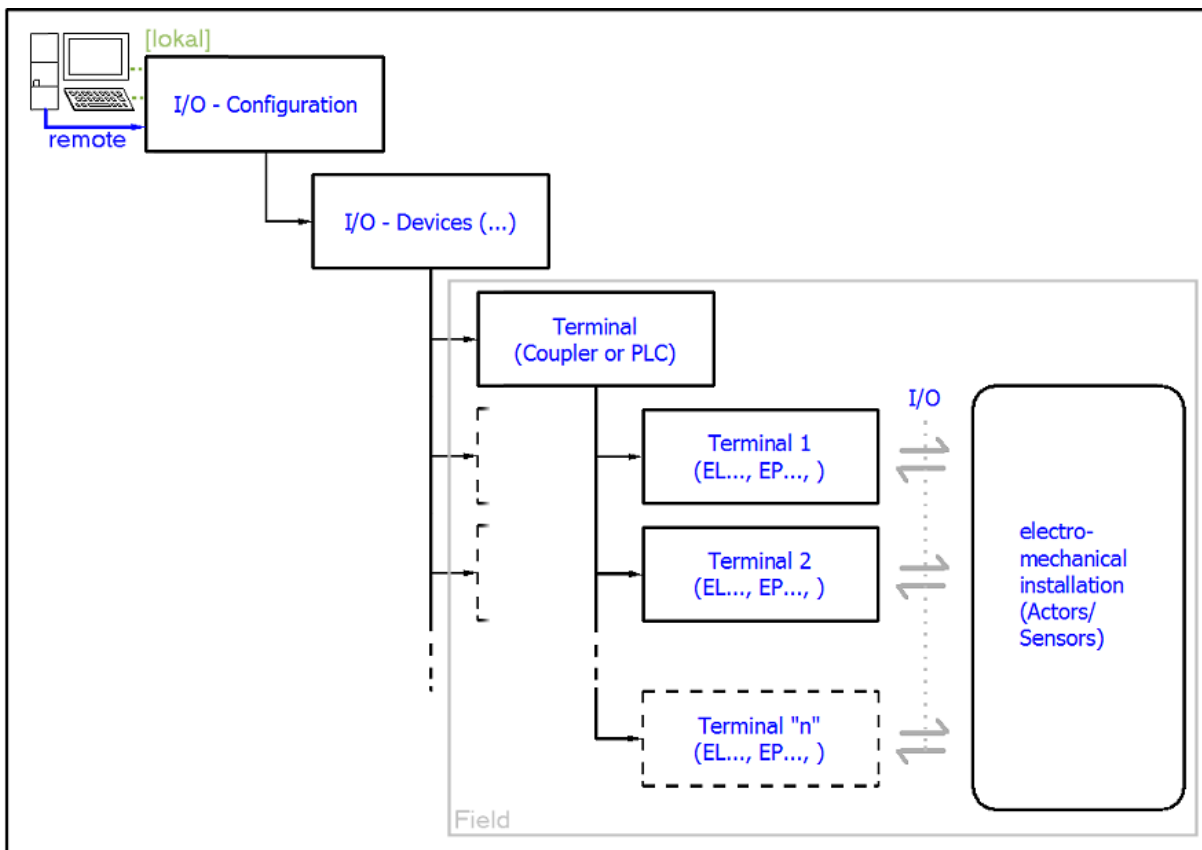


Fig. 23: Relationship between user side (commissioning) and installation

The user inserting of certain components (I/O device, terminal, box...) is the same in TwinCAT 2 and TwinCAT 3. The descriptions below relate to the online procedure.

### Sample configuration (actual configuration)

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

- Control system (PLC) **CX2040** including **CX2100-0004** power supply unit
- Connected to the CX2040 on the right (E-bus):  
**EL1004** (4-channel analog input terminal -10...+10 V)
- 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)

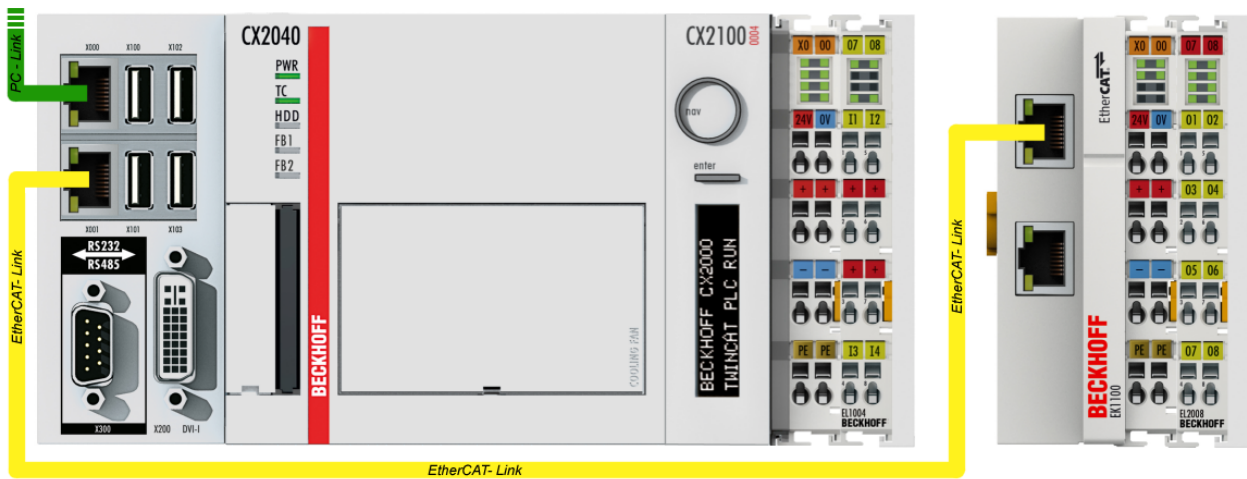


Fig. 24: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

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 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:

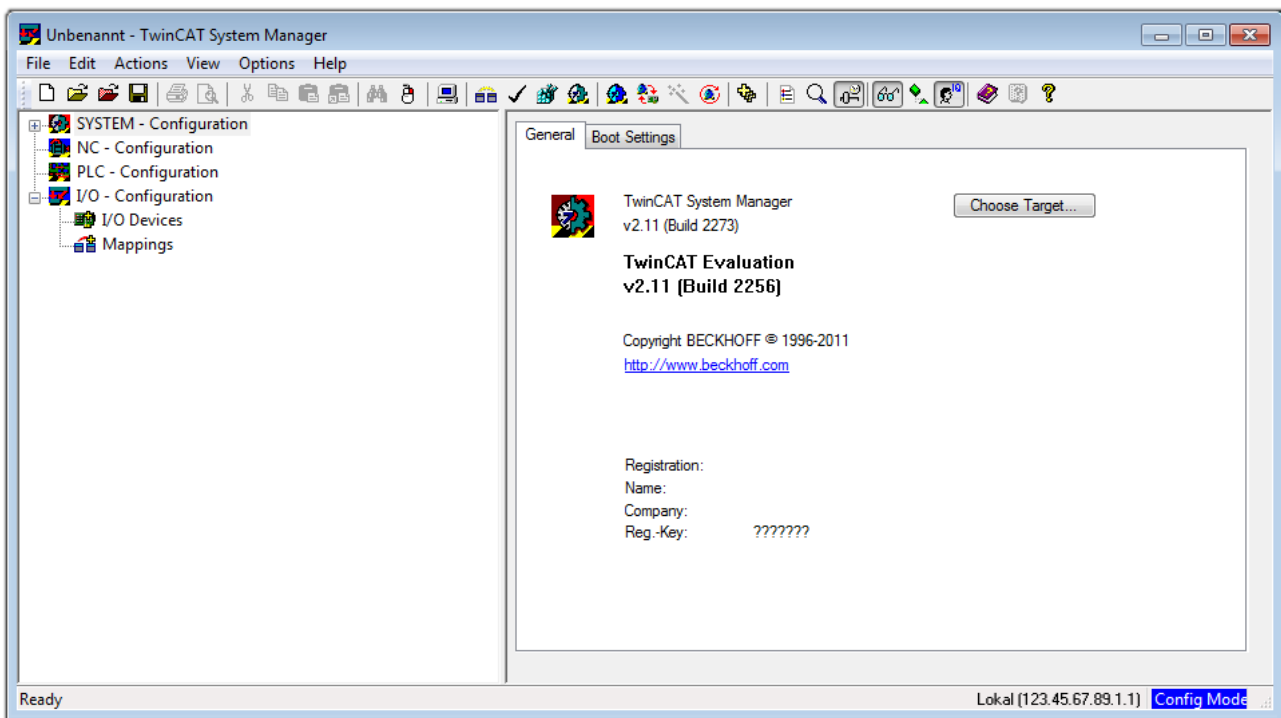



Fig. 25: Initial TwinCAT 2 user interface

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 thereby the next step is "Insert Device [▶ 35]".

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

"Actions" → "Choose Target System...", via the symbol  " or the "F8" key, open the following window:

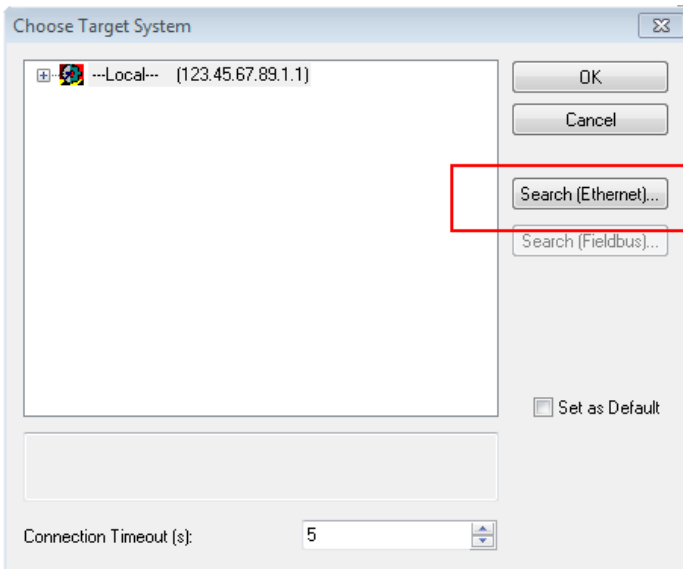


Fig. 26: Selection of the target system

Use "Search (Ethernet)..." to enter the target system. Thus a next 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.

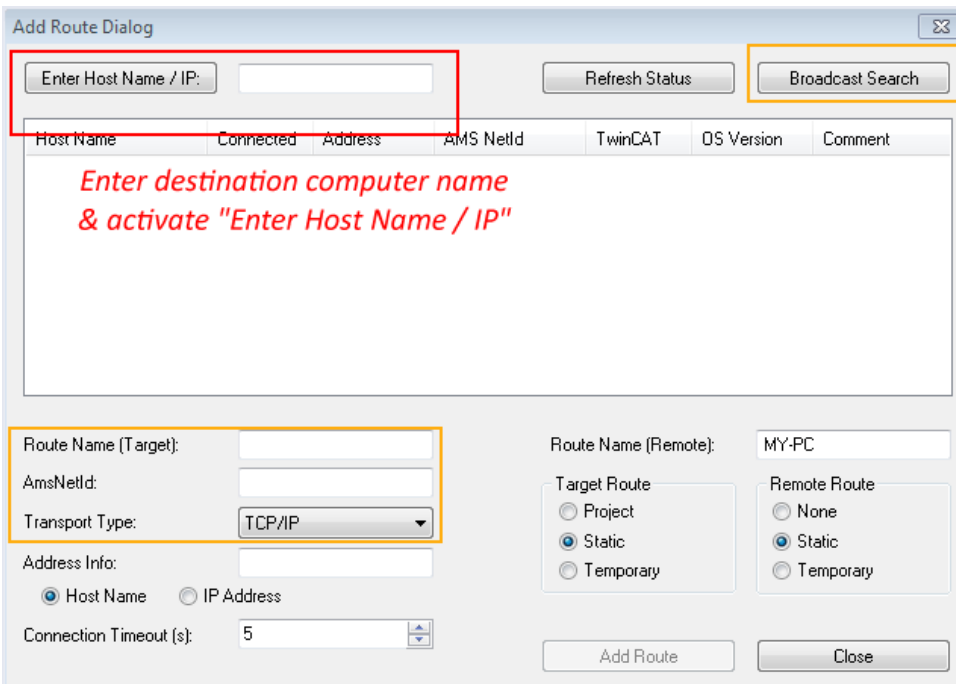
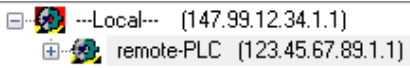


Fig. 27: Specify the PLC for access by the TwinCAT System Manager: selection of the target system



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



After confirmation with "OK" the target system can be accessed via the System Manager.

**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 . The TwinCAT System Manager may first have to be set to "Config mode" via  or via menu "Actions" → "Set/Reset TwinCAT to Config Mode..." (Shift + F4).

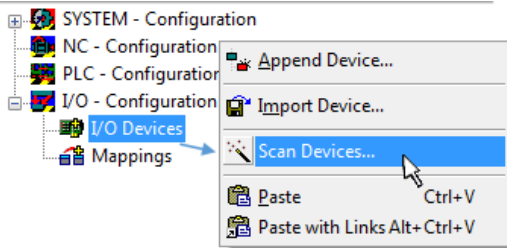


Fig. 28: Select "Scan Devices..."

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

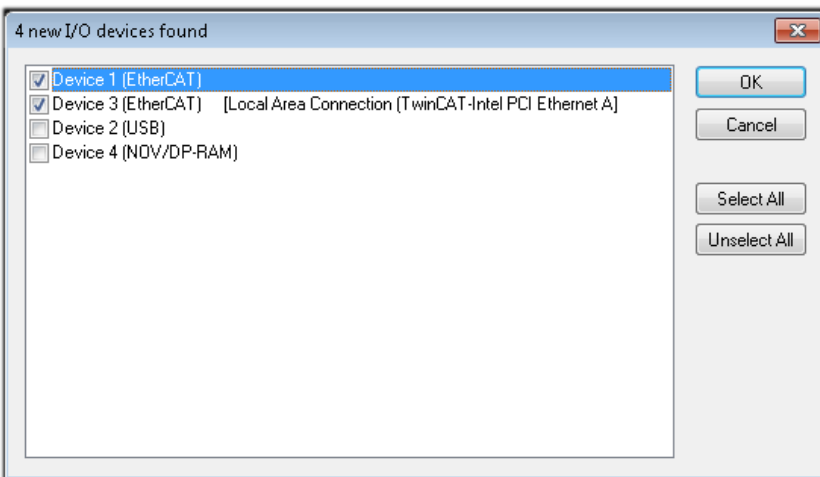


Fig. 29: Automatic detection of I/O devices: selection the devices to be integrated

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 [sample configuration \[▶ 32\]](#) described at the beginning of this section, the result is as follows:

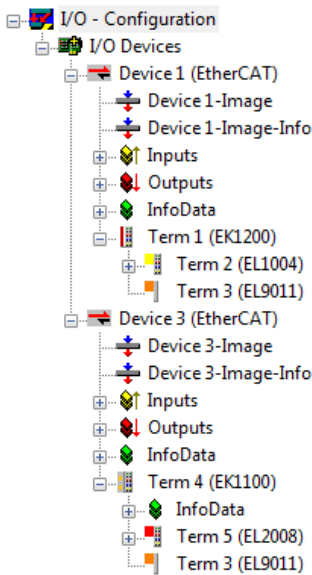


Fig. 30: Mapping of the configuration in the TwinCAT 2 System Manager

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

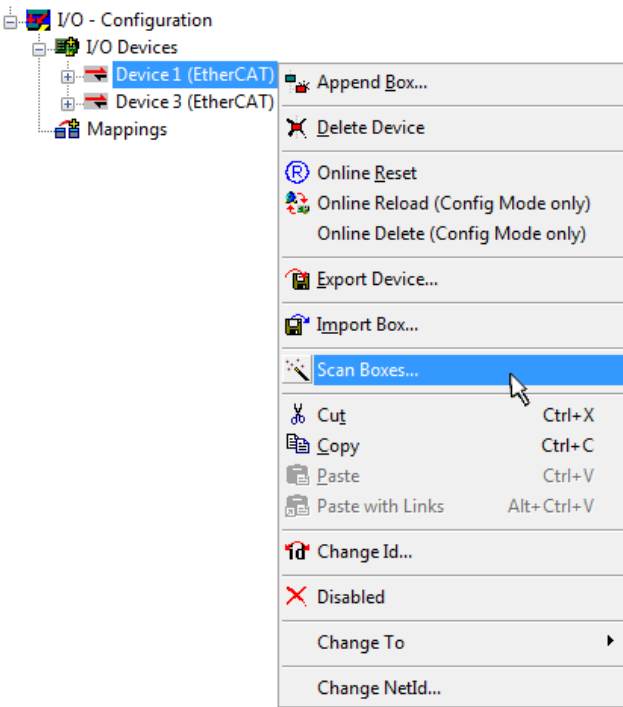


Fig. 31: Reading of individual terminals connected to a device

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 the creation of 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 to Structured Text (ST).

After starting TwinCAT PLC Control, the following user interface is shown for an initial project:

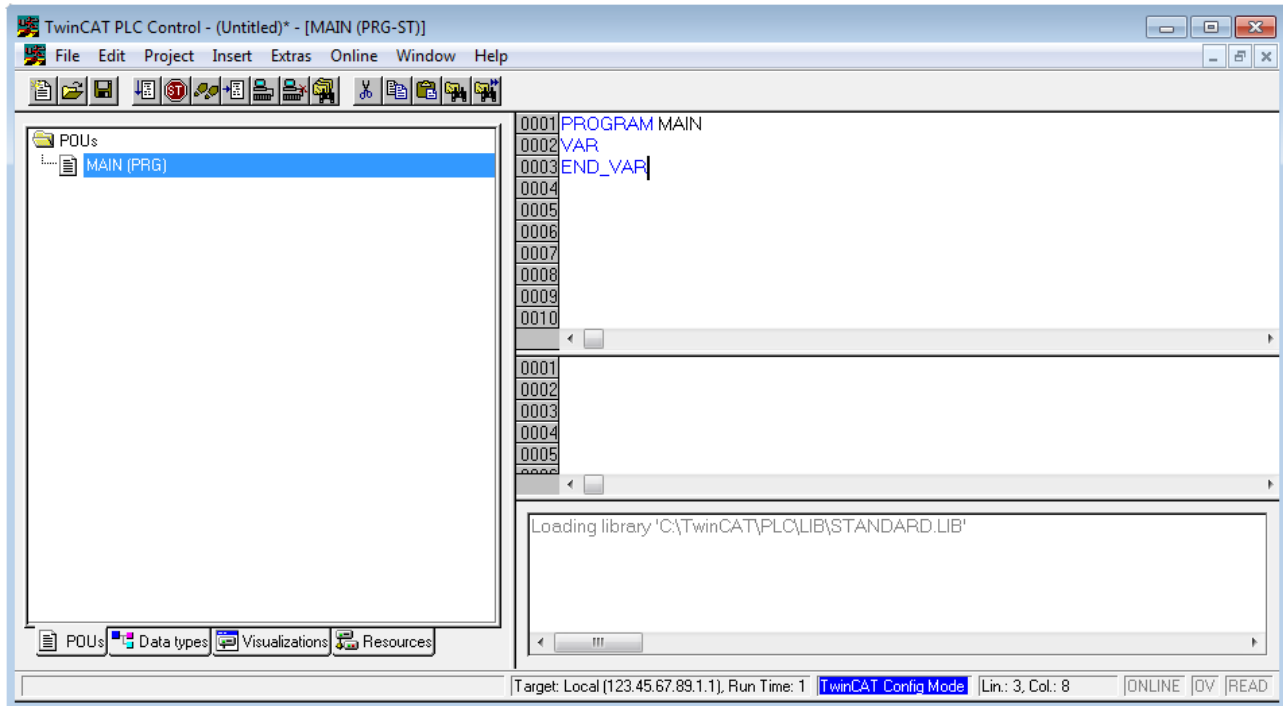


Fig. 32: TwinCAT PLC Control after startup

Sample variables and a sample program have been created and stored under the name "PLC\_example.pro":

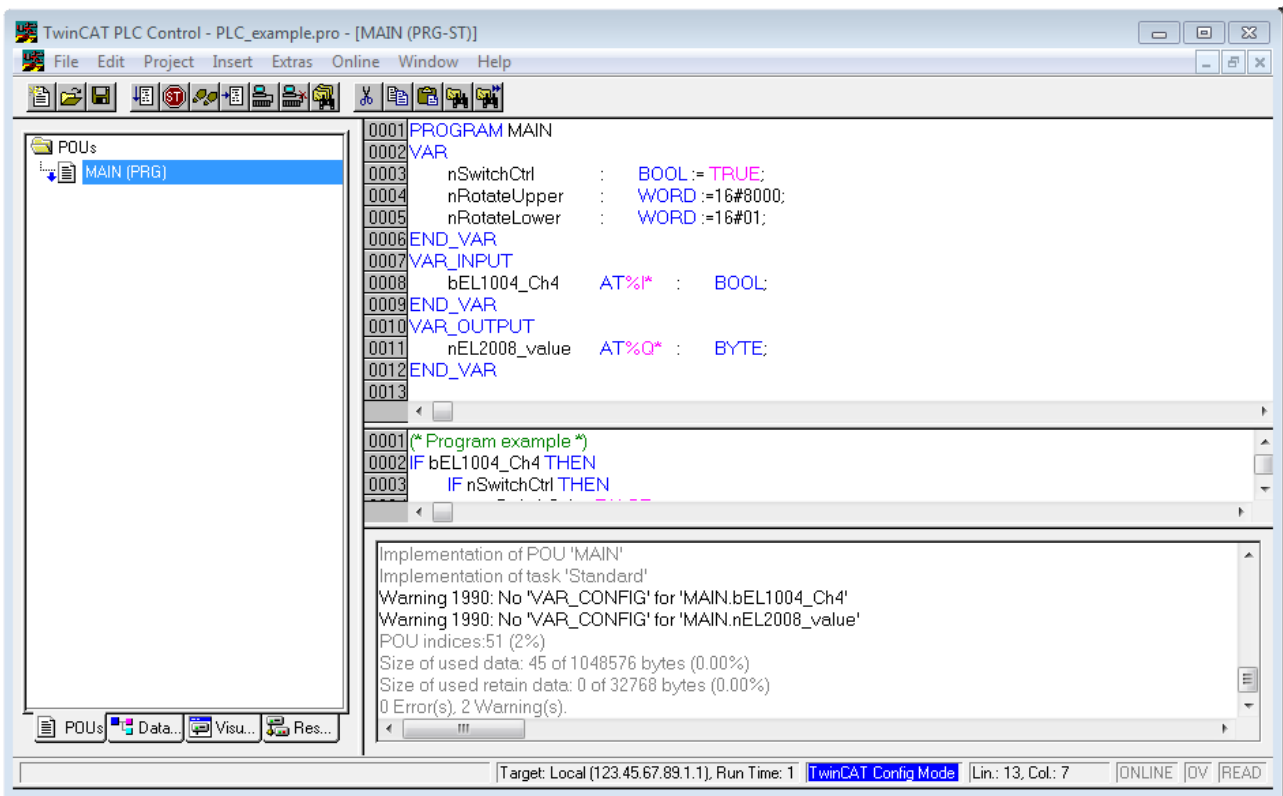


Fig. 33: Sample program with variables after a compile process (without variable integration)

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** via the context menu of the PLC configuration; right-click and select "Append PLC Project...":

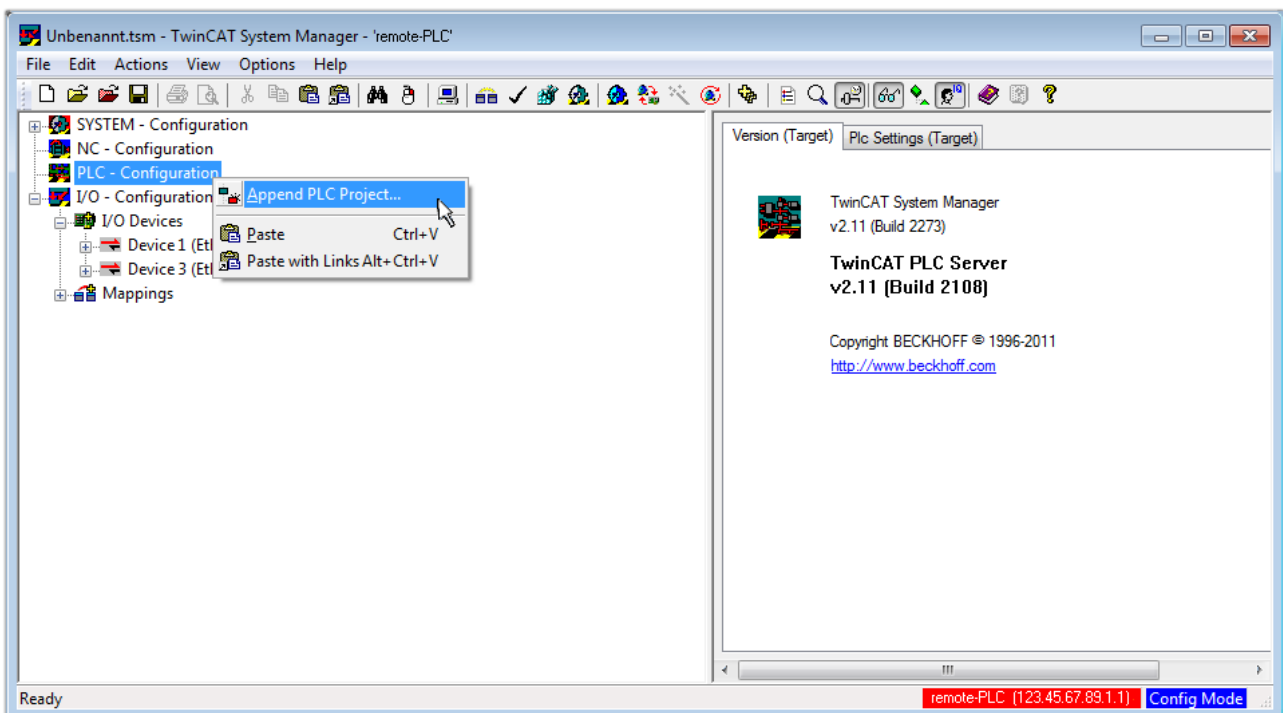


Fig. 34: Appending the TwinCAT PLC Control project

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:

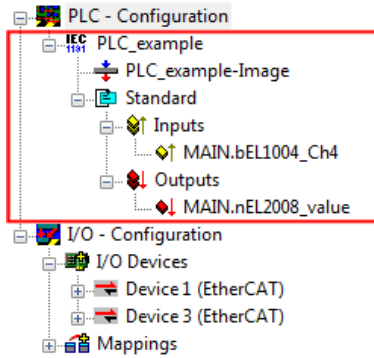


Fig. 35: PLC project integrated in the PLC configuration of the System Manager

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":

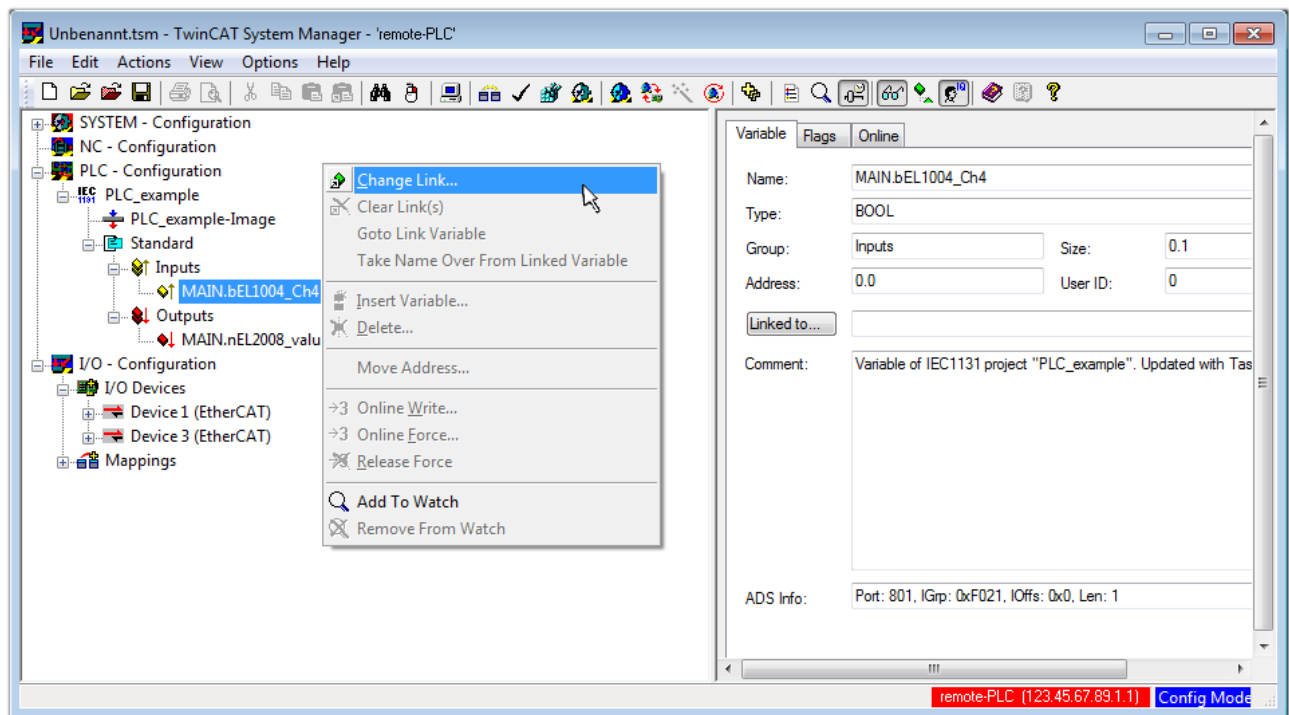


Fig. 36: Creating the links between PLC variables and process objects

In the window that opens, the process object for the variable "bEL1004\_Ch4" of type BOOL can be selected from the PLC configuration tree:

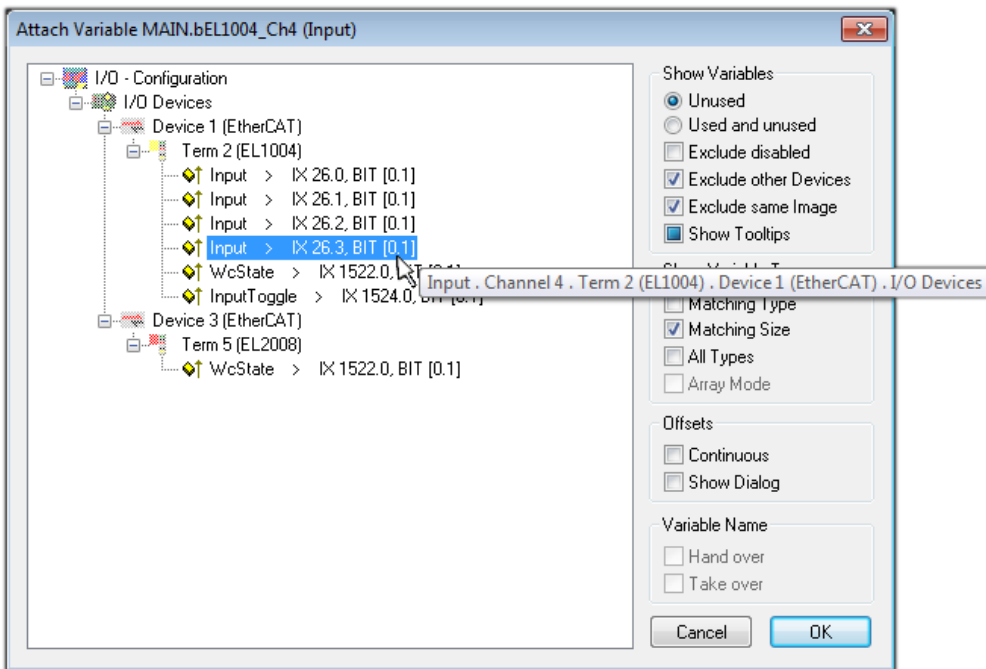


Fig. 37: Selecting PDO of type BOOL

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

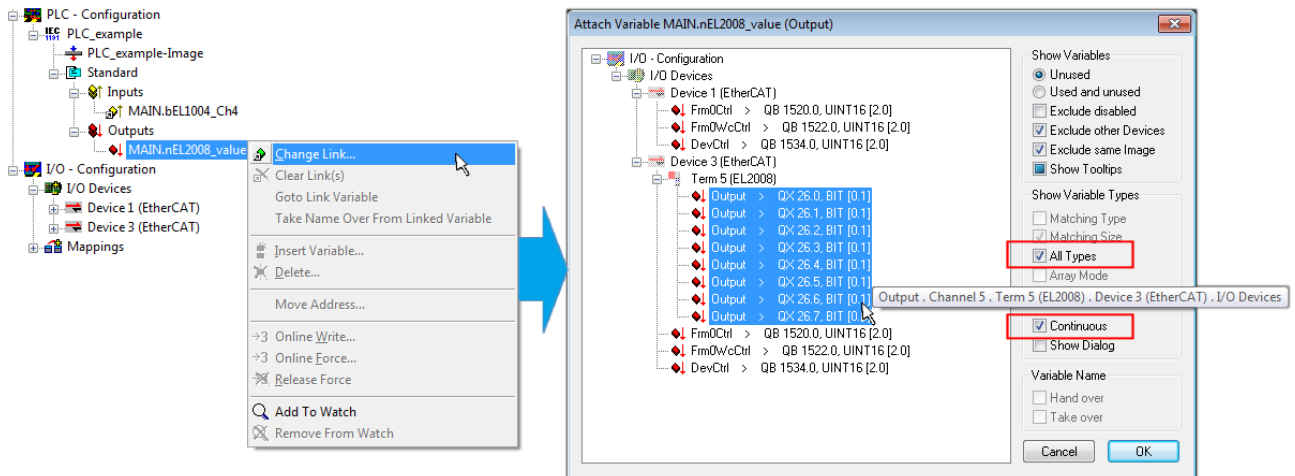



Fig. 38: Selecting several PDOs simultaneously: activate "Continuous" and "All types"

Note that the "Continuous" checkbox was also activated. This is designed to allocate the bits contained in the byte of the variable "nEL2008\_value" sequentially to all eight selected output bits of the EL2008 terminal. In this way it is 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 (  ) at the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting a "Goto Link Variable" from the context menu of a variable. The object opposite, in this case the PDO, is automatically selected:

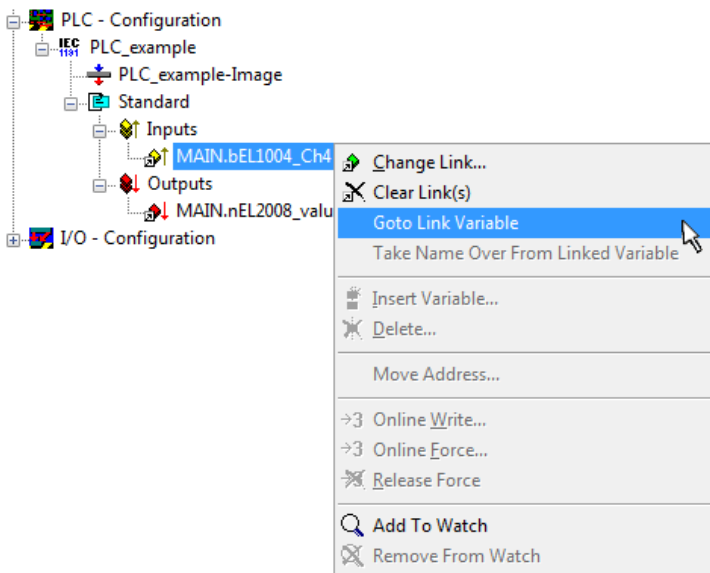

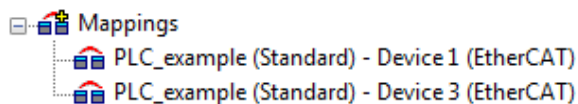


Fig. 39: Application of a "Goto Link" variable, using "MAIN.bEL1004\_Ch4" as a sample

The process of assigning variables to the PDO is completed via the menu selection "Actions" → "Generate Mappings", key Ctrl+M or by clicking on the symbol  in the menu.


This can be visualized in the configuration:




The process of creating links can also take place in the opposite direction, i.e. starting with individual PDOs to variable. However, in this example it would then 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 possible to allocate this a set of bit-standardised variables (type "BOOL"). Here, too, a "Goto Link Variable" from the context menu of a PDO 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 are overwritten!" and "Restart TwinCAT system in Run mode" with "OK".

A few seconds later the real-time status **RTime 0%** 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 Ethernet via "Online" → "Choose Run-Time System...":

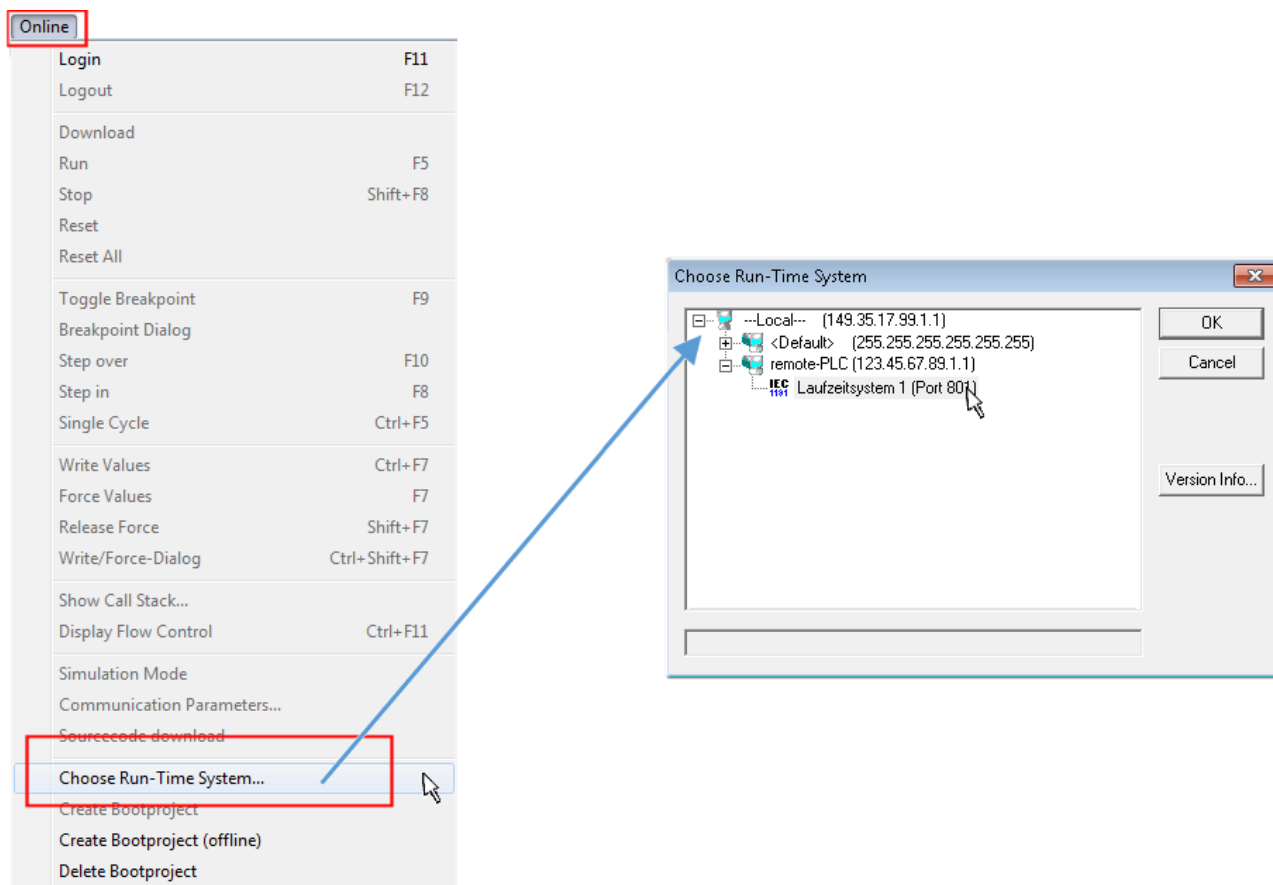

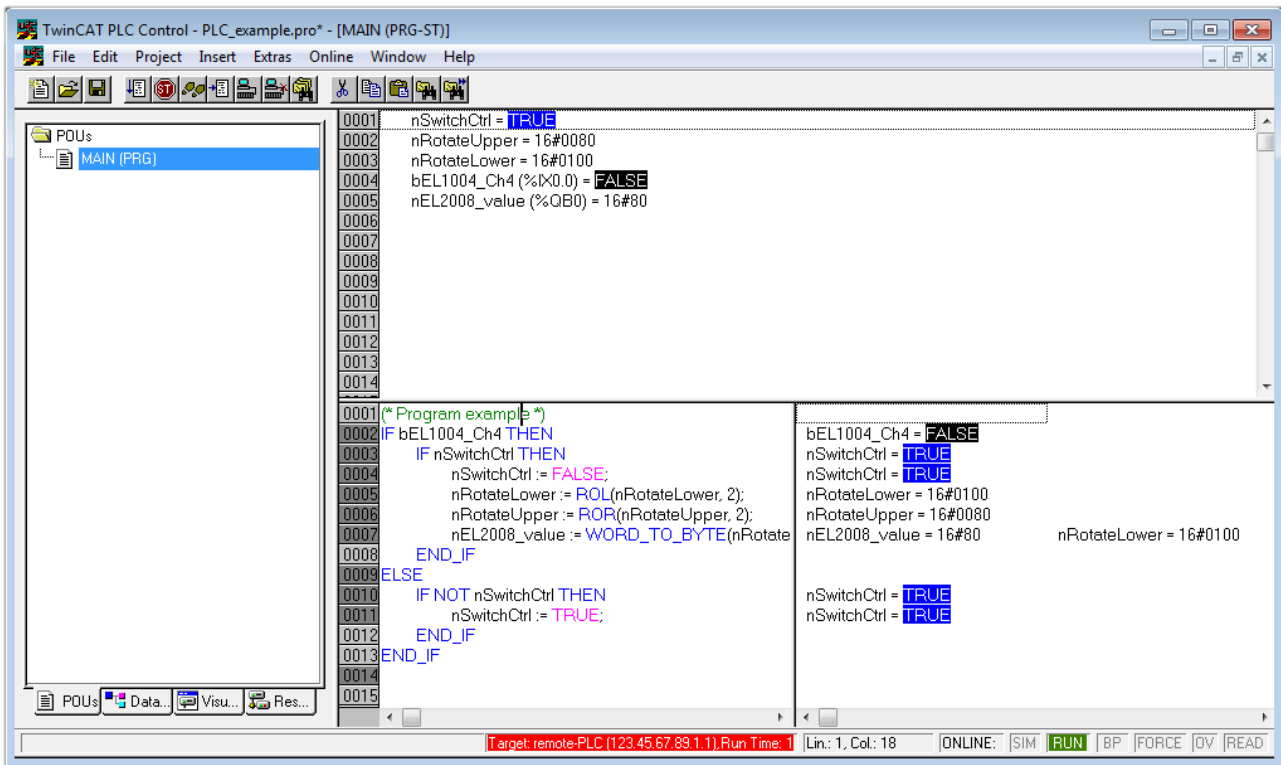


Fig. 40: Choose target system (remote)

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

system via 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 acknowledged with "Yes". The runtime environment is ready for the program start:



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

## 5.1.2 TwinCAT 3


### Startup

TwinCAT makes the development environment areas available together with Microsoft Visual Studio: after startup, the project folder explorer appears on the left in the general window area (cf. "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. 42: Initial TwinCAT 3 user interface

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

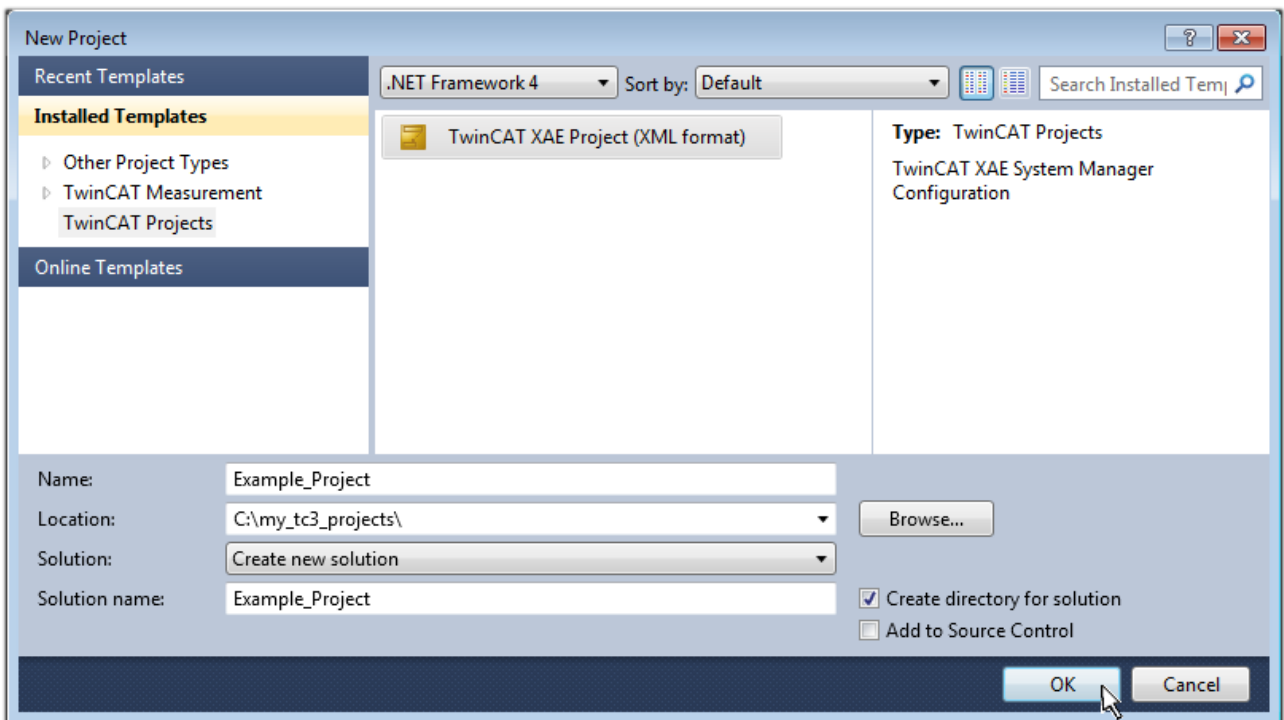


Fig. 43: Create new TwinCAT project

The new project is then available in the project folder explorer:

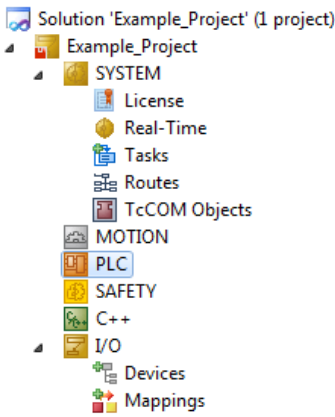
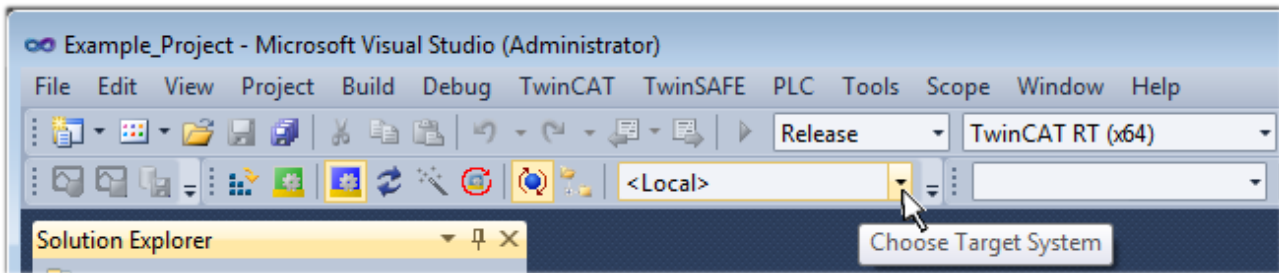


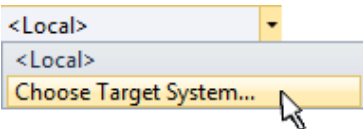
Fig. 44: New TwinCAT3 project in the project folder explorer

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, TwinCAT can be used in local mode and thereby the next step is "Insert Device [▶ 46]".

If the intention is to address the TwinCAT runtime environment installed on a PLC as development environment remotely from another system, 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:

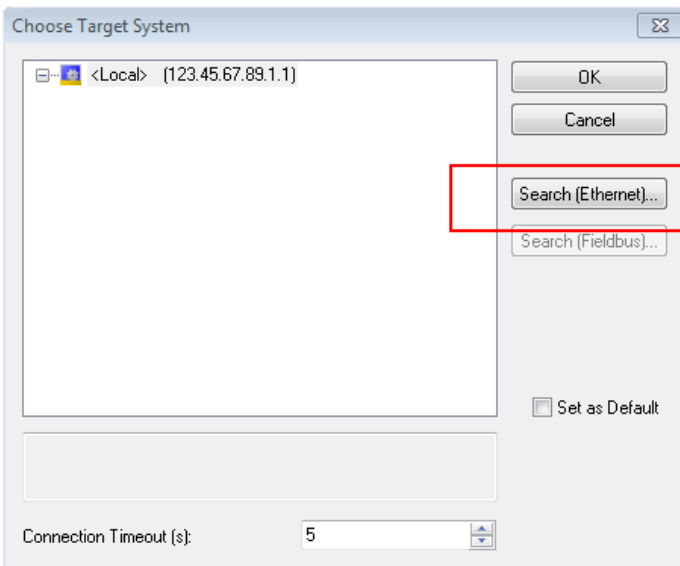


Fig. 45: Selection dialog: Choose the target system

Use "Search (Ethernet)..." to enter the target system. Thus a next 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.

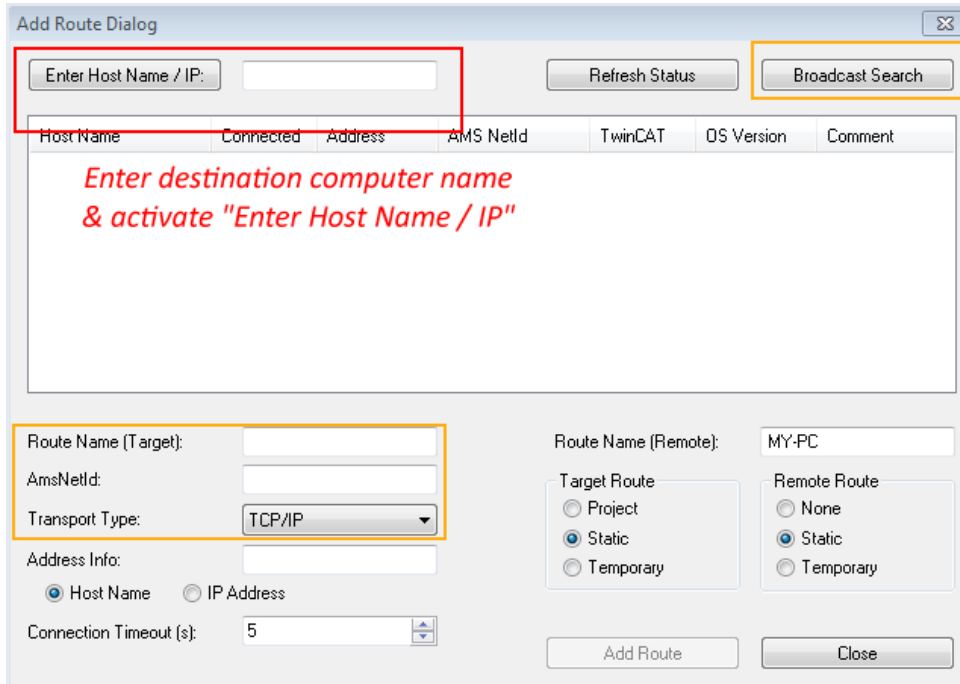
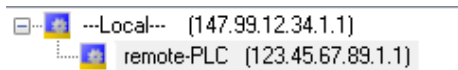


Fig. 46: Specify the PLC for access by the TwinCAT System Manager: selection of the target system


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




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

**Adding devices**

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

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

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

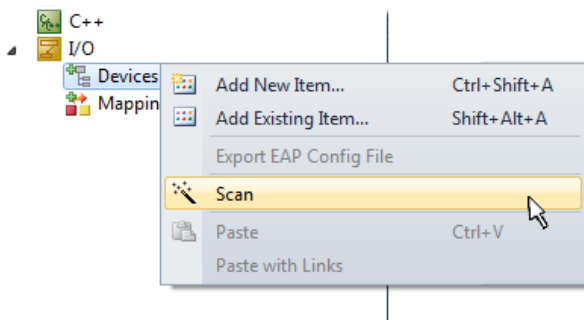


Fig. 47: Select "Scan"

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

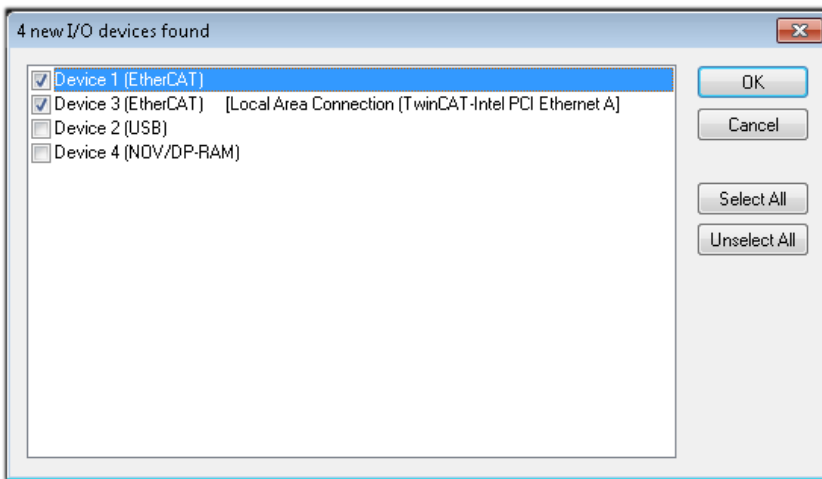


Fig. 48: Automatic detection of I/O devices: selection the devices to be integrated

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 [sample configuration \[▶ 32\]](#) described at the beginning of this section, the result is as follows:

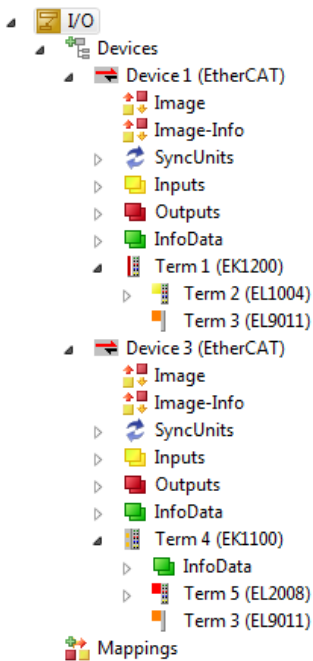


Fig. 49: Mapping of the configuration in VS shell of the TwinCAT3 environment

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

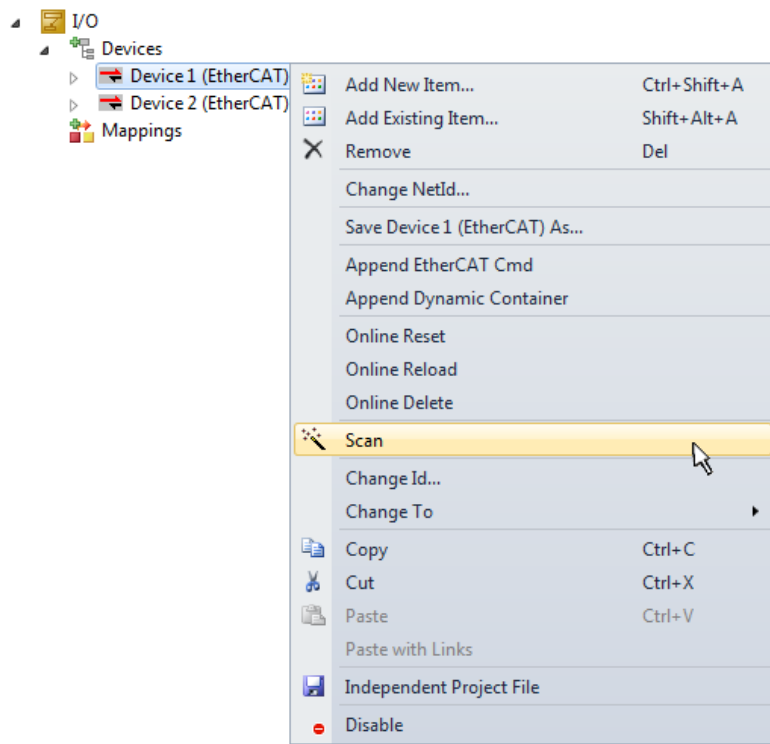


Fig. 50: Reading of individual terminals connected to a device

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

### Programming the PLC

TwinCAT PLC Control is the development environment for the creation of 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 to Structured Text (ST).

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

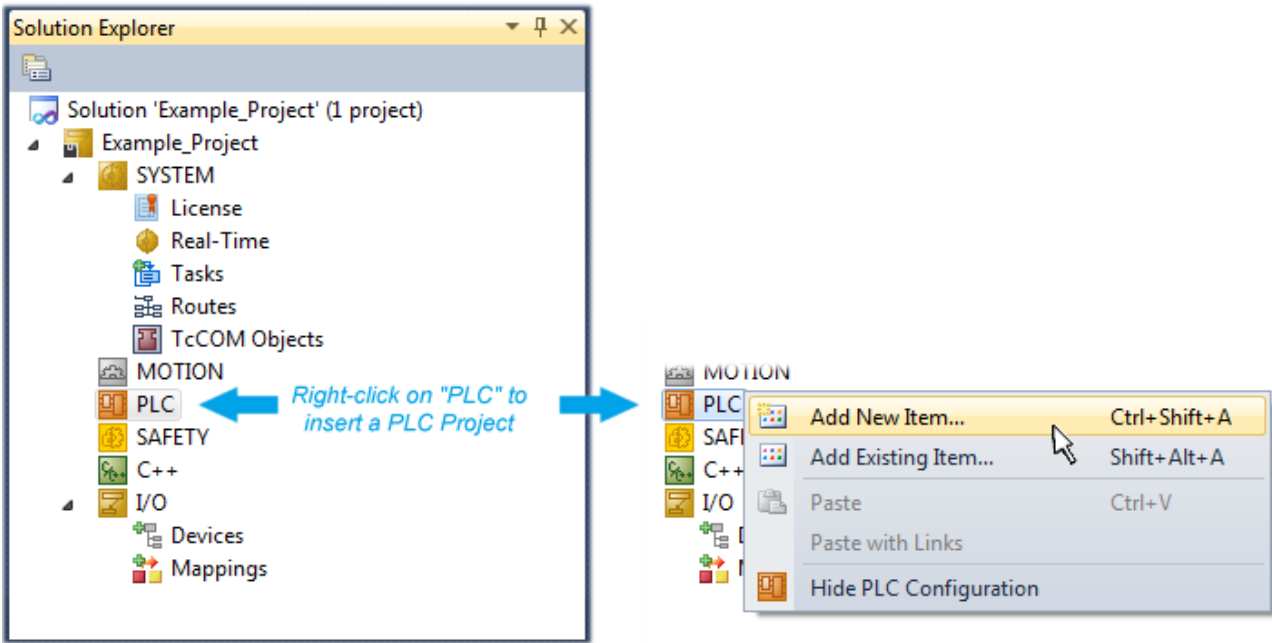


Fig. 51: Adding the programming environment in "PLC"

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

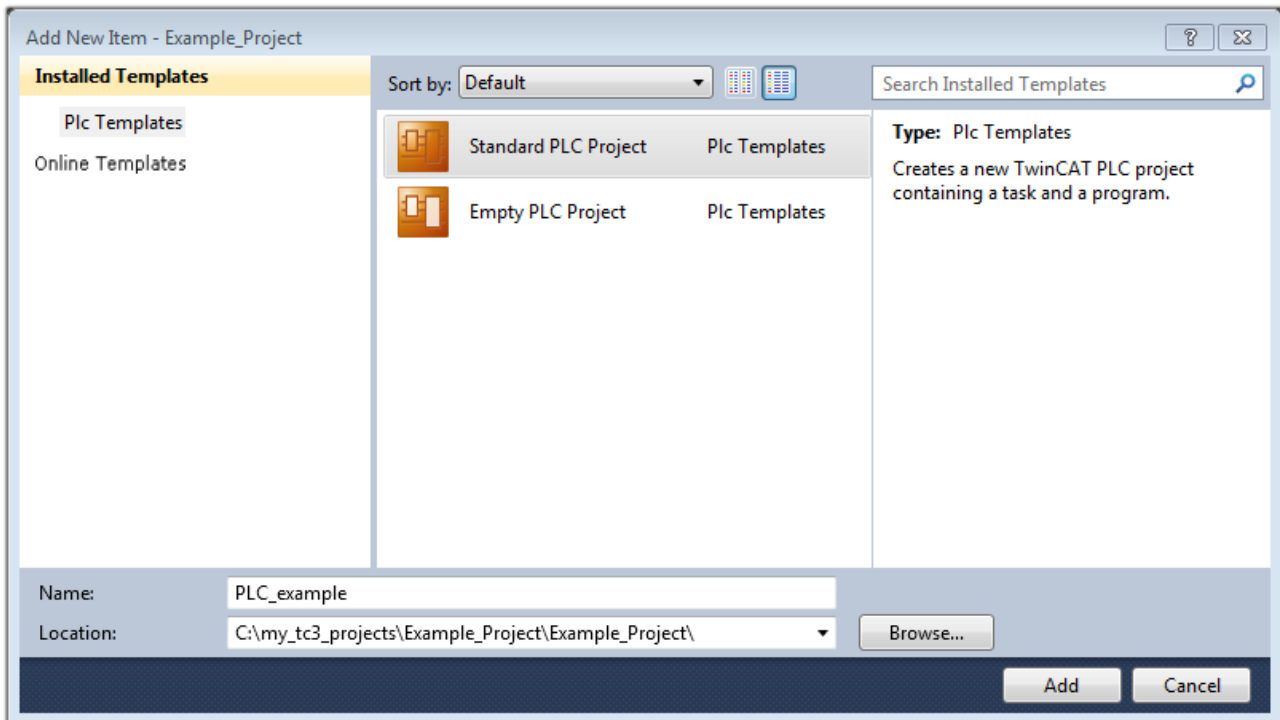


Fig. 52: Specifying the name and directory for the PLC programming environment

The "Main" program, which already exists by 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:

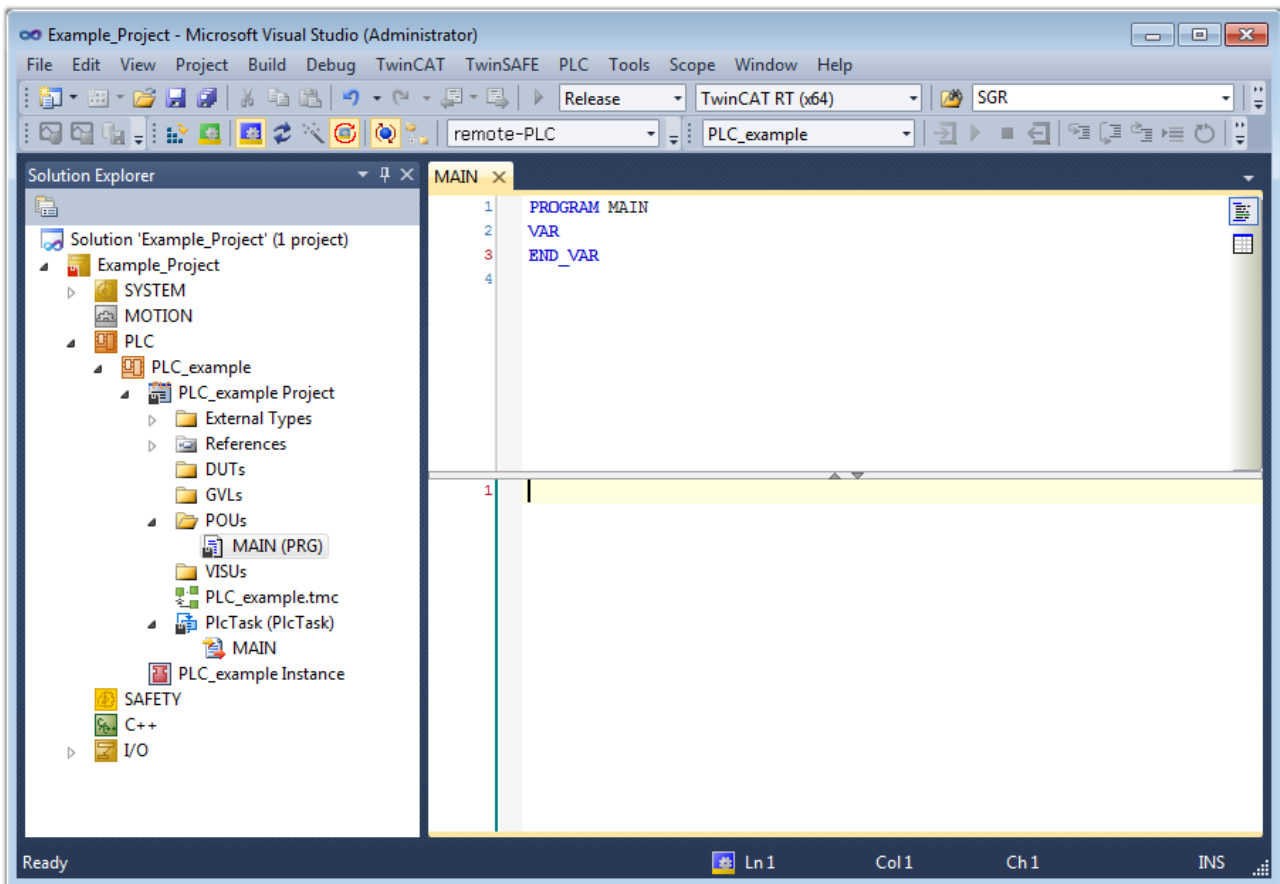


Fig. 53: Initial "Main" program of the standard PLC project

To continue, sample variables and a sample program have now been created:

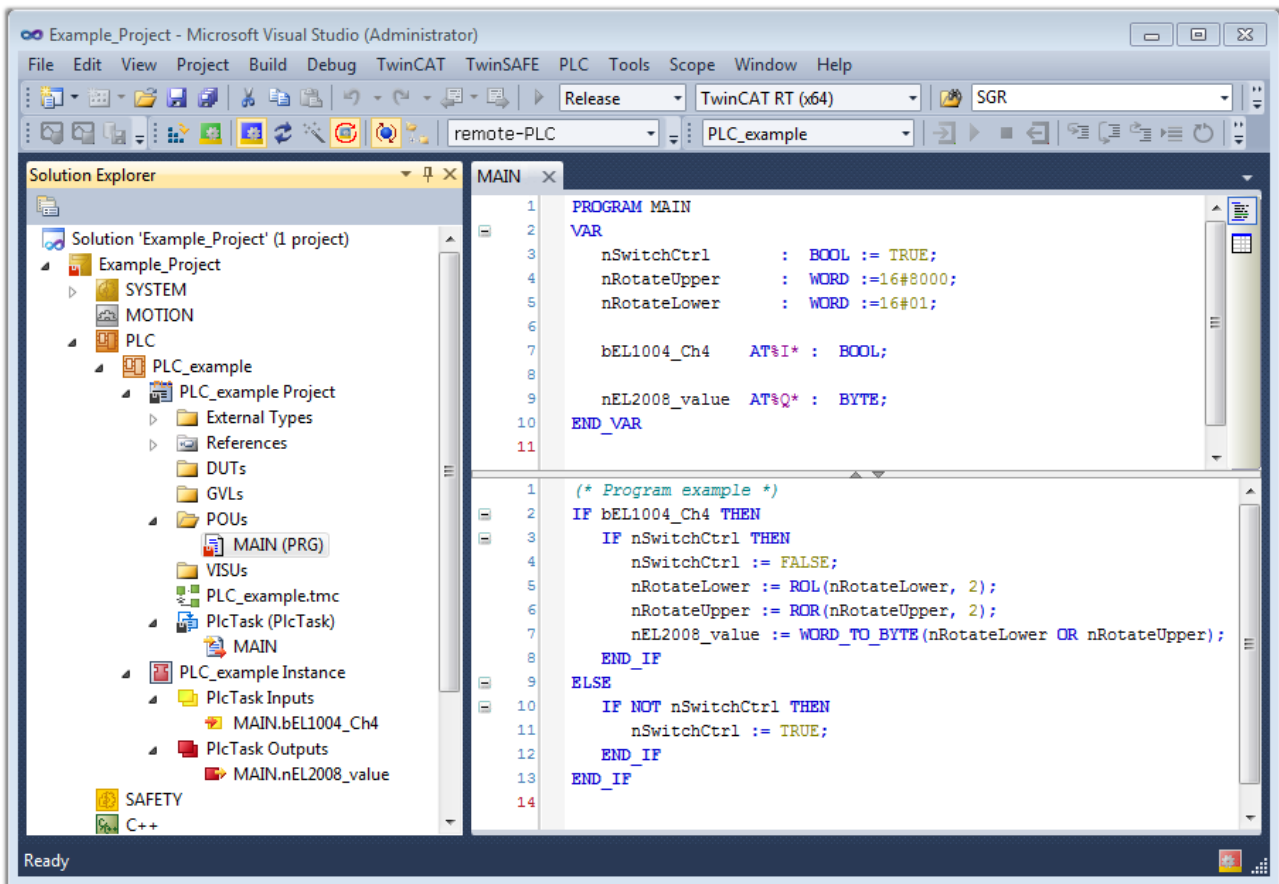


Fig. 54: Sample program with variables after a compile process (without variable integration)

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

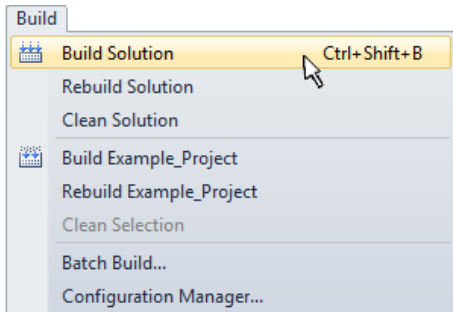
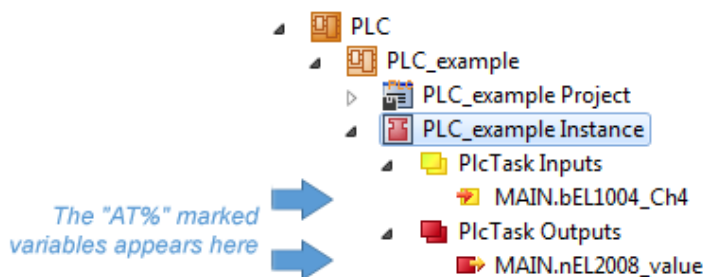


Fig. 55: Start program compilation

The following variables, identified in the ST/ PLC program with "AT%", are then available in 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 for selecting a suitable process object (PDO) for linking:

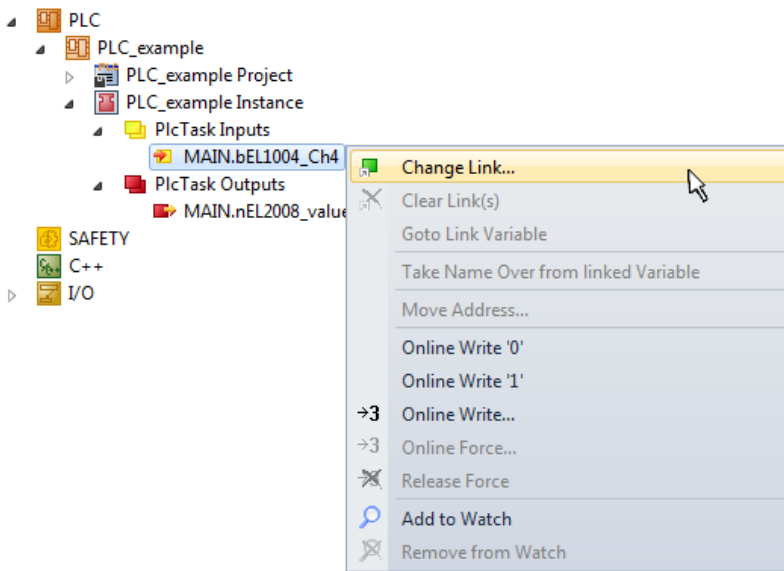


Fig. 56: Creating the links between PLC variables and process objects

In the window that opens, the process object for the variable "bEL1004\_Ch4" of type BOOL can be selected from the PLC configuration tree:

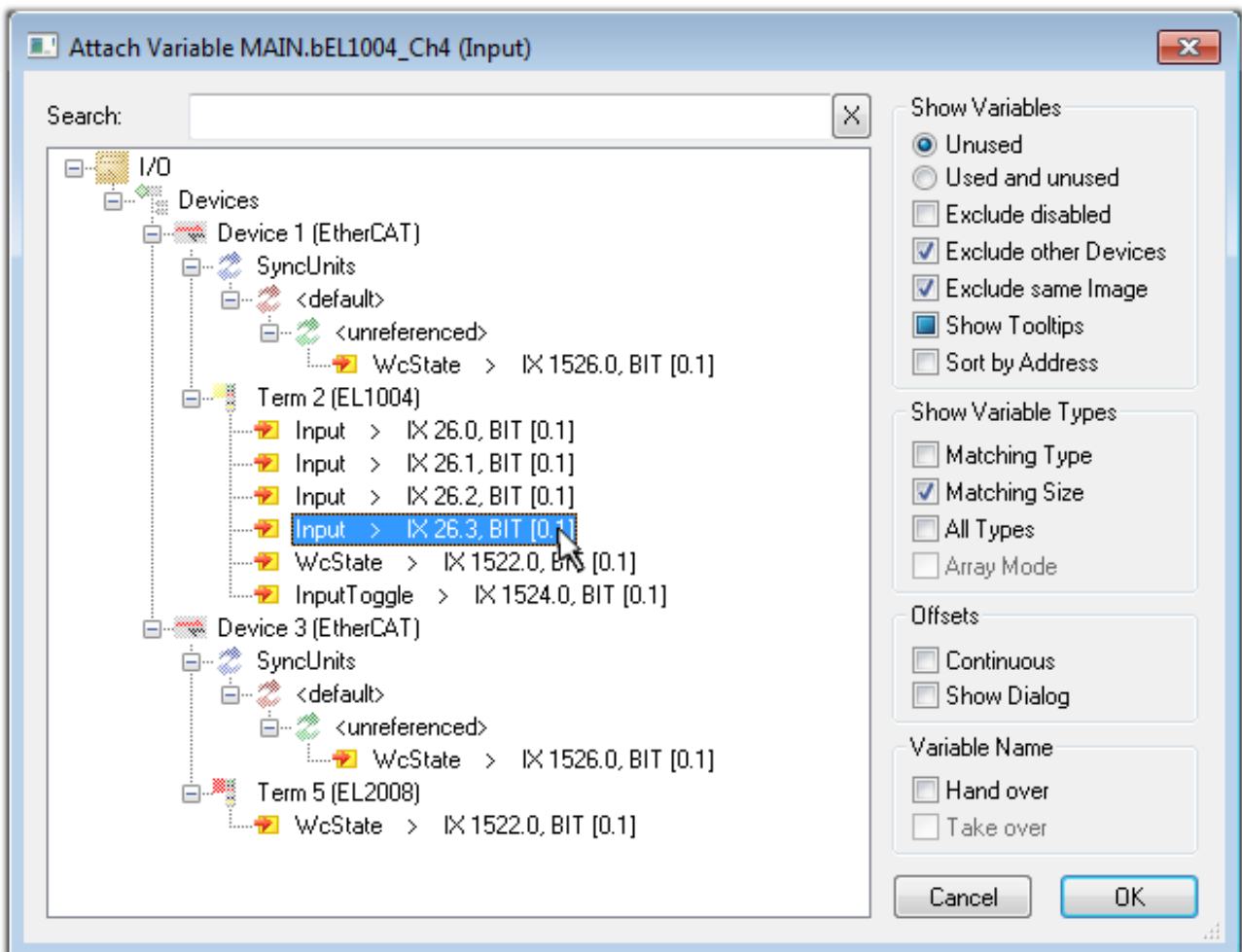


Fig. 57: Selecting PDO of type BOOL

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

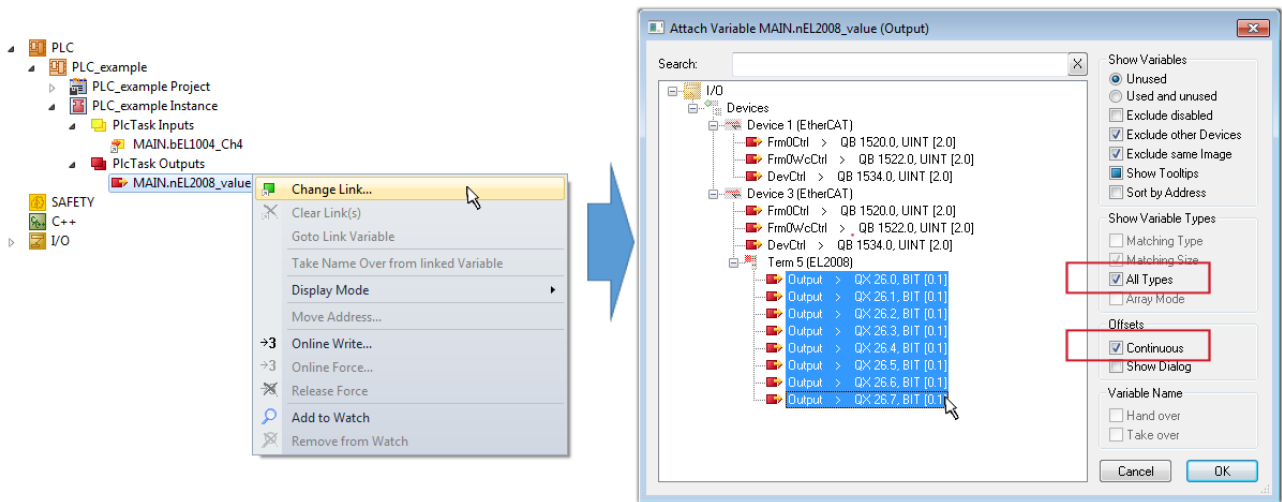



Fig. 58: Selecting several PDOs simultaneously: activate "Continuous" and "All types"

Note that the "Continuous" checkbox was also activated. This is designed to allocate the bits contained in the byte of the variable "nEL2008\_value" sequentially to all eight selected output bits of the EL2008 terminal. In this way it is 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 (  ) at the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting a "Goto Link Variable" from the context menu of a variable. The object opposite, in this case the PDO, is automatically selected:

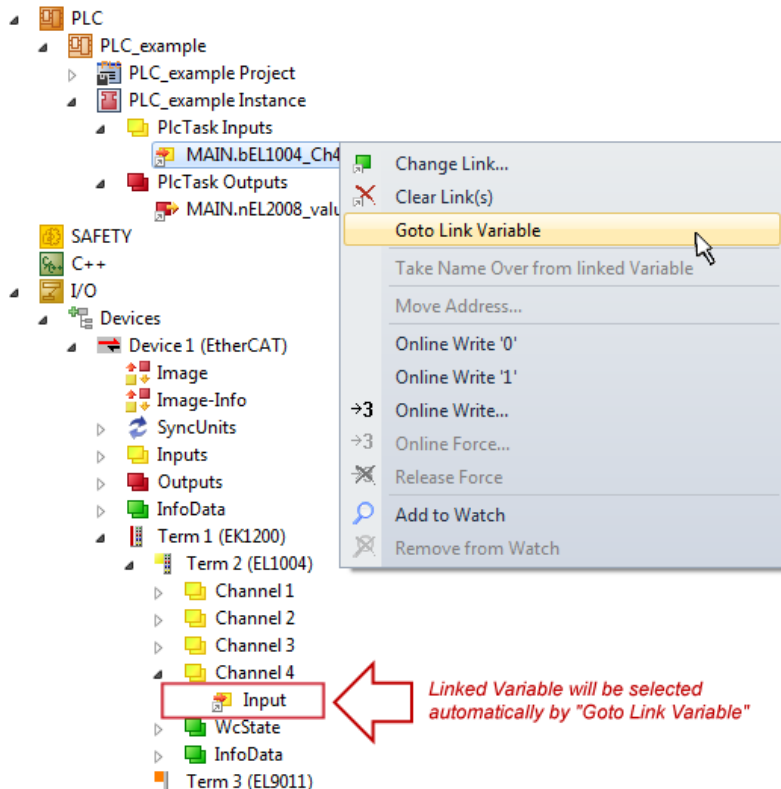


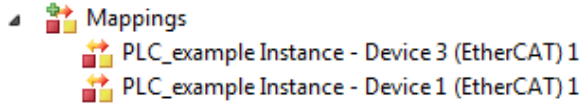
Fig. 59: Application of a "Goto Link" variable, using "MAIN.bEL1004\_Ch4" as a sample

The process of creating links can also take place in the opposite direction, i.e. starting with individual PDOs to variable. However, in this example it would then 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 possible to allocate this a set of bit-standardised variables (type "BOOL"). Here, too, a "Goto Link Variable" from the context menu of a PDO 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 with  or via the menu under "TwinCAT" in order to transfer settings of the development environment to the runtime system. Confirm the messages "Old configurations are 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

program start by click on symbol , the "F5" key or via "PLC" in the menu selecting "Start". The started programming environment shows the runtime values of individual variables:

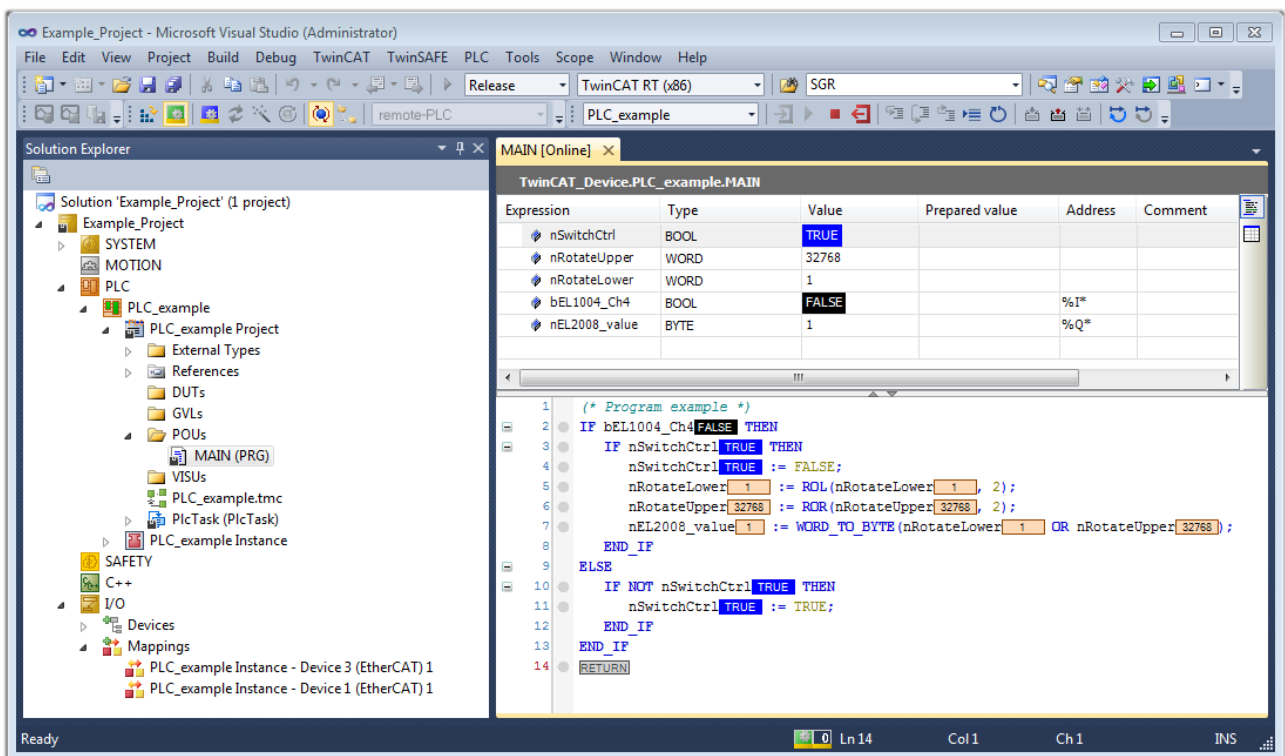


Fig. 60: TwinCAT development environment (VS shell): logged-in, after program startup

The two operator control elements for stopping  and logout  result in the required action (accordingly also for stop "Shift + F5", 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- und 64-Bit-Operatingsystem
  - 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. One option is described here.

In the System Manager call up the TwinCAT overview of the local network interfaces via Options → Show Real Time Ethernet Compatible Devices.

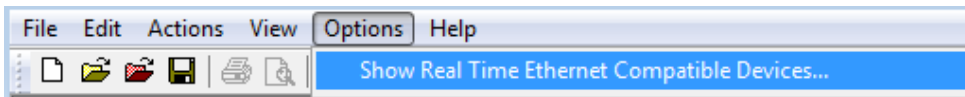


Fig. 61: System Manager "Options" (TwinCAT 2)

This has to be called up by the Menü "TwinCAT" within the TwinCAT 3 environment:

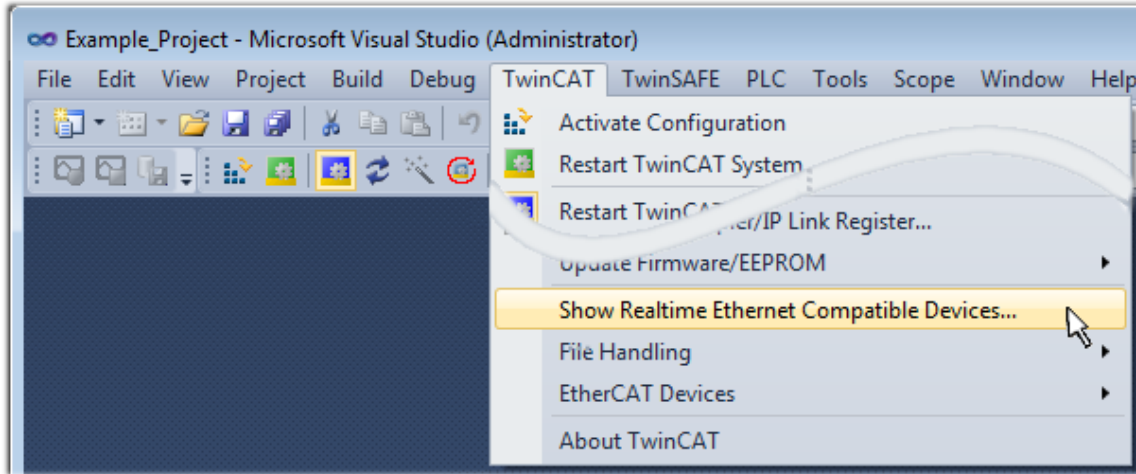


Fig. 62: Call up under VS Shell (TwinCAT 3)

The following dialog appears:

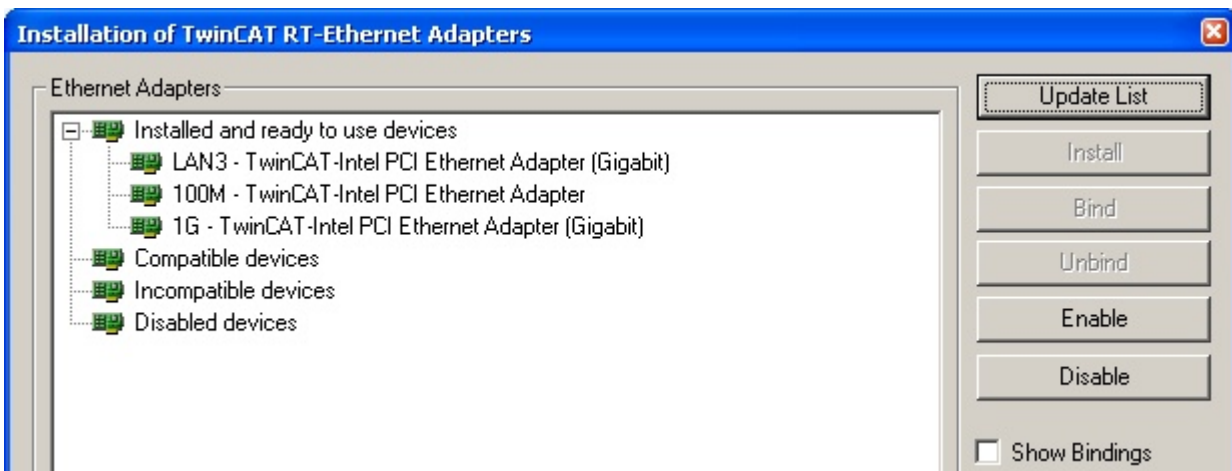


Fig. 63: Overview of network interfaces

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

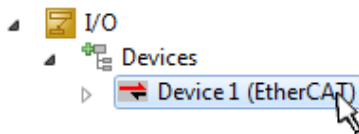
A Windows warning regarding the unsigned driver can be ignored.

**Alternatively** an EtherCAT-device can be inserted first of all as described in chapter [Offline configuration creation, section "Creating the EtherCAT device" \[▶ 66\]](#) in order to view the compatible ethernet ports via its EtherCAT properties (tab „Adapter“, button „Compatible Devices...“):



Fig. 64: 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)

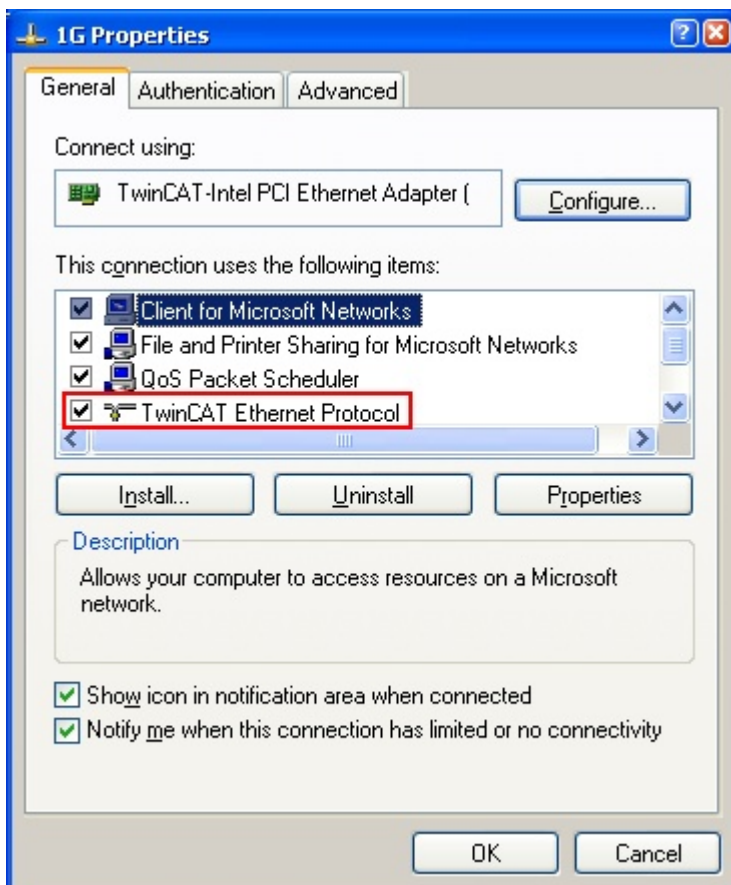


Fig. 65: Windows properties of the network interface

A correct setting of the driver could be:

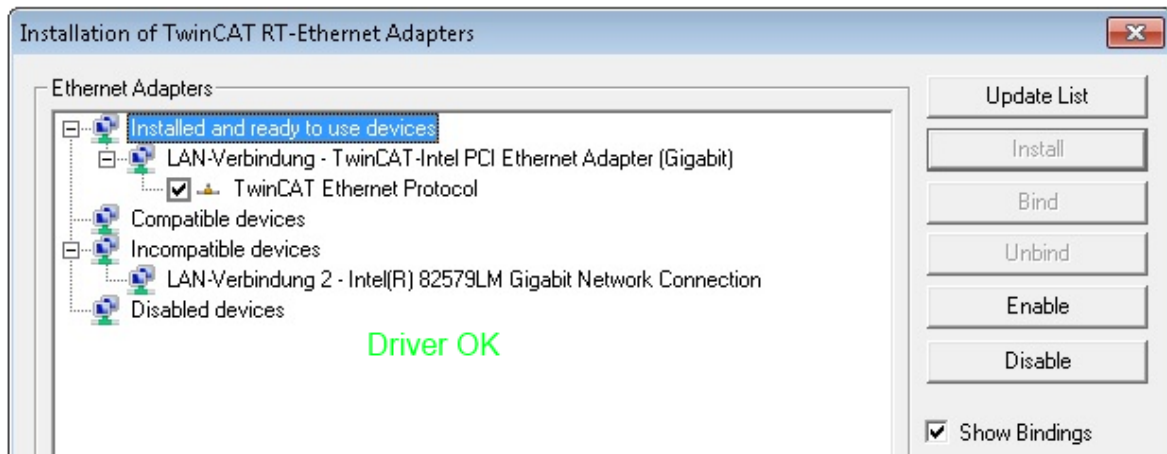


Fig. 66: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

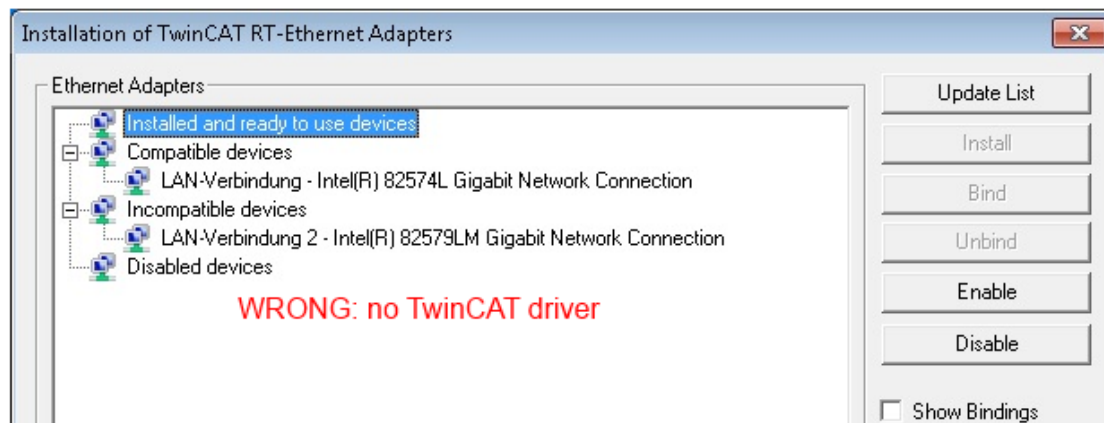
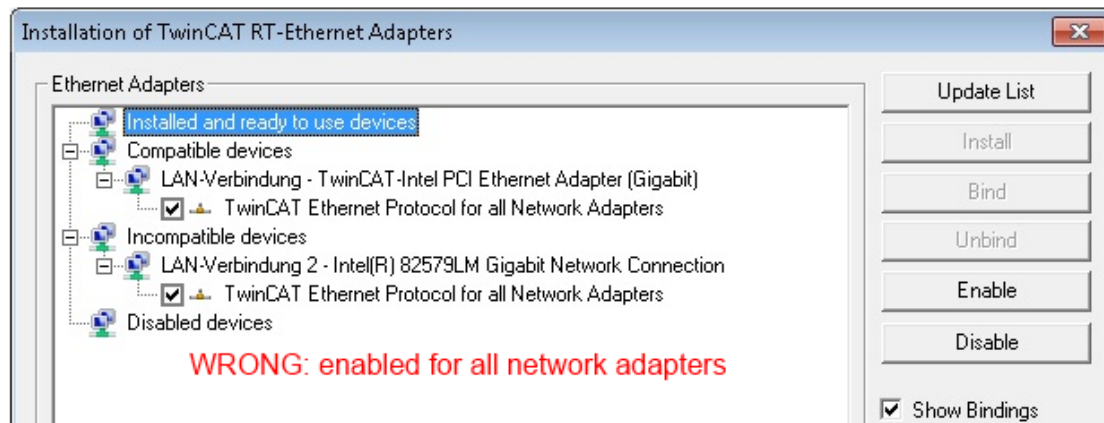
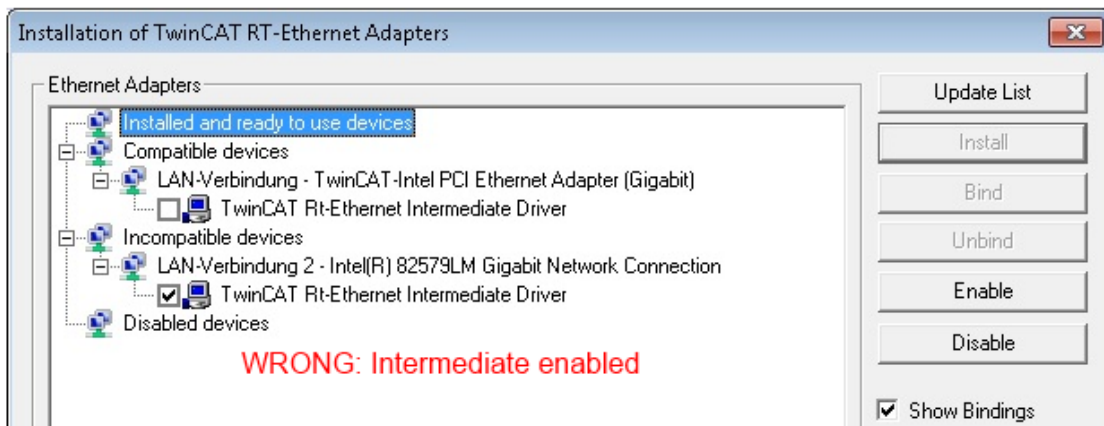
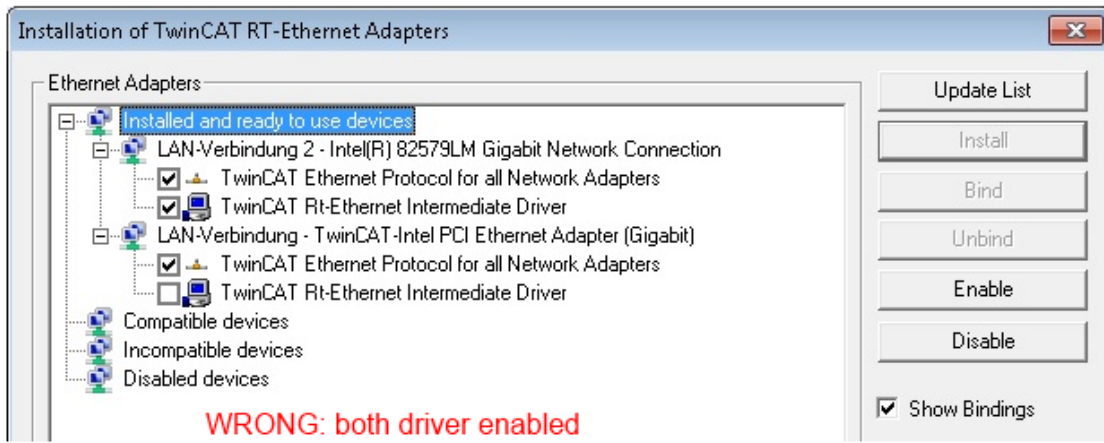


Fig. 67: Incorrect driver settings for the Ethernet port

## IP address of the port used



## Note

## 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.

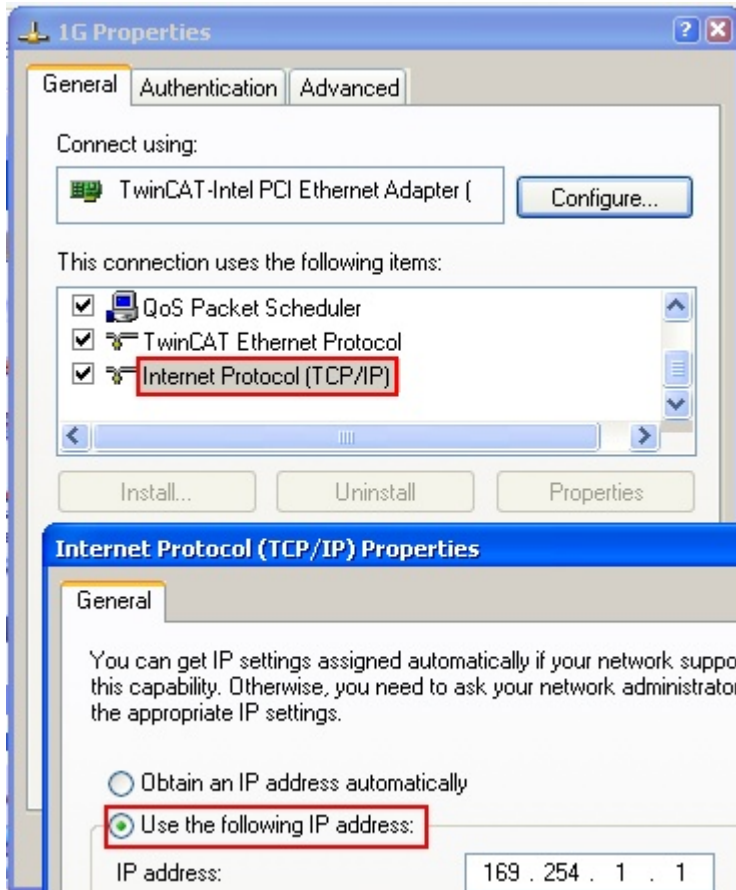


Fig. 68: TCP/IP setting for the Ethernet port

## 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 \[▶ 65\]](#) is available for this purpose.

 <b>Note</b>	<b>ESI</b> 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”

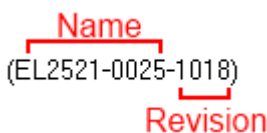


Fig. 69: Identifier structure

The order identifier consisting of name + type (here: EL2521-0010) 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 \[▶ 7\]](#).

## 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.

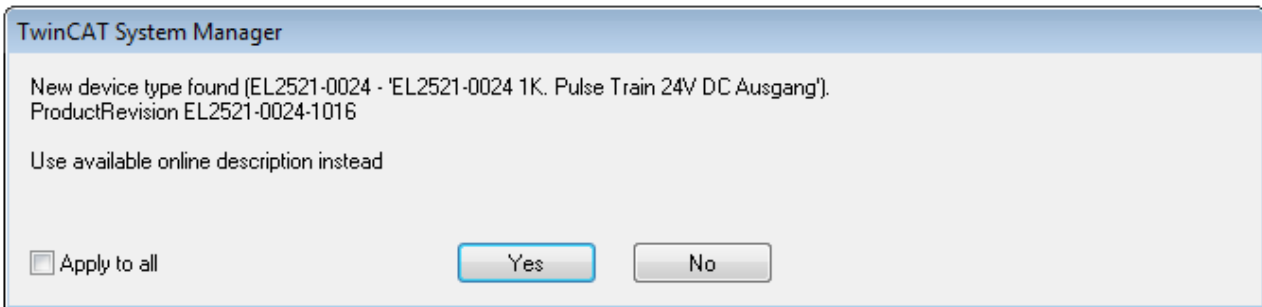


Fig. 70: *OnlineDescription information window (TwinCAT 2)*

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

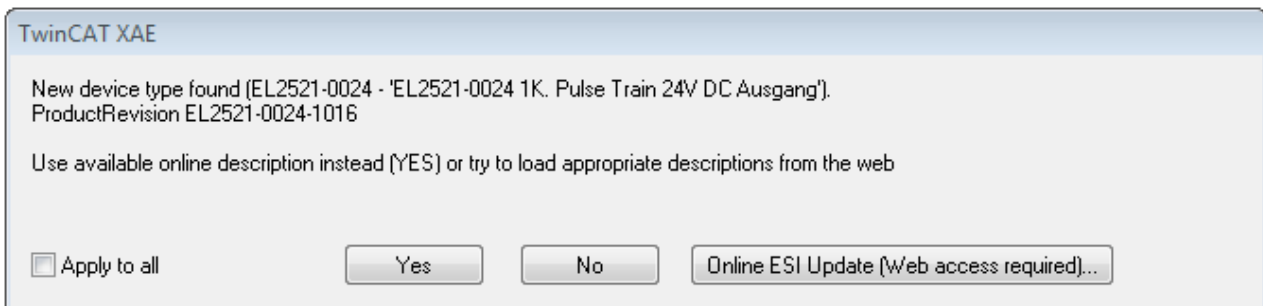



Fig. 71: *Information window OnlineDescription (TwinCAT 3)*

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.

 <b>Attention</b>	<p><b>Changing the 'usual' configuration through a scan</b></p> <ul style="list-style-type: none"> <li>✓ 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             <ol style="list-style-type: none"> <li>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).</li> <li>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.</li> </ol> </li> </ul>
---	---

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](#)' [▶ 66].

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.

OnlineDescriptionCache00000002.xml

Fig. 72: File OnlineDescription.xml created by the System Manager

If 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").

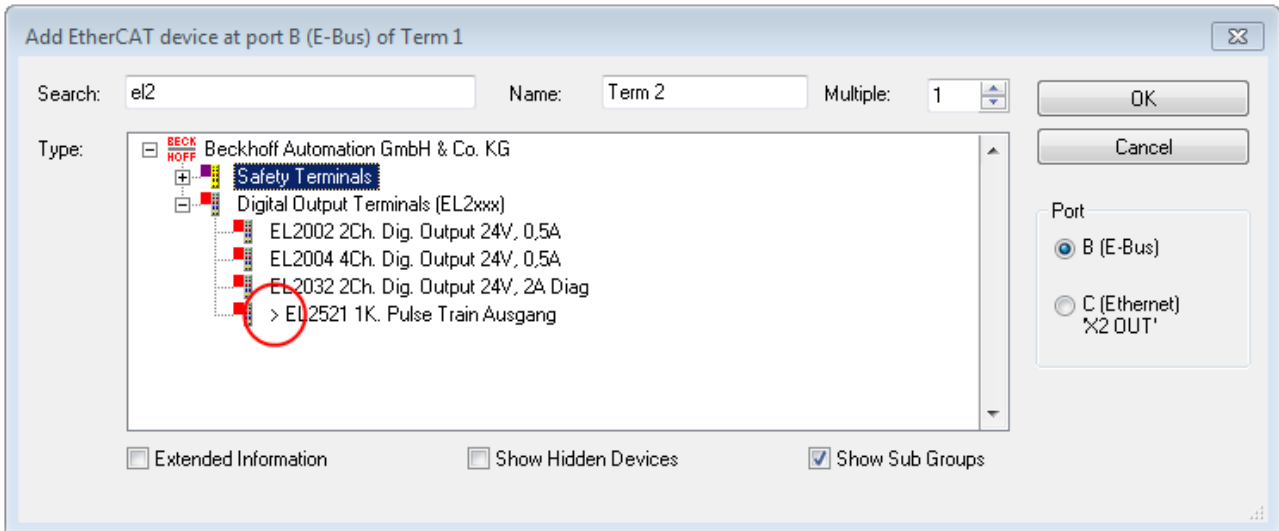



Fig. 73: Indication of an online recorded ESI of EL2521 as an example

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

 <b>Note</b>	<p><b>OnlineDescription for TwinCAT 3.x</b></p> <p>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:</p> <p><code>C:\User\{USERNAME}\AppData\Roaming\Beckhoff\TwinCAT3\Components\Base\EtherCATCache.xml</code></p> <p>(Please note the language settings of the OS!)</p> <p>You have to delete this file, too.</p>
--	---

**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.

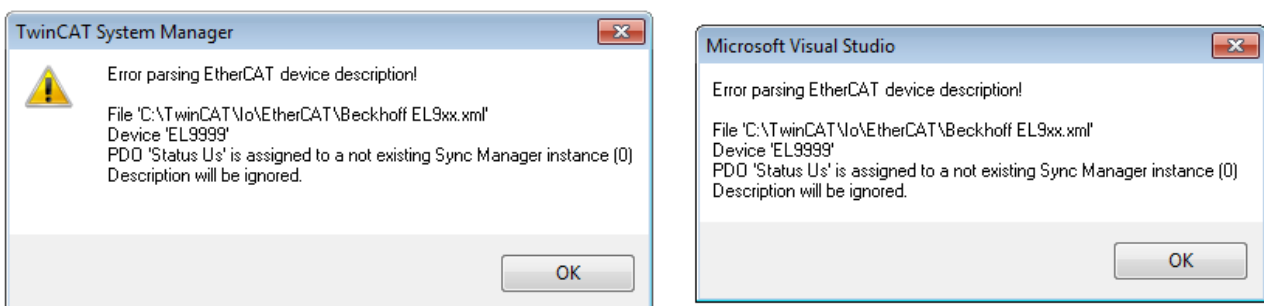


Fig. 74: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

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:

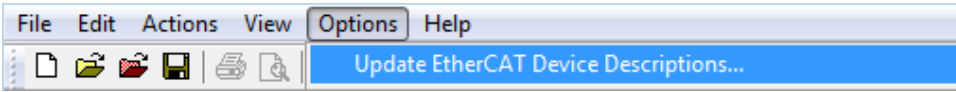


Fig. 75: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:

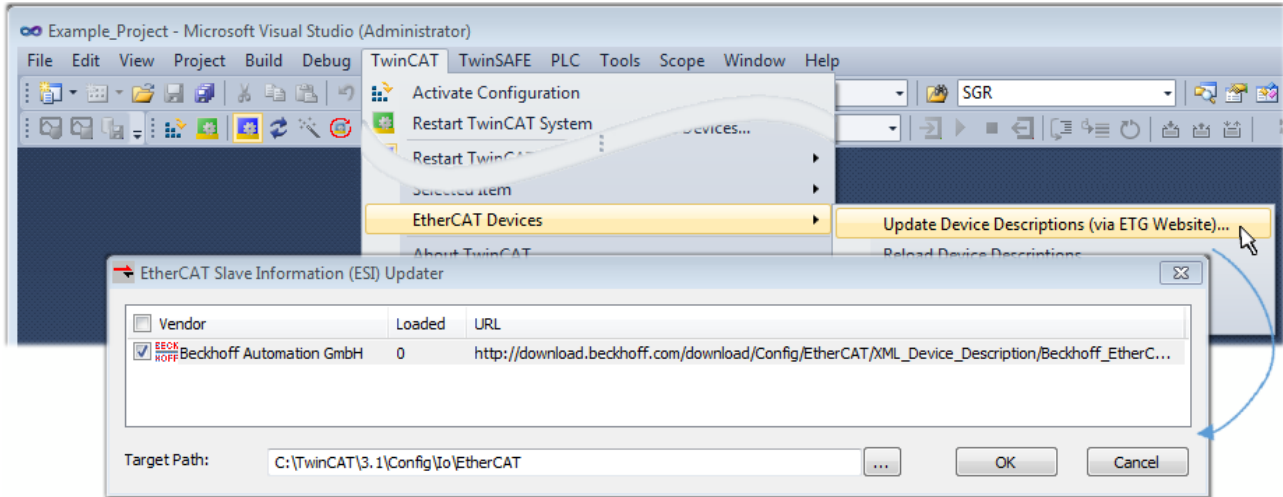


Fig. 76: Using the ESI Updater (TwinCAT 3)

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 directory (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” [▶ 61].

**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 [▶ 71] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 72]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 75]

The scan with existing configuration [▶ 76] 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.

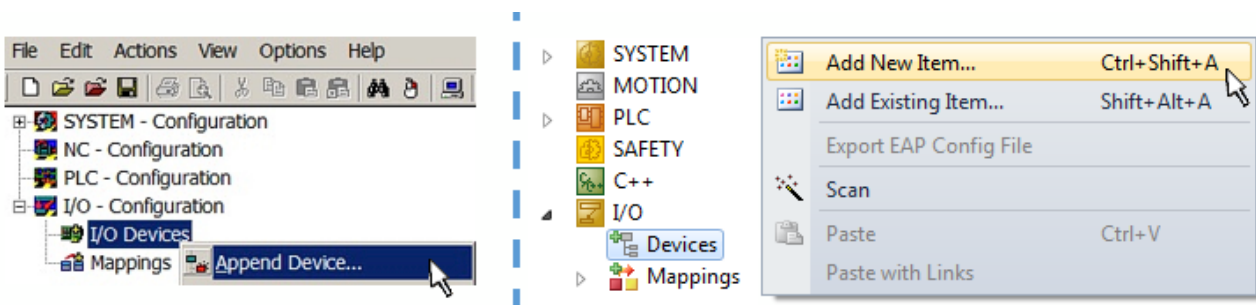


Fig. 77: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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".

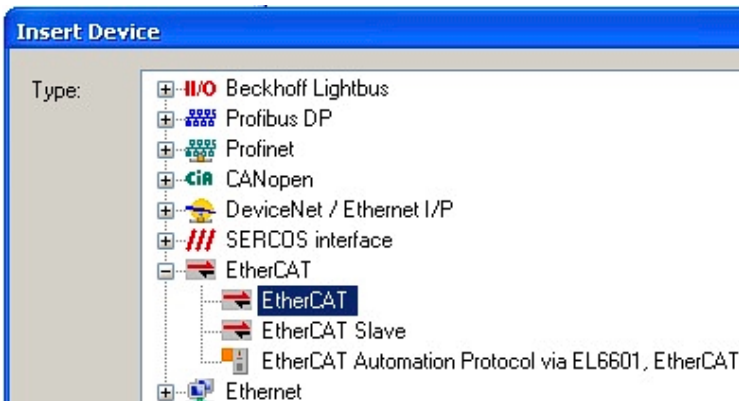


Fig. 78: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

Then assign a real Ethernet port to this virtual device in the runtime system.

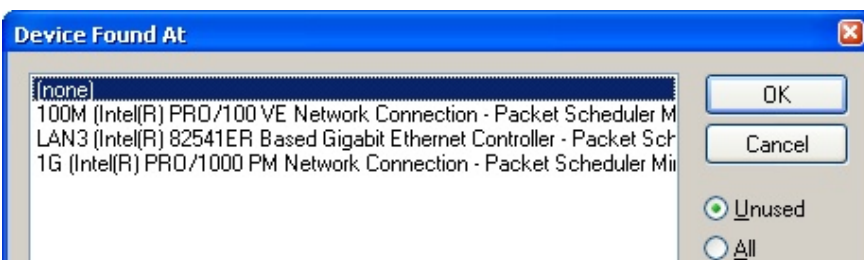


Fig. 79: Selecting the Ethernet port

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)”.

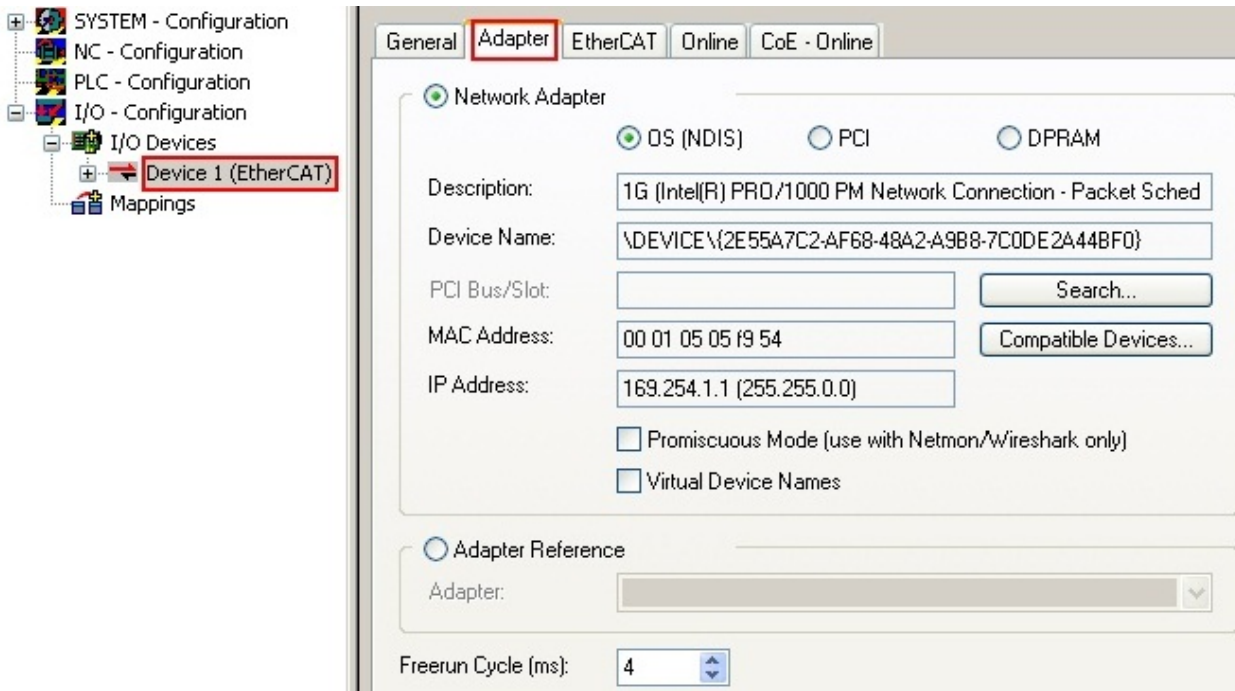
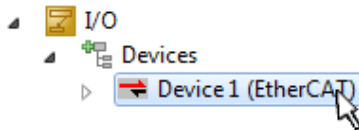



Fig. 80: EtherCAT device properties (TwinCAT 2)

TwinCAT 3: the properties of the EtherCAT device can be opened by double click on “Device .. (EtherCAT)” within the Solution Explorer under “I/O”:



 <b>Note</b>	<p><b>Selecting the Ethernet port</b></p> <p>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 <a href="#">installation page [p. 55]</a>.</p>
--	--

**Defining EtherCAT slaves**

Further devices can be appended by right-clicking on a device in the configuration tree.

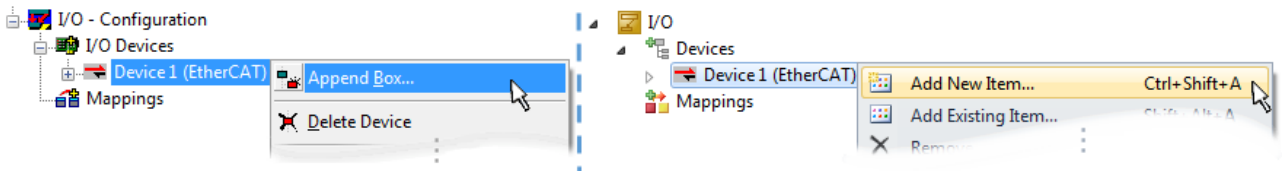


Fig. 81: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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: EK couplers, EP boxes, devices with RJ45/M8/M12 connector

- “E-Bus”: LVDS “terminal bus”, “EJ-module”: EL/ES terminals, various modular modules

The search field facilitates finding specific devices (since TwinCAT 2.11 or TwinCAT 3).

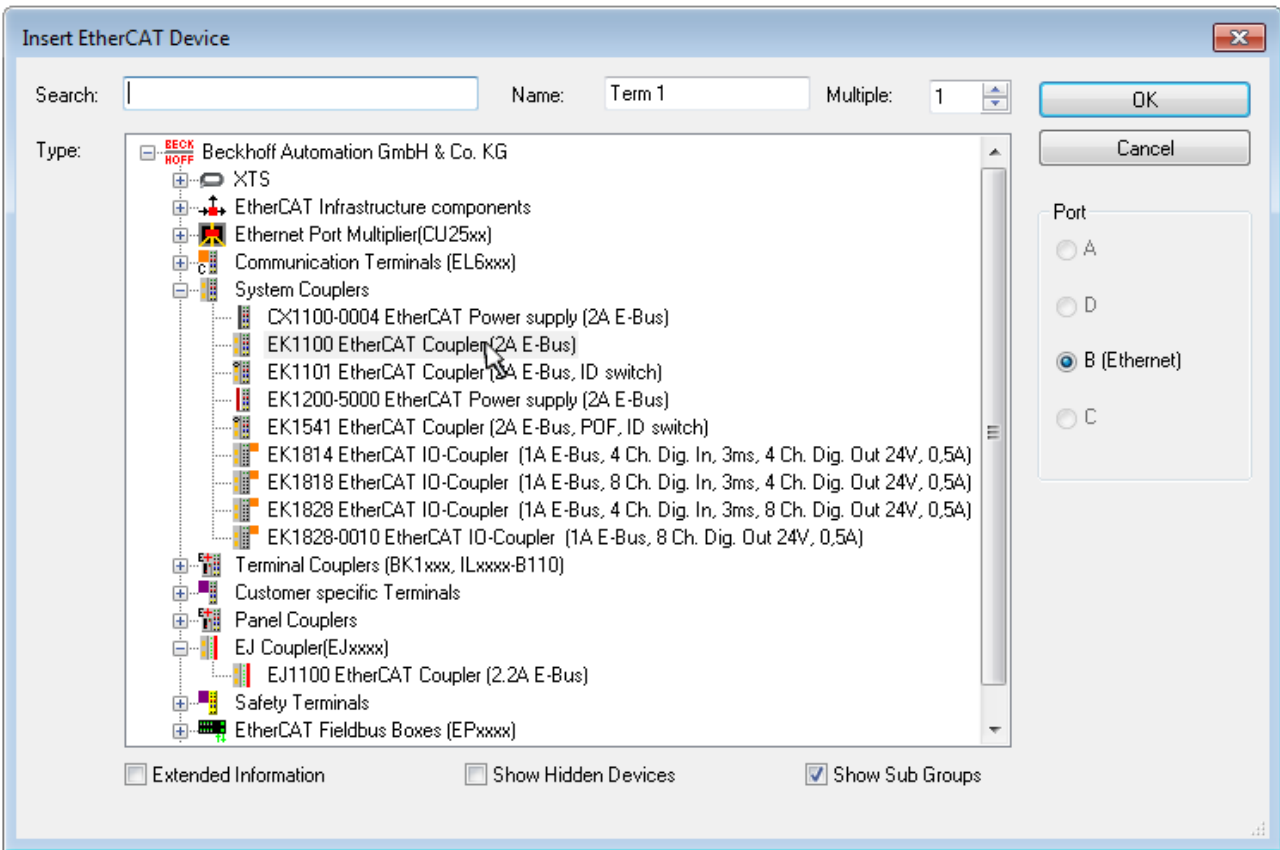


Fig. 82: Selection dialog for new EtherCAT device

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”.

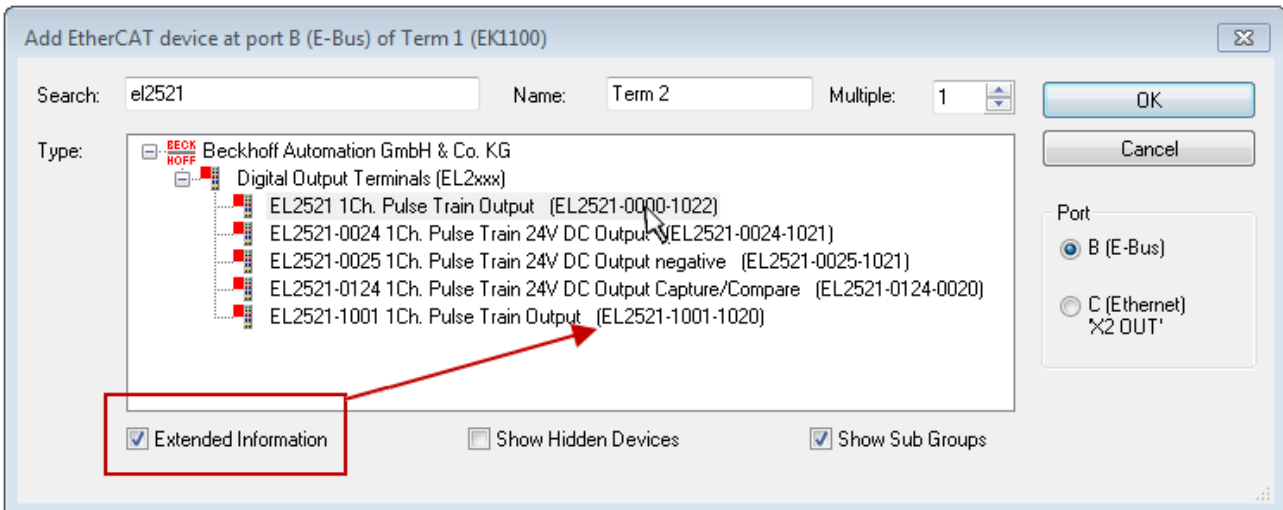


Fig. 83: Display of device revision

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”.

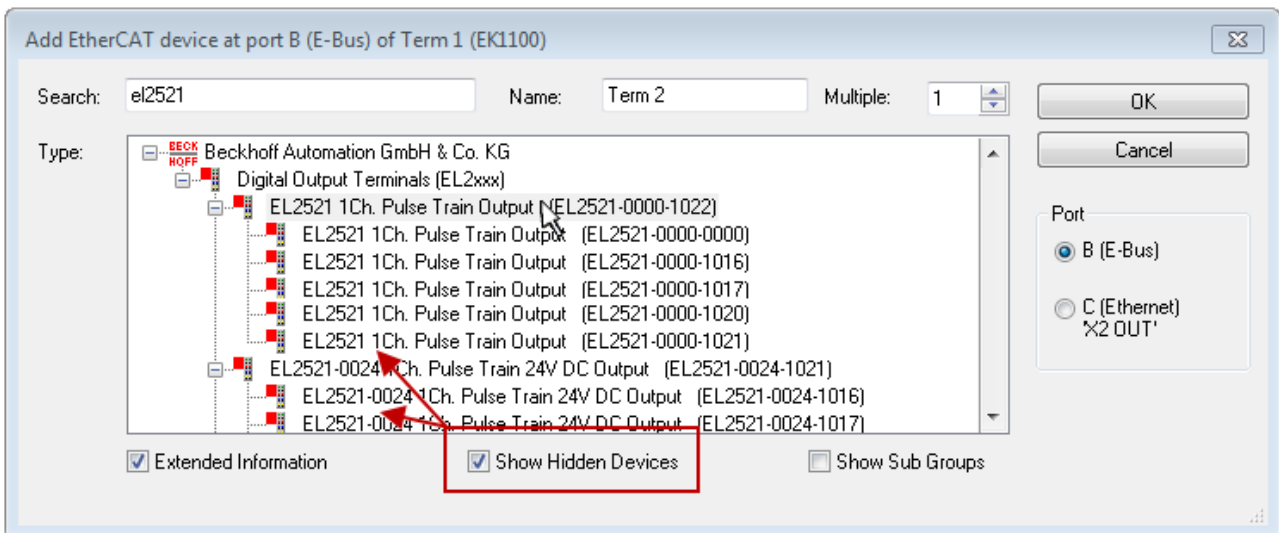



Fig. 84: Display of previous revisions

 <b>Note</b>	<p><b>Device selection based on revision, compatibility</b></p> <p>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:</p> <p><b>device revision in the system &gt;= device revision in the configuration</b></p> <p>This also enables subsequent replacement of devices without changing the configuration (different specifications are possible for drives).</p>
--	--

**Example:**

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

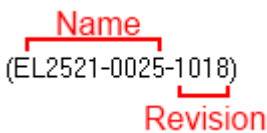


Fig. 85: Name/revision of the terminal

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 parameterised as follows: linking with the task, CoE/DC settings, plug-in definition, startup settings, ...

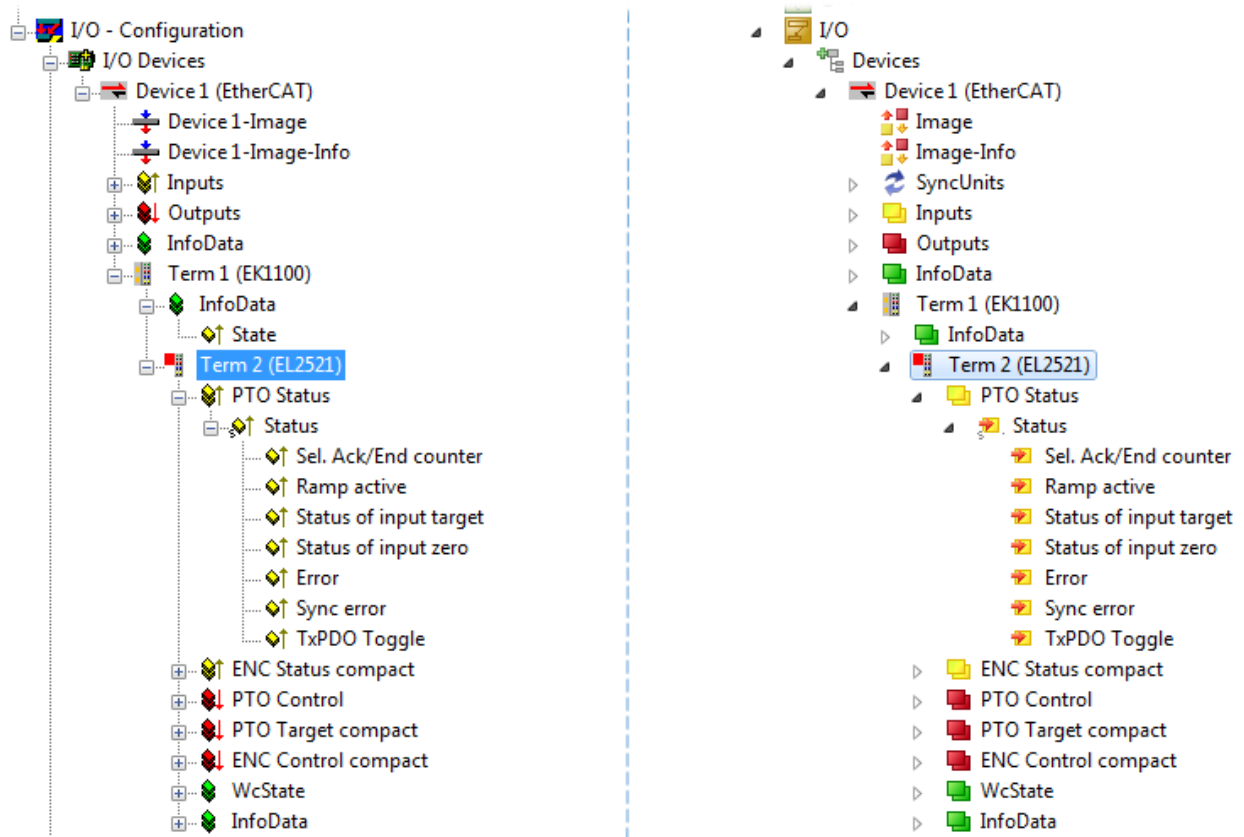




Fig. 86: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)



## 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:  .
- 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)”

 <b>Note</b>	<p><b>Online scanning in Config mode</b></p> <p>The online search is not available in RUN mode (production operation). Note the differentiation between TwinCAT programming system and TwinCAT target system.</p>
--	---

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.

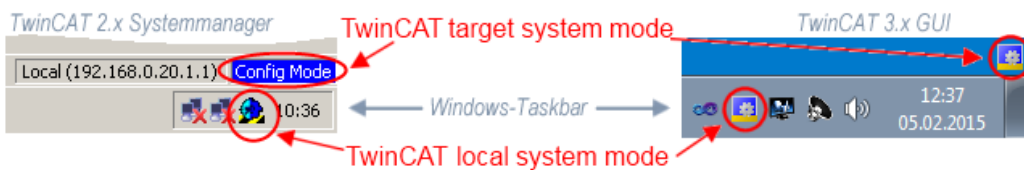


Fig. 87: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

Right-clicking on “I/O Devices” in the configuration tree opens the search dialog.

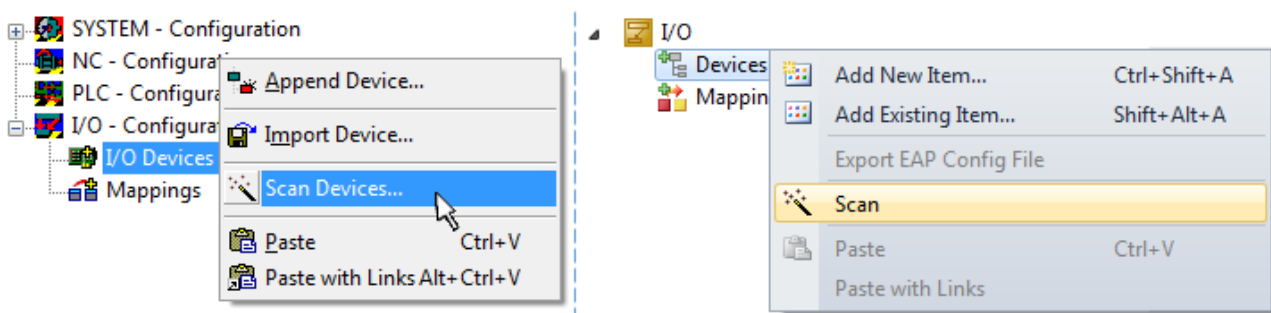


Fig. 88: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

This scan mode attempts to find not only EtherCAT devices (or Ethernet ports that are usable as such), but also NOVDRAM, fieldbus cards, SMB etc. However, not all devices can be found automatically.

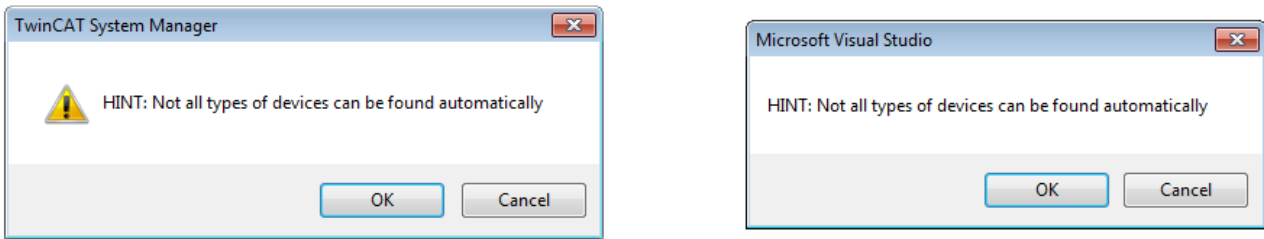


Fig. 89: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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” .

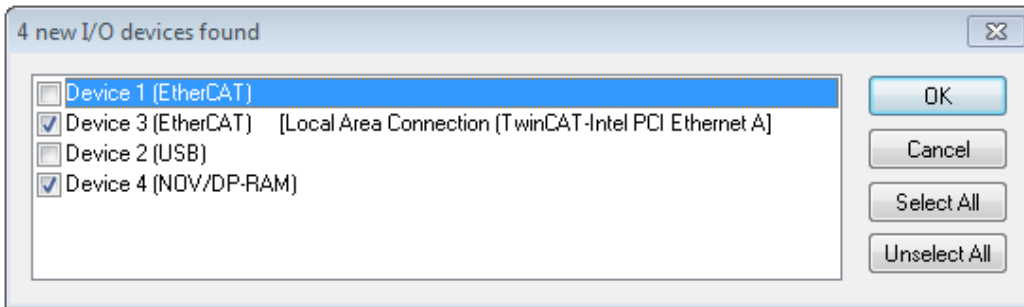




Fig. 90: Detected Ethernet devices

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”.

 <b>Note</b>	<p><b>Selecting the Ethernet port</b></p> <p>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 <a href="#">installation page</a>  &gt; 55].</p>
--	--

**Detecting/Scanning the EtherCAT devices**

 <b>Note</b>	<p><b>Online scan functionality</b></p> <p>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.</p>
--	--

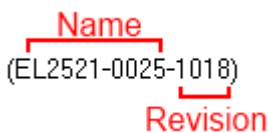



Fig. 91: Example default state

 <b>Attention</b>	<p><b>Slave scanning in practice in series machine production</b></p> <p>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 <a href="#">comparison</a>  &gt; 76] 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.</p>
---	--

**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:

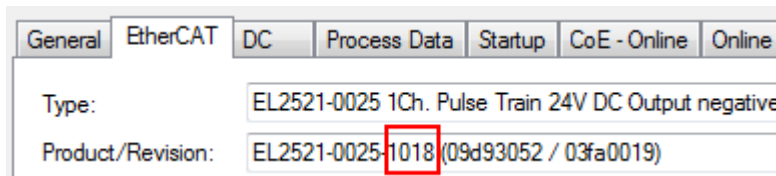


Fig. 92: Installing EtherCAT terminal with revision -1018

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 comparative scan [▶ 76] 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:

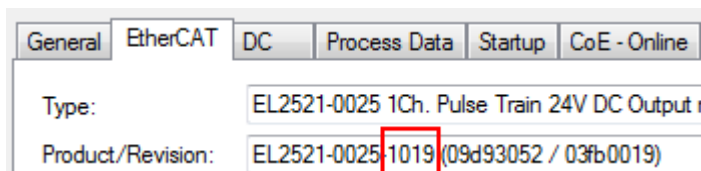


Fig. 93: Detection of EtherCAT terminal with revision -1019

This is usually not noticed by the commissioning engineers. TwinCAT cannot signal anything either, since virtually a new configuration is 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. 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. 94: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

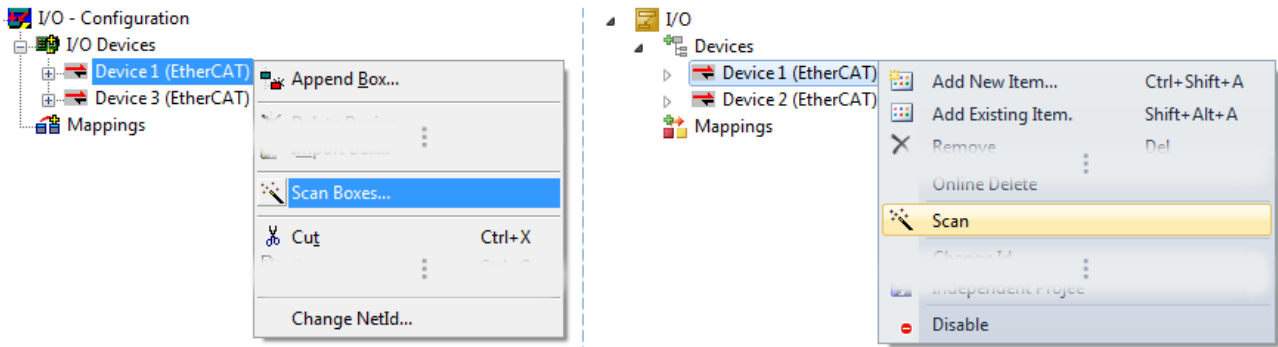


Fig. 95: Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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. 96: Scan progress exemplary by TwinCAT 2

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



Fig. 97: 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. 98: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar



Fig. 99: 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”.

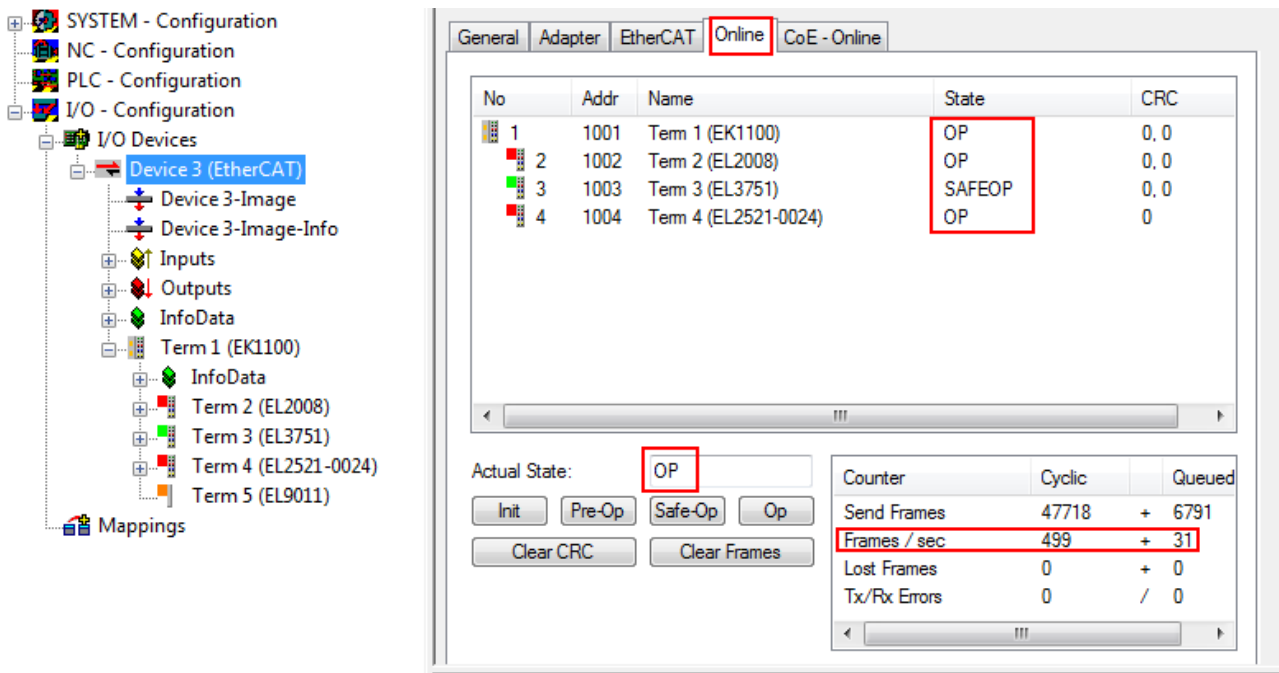


Fig. 100: Online display example

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 [manual procedure \[► 66\]](#).

### 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.

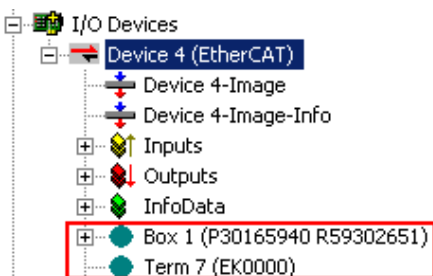



Fig. 101: Faulty identification

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**

 <b>Attention</b>	<p><b>Change of the configuration after comparison</b></p> <p>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.</p>
---	--

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.

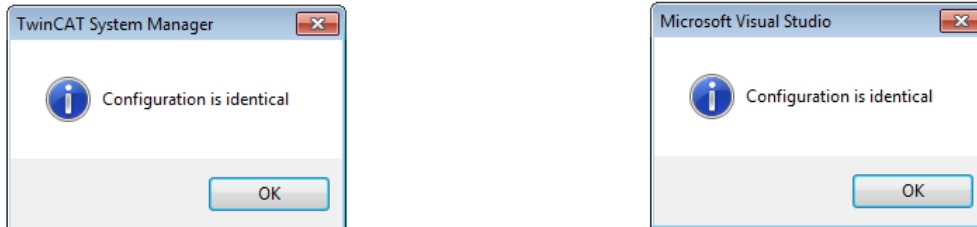


Fig. 102: *Identical configuration (left: TwinCAT 2; right: TwinCAT 3)*

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

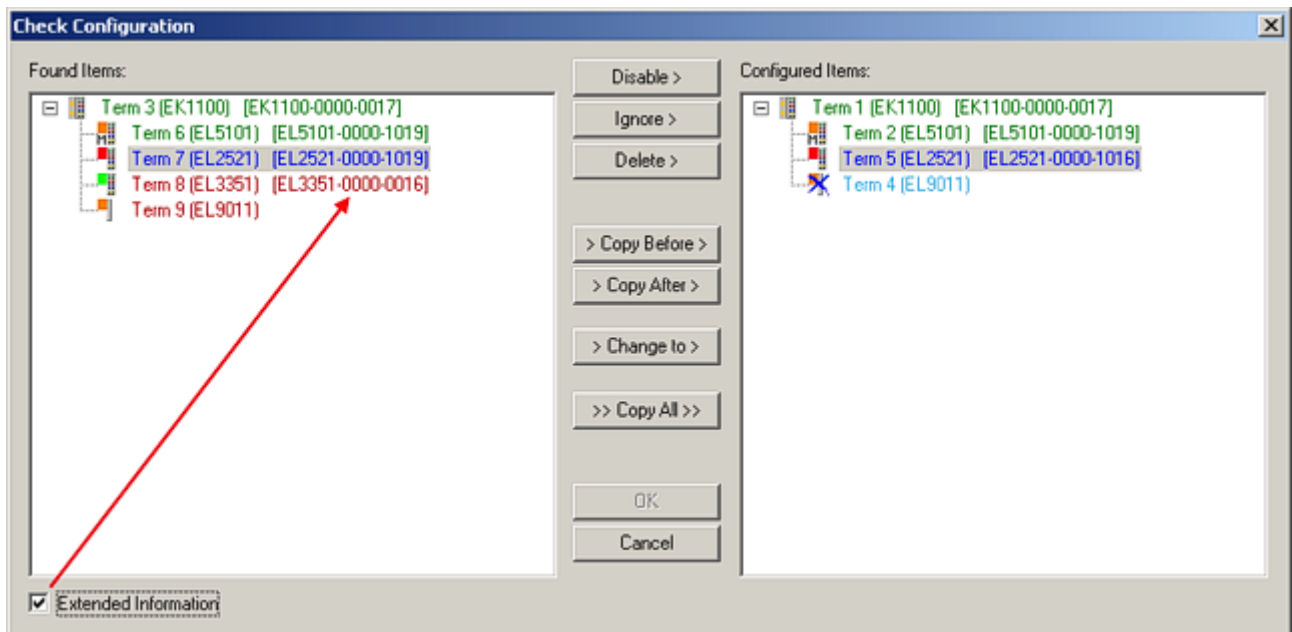



Fig. 103: *Correction dialog*

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

Colour	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	<ul style="list-style-type: none"> <li>This EtherCAT slave is not present on the other side.</li> <li>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.</li> </ul>

 <b>Note</b>	<p><b>Device selection based on revision, compatibility</b></p> <p>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:</p> <p><b>device revision in the system &gt;= device revision in the configuration</b></p> <p>This also enables subsequent replacement of devices without changing the configuration (different specifications are possible for drives).</p>
--	--

**Example:**

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

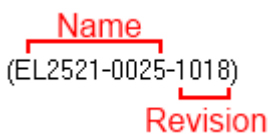


Fig. 104: Name/revision of the terminal

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 parameterised as follows: linking with the task, CoE/DC settings, plug-in definition, startup settings, ...

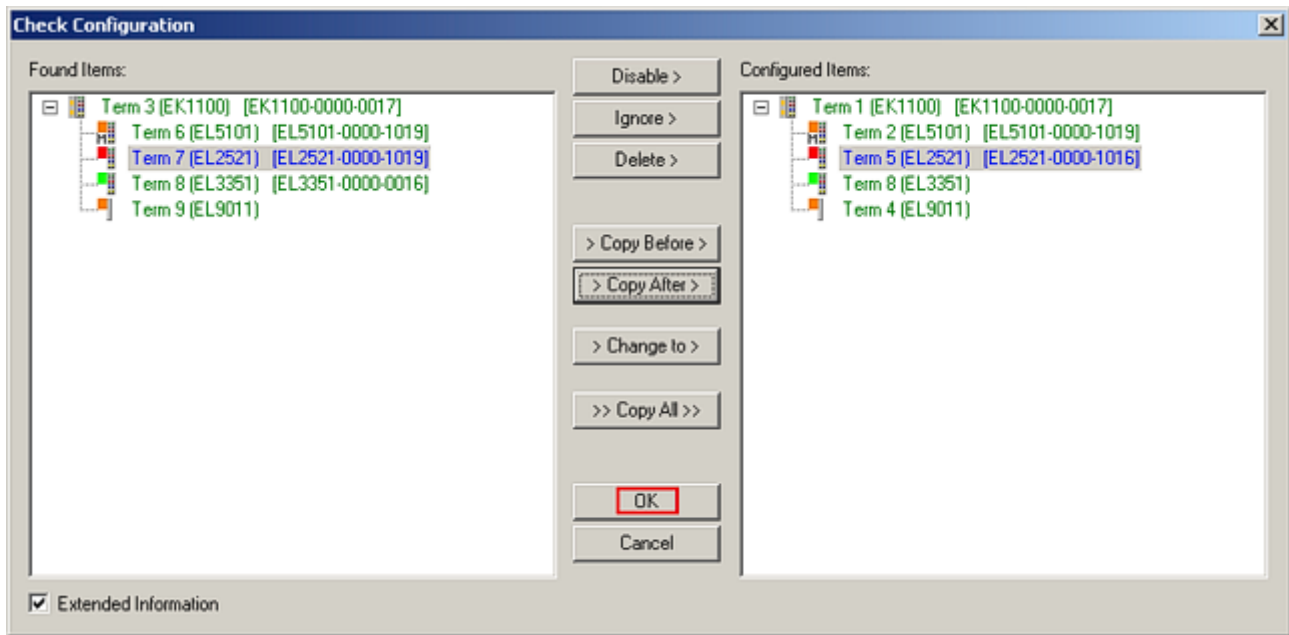


Fig. 105: Correction dialog with modifications

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

### Change to compatible device

TwinCAT offers a function “Change to Compatible Type...” for the exchange of a device whilst retaining the links in the task.

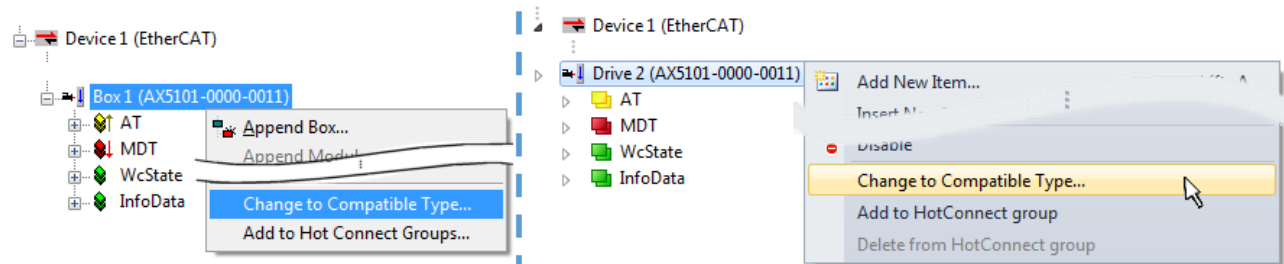


Fig. 106: Dialog “Change to Compatible Type...” (left: TwinCAT 2; right: TwinCAT 3)

This function is preferably to be used on AX5000 devices. If called, the System Manager suggests the devices that it finds in the associated sub-folder; in the case of the AX5000, for example, in `\TwinCAT\IO\EtherCAT\Beckhoff AX5xxx`.

### Change to Alternative Type

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

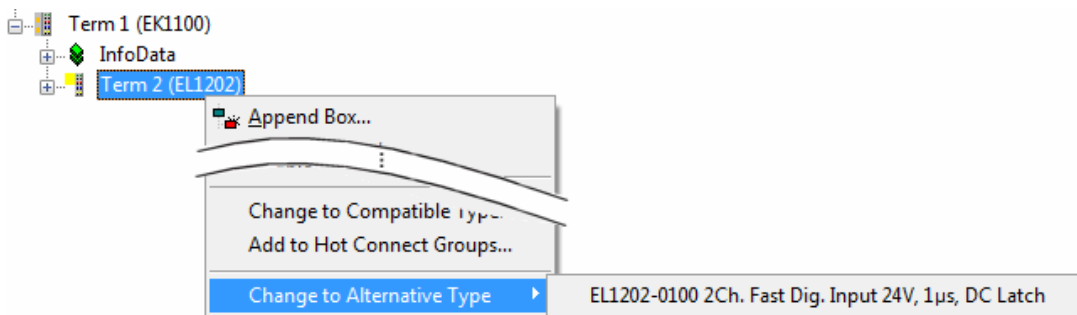


Fig. 107: TwinCAT 2 Dialog Change to Alternative Type

If called, the System Manager searches in the procured device ESI (in this example: EL 1202-0000) 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).

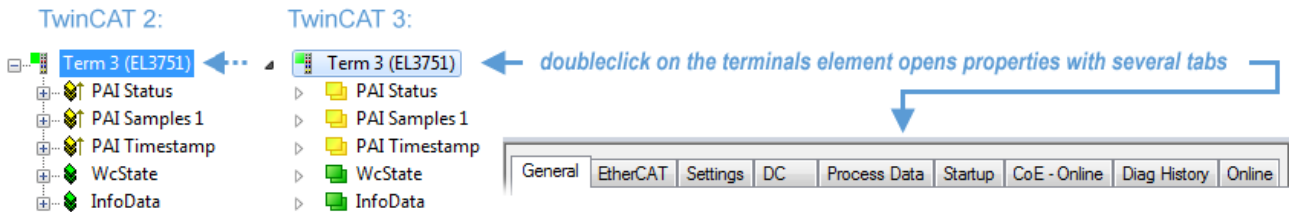


Fig. 108: Branch element as terminal EL3751

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

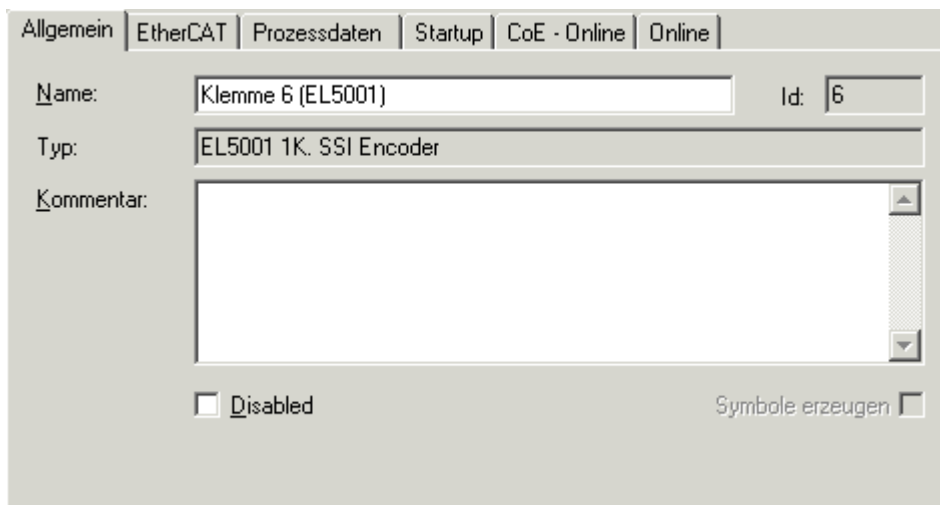


Fig. 109: "General" tab

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

## „EtherCAT“ tab

Fig. 110: „EtherCAT“ tab

<b>Type</b>	EtherCAT device type
<b>Product/Revision</b>	Product and revision number of the EtherCAT device
<b>Auto Inc Addr.</b>	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 <sub>hex</sub> . For each further slave the address is decremented by 1 (FFFF <sub>hex</sub> , FFFE <sub>hex</sub> etc.).
<b>EtherCAT Addr.</b>	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.
<b>Previous Port</b>	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.
<b>Advanced Settings</b>	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.

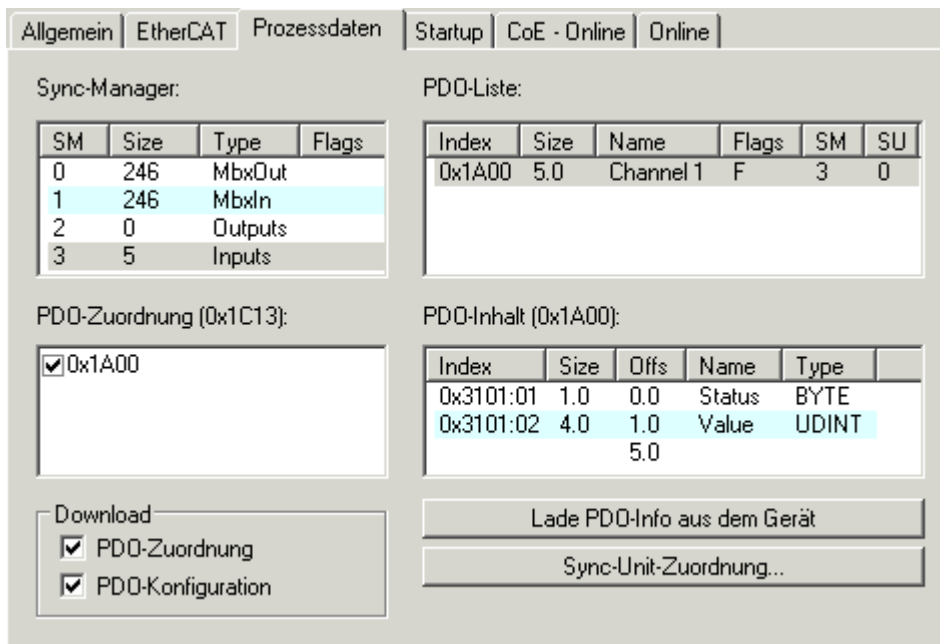


Fig. 111: "Process Data" tab

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").

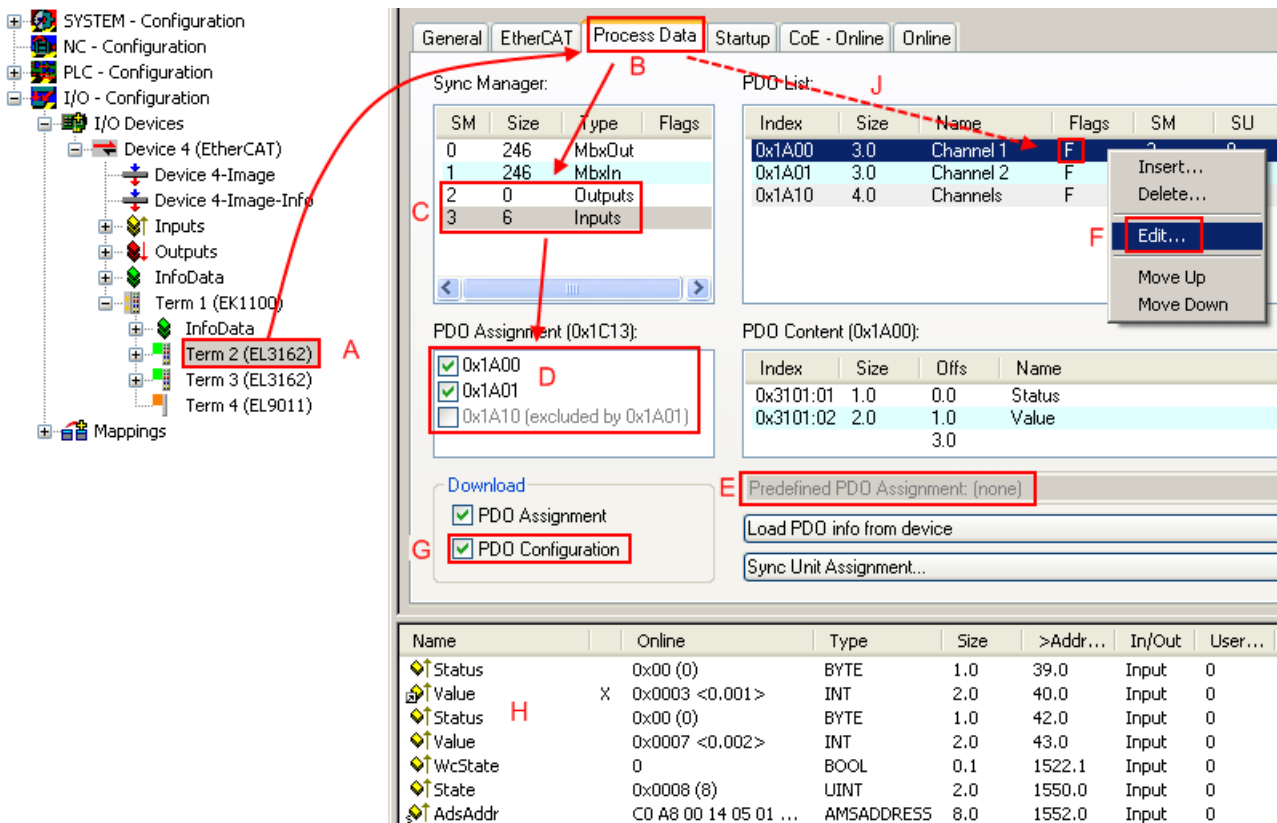


Fig. 112: Configuring the process data

i

**Note**

**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.

A detailed description [▶ 87] can be found at the end of this section.

**„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.

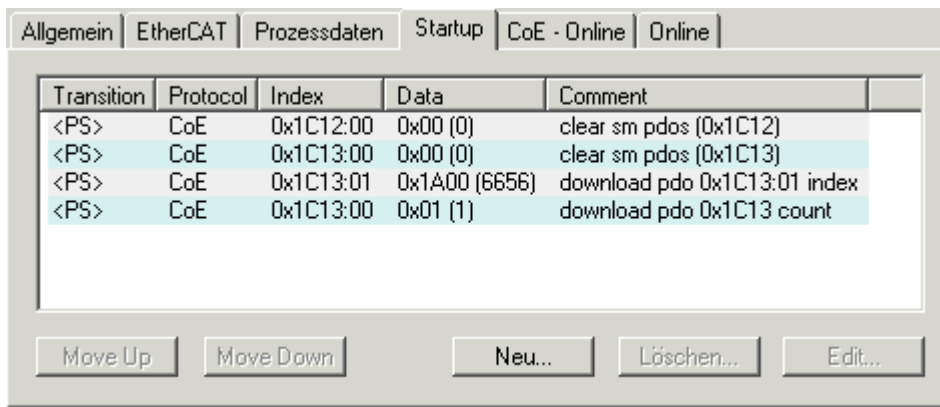


Fig. 113: „Startup“ tab

Column	Description
Transition	Transition to which the request is sent. This can either be <ul style="list-style-type: none"> <li>• the transition from pre-operational to safe-operational (PS), or</li> <li>• the transition from safe-operational to operational (SO).</li> </ul> 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.

Index	Name	Flags	Wert
1000	Device type	RO	0x00000000 (0)
1008	Device name	RO	EL5001-0000
1009	Hardware version	RO	V00.01
100A	Software version	RO	V00.07
1011:0	Restore default parame...	RW	> 1 <
1011:01	Restore all	RW	0
1018:0	Identity object	RO	> 4 <
1018:01	Vendor id	RO	0x00000002 (2)
1018:02	Product code	RO	0x13893052 (327757906)
1018:03	Revision number	RO	0x00000000 (0)
1018:04	Serial number	RO	0x00000001 (1)
1A00:0	TxPDO 001 mapping	RO	> 2 <
1A00:01	Subindex 001	RO	0x3101:01, 8
1A00:02	Subindex 002	RO	0x3101:02, 32
1C00:0	SM type	RO	> 4 <
1C00:01	Subindex 001	RO	0x01 (1)
1C00:02	Subindex 002	RO	0x02 (2)
1C00:03	Subindex 003	RO	0x03 (3)
1C00:04	Subindex 004	RO	0x04 (4)
1C13:0	SM 3 PDO assign (inputs)	RW	> 1 <
1C13:01	Subindex 001	RW	0x1A00 (6656)
3101:0	Inputs	RO P	> 2 <
3101:01	Status	RO P	0x41 (65)
3101:02	Value	RO P	0x00000000 (0)
4061:0	Feature bits	RW	> 4 <
4061:01	disable frame error	RW	FALSE
4061:02	enable power failure Bit	RW	FALSE
4061:03	enable inhibit time	RW	FALSE
4061:04	enable test mode	RW	FALSE
4066	SSI-coding	RW	Gray code (1)
4067	SSI-baudrate	RW	500 kBaud (3)
4068	SSI-frame type	RW	Multiturn 25 bit (0)
4069	SSI-frame size	RW	0x0019 (25)
406A	Data length	RW	0x0018 (24)
406B	Min. inhibit time[μs]	RW	0x0000 (0)

Fig. 114: "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.

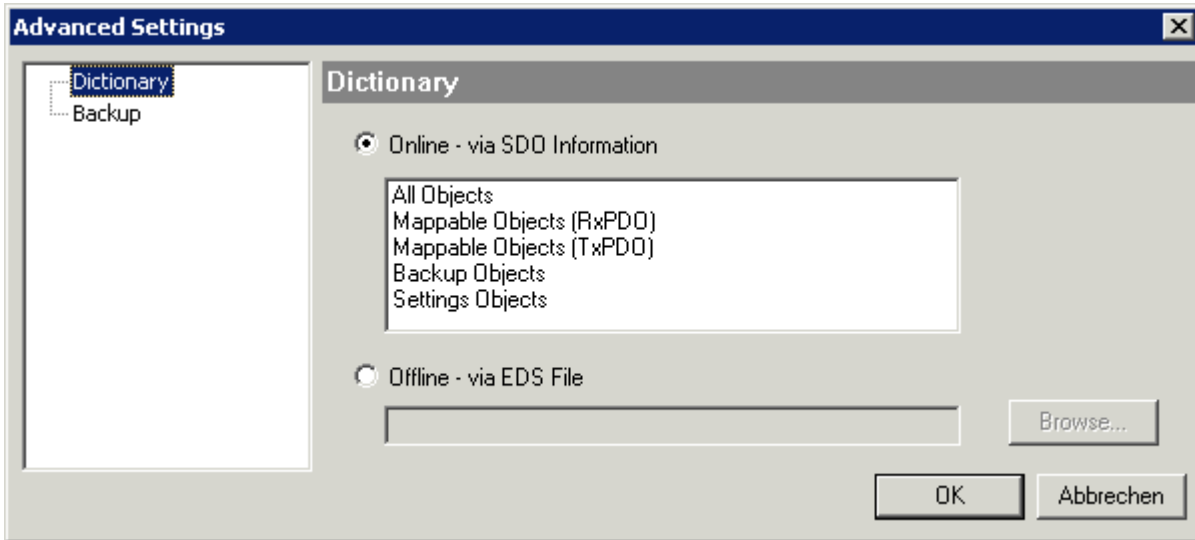


Fig. 115: Dialog “Advanced settings”

<p><b>Online - via SDO Information</b></p>	<p>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.</p>
<p><b>Offline - via EDS File</b></p>	<p>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.</p>

## „Online“ tab

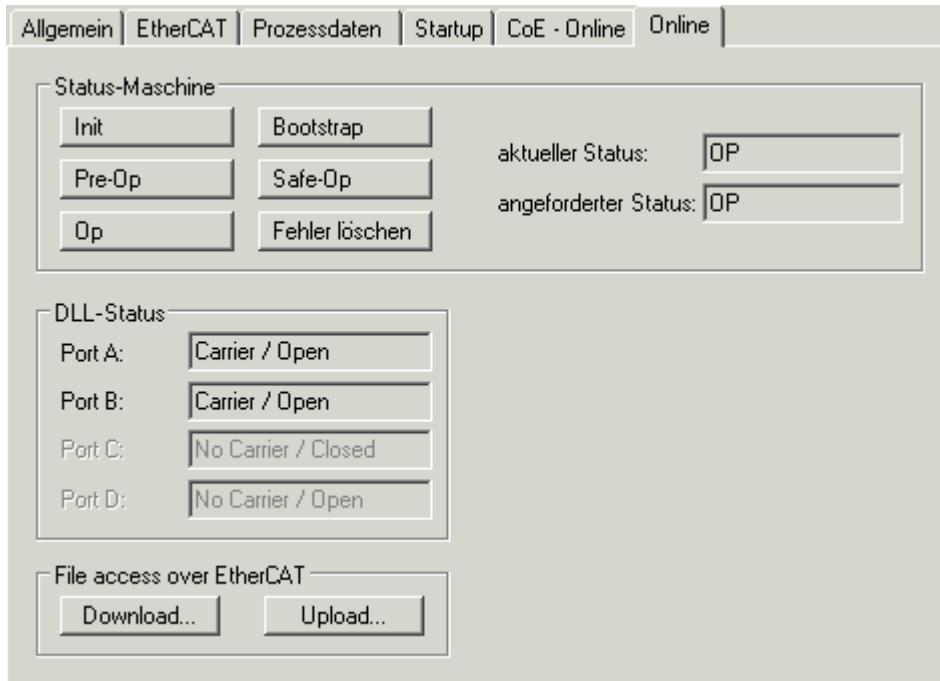


Fig. 116: „Online“ tab

**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)**

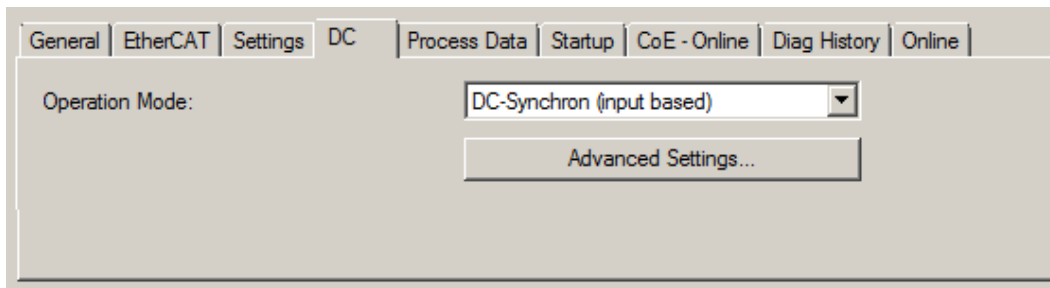


Fig. 117: "DC" tab (Distributed Clocks)

**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 are specified on <http://infosys.beckhoff.com>:

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

**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.

 <b>Note</b>	<p><b>Activation of PDO assignment</b></p> <ul style="list-style-type: none"> <li>✓ If you have changed the PDO assignment, in order to activate the new PDO assignment,             <ol style="list-style-type: none"> <li>a) the EtherCAT slave has to run through the PS status transition cycle (from pre-operational to safe-operational) once (see <a href="#">Online tab [► 86]</a>),</li> <li>b) and the System Manager has to reload the EtherCAT slaves</li> </ol> </li> </ul> <p>(  button for TwinCAT 2 or  button for TwinCAT 3)</p>
--	---

### 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 <i>PDO Assignment</i> 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 [Startup \[► 82\]](#) 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.

### 5.3 General Notes - EtherCAT Slave Application

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.

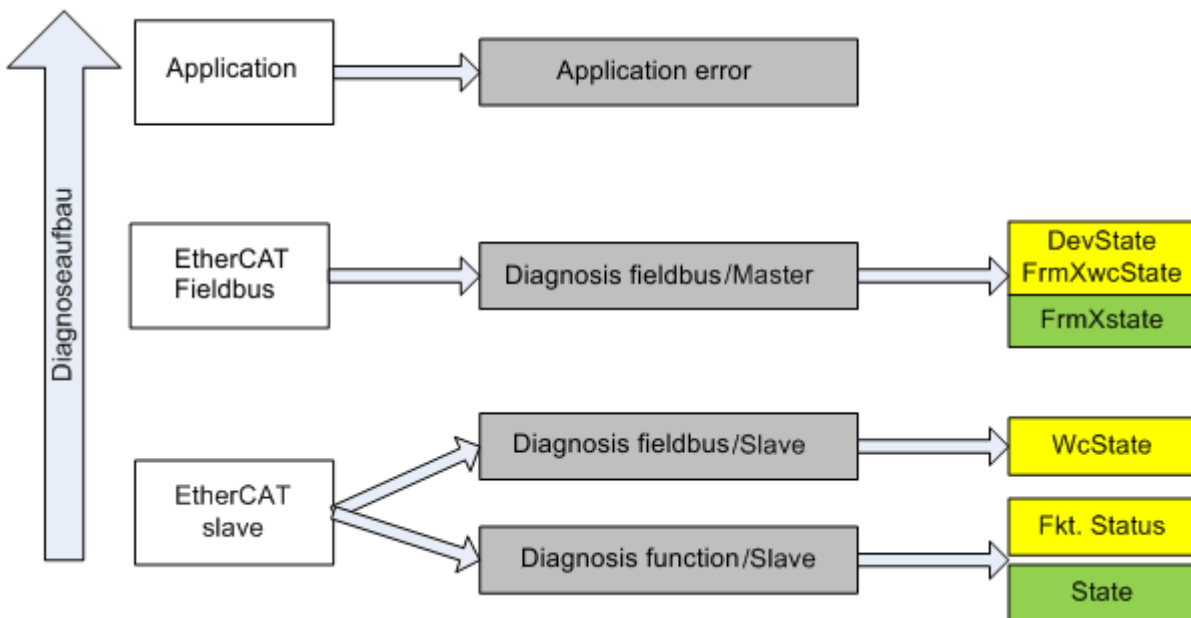


Fig. 118: Selection of the diagnostic information of an EtherCAT Slave

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.

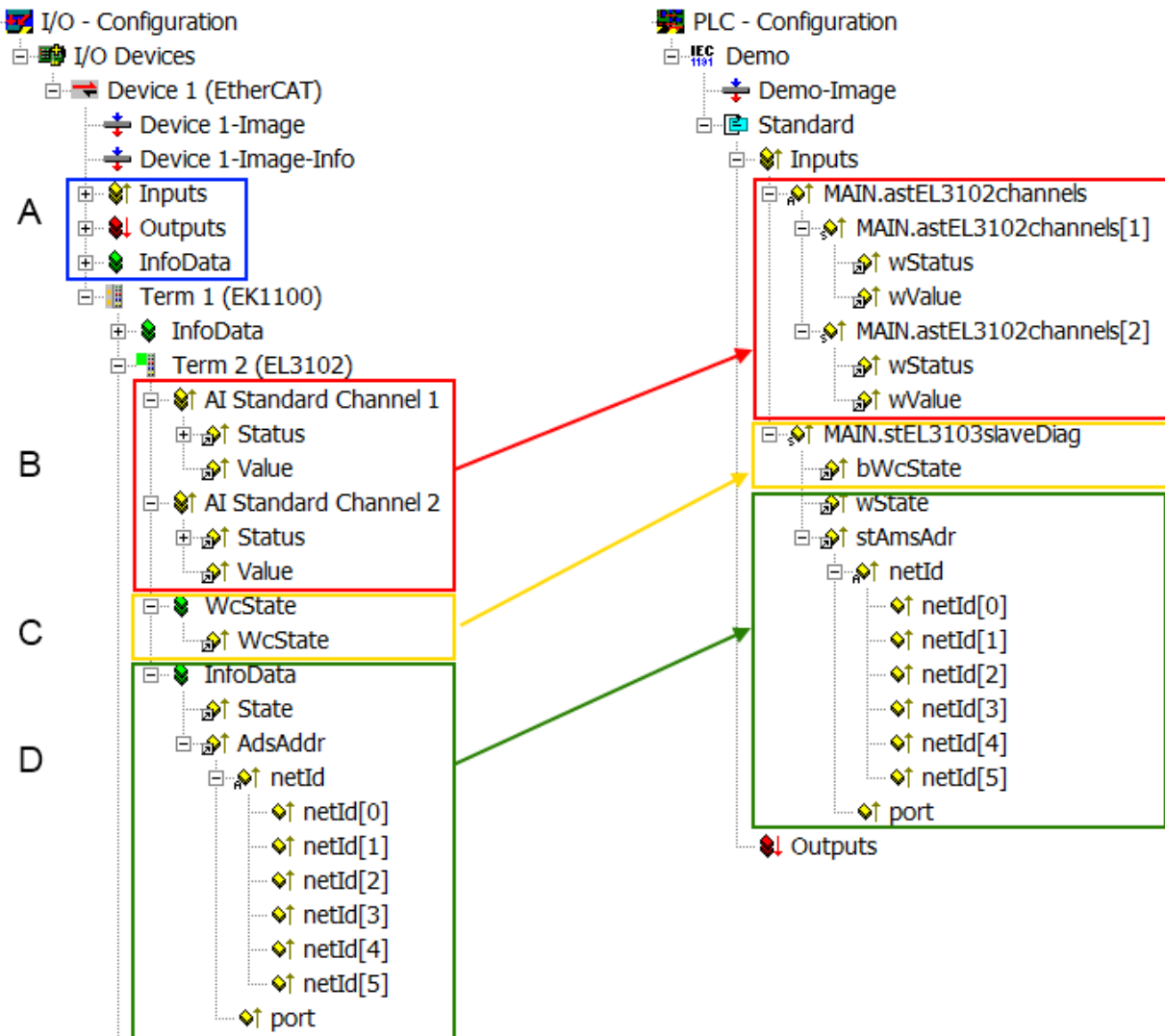



Fig. 119: Basic EtherCAT Slave Diagnosis in the PLC

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: <ul style="list-style-type: none"> <li>• CoE in the Master for communication with/through the Slaves</li> <li>• Functions from <i>TcEtherCAT.lib</i></li> <li>• Perform an OnlineScan</li> </ul>
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 <ul style="list-style-type: none"> <li>• the bit significations may be found in the device documentation</li> <li>• other devices may supply more information, or none that is typical of a slave</li> </ul>	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 <ol style="list-style-type: none"> <li>1. at the EtherCAT Slave, and, with identical contents</li> <li>2. as a collective variable at the EtherCAT Master (see Point A)</li> </ol> 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 <ul style="list-style-type: none"> <li>• is only rarely/never changed, except when the system starts up</li> <li>• is itself determined acyclically (e.g. EtherCAT Status)</li> </ul>	State current Status (INIT..OP) of the Slave. The Slave must be in OP (=8) when operating normally. <i>AdsAddr</i> 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 <i>port</i> (= 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.

 <b>Attention</b>	<p><b>Diagnostic information</b></p> <p>It is strongly recommended that the diagnostic information made available is evaluated so that the application can react accordingly.</p>
---	---

**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".

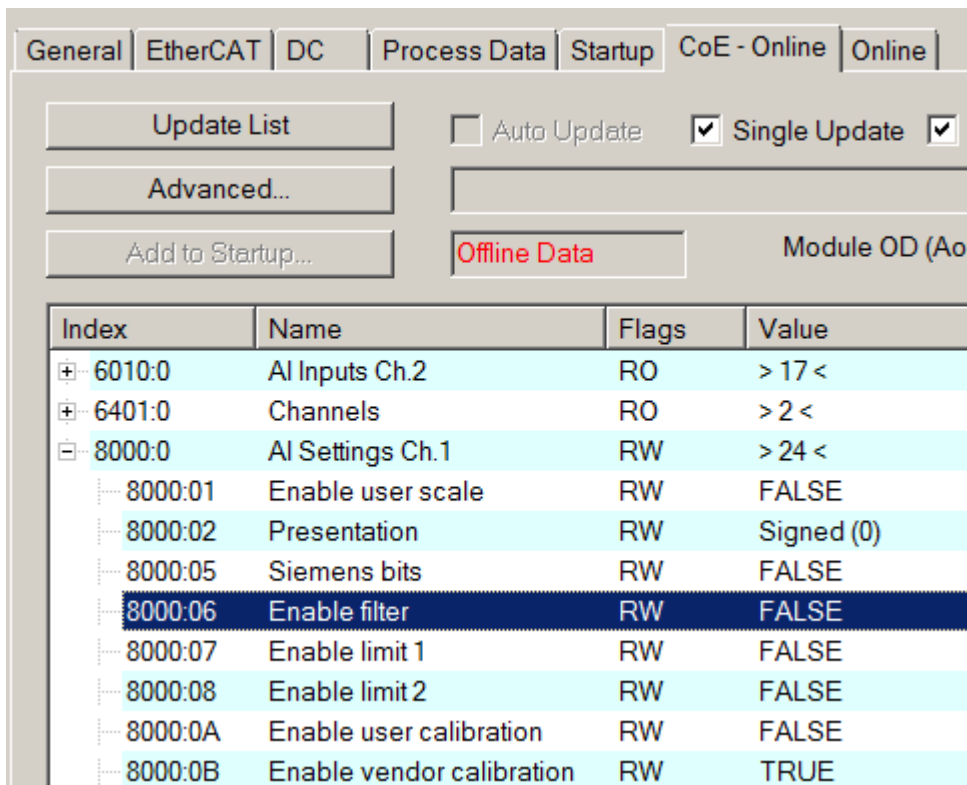



Fig. 120: EL3102, CoE directory

 <b>Note</b>	<p><b>EtherCAT System Documentation</b></p> <p>The comprehensive description in the <a href="#">EtherCAT System Documentation</a> (EtherCAT Basics --&gt; CoE Interface) must be observed!</p>
--	--

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.

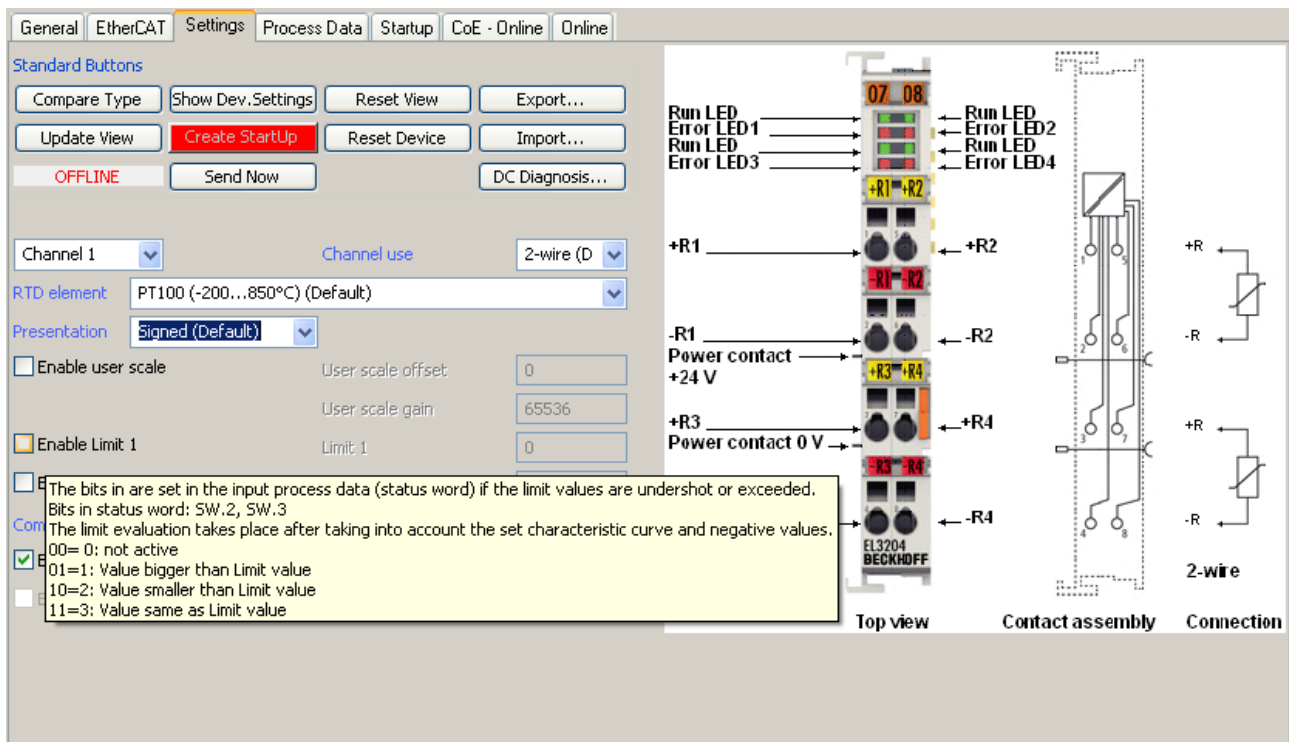


Fig. 121: Example of commissioning aid for a EL3204

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 \[► 16\]](#)" 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.

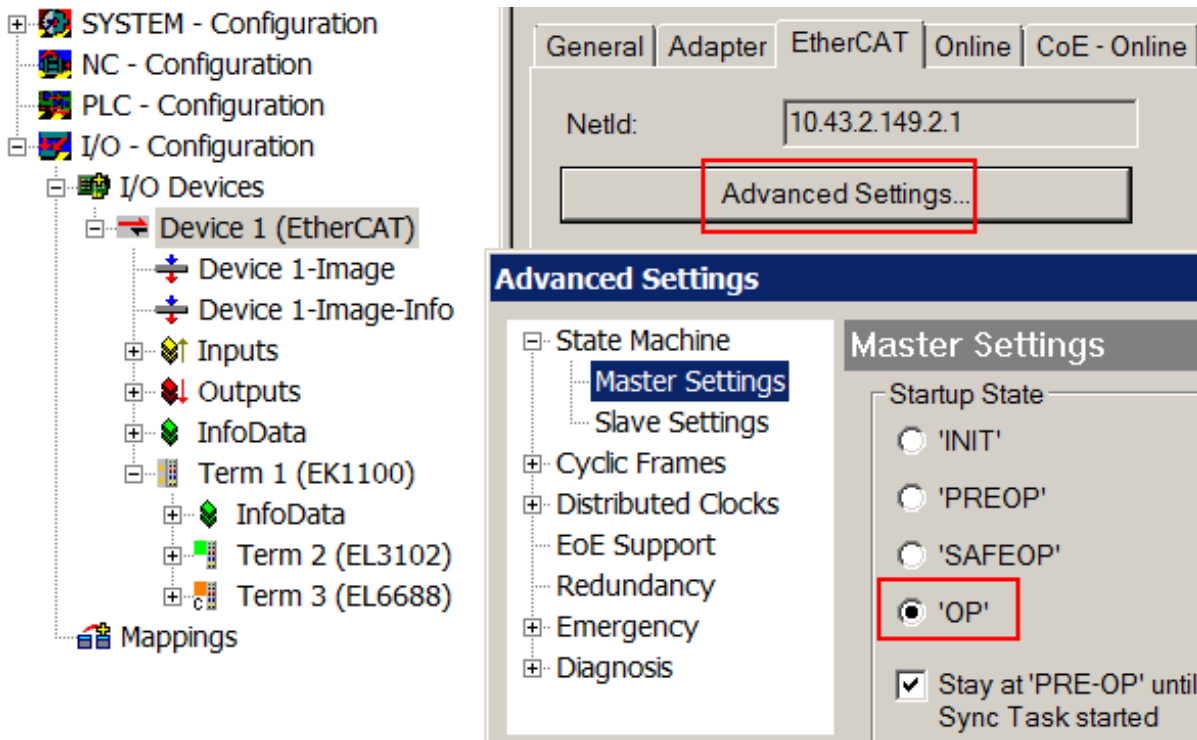


Fig. 122: Default behaviour of the System Manager

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

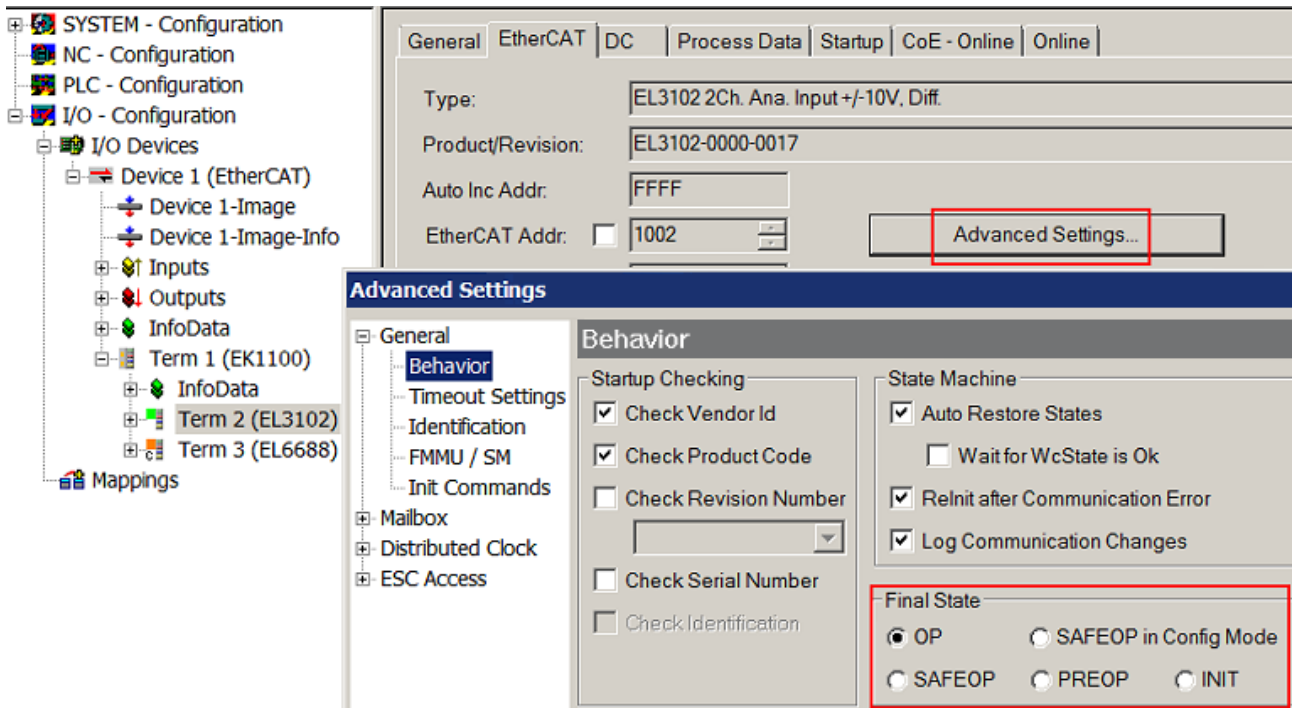


Fig. 123: Default target state in the Slave

**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.

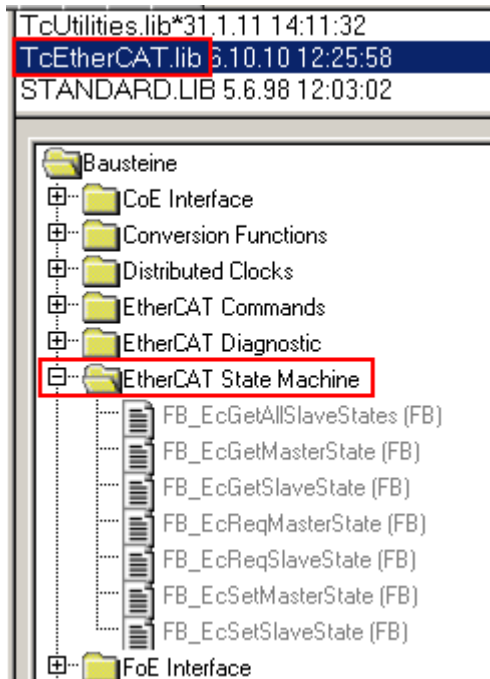


Fig. 124: PLC function blocks

**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.

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

Fig. 125: 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:

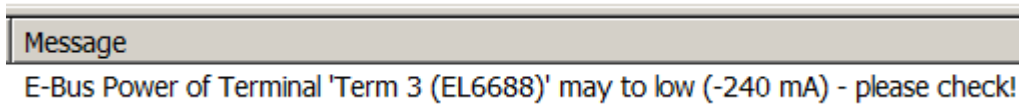



Fig. 126: Warning message for exceeding E-Bus current

 <b>Attention</b>	<p><b>Caution! Malfunction possible!</b></p> <p>The same ground potential must be used for the E-Bus supply of all EtherCAT terminals in a terminal block!</p>
---	--

## 5.4 Basic function principles

The EL6720 Lightbus master terminal is based on the characteristics of the FC2001/FC2002 master cards. The fieldbus cards are installed in the PC and are controlled directly by TwinCAT. The EL6720, however, is operated via the EtherCAT fieldbus.

For the EL6720 please note:

- Max. 3 CDLs (communication description lists, logical communication channels, allocated to a particular cycle time/task)  
Each CDL has a number; the higher the number, the higher the priority. The FC200x has 8 CDL, counted as 1..8, the EL6720 therefore has the 3 CDLs 8, 7 and 6.
- Lightbus IOs at the EL6720 require 1 cycle more in terms of response time compared with Lightbus IOs at a FC200x, since the EtherCAT fieldbus has to be taken into account
- The handling on error differs from the FC200x, see [Operation in the event of a Lightbus error \[► 99\]](#)
- See also Lightbus information in the [FC200x](#) documentation
- See also Lightbus information in the [C1220](#) documentation

### Basic information on diagnostics

The following information is available for Lightbus diagnostics via the EL6720:

- **Real-time diagnostics "cycle-current"**: Each Lightbus CDL has an
  - InputToggle: toggles whenever new valid Lightbus telegram was received
  - CDL state: if in this cycle invalid Lightbus data were received (e.g. incorrect checksum, attenuation too high, ...) or the transfer via the EtherCAT fieldbus failed, the CDLstate for this cycle is set to 1.  
During normal operation the CDLstate is 0

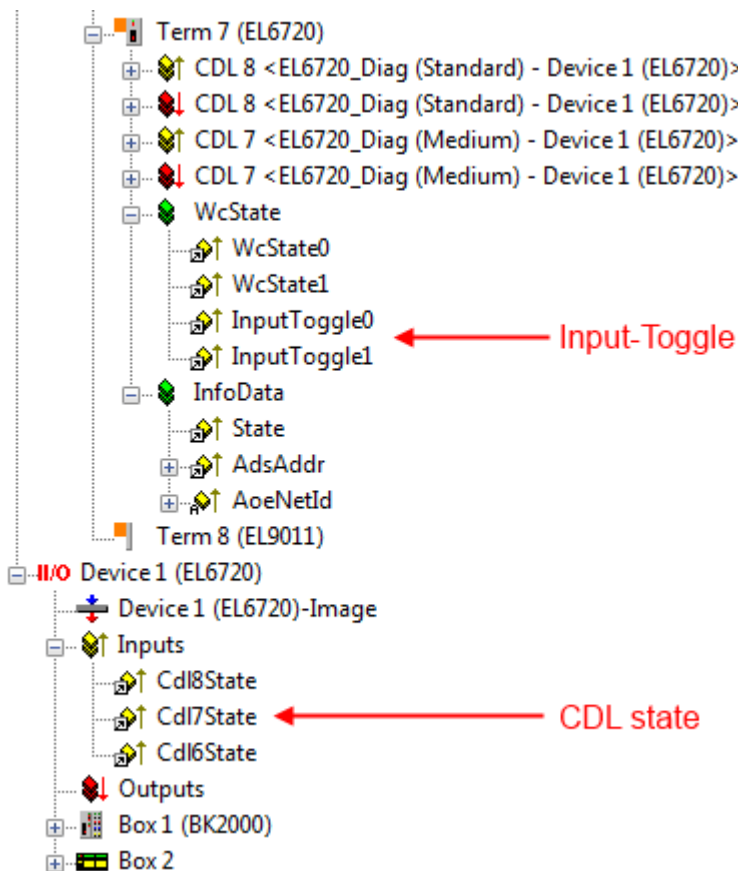


Fig. 127: EL6720 – Input Toggle and CDLstate in the TwinCAT tree

- Acyclic information:** A range of status information can be read from the CoE list of the EL6720. The EL6720 Rev. 0017 has no offline dictionary, i.e. the CoE can only be viewed in online mode. Of particular interest are the objects 0xFA00 and 0xFA01. If a break in the optical fiber ring occurs, xFA00:04 *BreakLocation* contains the number (1...) of the IO box after which the break occurred, for example (e.g. 2: interruption after box 2; 255: interruption point cannot be localized).

The screenshot shows the 'CoE - Online' tab in the Beckhoff software. At the top, there are control buttons: 'Update list', 'Auto Update' (unchecked), 'Single Update' (checked), and 'Show Offline Data' (unchecked). Below these are 'Advanced...' and 'Add to Startup...' buttons, along with a text field for 'All Objects' and a 'Module OD (AoE Port): 0' field.

The main area contains a table with the following columns: Index, Name, Flags, Wert, and Einheit. The table lists various CoE objects, with a red box highlighting the FA00:05 to FA00:0A range. Red callouts provide additional context for several rows:

- LWL OK -> 0** and **LWL ERR -> 1**: Callouts pointing to the 'Wert' column of the first two rows (1C13:0 and 1C14:0).
- Error counter**: Callout pointing to FA00:05.
- Parity error localization**: Callout pointing to FA00:0A.
- Boxtype 0 -> m module 1 -> BC/BK/AX**: Callout pointing to FA01:01.
- Number of telegrams**: Callout pointing to FA09:0.
- Number of cycles**: Callout pointing to FA10:02.

Index	Name	Flags	Wert	Einheit
1C13:0	TxPDO Assign	RW P	> 2 <	
1C14:0	RxPDO Assign 1	RW P	> 0 <	
1C15:0	TxPDO Assign 1	RW P	> 0 <	
1C16:0	RxPDO Assign 2	RW P	> 0 <	
1C17:0	TxPDO Assign 2	RW P	> 0 <	
F000:0	Modular Device Profile	RO	> 4 <	
F008	Code Word	RW	0x00000000 (0)	
F009:0	Module Control	RO	> 2 <	
F800:0	IIO Config	RO	> 1 <	
FA00:0	IIO Info	RO	> 13 <	
FA00:01	Status	RO	0x0000 (0)	
FA00:02	# of Boxes	RO	0x0002 (2)	
FA00:03	Loop Time(µs)	RO	0x007A (122)	
FA00:04	Break Location	RO	0x0000 (0)	
FA00:05	Err Handler	RO	0x0008 (8)	
FA00:06	CRC	RO	0x0000 (0)	
FA00:07	TimeOut	RO	0x0000 (0)	
FA00:08	Header	RO	0x0004 (4)	
FA00:09	DMA Ovr	RO	0x0000 (0)	
FA00:0A	Cnt	RO	0x0000 (0)	
FA00:0B	AD/CTRL Q	RO	0x0230 (560)	
FA00:0C	AD/CTRL Rxd	RO	0xFF40 (65344)	
FA00:0D	AC Err Box	RO	0x0000 (0)	
FA01:0	IIO Parity Info	RO	> 2 <	
FA01:01	SubIndex 001	RO	0x00 (0)	
FA01:02	SubIndex 002	RO	0x00 (0)	
FA08:0	IIO Box Info	RO	> 2 <	
FA08:01	SubIndex 001	RO	0x0001 (1)	
FA08:02	SubIndex 002	RO	0x0001 (1)	
FA09:0	IIO Box Info	RO	> 0 <	
FA10:0	CDL 8 Info	RO	> 2 <	
FA10:01	Size	RO	0x0004 (4)	
FA10:02	Done	RO	0x379F (14239)	
FA11:0	CDL 7 Info	RO	> 2 <	
FA12:0	CDL 6 Info	RO	> 2 <	
FA17:0	Acyclic CDL Info	RO	> 2 <	
FA18:0	AC CDL Info	RO	> 2 <	
FA19:0	Sys CDL Info	RO	> 2 <	

Fig. 128: EL6720 – Status information in the CoE-Online tab

In addition it is advisable to diagnose the actual IO boxes, e.g. BK2000 coupler diagnosis in this example.

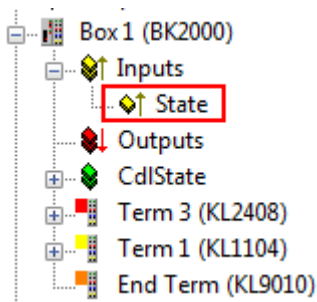


Fig. 129: BK2000 – Status information

**Operation in the event of a Lightbus error**

If a Lightbus device fails or the Lightbus connection is interrupted, continuous communication is no longer possible. The Lightbus master where the line interruption can be found. Once the interruption has been rectified, the Lightbus master has to be restarted.

Recommended procedure in the event of a fault:

1. Read the diagnostic information from the CoE list of the EL6720, e.g. from the PLC via ADS  
See the diagnostics example program. (Download)
2. Repair or rectification of the fault
3. The EL6720 is then in EtherCAT state *SafeOp\_Err* and has to be reset:
  - Clear Error
  - Carry out the LowIntensityTest in the EL6720 (see [CoE-overview xFA00:01 |> 105](#)] for any warnings)
  - Set to OP  
This can be done manually in the System Manager or, for example, from the PLC via *FB\_EcReqSlaveState* from the *TcEtherCAT.lib*.

**Optical fiber reset/attenuation test**

The Lightbus route can be subjected to an attenuation test via the EL6720. This attenuation test takes place by sending selected test telegrams at 80% transmission power for all modules

- after the EL6720 is first switched on (PowerOn)
- in the Preop/SafeOp state transition if an optical fiber error had previously occurred (see above)
- after a Lightbus error reset (see above)

**AoE**

If ADS-capable devices are integrated in the Lightbus ring, they can be addressed in the chain *TwinCAT --> EtherCAT --> EL6720 -> IO box* (and back) with ADS and the ADS address. To enter the ADS route via the EL6720, the EL6720 offers the AoE address in the configuration:

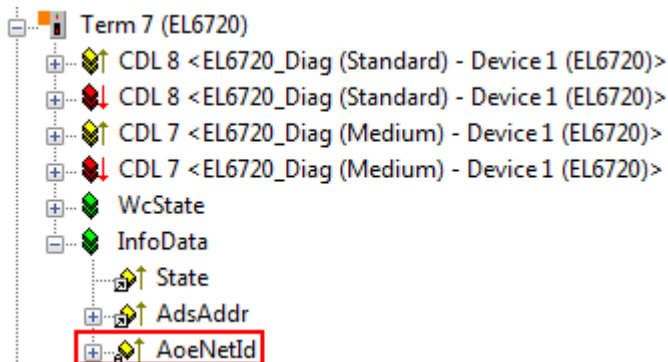



Fig. 130: AoE Address

**Example program for Lightbus diagnostics**

This example program

 (<http://infosys.beckhoff.com/content/1033/el6720/Resources/zip/2491293195.zip>) illustrates how the PLC can read information from the EL6720 via the Lightbus mode. The reset process in the event of an error is also demonstrated. It is based on the data from the FC200x. The FB consolidates the data from *EL6720/EtherCAT* and *DeviceLightbus* to a diagnostic structure.

The analysis of the maximum number of CDL (3) is demonstrated by linking a PLC project with 3 tasks with cycle times of 1, 10 and 100 ms.

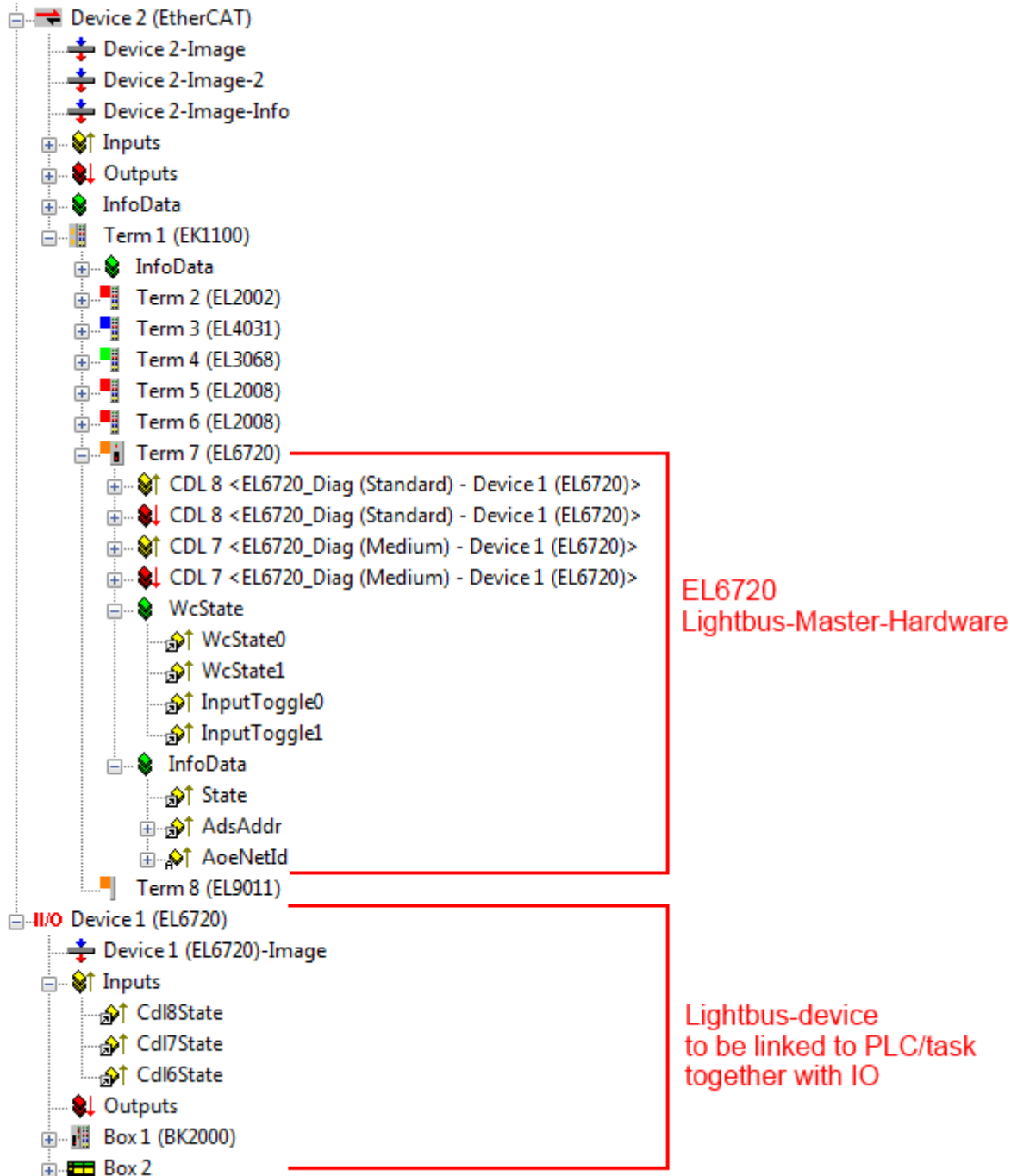


Fig. 131: Display EL6720 Lightbus master hardware and Lightbus device in the TwinCAT tree

## 5.5 Preparing a terminal configuration with EL6720



Note

### Installation of the latest XML device description

Please ensure that you have installed the corresponding latest XML device description in TwinCAT. This can be found in the download area of the [Beckhoff website](#). Follow the installation instructions.

### Automatic scanning in of the terminal

Based on an empty configuration, carry out an online scan to find the basic EtherCAT system and the Lightbus configuration.

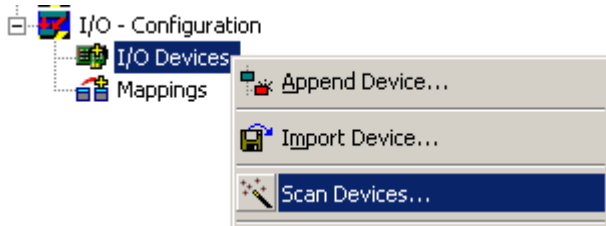


Fig. 132: Scanning existing devices

Once the EtherCAT system with its components, including the existing EL6720, has been created, the Lightbus system can be scanned.

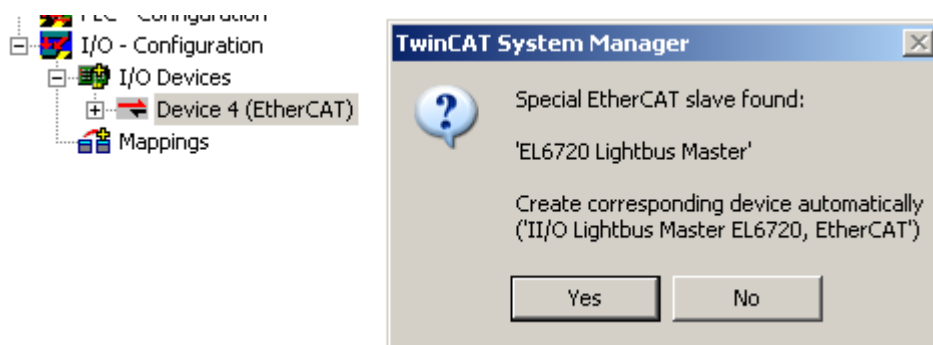


Fig. 133: Scanning existing Lightbus devices

Then set the bus settings via the Lightbus device in the configuration. On startup they are downloaded to the EL6720.

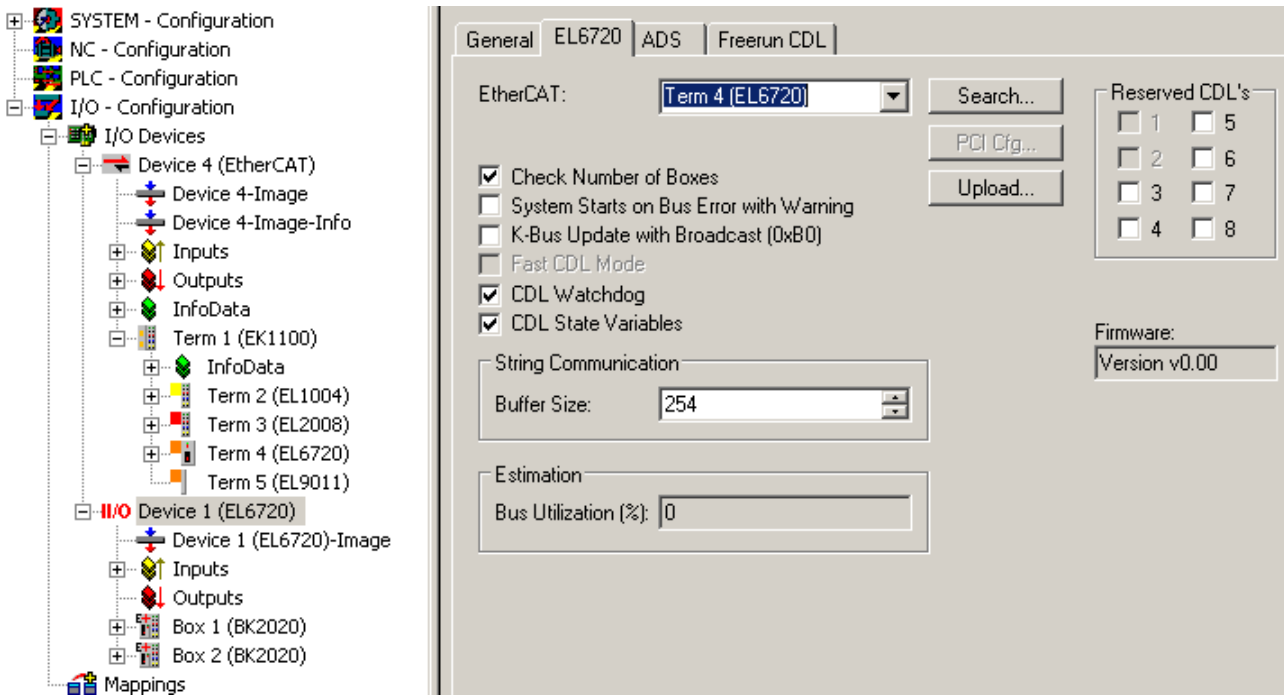


Fig. 134: Default setting of an EL6720 Lightbus device

**Additional information regarding the EL6720 Lightbus implementation**

1. As usual in TwinCAT, the Lightbus components require at least one link to a control task, otherwise is the bus is not triggered.
2. If no link exists, TwinCAT issues the message "needs sync master (at least one variable linked to a task variable)".

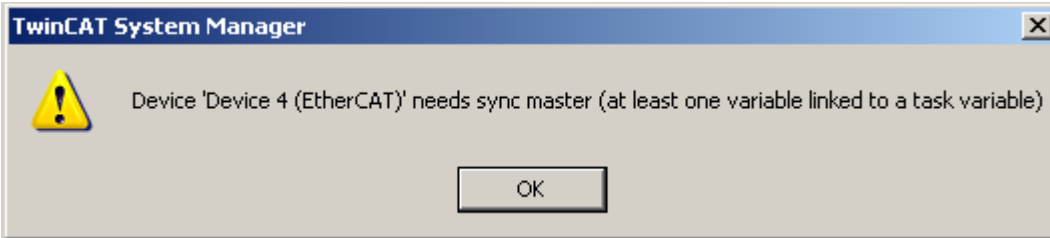


Fig. 135: Missing link message

3. In contrast to the EtherCAT system, the Lightbus system only sends telegrams for process data that are linked.  
*Sample:* A terminal has 4 digital inputs for BOOL variables, e.g. KL1004 or EL1004. In EtherCAT system a query with 4 bits would be sent to this terminal, regardless of whether and how many of the 4 inputs are actually linked with a task. In Lightbus system only 1 bit is used in the cyclic telegrams, if only 1 channel is linked from the terminal.  
 It follows that the circulating telegram length/data quantity/occupied CDL naturally depends on the topology, but also on the links.
4. The detailed display shows the used Lightbus capacity, i.e. how many data are currently sent in the selected cycle time.

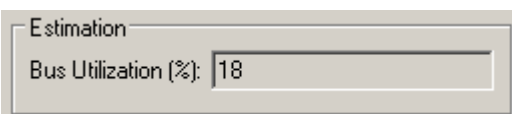


Fig. 136: Bus load display

More detailed information for the individual CDLs can be displayed in the "own" CDL view.

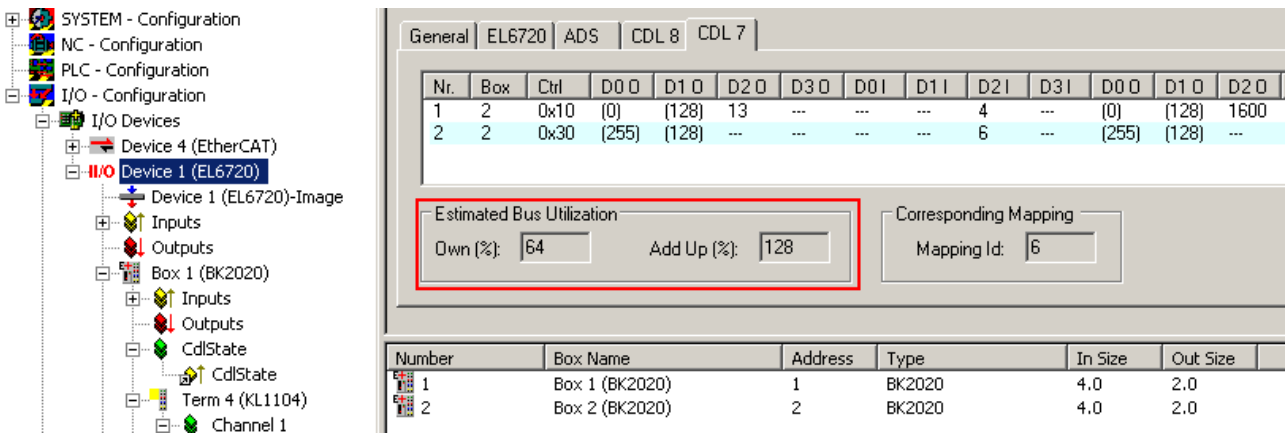


Fig. 137: CDL load display

In this example the cycle time of 200  $\mu$ s is too short relative to the (linked) process data, resulting in Lightbus overload.

A telegram can be assumed to be approx. 30  $\mu$ s long.

The System Manager reports this on activation.

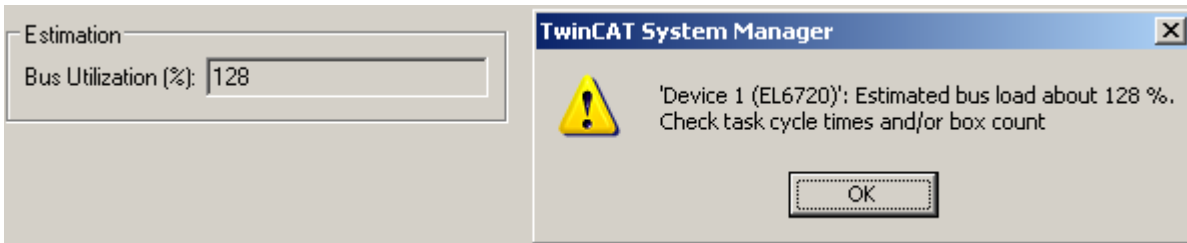


Fig. 138: Warning message System Manager

If such a system is then operated in Run mode, the lower-priority CDL cannot always be executed, and the diagnostic tool *CdIState* at times signals invalid input data with *CdIState* = 1.

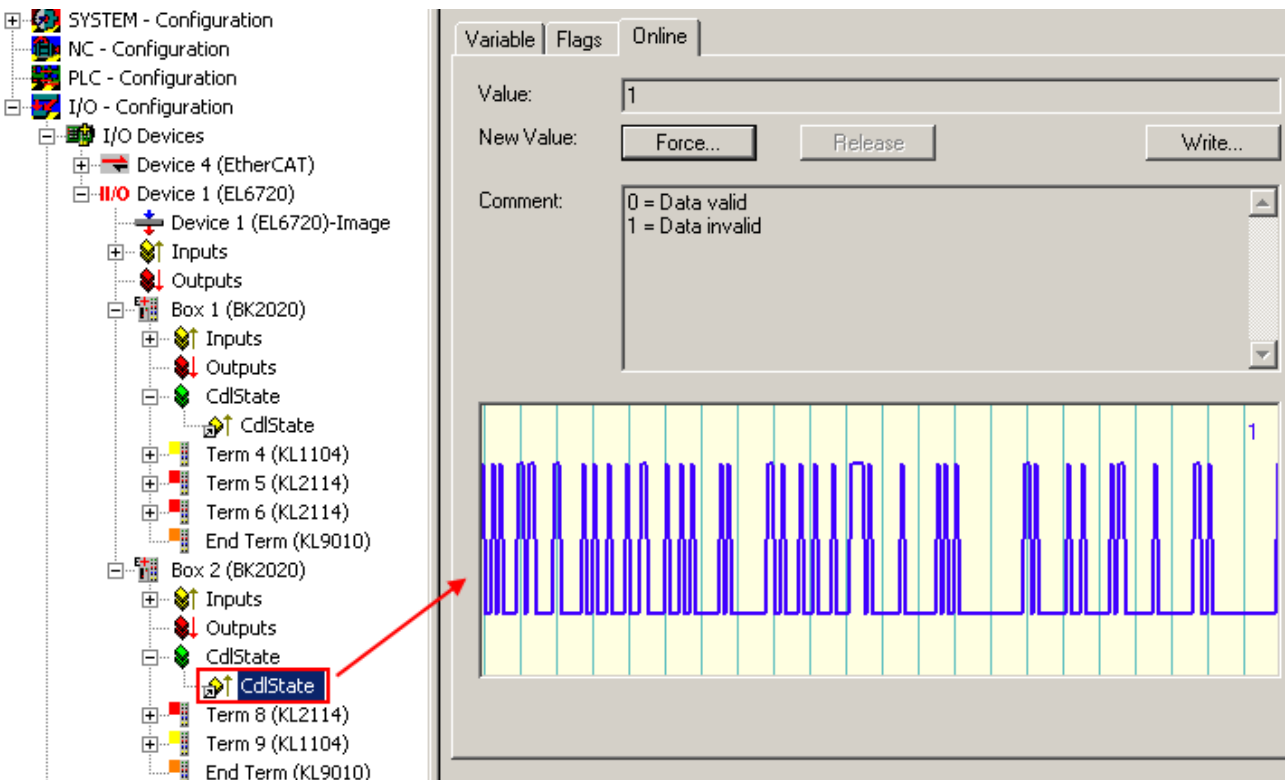


Fig. 139: DataInvalid overload display

- The number of Lightbus devices (maximum 254) is checked routinely. If the number of Couplers or Boxes is incorrect, the EL6720 does not start in OP: "SAFEOP to OP failed".

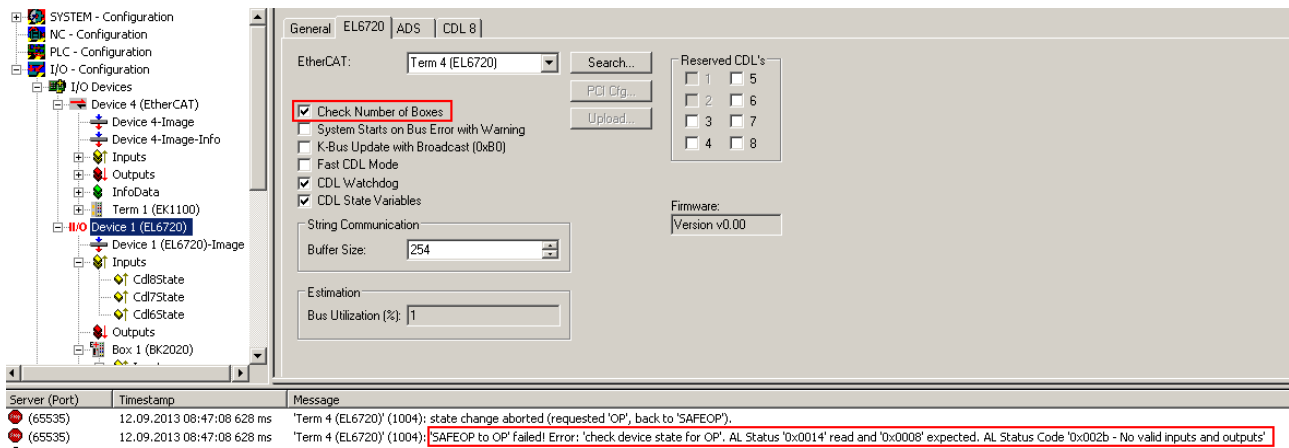


Fig. 140: Test box number

- The EL6720 has further diagnostic and configuration variables, which are shown in the [CoE overview](#) [▶ 105].

Index	Name	Flags	Value
[-] F800:0	IIO Config	RO	> 1 <
F800:01	# of configured Boxes	R'W	0x0002 (2)
[-] FA00:0	IIO Info	RO	> 15 <
FA00:01	Status	RO	0x0000 (0)
FA00:02	# of Boxes	RO	0x0002 (2)
FA00:03	Loop Time(µs)	RO	0x007A (122)
FA00:04	Break Location	RO	0x0000 (0)
FA00:05	Err Handler	RO	0x0005 (5)
FA00:06	CRC	RO	0x0000 (0)
FA00:07	TimeOut	RO	0x0000 (0)
FA00:08	Header	RO	0x0002 (2)
FA00:09	DMA Ovr	RO	0x0000 (0)
FA00:0A	Cnt	RO	0x0000 (0)
FA00:0B	AD/CTRL Q	RO	0x0140 (320)
FA00:0C	AD/CTRL Rxd	RO	0xFF40 (65344)
FA00:0D	AC Err Box	RO	0x0000 (0)
[-] FA01:0	IIO Parity Info	RO	> 2 <
FA01:01	SubIndex 001	RO	0x00 (0)
FA01:02	SubIndex 002	RO	0x00 (0)
[-] FA08:0	IIO Box Info	RO	> 2 <
FA08:01	SubIndex 001	RO	0x0001 (1)
FA08:02	SubIndex 002	RO	0x0001 (1)
[+] FA09:0	IIO Box Info	RO	> 0 <
[-] FA10:0	CDL 8 Info	RO	> 2 <
FA10:01	Size	RO	0x0002 (2)
FA10:02	Done	RO	0xD935 (55605)
[+] FA11:0	CDL 7 Info	RO	> 2 <
[+] FA12:0	CDL 6 Info	RO	> 2 <
[-] FA17:0	Acyclic CDL Info	RO	> 2 <
FA17:01	Size	RO	0x0001 (1)
FA17:02	Done	RO	0x01F1 (497)
[-] FA18:0	AC CDL Info	RO	> 2 <
FA18:01	Size	RO	0x0008 (8)
FA18:02	Done	RO	0x458C (17804)
[-] FA19:0	Sys CDL Info	RO	> 2 <
FA19:01	Size	RO	0x0001 (1)
FA19:02	Done	RO	0x0219 (537)

Fig. 141: Overview of Lightbus-relevant CoE entries

3. Advanced Settings:

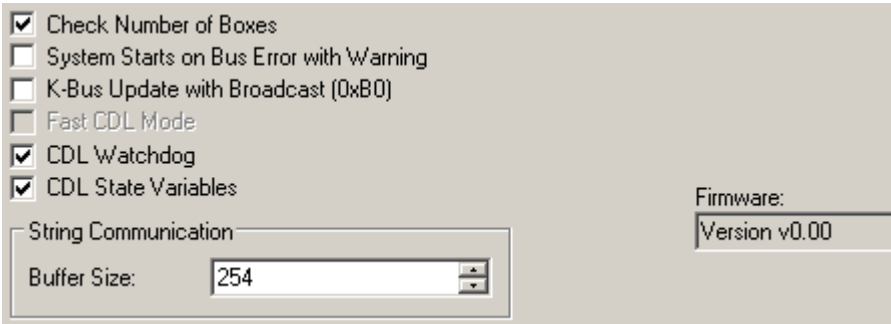




Fig. 142: Advanced Settings

- Check Number of Boxes: see above
- System Starts ...: Without function
- K-Bus Update ...: Without function
- CDL watchdog
- CDL state variables: activates the CDL process data in the configuration tree
- Firmware display: in the EL6720 without function; the firmware version can be read in the CoE of the EL6720
- String communication: In the EL6720 the string communication of the FC20xx cards was changed to AoE and is therefore no longer relevant here.

**Appending a terminal manually**

The manual configuration setup is based on the steps described above.

## 5.6 Object description and parameterization

 <b>Note</b>	<p><b>EtherCAT XML Device Description</b></p> <p>The display matches that of the CoE objects from the EtherCAT <u>XML</u> Device Description. We recommend downloading the latest XML file from the download area of the Beckhoff website and installing it according to installation instructions.</p>
 <b>Note</b>	<p><b>Parameterization via the CoE list (CAN over EtherCAT)</b></p> <p>The EtherCAT device is parameterized via the <u>CoE - Online tab [▶ 83]</u> (double-click on the respective object) or via the <u>Process Data tab [▶ 80]</u>(allocation of PDOs). Please note the following general <u>CoE notes [▶ 17]</u> when using/manipulating the CoE parameters:</p> <ul style="list-style-type: none"> <li>- Keep a startup list if components have to be replaced</li> <li>- Differentiation between online/offline dictionary, existence of current XML description</li> <li>- use “CoE reload” for resetting changes</li> </ul>

The EL6720-0000-0016/0017 has no offline dictionary in the ESI XML. This means the CoE entries can only be viewed in online state.

### 5.6.1 Restore object

**Index 1011 Restore default parameters**

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters	Restore default parameters	UINT8	RO	0x01 (1 <sub>dec</sub> )
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 <sub>dec</sub> )

## 5.6.2 Configuration and diagnostic data

### Index F800

Index (hex)	Name	Meaning	Data type	Flags	Default
F800:0	IIO Config	Max. subindex	UINT8	RO	0x1 (1 <sub>dec</sub> )
F800:01	# of configured Boxes	Number of configured boxes/couplers		RO	0

### Index FA00

Index (hex)	Name	Meaning	Data type	Flags	Default
FA00:0	IIO Info	Max. subindex	UINT8	RO	0xF (15 <sub>dec</sub> )
FA00:01	Status	The status has to be interpreted bit by bit: 0x0001 general optical fiber error (Lightbus no longer runs) 0x0002 too many errors during address check 0x0010 setting of the transmission power for the IIO boxes to 100% failed 0x0020 resetting of the box addresses failed 0x0030 break location detected 0x0040 setting of box addresses failed 0x0050 testing of box addresses failed 0x0060 testing of box addresses failed 0x0070 testing of box addresses failed  Warnings from the LowIntensityTest (80% transmission power) for information 0x0100 setting of transmission power for the IIO boxes to 80% failed, e.g. during reset 0x0110 box test with extreme test telegrams failed 0x0120 box test with extreme test telegrams failed 0x0130 box test with extreme test telegrams failed 0x0140 setting of the transmission power for the IIO boxes to 100% failed	UINT32	RO	
FA00:02	# of Boxes	Actually found boxes/couplers		RO	
FA00:03	Loop time (µs)	Actual telegram runtime via the circuit		RO	
FA00:04	Break Location	Number/address of the box at which an error occurred		RO	
FA00:05	Err Handler	If an error occurs after FA00:06 to FA00:09, an internal error handling routine is started. This counter indicates how often this routine was started and should be <= the sum of :06 to :09.		RO	
FA00:06	CRC	Number of detected CRC errors (parity errors)		RO	
FA00:07	TimeOut	Number of detected timeout errors (2x loop time is not adhered to)		RO	
FA00:08	Header	Number of detected header errors (address and header check)		RO	
FA00:09	DMA Ovr	Number of detected DMA errors (internal)		RO	
FA00:0A	Cnt	For each ErrHandler error after FA00:05 this counter is incremented +4, for each "good" received telegram -1 If the value exceeds 200 the Lightbus communication is aborted with "Fault"		RO	
FA00:0B	AD/CTRL Q	Internal address check		RO	
FA00:0C	AD/CTRL Rxd	Internal address check		RO	
FA00:0D	AC Err Box	Number of the box at which the cyclic address check detected an error		RO	

### Index FA01

Each Lightbus box has a byte counter, which counts the CRC/parity errors. In the EL6720 these local counters can be shown in the respective subindices for 0xFA01.

Index (hex)	Name	Meaning	Data type	Flags	Default
FA01:0	IIO Parity Info	Maximum subindex = no. of devices	UINT8	RO	0x2 (2 <sub>dec</sub> )
FA01:nn	Subindex 0nn	Local CRC counter from the respective device. This can be used to locate a CRC fault. :01 maps device 1 :02 maps device 2 etc.		RO	

**Index FA08**

Each existing Lightbus device has a subindex that indicates the device type.

Index (hex)	Name	Meaning	Data type	Flags	Default
FA08:0	IIO Box Info	Maximum subindex = no. of devices	UINT8	RO	
FA08:nn	Subindex 0nn	Device type: 0 = M-module, 1 = Coupler		RO	

**Index FA09**

Index (hex)	Name	Meaning	Data type	Flags	Default
FA09:0	IIO Box Info	Max. subindex	UINT8	RO	

**Index FA10 ... FA12**

Each of the up to 3 cyclic process data CDLs of the EL6720 shows live data here.

Index (hex)	Name	Meaning	Data type	Flags	Default
FA10:0	CDL8 Info	Max. subindex	UINT8	RO	0x2 (2 <sub>dec</sub> )
FA10:01	Size	Number of telegrams in the CDL		RO	
FA10:02	Done	Counter indicating how often this CDL has already been sent. Increments during operation.		RO	

**Index FA17**

Live data of the hidden CDLs, e.g. for string communication

Index (hex)	Name	Meaning	Data type	Flags	Default
FA17:0	Acyclic CDL information	Max. subindex	UINT8	RO	0x2 (2 <sub>dec</sub> )
FA17:01	Size	Number of telegrams in the CDL		RO	
FA17:02	Done	Counter indicating how often this CDL has already been sent. Increments during operation.		RO	

**Index FA18**

Live data of the hidden CDL for permanent address check

Index (hex)	Name	Meaning	Data type	Flags	Default
FA18:0	AC CDL Info	Max. subindex	UINT8	RO	0x2 (2 <sub>dec</sub> )
FA18:01	Size	Number of telegrams in the CDL		RO	
FA18:02	Done	Counter indicating how often this CDL has already been sent. Increments during operation.		RO	

**Index FA19**

Live data of the hidden system CDLs, e.g. for start control.

Index (hex)	Name	Meaning	Data type	Flags	Default
FA19:0	Sys CDL Info	Max. subindex	UINT8	RO	0x2 (2 <sub>dec</sub> )
FA19:01	Size	Number of telegrams in the CDL		RO	
FA19:02	Done	Counter indicating how often this CDL has already been sent. Increments during operation.		RO	

**5.6.3 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 <sub>dec</sub> )

**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 <sub>dec</sub> )
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2 <sub>dec</sub> )
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x0D1C3052 (219951186 <sub>dec</sub> )
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 <sub>dec</sub> )
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 <sub>dec</sub> )

## 6 Appendix

### 6.1 UL notice

	<p><b>Application</b> Beckhoff EtherCAT modules are intended for use with Beckhoff's UL Listed EtherCAT System only.</p>
	<p><b>Examination</b> For cULus examination, the Beckhoff I/O System has only been investigated for risk of fire and electrical shock (in accordance with UL508 and CSA C22.2 No. 142).</p>
	<p><b>For devices with Ethernet connectors</b> Not for connection to telecommunication circuits.</p>

#### Basic principles

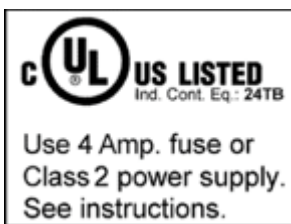
Two UL certificates are met in the Beckhoff EtherCAT product range, depending upon the components:

- UL certification according to UL508  
Devices with this kind of certification are marked by this sign:



Almost all current EtherCAT products (as at 2010/05) are UL certified without restrictions.

- UL certification according to UL508 with limited power consumption  
The current consumed by the device is limited to a max. possible current consumption of 4 A. Devices with this kind of certification are marked by this sign:



Almost all current EtherCAT products (as at 2010/05) are UL certified without restrictions.

#### Application

If terminals certified *with restrictions* are used, then the current consumption at 24 V<sub>DC</sub> must be limited accordingly by means of supply

- from an isolated source protected by a fuse of max. 4A (according to UL248) or
- from a voltage supply complying with *NEC class 2*.  
A voltage source complying with *NEC class 2* may not be connected in series or parallel with another *NEC class 2* compliant voltage supply!


These requirements apply to the supply of all EtherCAT bus couplers, power adaptor terminals, Bus Terminals and their power contacts.

## 6.2 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.

 <b>Attention</b>	<p><b>Risk of damage to the device!</b></p> <p>Pay attention to the instructions for firmware updates on the <a href="#">separate page</a> [▶ 110]. 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!</p>
---	--

EL6720			
Hardware (HW)	Firmware (FW)	Revision no.	Release date
00 - 14*	01		2008/03
		EL6720-0000-0016	2008/11
	02		2009/01
		03-04	
	05		2012/10
		EL6720-0000-0017	2012/10
		EL6720-0000-0018	2013/12
	06*	EL6720-0000-0019	2014/07
	EL6720-0000-0020	2014/12	

\*) 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.

## 6.3 Firmware Update EL/ES/EM/EPxxxx

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

### Storage locations

An EtherCAT slave stores operating data in up to 3 locations:

- 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.
- In addition, each EtherCAT slave has a memory chip, a so-called **ESI-EEPROM**, for storing its own device description (ESI: EtherCAT Slave Information). On power-up this description is loaded and the EtherCAT communication is set up accordingly. The device description is available from the download area of the Beckhoff website at (<http://www.beckhoff.de>). All ESI files are accessible there as zip files.

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 3 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. ELxxx-xxx\_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

 <b>Attention</b>	<p><b>Risk of damage to the device!</b></p> <p>Note the following when downloading new device files</p> <ul style="list-style-type: none"> <li>• Firmware downloads to an EtherCAT device must not be interrupted</li> <li>• Flawless EtherCAT communication must be ensured. CRC errors or LostFrames must be avoided.</li> <li>• The power supply must adequately dimensioned. The signal level must meet the specification.</li> </ul> <p>In the event of malfunctions during the update process the EtherCAT device may become unusable and require re-commissioning by the manufacturer.</p>
---	---

**Device description ESI file/XML**

 <b>Attention</b>	<p><b>Notice regarding update of the ESI description/EEPROM</b></p> <p>Some slaves have stored calibration and configuration data from the production in the EEPROM. These are irretrievably overwritten during an update.</p>
---	--

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:

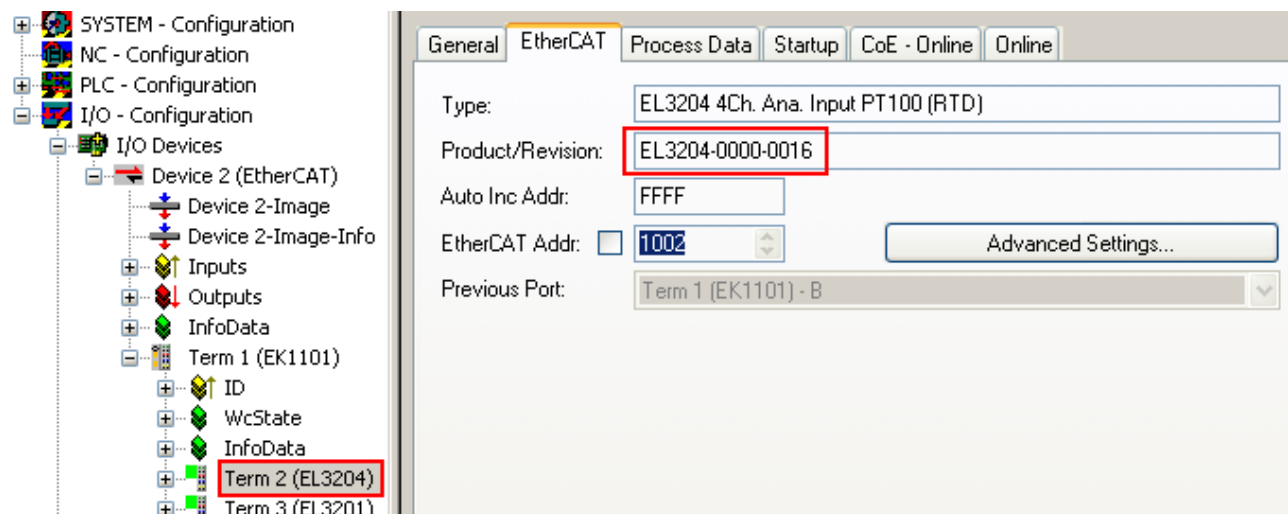



Fig. 143: Device identifier consisting of name EL3204-0000 and revision -0016

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](#).

 <b>Note</b>	<p><b>Update of XML/ESI description</b></p> <p>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.</p>
--	--

### 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:

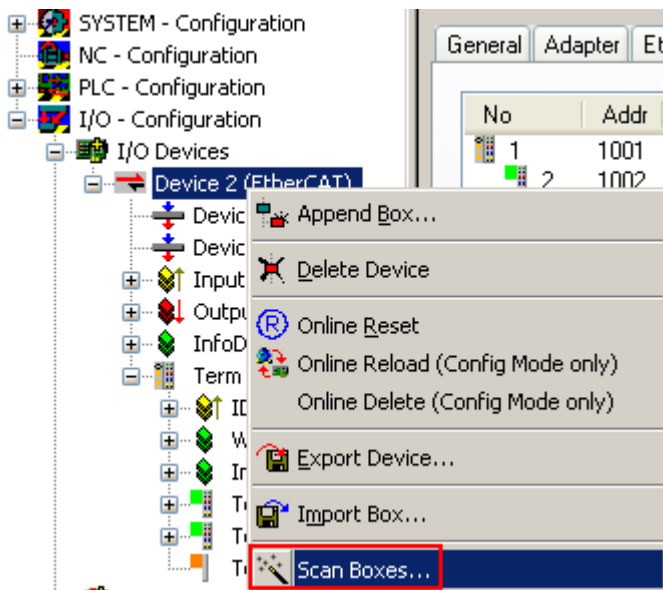


Fig. 144: Scan the subordinate field by right-clicking on the EtherCAT device in Config/FreeRun mode

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



Fig. 145: Configuration is identical

otherwise a change dialog appears for entering the actual data in the configuration.

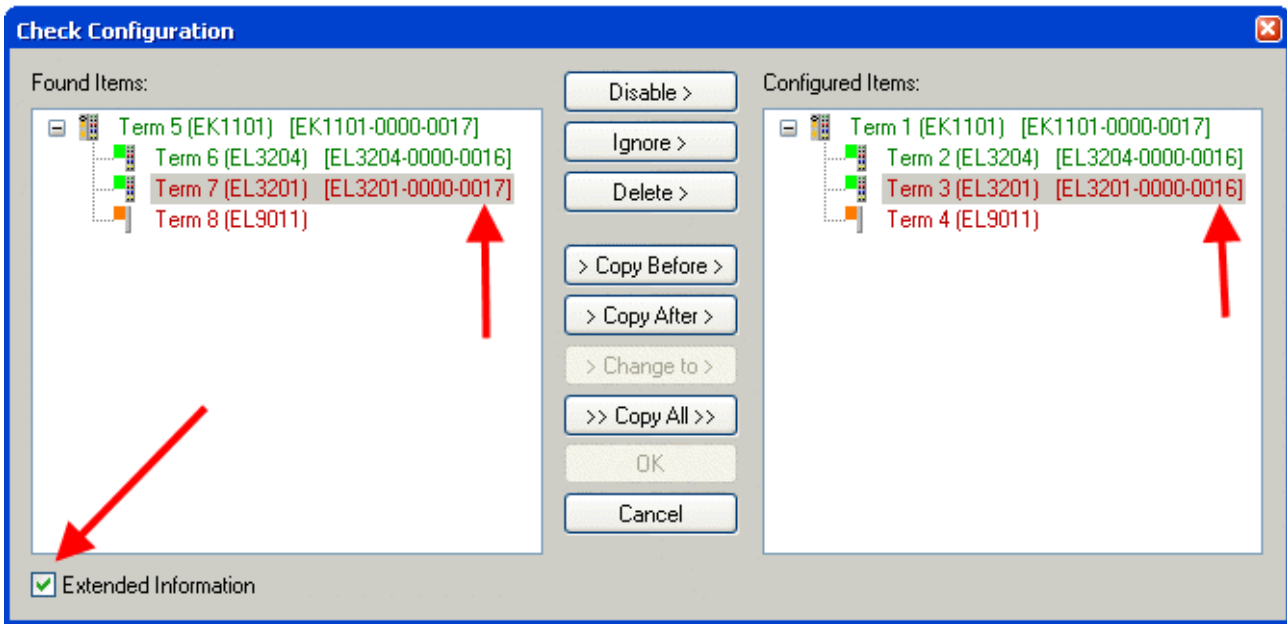


Fig. 146: Change dialog

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"

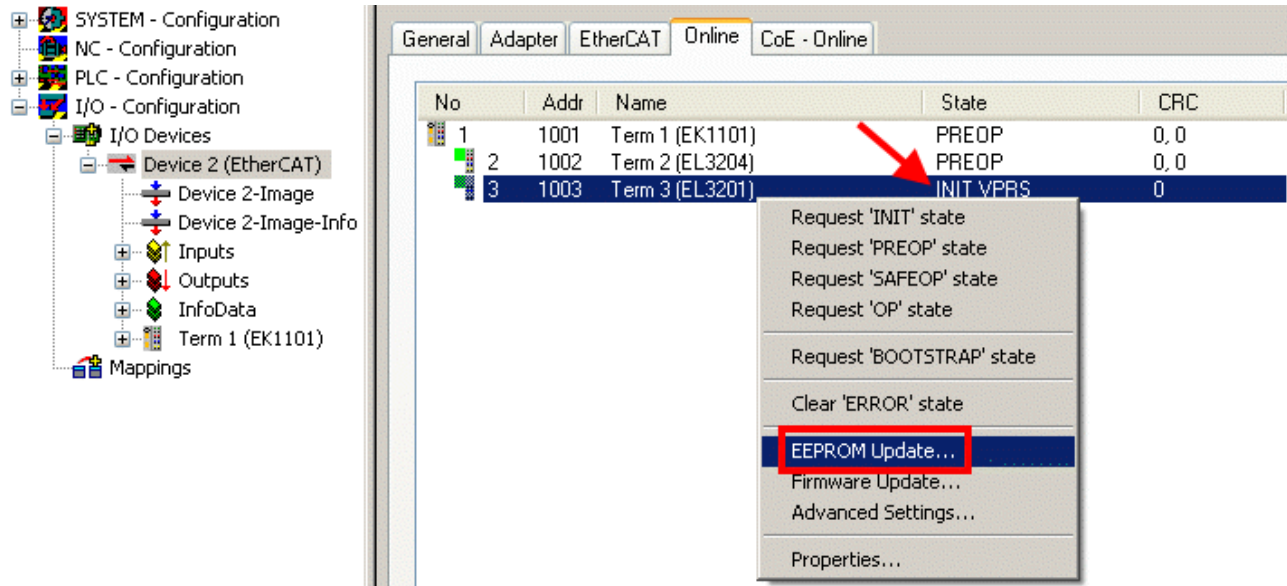


Fig. 147: EEPROM Update

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.

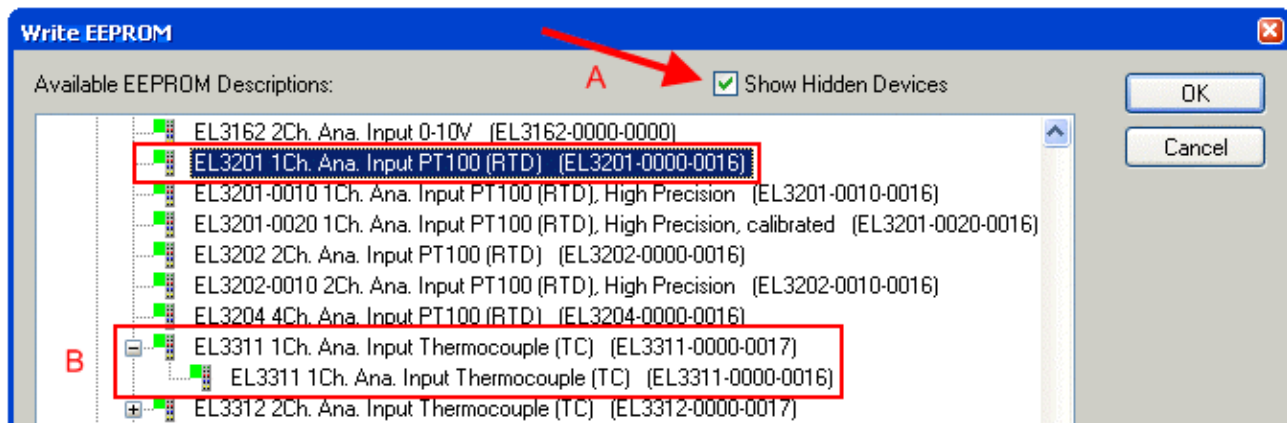



Fig. 148: *Selecting the new ESI*

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

 <b>Note</b>	<p><b>The change only takes effect after a restart.</b></p> <p>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.</p>
--	--

### Determining the firmware version

#### Determining the version on laser inscription

Beckhoff EtherCAT slaves feature serial numbers applied by laser. The serial number has the following structure: **KK YY FF HH**


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

Example with ser. no.: 12 10 03 02:

12 - week of production 12  
 10 - year of production 2010  
 03 - firmware version 03  
 02 - hardware version 02

#### 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).

 <b>Note</b>	<p><b>CoE Online and Offline CoE</b></p> <p>Two CoE directories are available:</p> <ul style="list-style-type: none"> <li>• <b>online</b>: This is offered in the EtherCAT slave by the controller, if the EtherCAT slave does supported it. This CoE directory can only be displayed if a slave is connected and operational.</li> <li>• <b>offline</b>: 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").</li> </ul> <p>The Advanced button must be used for switching between the two views.</p>
--	--

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

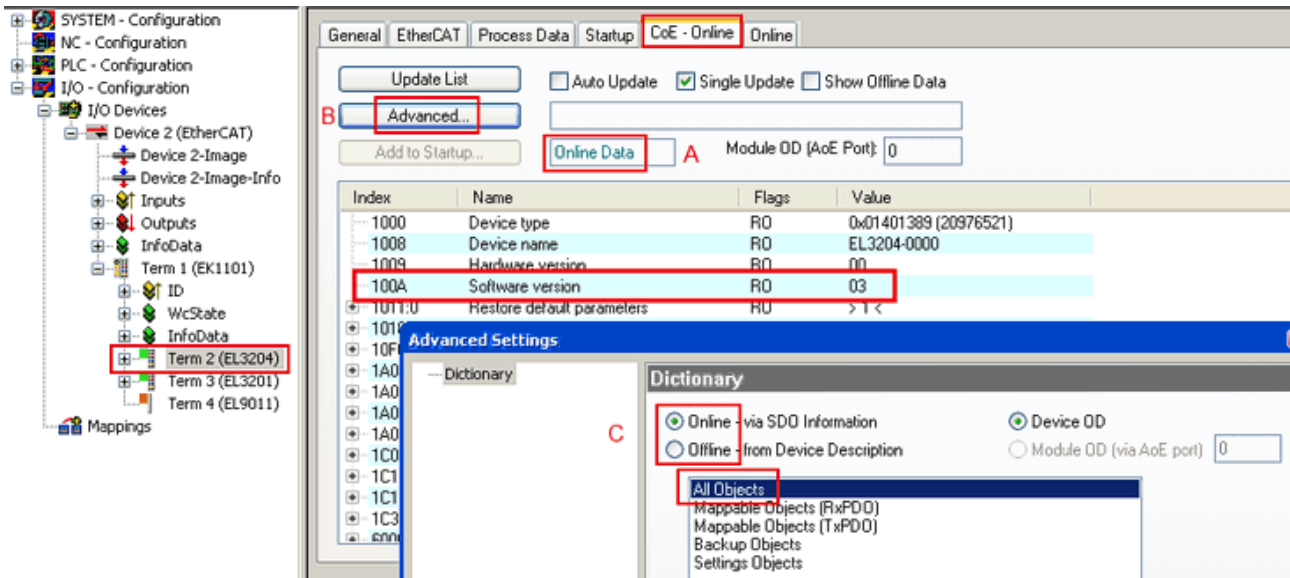


Fig. 149: Display of EL3204 firmware version

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*.

**Updating controller firmware \*.efw**

i

**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.

**Note**

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

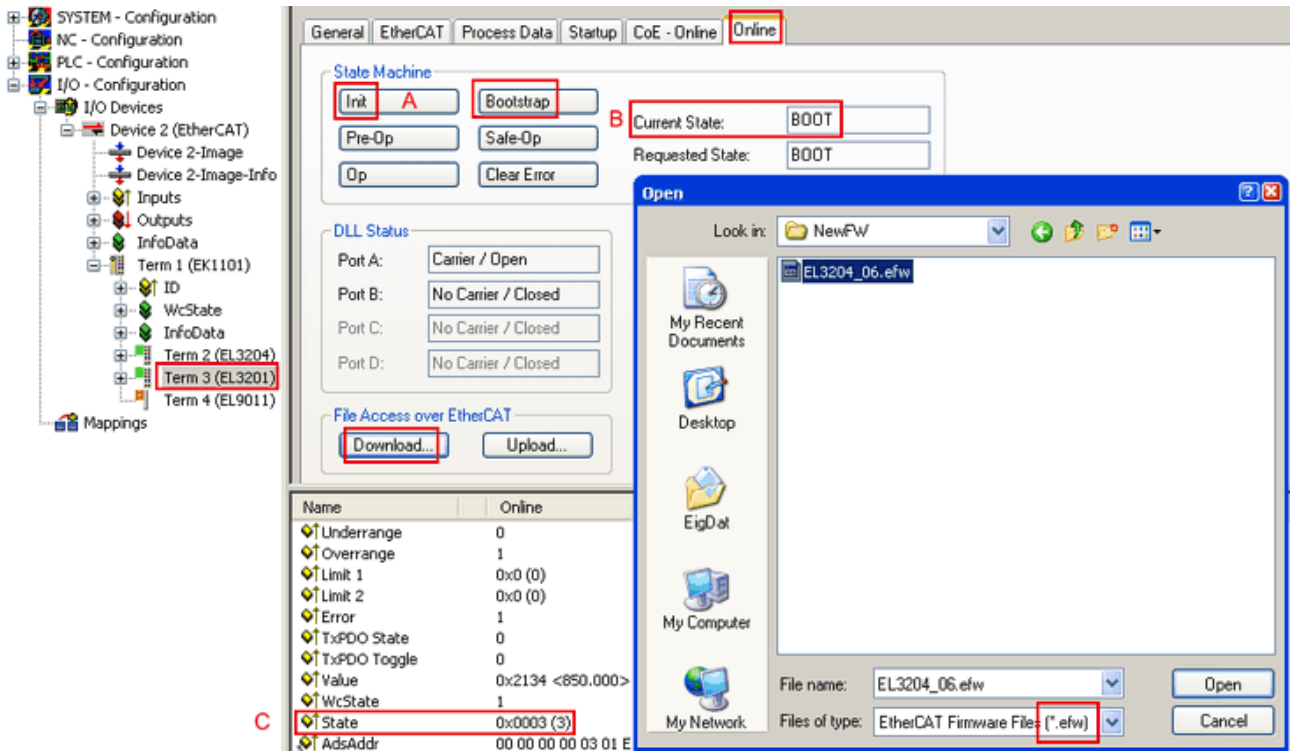


Fig. 150: Firmware Update

Proceed as follows, unless instructed otherwise by Beckhoff support.

- Switch slave to INIT (A)
- Switch slave to BOOTSTRAP
- Check the current status (B, C)
- Download the new \*efw file
- After the download switch to INIT, then OP
- Switch off the slave briefly

### 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.

The screenshot shows the TwinCAT System Manager interface. The left sidebar displays a tree view of the system configuration, with 'I/O - Configuration' expanded to show 'I/O Devices' and 'Device 2 (EtherCAT)' selected. The main window has tabs for 'General', 'Adapter', 'EtherCAT', and 'Online', with 'Online' currently active. A table lists the following data:

No	Addr	Name	State	CRC	Reg:0002
1	1001	Term 1 (EK1100)	OP	0	0x0002 (11)
2	1002	Term 2 (EL1012)	OP	0	0x0002 (10)
3	1003	Term 3 (EL2004)	OP	0	0x0002 (11)
4	1004	Term 4 (EL3102)	OP	0	0x0002 (10)
5	1005	Term 5 (EL4102)	OP	0	0x000B (11)
6	1006	Term 6 (EL5001)	OP	0	0x0002 (11)
7	1007	Term 7 (EL6751)	OP	0	0x000C (12)

Below the table, the 'Actual State' is 'OP', 'Send Frames' is 74237, 'Frames / sec' is 329, and 'Lost Frames' is 0. At the bottom, there are buttons for 'Init', 'Pre-Op', 'Safe-Op', 'Op', 'Clear CRC', and 'Clear Frames'. A status bar at the bottom right shows 'Local ()' and a red 'Free Run' button.

Fig. 151: FPGA firmware version definition

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

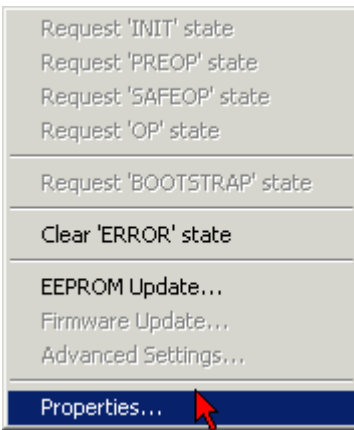


Fig. 152: Context menu Properties

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

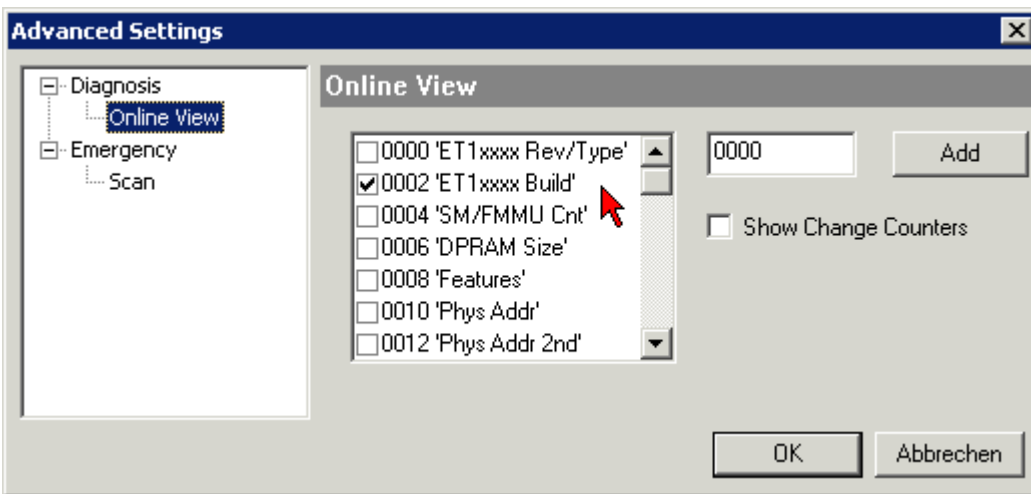


Fig. 153: Dialog Advanced Settings

**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**

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.

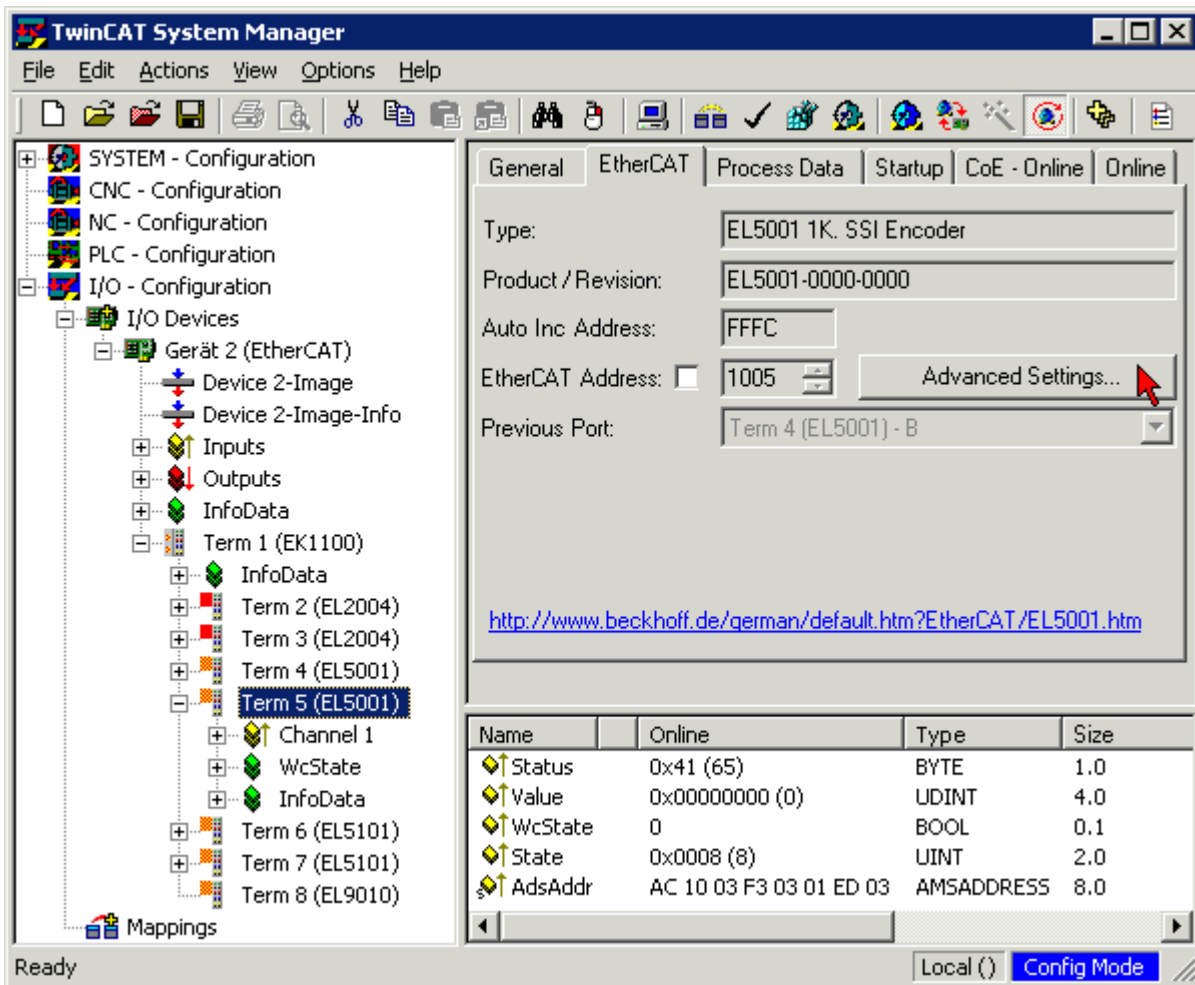


Fig. 154: Select dialog Advanced Settings

The Advanced Settings dialog appears. Under ESC Access/E<sup>2</sup>PROM/FPGA click on Write FPGA button,

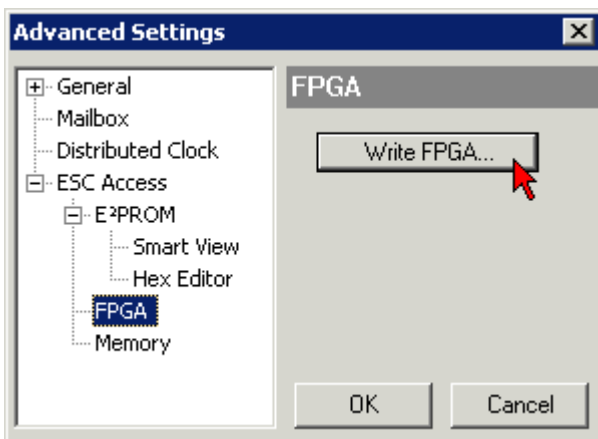


Fig. 155: Select dialog Write FPGA

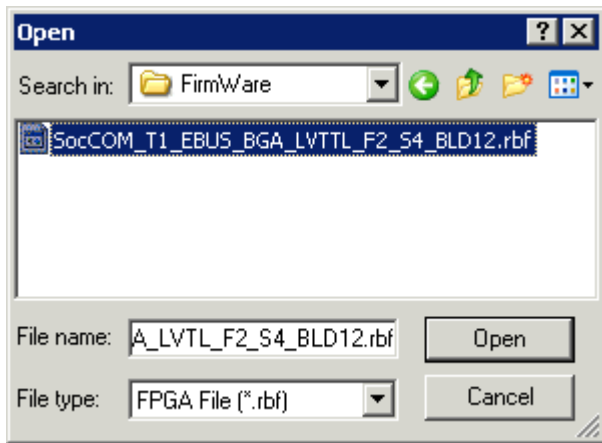



Fig. 156: *Select file*

Select the file (\*.rbf) with the new FPGA firmware, and transfer it to the EtherCAT device.

 <b>Attention</b>	<p><b>Risk of damage to the device!</b></p> <p>A firmware download to an EtherCAT device must never be interrupted! If this process is cancelled, the supply voltage switched off or the Ethernet connection interrupted, the EtherCAT device can only be recommissioned by the manufacturer!</p>
---	---

In order to activate the new FPGA firmware a restart (switching the power supply off and on again) of the EtherCAT device is required.

**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.

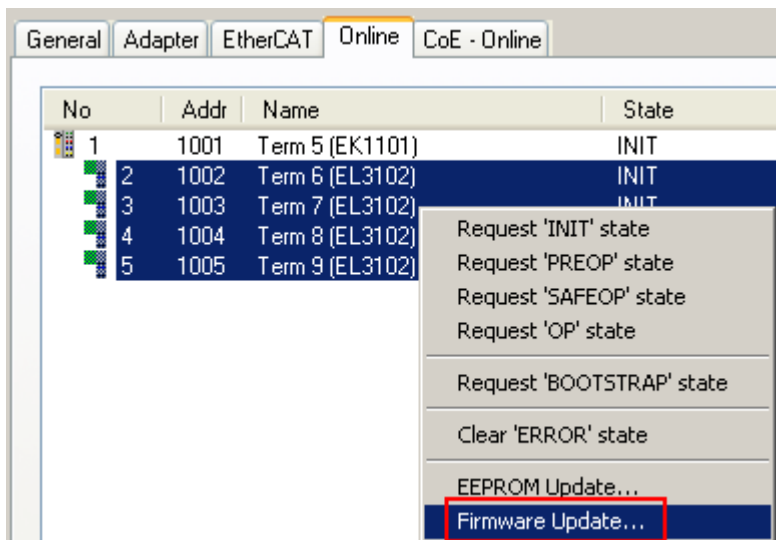


Fig. 157: *Multiple selection and firmware update*

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

## 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:

<http://www.beckhoff.com>

You will also find further documentation for Beckhoff components there.

### Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20  
33415 Verl  
Germany

Phone:	+49(0)5246/963-0
Fax:	+49(0)5246/963-198
e-mail:	info@beckhoff.com

### Beckhoff Support

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(0)5246/963-157
Fax:	+49(0)5246/963-9157
e-mail:	support@beckhoff.com

### Beckhoff 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(0)5246/963-460
Fax:	+49(0)5246/963-479
e-mail:	service@beckhoff.com

# List of illustrations

Fig. 1	EL5021 EL terminal, standard IP20 IO device with batch number and revision ID (since 2014/01).....	9
Fig. 2	EK1100 EtherCAT coupler, standard IP20 IO device with batch number .....	9
Fig. 3	CU2016 switch with batch number .....	9
Fig. 4	EL3202-0020 with batch numbers 26131006 and unique ID-number 204418 .....	10
Fig. 5	EP1258-00001 IP67 EtherCAT Box with batch number 22090101 and unique serial number 158102.....	10
Fig. 6	EP1908-0002 IP67 EtherCAT Safety Box with batch number 071201FF and unique serial number 00346070.....	10
Fig. 7	EL2904 IP20 safety terminal with batch number/date code 50110302 and unique serial number 00331701.....	10
Fig. 8	ELM3604-0002 terminal with ID number (QR code) 100001051 and unique serial number 44160201.....	11
Fig. 9	EL6720 .....	12
Fig. 10	Lightbus LWL-Ring .....	14
Fig. 11	Lightbus telegram structure .....	14
Fig. 12	States of the EtherCAT State Machine.....	16
Fig. 13	"CoE Online " tab .....	18
Fig. 14	Startup list in the TwinCAT System Manager .....	19
Fig. 15	Offline list.....	20
Fig. 16	Online list .....	21
Fig. 17	Spring contacts of the Beckhoff I/O components.....	23
Fig. 18	Correct configuration .....	26
Fig. 19	Incorrect configuration .....	26
Fig. 20	Recommended distances for standard installation position .....	28
Fig. 21	Other installation positions .....	29
Fig. 22	LEDs.....	30
Fig. 23	Relationship between user side (commissioning) and installation.....	32
Fig. 24	Control configuration with Embedded PC, input (EL1004) and output (EL2008) .....	33
Fig. 25	Initial TwinCAT 2 user interface.....	33
Fig. 26	Selection of the target system .....	34
Fig. 27	Specify the PLC for access by the TwinCAT System Manager: selection of the target system ..	34
Fig. 28	Select "Scan Devices..." .....	35
Fig. 29	Automatic detection of I/O devices: selection the devices to be integrated.....	35
Fig. 30	Mapping of the configuration in the TwinCAT 2 System Manager.....	36
Fig. 31	Reading of individual terminals connected to a device.....	36
Fig. 32	TwinCAT PLC Control after startup .....	37
Fig. 33	Sample program with variables after a compile process (without variable integration).....	38
Fig. 34	Appending the TwinCAT PLC Control project .....	38
Fig. 35	PLC project integrated in the PLC configuration of the System Manager .....	39
Fig. 36	Creating the links between PLC variables and process objects.....	39
Fig. 37	Selecting PDO of type BOOL .....	40
Fig. 38	Selecting several PDOs simultaneously: activate "Continuous" and "All types".....	40
Fig. 39	Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample .....	41
Fig. 40	Choose target system (remote) .....	42
Fig. 41	PLC Control logged in, ready for program startup .....	43

Fig. 42	Initial TwinCAT 3 user interface.....	44
Fig. 43	Create new TwinCAT project.....	44
Fig. 44	New TwinCAT3 project in the project folder explorer .....	45
Fig. 45	Selection dialog: Choose the target system .....	45
Fig. 46	Specify the PLC for access by the TwinCAT System Manager: selection of the target system ..	46
Fig. 47	Select "Scan" .....	46
Fig. 48	Automatic detection of I/O devices: selection the devices to be integrated.....	47
Fig. 49	Mapping of the configuration in VS shell of the TwinCAT3 environment.....	47
Fig. 50	Reading of individual terminals connected to a device.....	48
Fig. 51	Adding the programming environment in "PLC" .....	49
Fig. 52	Specifying the name and directory for the PLC programming environment .....	49
Fig. 53	Initial "Main" program of the standard PLC project.....	50
Fig. 54	Sample program with variables after a compile process (without variable integration).....	51
Fig. 55	Start program compilation.....	51
Fig. 56	Creating the links between PLC variables and process objects .....	52
Fig. 57	Selecting PDO of type BOOL .....	52
Fig. 58	Selecting several PDOs simultaneously: activate "Continuous" and "All types".....	53
Fig. 59	Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample .....	53
Fig. 60	TwinCAT development environment (VS shell): logged-in, after program startup.....	54
Fig. 61	System Manager "Options" (TwinCAT 2).....	56
Fig. 62	Call up under VS Shell (TwinCAT 3) .....	56
Fig. 63	Overview of network interfaces .....	56
Fig. 64	EtherCAT device properties(TwinCAT 2): click on „Compatible Devices...“ of tab “Adapter” .....	57
Fig. 65	Windows properties of the network interface.....	57
Fig. 66	Exemplary correct driver setting for the Ethernet port .....	58
Fig. 67	Incorrect driver settings for the Ethernet port .....	59
Fig. 68	TCP/IP setting for the Ethernet port .....	60
Fig. 69	Identifier structure .....	61
Fig. 70	OnlineDescription information window (TwinCAT 2) .....	62
Fig. 71	Information window OnlineDescription (TwinCAT 3).....	62
Fig. 72	File OnlineDescription.xml created by the System Manager .....	63
Fig. 73	Indication of an online recorded ESI of EL2521 as an example.....	63
Fig. 74	Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3).....	63
Fig. 75	Using the ESI Updater (>= TwinCAT 2.11).....	65
Fig. 76	Using the ESI Updater (TwinCAT 3).....	65
Fig. 77	Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3) .....	66
Fig. 78	Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3).....	66
Fig. 79	Selecting the Ethernet port .....	66
Fig. 80	EtherCAT device properties (TwinCAT 2) .....	67
Fig. 81	Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3).....	67
Fig. 82	Selection dialog for new EtherCAT device .....	68
Fig. 83	Display of device revision .....	68
Fig. 84	Display of previous revisions .....	69
Fig. 85	Name/revision of the terminal.....	69
Fig. 86	EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3).....	70
Fig. 87	Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3).....	71

Fig. 88	Scan Devices (left: TwinCAT 2; right: TwinCAT 3).....	71
Fig. 89	Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3).....	72
Fig. 90	Detected Ethernet devices .....	72
Fig. 91	Example default state .....	72
Fig. 92	Installing EthetCAT terminal with revision -1018 .....	73
Fig. 93	Detection of EtherCAT terminal with revision -1019.....	73
Fig. 94	Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3) .....	74
Fig. 95	Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3).....	74
Fig. 96	Scan progressexemplary by TwinCAT 2 .....	74
Fig. 97	Config/FreeRun query (left: TwinCAT 2; right: TwinCAT 3).....	74
Fig. 98	Displaying of “Free Run” and “Config Mode” toggling right below in the status bar .....	74
Fig. 99	TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3) .....	74
Fig. 100	Online display example .....	75
Fig. 101	Faulty identification .....	75
Fig. 102	Identical configuration (left: TwinCAT 2; right: TwinCAT 3).....	76
Fig. 103	Correction dialog .....	76
Fig. 104	Name/revision of the terminal.....	77
Fig. 105	Correction dialog with modifications .....	78
Fig. 106	Dialog “Change to Compatible Type...” (left: TwinCAT 2; right: TwinCAT 3).....	78
Fig. 107	TwinCAT 2 Dialog Change to Alternative Type .....	78
Fig. 108	Branch element as terminal EL3751.....	79
Fig. 109	“General” tab.....	79
Fig. 110	„EtherCAT“ tab.....	80
Fig. 111	“Process Data” tab.....	81
Fig. 112	Configuring the process data.....	82
Fig. 113	„Startup“ tab.....	83
Fig. 114	“CoE – Online” tab .....	84
Fig. 115	Dialog “Advanced settings”.....	85
Fig. 116	„Online“ tab .....	86
Fig. 117	"DC" tab (Distributed Clocks).....	87
Fig. 118	Selection of the diagnostic information of an EtherCAT Slave .....	89
Fig. 119	Basic EtherCAT Slave Diagnosis in the PLC.....	90
Fig. 120	EL3102, CoE directory .....	92
Fig. 121	Example of commissioning aid for a EL3204 .....	93
Fig. 122	Default behaviour of the System Manager .....	94
Fig. 123	Default target state in the Slave .....	94
Fig. 124	PLC function blocks .....	95
Fig. 125	Illegally exceeding the E-Bus current .....	96
Fig. 126	Warning message for exceeding E-Bus current .....	96
Fig. 127	EL6720 – Input Toggle and CDLstate in the TwinCAT tree.....	97
Fig. 128	EL6720 – Status information in the CoE-Online tab .....	98
Fig. 129	BK2000 – Status information .....	99
Fig. 130	AoE Address.....	99
Fig. 131	Display EL6720 Lightbus master hardware and Lightbus device in the TwinCAT tree .....	100

Fig. 132	Scanning existing devices .....	101
Fig. 133	Scanning existing Lightbus devices .....	101
Fig. 134	Default setting of an EL6720 Lightbus device .....	102
Fig. 135	Missing link message.....	102
Fig. 136	Bus load display.....	102
Fig. 137	CDL load display.....	103
Fig. 138	Warning message System Manager.....	103
Fig. 139	DataInvalid overload display .....	103
Fig. 140	Test box number .....	104
Fig. 141	Overview of Lightbus-relevant CoE entries .....	104
Fig. 142	Advanced Settings .....	105
Fig. 143	Device identifier consisting of name EL3204-0000 and revision -0016 .....	111
Fig. 144	Scan the subordinate field by right-clicking on the EtherCAT device in Config/FreeRun mode .	112
Fig. 145	Configuration is identical .....	112
Fig. 146	Change dialog .....	113
Fig. 147	EEPROM Update .....	113
Fig. 148	Selecting the new ESI .....	114
Fig. 149	Display of EL3204 firmware version .....	115
Fig. 150	Firmware Update .....	115
Fig. 151	FPGA firmware version definition .....	116
Fig. 152	Context menu Properties .....	117
Fig. 153	Dialog Advanced Settings .....	117
Fig. 154	Select dialog Advanced Settings .....	118
Fig. 155	Select dialog Write FPGA .....	118
Fig. 156	Select file .....	119
Fig. 157	Multiple selection and firmware update .....	119