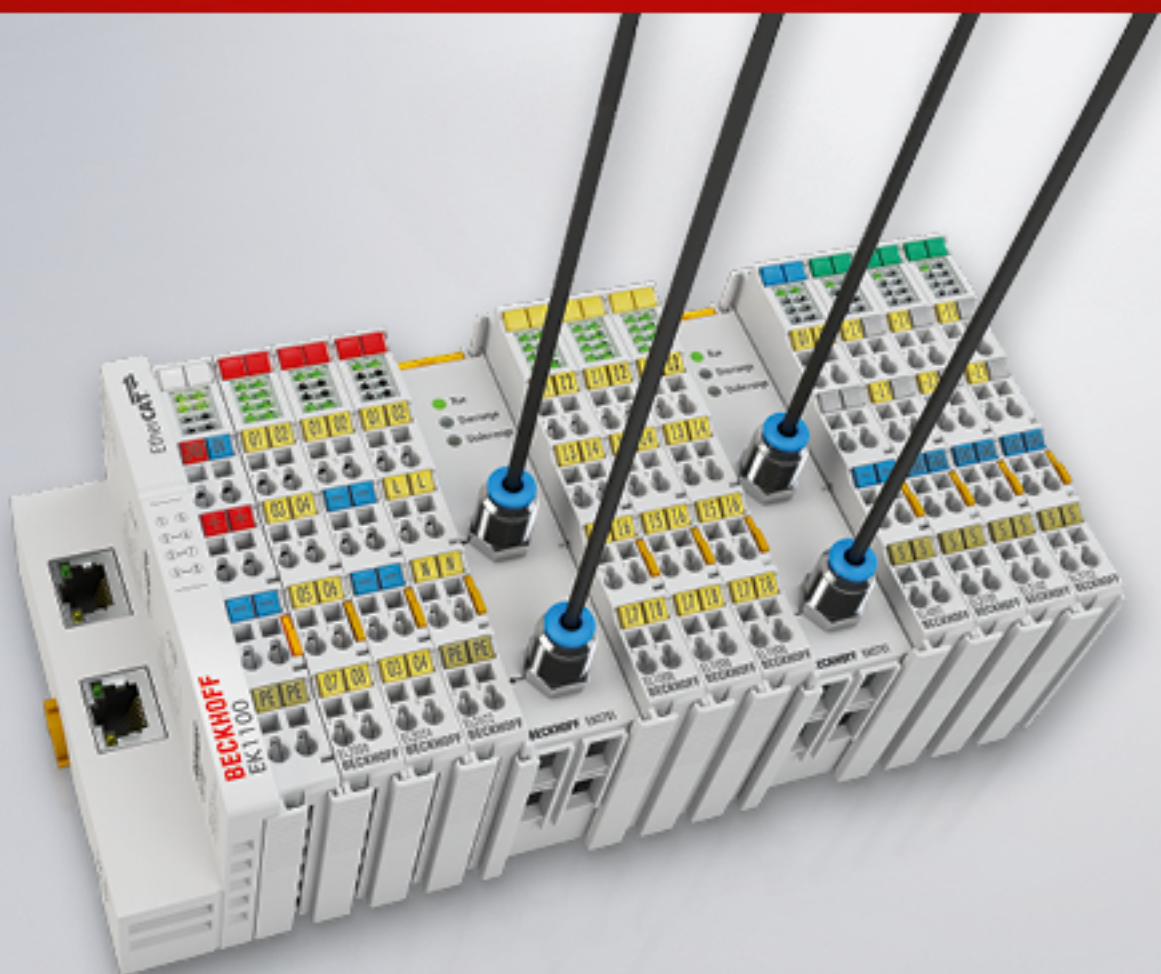# BECKHOFF New Automation Technology

Documentation | EN

EM37xx

Pressure Measuring Modules

Table of contents

1   Foreword

1.1   Product overview Pressure Measuring Terminal

1.2   Notes on the documentation

Intended audience

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

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

Disclaimer

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

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

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

Trademarks

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

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

Copyright

Third-party brands

Trademarks of third parties may be used in this documentation. You can find the trademark notices here:
https://www.beckhoff.com/trademarks

1.3   Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Signal words

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

Personal injury warnings

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

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

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

Warning of damage to property or environment

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

Information on handling the product

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

1.4   Guide through documentation

NOTICE	
	Further components of documentation This documentation describes device-specific content. It is part of the modular documentation concept for Beckhoff I/O components. For the use and safe operation of the device / devices described in this documentation, additional cross-product descriptions are required, which can be found in the following table.

Title	Description
EtherCAT System Documentation (PDF)	• System overview • EtherCAT basics • Cable redundancy • Hot Connect • EtherCAT devices configuration
I/O Analog Manual (PDF)	Notes on I/O components with analog in and outputs
Infrastructure for EtherCAT/Ethernet (PDF)	Technical recommendations and notes for design, implementation and testing
Software Declarations I/O (PDF)	Open source software declarations for Beckhoff I/O components

The documentations can be viewed at and downloaded from the Beckhoff website (www.beckhoff.com) via:

- the "Documentation and Download" area of the respective product page,
- the Download finder,
- the Beckhoff Information System.

If you have any suggestions or proposals for our documentation, please send us an e-mail stating the documentation title and version number to: documentation@beckhoff.com

BECKHOFF

1.5   Documentation issue status

Version	Comment
1.6.0	• Update chapter "Basic function principles" • Update structure
1.5	• Chapter "Electronic access to the BIC (eBIC)" added • Update Technical data • Update chapter "Application examples" • Chapter "Disposal" added • Update Revision status • Update structure
1.4	• New front pate • Update revision status • Update structure
1.3	• Update chapter "UL notice" • Update chapter "Firmware compatibility" • Update structure
1.2	• Update chapter "Notes on the documentation" • Update chapter "Technical data" • Addenda chapter "Instructions for ESD protection" • Chapter "Analog technical notices - specifications" replaced by chapter "Notices on analog specifications" • Addenda chapter "UL notice" • Update revision status
1.1	• EM3712 added • Update title page • Addenda temperature coefficient in chapter "Analog technical notices - specifications" • Update chapter "EtherCAT AL Status Codes" • Update chapter "TwinCAT 2.1x" -> "TwinCAT Development Environment" • "TwinCAT Quick Start" added
1.0	• Documentation recreated

1.6 Version identification of EtherCAT devices

1.6.1 General notes on marking

Designation

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

- family key
- type
- version
- revision

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

Notes

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

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

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

- The **revision** -0016 shows the technical progress, such as the extension of features with regard to the EtherCAT communication, and is managed by Beckhoff.
  In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation.
  Associated and synonymous with each revision there is usually a description (ESI, EtherCAT Slave Information) in the form of an XML file, which is available for download from the Beckhoff web site.
  From 2014/01 the revision is shown on the outside of the IP20 terminals, see Fig. *"EL2872 with revision 0022 and serial number 01200815"*.

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

1.6.2   Version identification of EL terminals

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

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

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

Example with serial number 12 06 3A 02:

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

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

1.6.3   Beckhoff Identification Code (BIC)

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

Fig. 2: BIC as data matrix code (DMC, code scheme ECC200)
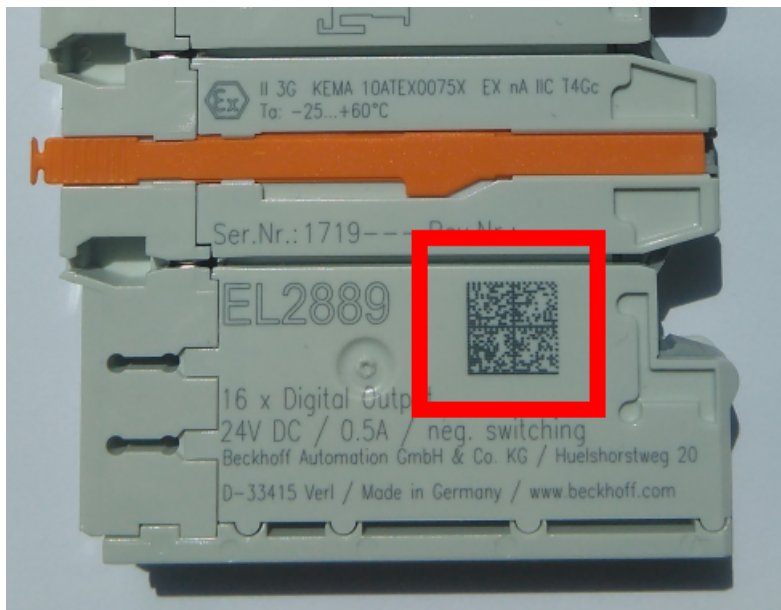
The BIC will be introduced step by step across all product groups.

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

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

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

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

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

Posi-tion	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	**Beckhoff order number**	1P	8	**1P**072222
2	Beckhoff Traceability Number (BTN**)**	**Unique serial number, see note below**	SBTN	12	**SBTN**k4p562d7
3	Article description	**Beckhoff article description, e.g. EL1008**	1K	32	**1K**EL1809
4	Quantity	**Quantity in packaging unit, e.g. 1, 10, etc.**	Q	6	**Q**1
5	Batch number	Optional: Year and week of production	2P	14	**2P**401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	**51S**678294
7	Variant number	Optional: Product variant number on the basis of standard products	30P	12	**30P**F971, 2*K183
...					

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

Structure of the BIC

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

1P072222**SBTN**k4p562d7**1K**EL1809 **Q**1 **51S**678294

Accordingly as DMC:

Fig. 3: Example DMC **1P**072222**SBTN**k4p562d7**1K**EL1809 **Q**1 **51S**678294

BTN

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

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

1.6.4   Electronic access to the BIC (eBIC)

Electronic BIC (eBIC)

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

The interface that the product can be electronically addressed by is crucial for the electronic readout.

K-bus devices (IP20, IP67)

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

EtherCAT devices (IP20, IP67)

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

Beckhoff also stores the eBIC in the ESI-EEPROM. The eBIC was introduced into Beckhoff IO production (terminals, box modules) in 2020; as of 2023, implementation is largely complete.

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

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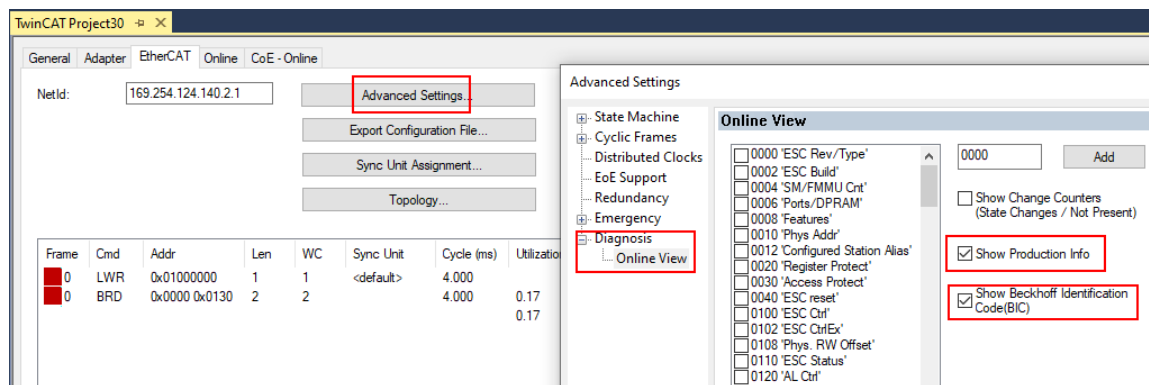
    - The BTN and its contents are then displayed:

    - Note: As shown in the figure, the production data HW version, FW version, and production date, which have been programmed since 2012, can also be displayed with "Show production info".
    - Access from the PLC: From TwinCAT 3.1. build 4024.24, the functions *FB_EcReadBIC* and *FB_EcReadBTN* for reading into the PLC are available in the Tc2_EtherCAT library from v3.3.19.0.
- EtherCAT devices with a CoE directory may also have the object 0x10E2:01 to display their own eBIC, which can also be easily accessed by the PLC:

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

Index	Name	Flags	Value
1000	Device type	RO	0x015E1389 (22942601)
1008	Device name	RO	ELM3704-0000
1009	Hardware version	RO	00
100A	Software version	RO	01
100B	Bootloader version	RO	J0.1.27.0
1011:0	Restore default parameters	RO	> 1 <
1018:0	Identity	RO	> 4 <
10E2:0	Manufacturer-specific Identification C...	RO	> 1 <
10E2:01	SubIndex 001	RO	1P158442SBTN0008ekp1KELM3704 Q1 2P482001000016
10F0:0	Backup parameter handling	RO	> 1 <
10F3:0	Diagnosis History	RO	> 21 <
10F8	Actual Time Stamp	RO	0x170bfb277e

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

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

• The following auxiliary functions are available for processing the BIC/BTN data in the PLC in *Tc2_Utilities* as of TwinCAT 3.1 build 4024.24

   ◦ F_SplitBIC: The function splits the Beckhoff Identification Code (BIC) sBICValue into its components using known identifiers and returns the recognized substrings in the ST_SplittedBIC structure as a return value

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

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

• Technical background
The new BIC information is written as an additional category in the ESI-EEPROM during device production. The structure of the ESI content is largely dictated by the ETG specifications, therefore the additional vendor-specific content is stored using a category in accordance with the ETG.2010. ID 03 tells all EtherCAT masters that they may not overwrite these data in the event of an update or restore the data after an ESI update.
The structure follows the content of the BIC, see here. The EEPROM therefore requires approx. 50..200 bytes of memory.

• Special cases

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

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

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

PROFIBUS; PROFINET, and DeviceNet devices

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

2   EM37xx - Product description

2.1   EM3701 - Introduction

Run LED
Overrange LED
Underrange LED

○ Run
● Overrange
● Underrange

X1 / P+

X2 / P-

BECKHOFF EM3701
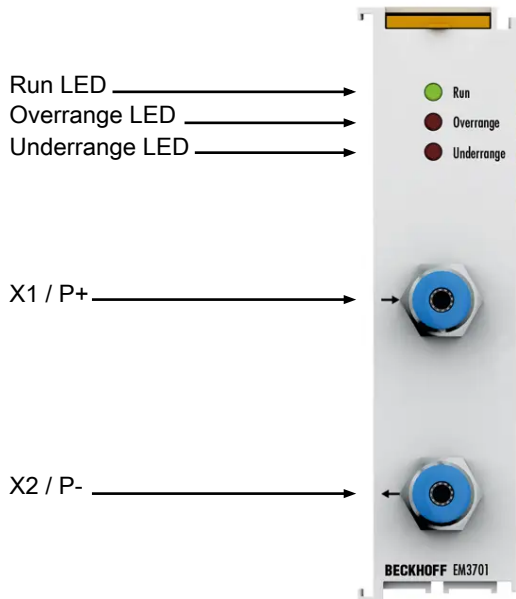
Fig. 4: EM3701 Top view

Single channel differential pressure measuring module

The EM3701 differential pressure measuring module enables direct measurement of pressure differences between two hose connections. The pressure difference is available in the fieldbus as a 16 bit value in the range between -100 hPa to +100 hPa (-100 mbar to +100 mbar). The status LEDs indicate proper function or errors such as overrange.

2.2   EM3702 - Introduction
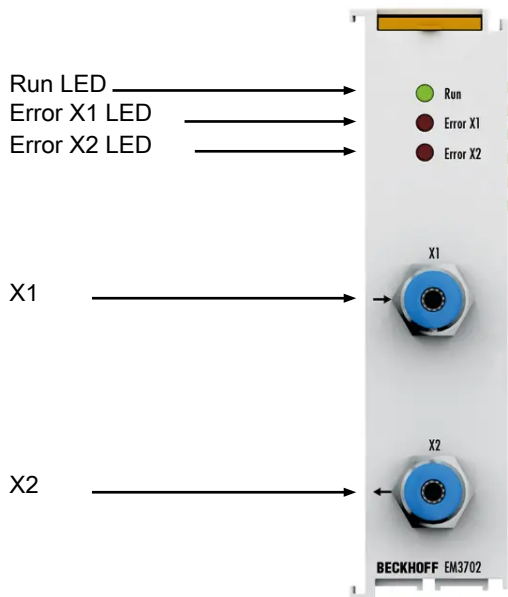
Fig. 5: EM3702 Top view

Dual channel relative pressure measuring module for 0 to 7500 hPa (0 to 7.5 bar)

The EM3702 relative pressure measuring module enables direct measurement of two pressure values at the hose connections. The pressure is determined as a pressure difference to the ambiance of the EM3702 and is available in the fieldbus as a 16 bit value. The status LEDs indicate proper function or errors such as overrange.

2.3   EM3712 - Introduction
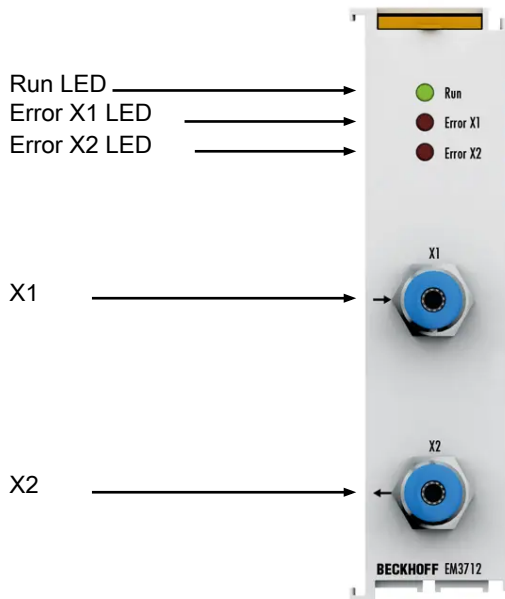
Fig. 6: EM3712 Top view

Dual channel relative pressure measuring module for -1000 hPa to +1000 hPa (-1 bar to +1 bar)

The relative pressure measuring module EM3712 allows the direct measurement of two negative pressure values on the hose connections. The pressure is determined as the difference to the environment of the EM3712 and is available in the field bus with 16 bit resolution. The status LEDs indicate proper function or errors such as range exceedance.

2.4   Quick links

- Mounting and wiring [▶ 29]
- Process data and settings [▶ 120]
- Application examples [▶ 117]
- Object description and parameterization [▶ 128]

2.5   EM37xx - Technical data

Technical data	EM3701	EM3702	EM3712
Number of inputs	1	2	2
Technology	Differential pressure measurement	Relative pressure measurement	Relative pressure measurement
Measuring range **)	-100 hPa to +100 hPa (-100 mbar to +100 mbar)	0 hPa to 7500 hPa (0 bar to 7.5 bar)	-1000 hPa to +1000 hPa (-1 bar to +1 bar)
Permissible overpressure	max. ±500 hPa differential	max. +10,000 hPa	+5000 hPa
permissible media	non-aggressive gases		
Resolution	0.1 hPa (0.1 mbar) per digit	1 hPa (1 mbar) per digit	1 hPa (1 mbar) per digit
Measurement error	±3% (of the full scale value [▶ 134])		
Measuring speed	typically 4 ms		
Pressure connectors	screwing plug [▶ 36], M12 x 1		
Power supply for the electronics	via the E-bus		
Current consumption via E-bus	70 mA	60 mA	70 mA
	The pre-calculated maximum E-Bus current [▶ 18] is displayed in the TwinCAT System Manager as a column value.		
Electrical isolation	500 V (E-bus/signal voltage)		
Bit width in process image	2 bytes status, 2 bytes value per channel		
Dimensions without tubes (w x h x d)	approx. 26,5 mm x 100 mm x 52 mm (width aligned: 24 mm)		
Weight	approx. 95 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		
Assembly	on a 35 mm mounting rail [▶ 29] (e.g. DIN rail TH 35-7.5 conforming to EN 60715)		
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27, EN 60068-2-29		
EMC immunity/ emission	conforms to EN 61000-6-2 / EN 61000-6-4		
Protection class	IP20		
Installation position	variable		
Approvals/markings *)	CE, EAC, UKCA cULus [▶ 37]		

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

**) Media compatibility: Use clean and dry gas!

3   Basics communication

3.1   EtherCAT basics

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

3.2   EtherCAT cabling – wire-bound

The cable length between two EtherCAT devices must not exceed 100 m. This results from the FastEthernet technology, which, above all for reasons of signal attenuation over the length of the cable, allows a maximum link length of 5 + 90 + 5 m if cables with appropriate properties are used. See also the Design recommendations for the infrastructure for EtherCAT/Ethernet.

Cables and connectors

For connecting EtherCAT devices only Ethernet connections (cables + plugs) that meet the requirements of at least category 5 (CAt5) according to EN 50173 or ISO/IEC 11801 should be used. EtherCAT uses 4 wires for signal transfer.

EtherCAT uses RJ45 plug connectors, for example. The pin assignment is compatible with the Ethernet standard (ISO/IEC 8802-3).

Pin	Color of conductor	Signal	Description
1	yellow	TD +	Transmission Data +
2	orange	TD -	Transmission Data -
3	white	RD +	Receiver Data +
6	blue	RD -	Receiver Data -

Due to automatic cable detection (auto-crossing) symmetric (1:1) or cross-over cables can be used between EtherCAT devices from Beckhoff.

> ● **Recommended cables**
>
> **i** It is recommended to use the appropriate Beckhoff components e.g.
> - cable sets ZK1090-9191-xxxx respectively
> - RJ45 connector, field assembly ZS1090-0005
> - EtherCAT cable, field assembly ZB9010, ZB9020
>
> Suitable cables for the connection of EtherCAT devices can be found on the Beckhoff website!
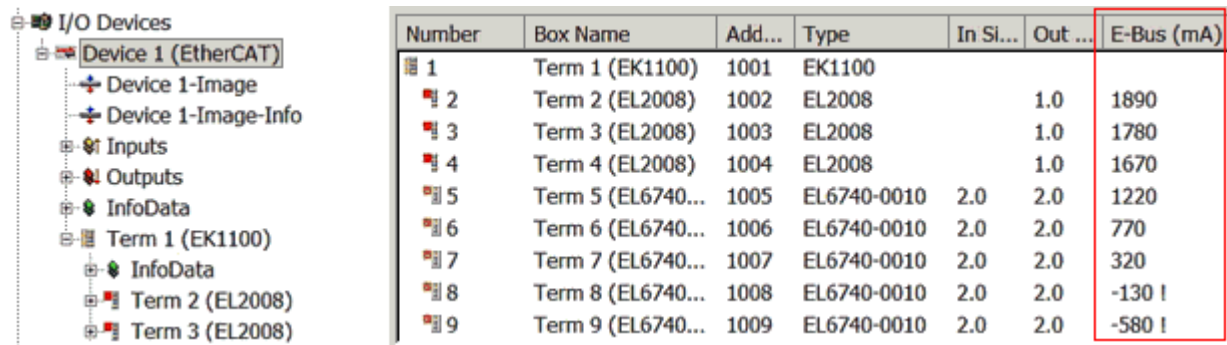
E-Bus supply

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 (see details in respective device documentation).
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. 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.

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

Fig. 7: System manager current calculation

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

BECKHOFF

3.3 General notes for setting the watchdog

The EtherCAT terminals are equipped with a safety device (watchdog) which, e. g. in the event of interrupted process data traffic, switches the outputs (if present) to a presettable state after a presettable time, depending on the device and setting, e. g. to FALSE (off) or an output value.

The EtherCAT slave controller features two watchdogs:

- Sync Manager (SM) watchdog (default: 100 ms)
- Process Data (PDI) watchdog (default: 100 ms)

Their times are individually parameterized in TwinCAT as follows:
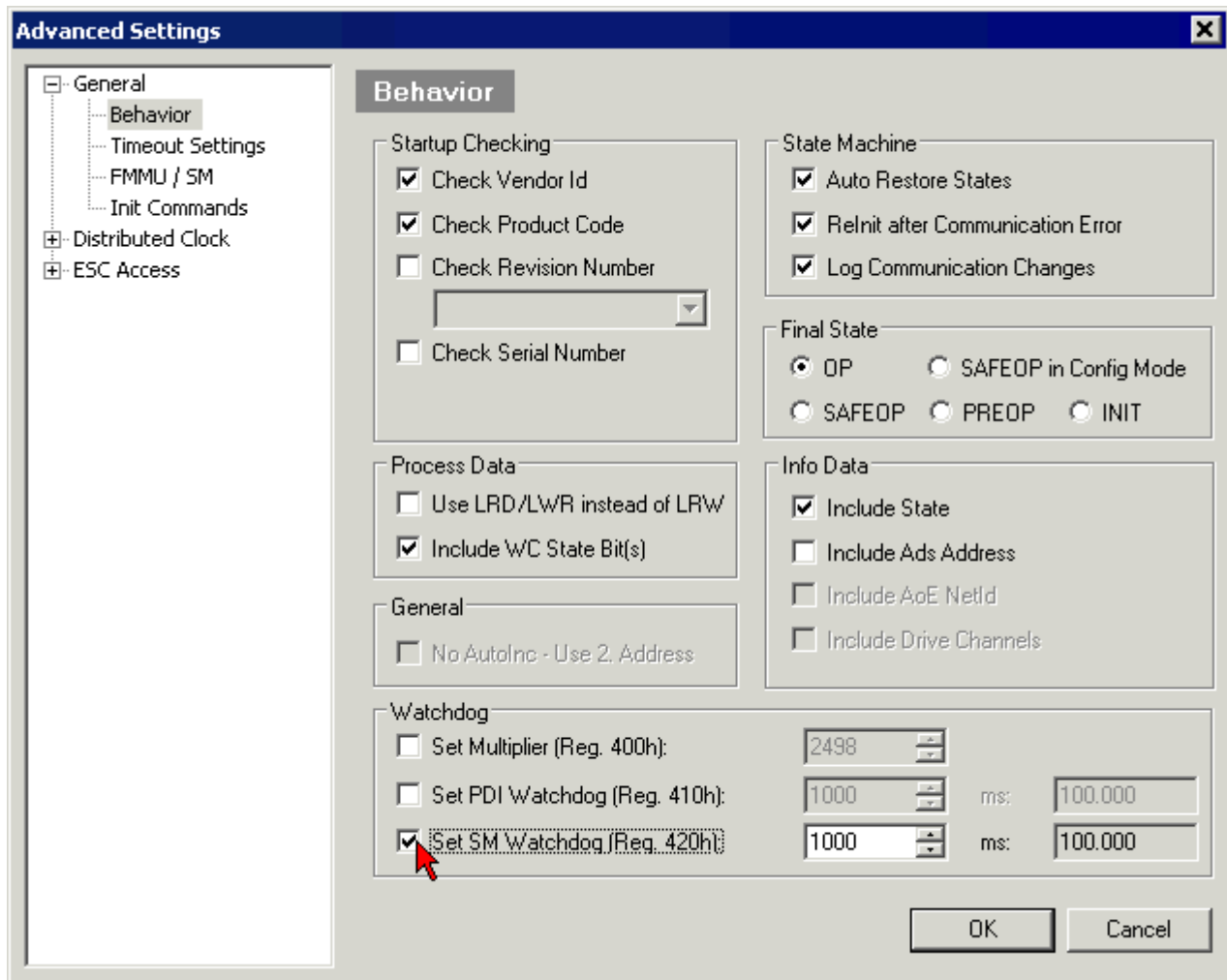
Fig. 8: eEtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

- the Multiplier Register 400h (hexadecimal, i. e. 0x0400) is valid for both watchdogs.
- each watchdog has its own timer setting 410h or 420h, which together with the Multiplier results in a resulting time.
- important: the Multiplier/Timer setting is only loaded into the slave at EtherCAT startup if the checkbox in front of it is activated.
- if it is not checked, nothing is downloaded and the setting located in the ESC remains unchanged.
- the downloaded values can be seen in the ESC registers 400h, 410h and 420h: ESC Access -> Memory

SM watchdog (SyncManager Watchdog)

The SyncManager watchdog is reset with each successful EtherCAT process data communication with the terminal. If, for example, no EtherCAT process data communication with the terminal takes place for longer than the set and activated SM watchdog time due to a line interruption, the watchdog is triggered. The status of the terminal (usually OP) remains unaffected. The watchdog is only reset again by a successful EtherCAT process data access.

The SyncManager watchdog is therefore a monitoring for correct and timely process data communication with the ESC from the EtherCAT side.

The maximum possible watchdog time depends on the device. For example, for "simple" EtherCAT slaves (without firmware) with watchdog execution in the ESC it is usually up to 170 seconds. For complex EtherCAT slaves (with firmware) the SM watchdog function is usually parameterized via register 400h/420h but executed by the microcontroller (µC) and can be significantly lower. In addition, the execution may then be subject to a certain time uncertainty. Since the TwinCAT dialog may allow inputs up to 65535, a test of the desired watchdog time is recommended.

PDI watchdog (Process Data Watchdog)

If there is no PDI communication with the ESC for longer than the set and activated Process Data Interface (PDI) watchdog time, this watchdog is triggered.

The PDI is the internal interface of the ESC, e.g. to local processors in the EtherCAT slave. With the PDI watchdog this communication can be monitored for failure.

The PDI watchdog is therefore a monitoring for correct and timely process data communication with the ESC, but viewed from the application side.

Calculation

Watchdog time = [1/25 MHz * (Watchdog multiplier + 2)] * SM/PDI watchdog

Example: default setting Multiplier = 2498, SM watchdog = 1000 => 100 ms

The value in "Watchdog multiplier + 2" in the formula above corresponds to the number of 40ns base ticks representing one watchdog tick.

⚠ **CAUTION**
Undefined state possible!
The function for switching off the SM watchdog via SM watchdog = 0 is only implemented in terminals from revision -0016. In previous versions this operating mode should not be used.

⚠ **CAUTION**
Damage of devices and undefined state possible!
If the SM watchdog is activated and a value of 0 is entered the watchdog switches off completely. This is the deactivation of the watchdog! Set outputs are NOT set in a safe state if the communication is interrupted.

3.4   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
- Operational

- Bootstrap

The regular state of each EtherCAT slave after bootup is the OP state.
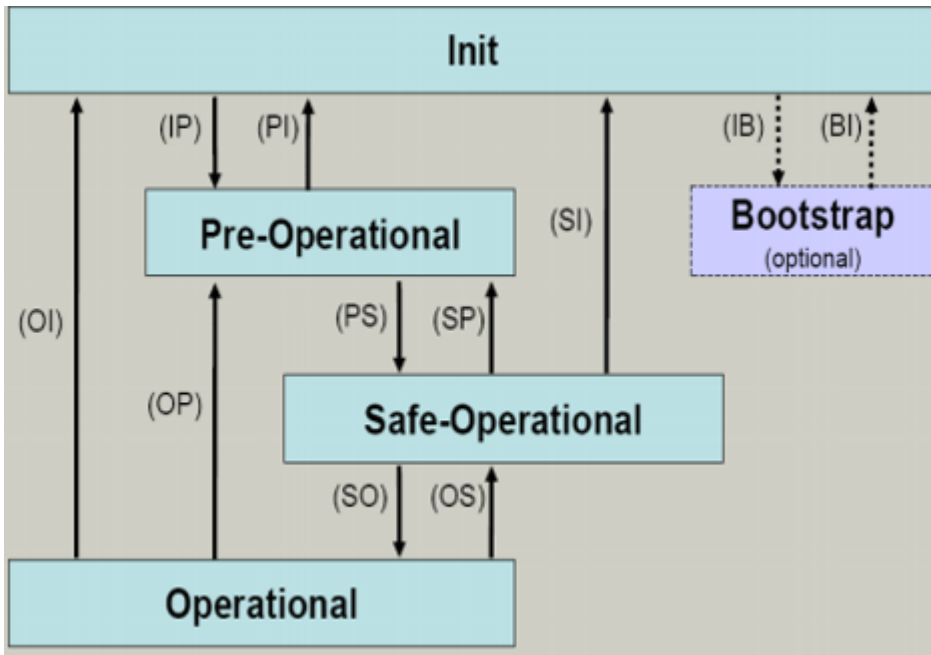

Fig. 9: States of the EtherCAT State Machine

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 Fieldbus Memory Management Unit (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 Dual Port (DP)-RAM areas of the ESC.

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

ⓘ Outputs in SAFEOP state

The default set watchdog monitoring sets the outputs of the ESC 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 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.

3.5 CoE Interface

General description

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

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

The value ranges are

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

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

The relevant ranges for EtherCAT fieldbus users are:

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

Other important ranges are:

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

Availability

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

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

Fig. 10: "CoE Online" tab

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

NOTICE

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

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

- Keep a startup list if components have to be replaced,

- Distinction between online/offline dictionary,

- Existence of current XML description (download from the Beckhoff website),

- "CoE-Reload" for resetting the changes

- Program access during operation via PLC (see TwinCAT 3 | PLC Library: "Tc2_EtherCAT" and Example program R/W CoE)

Data management and function "NoCoeStorage"

Some parameters, particularly the setting parameters of the slave, are configurable and writeable,

- via the System Manager (Fig. "CoE Online" tab) by clicking.
  This is useful for commissioning of the system or slaves. Click on the row of the index to be parameterized and enter a value in the "SetValue" dialog.

- from the control system or PLC via ADS, e.g. through blocks from the TcEtherCAT.lib library.
  This is recommended for modifications while the system is running or if no System Manager or operating staff are available.

Data management

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

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

- If the function is supported: the function is activated by entering the code word 0x12345678 once in CoE index 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.

- If the function is not supported: continuous changing of CoE values is not permissible in view of the lifetime limit.

Startup list

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

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

Recommended approach for manual modification of CoE parameters

- Make the required change in the System Manager
  (the values are stored locally in the EtherCAT slave).

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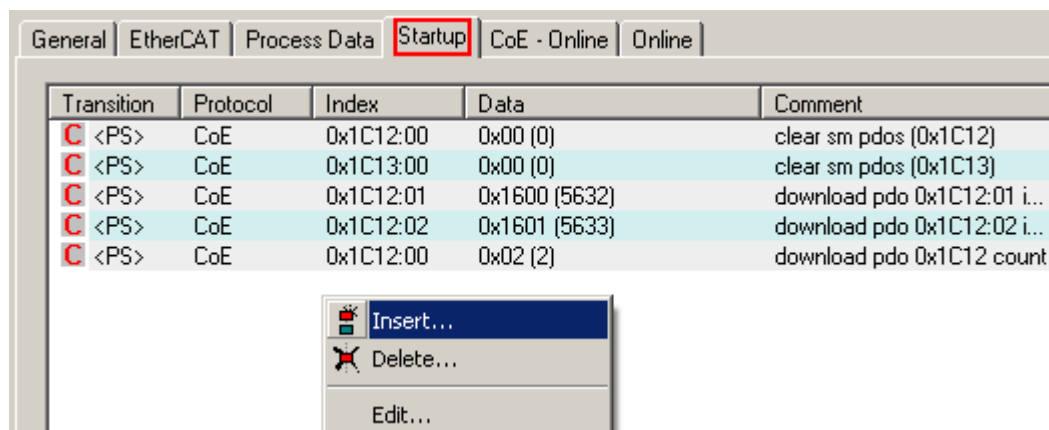 The order of the Startup entries is usually irrelevant.

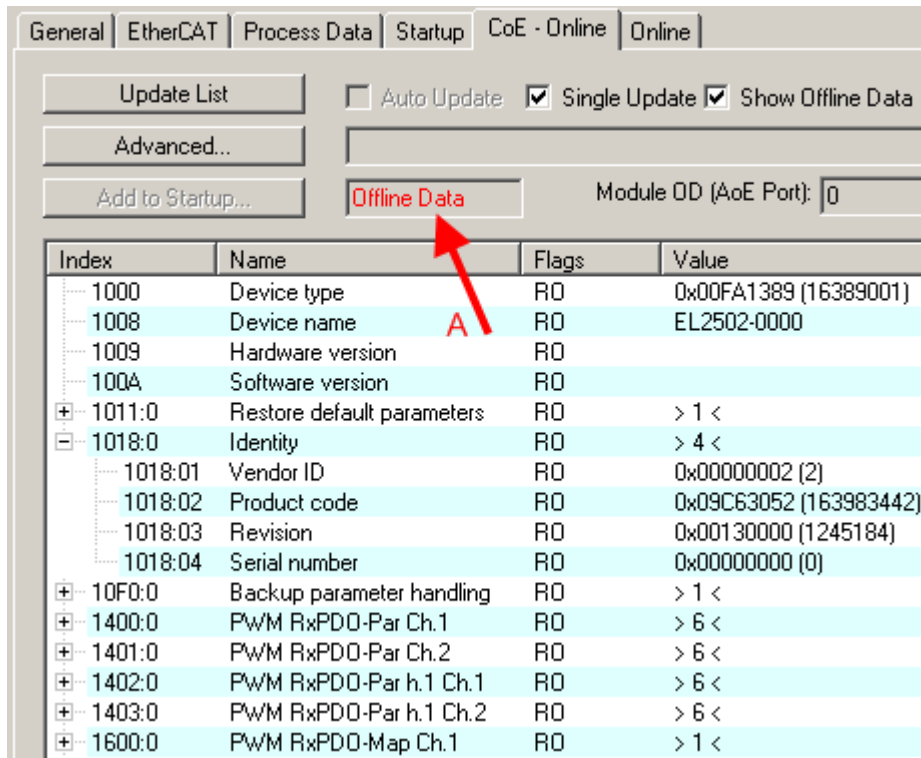Fig. 11: Startup list in the TwinCAT System Manager

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

Online / offline list

When working with the TwinCAT System Manager, a distinction must be made as to whether the EtherCAT device is currently "available", i.e. switched on and connected via EtherCAT - i.e. **online** - or whether a configuration is created **offline** without slaves being connected.

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 Data** is shown in red.

Fig. 12: 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 status of the device is displayed in the CoE.
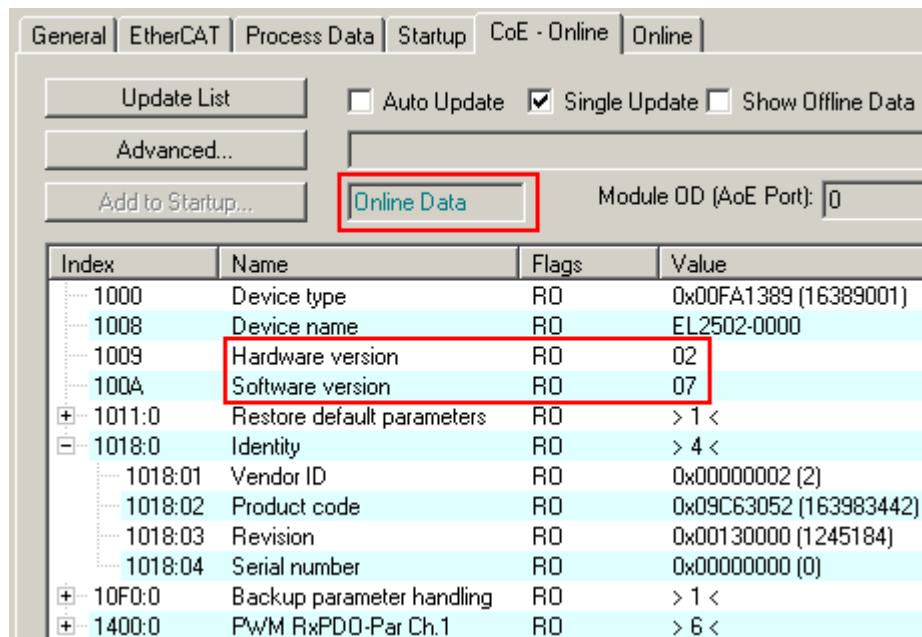 - ◦ **Online Data** is shown in green.

Fig. 13: 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 input terminal also has four logical channels and therefore four identical sets of parameter data for the channels. In order to avoid having to list each channel in the documentation, the placeholder "n" tends to be used for the individual channel numbers.

In the CoE system 16 indices, each with 255 subindices, are generally sufficient for representing all channel parameters. The channel-based order is therefore arranged in 16_{dec} or 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.

3.6   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

NOTICE
Destruction of the devices by electrostatic discharge possible!
The devices contain components at risk from electrostatic discharge caused by improper handling.

- When handling the components, ensure that there is no electrostatic discharge; also avoid touching the spring contacts directly (see illustration).
- Contact with highly insulating materials (synthetic fibers, plastic films, etc.) should be avoided when handling components at the same time.
- When handling the components, ensure that the environment (workplace, packaging and persons) is properly earthed.
- Each bus station must be terminated on the right-hand side with the EL9011 or EL9012 end cap to ensure the degree of protection and ESD protection.
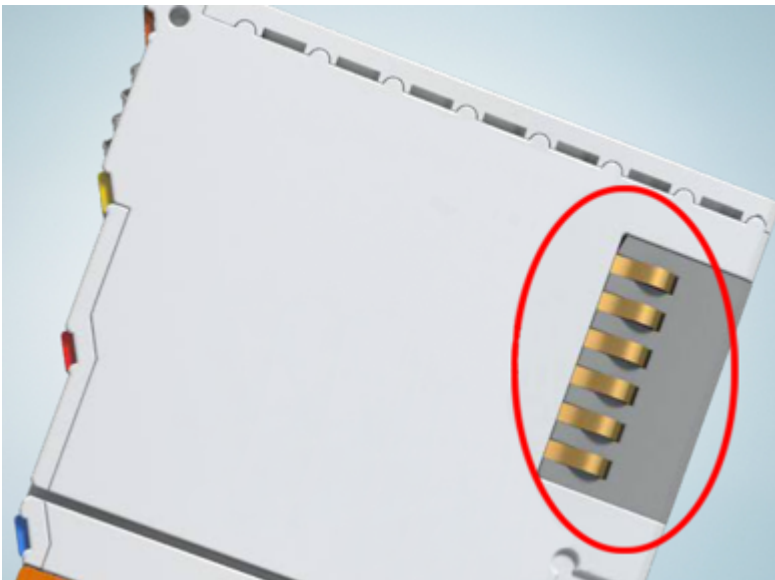

Fig. 14: Spring contacts of the Beckhoff I/O components

4.2 Recommended mounting rails

Terminal Modules and 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:

- mounting rail TH 35-7.5 with 1 mm material thickness (according to EN 60715)
- mounting rail TH 35-15 with 1.5 mm material thickness
- mounting rail TH 35-15 with 2.2 to 2.5 mm material thickness (according to EN 60715)

	For older modules pay attention to the material thickness of the mounting rail
i	Modules of KM10x4, KM10x8, KM2004, KM2008, KM26x4 and KM2774 series, do not fit to the mounting rail TH 35-15 with 2.2 to 2.5 mm material thickness (according to EN 60715)!

BECKHOFF

4.3 Mounting and demounting - top front unlocking

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

ℹ️ Fixing of mounting rails

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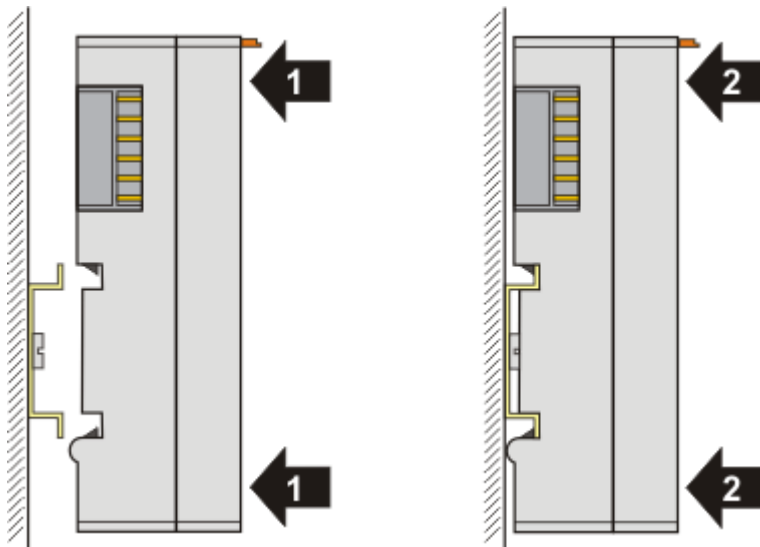

⚠ WARNING

Risk of electric shock and damage of device!

Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the Bus Terminals!

Mounting

- Fit the mounting rail to the planned assembly location.

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

- Attach the cables.

Demounting

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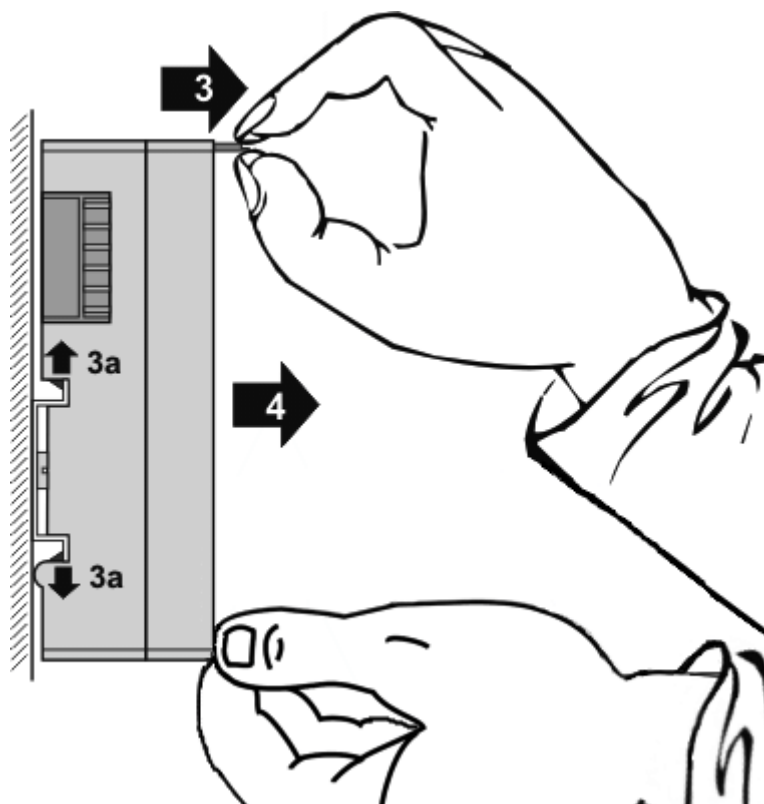
- Pull (4) the terminal module away from the mounting surface.
  Avoid canting of the module; you should stabilize the module with the other hand, if required.

4.4   Installation positions

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

Optimum installation position (standard)

The optimum installation position requires the mounting rail to be installed horizontally and the connection surfaces of the EL- / KL terminals to face forward (see Fig. "Recommended distances for standard installation position"). The terminals are ventilated from below, which enables optimum cooling of the electronics through convection. "From below" is relative to the acceleration of gravity.
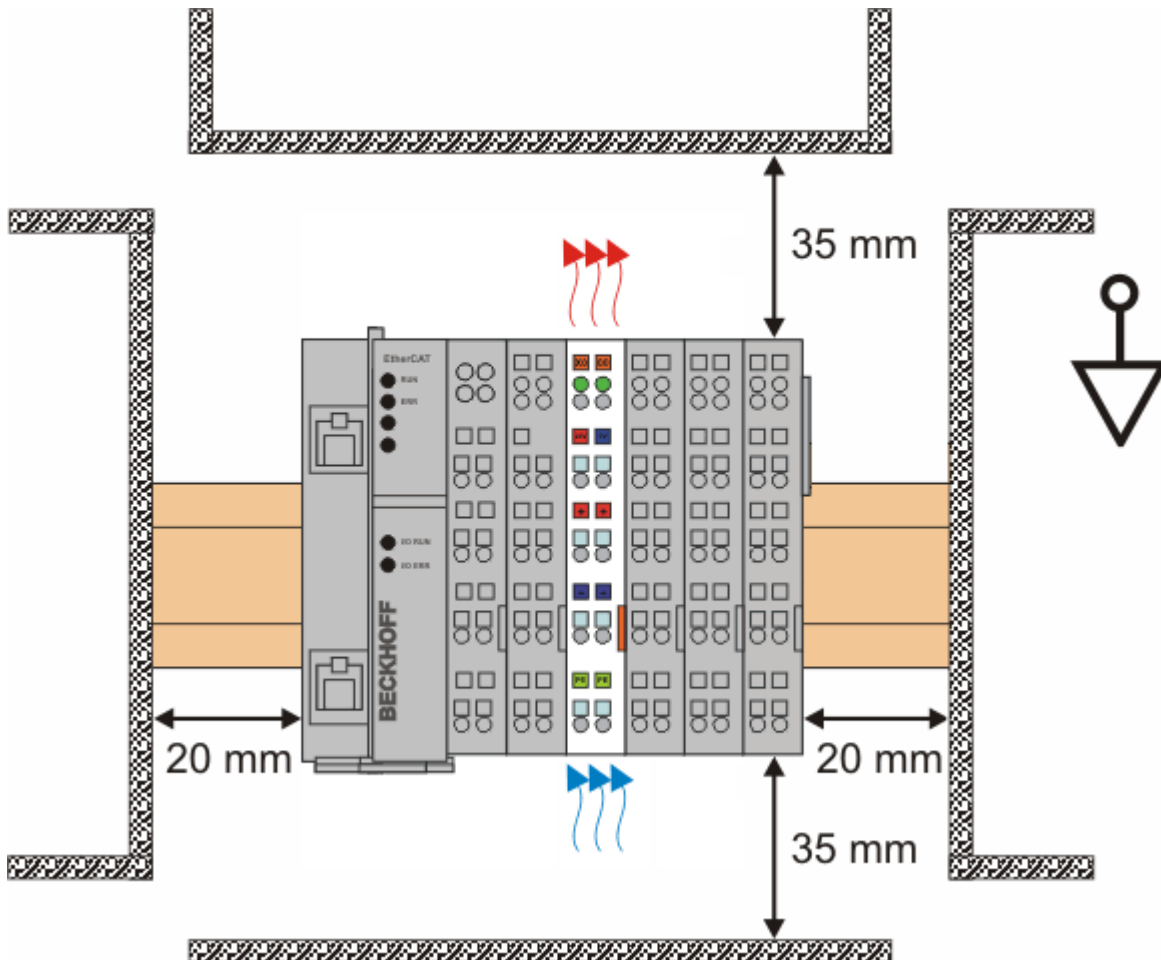
Fig. 15: Recommended distances for standard installation position

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".

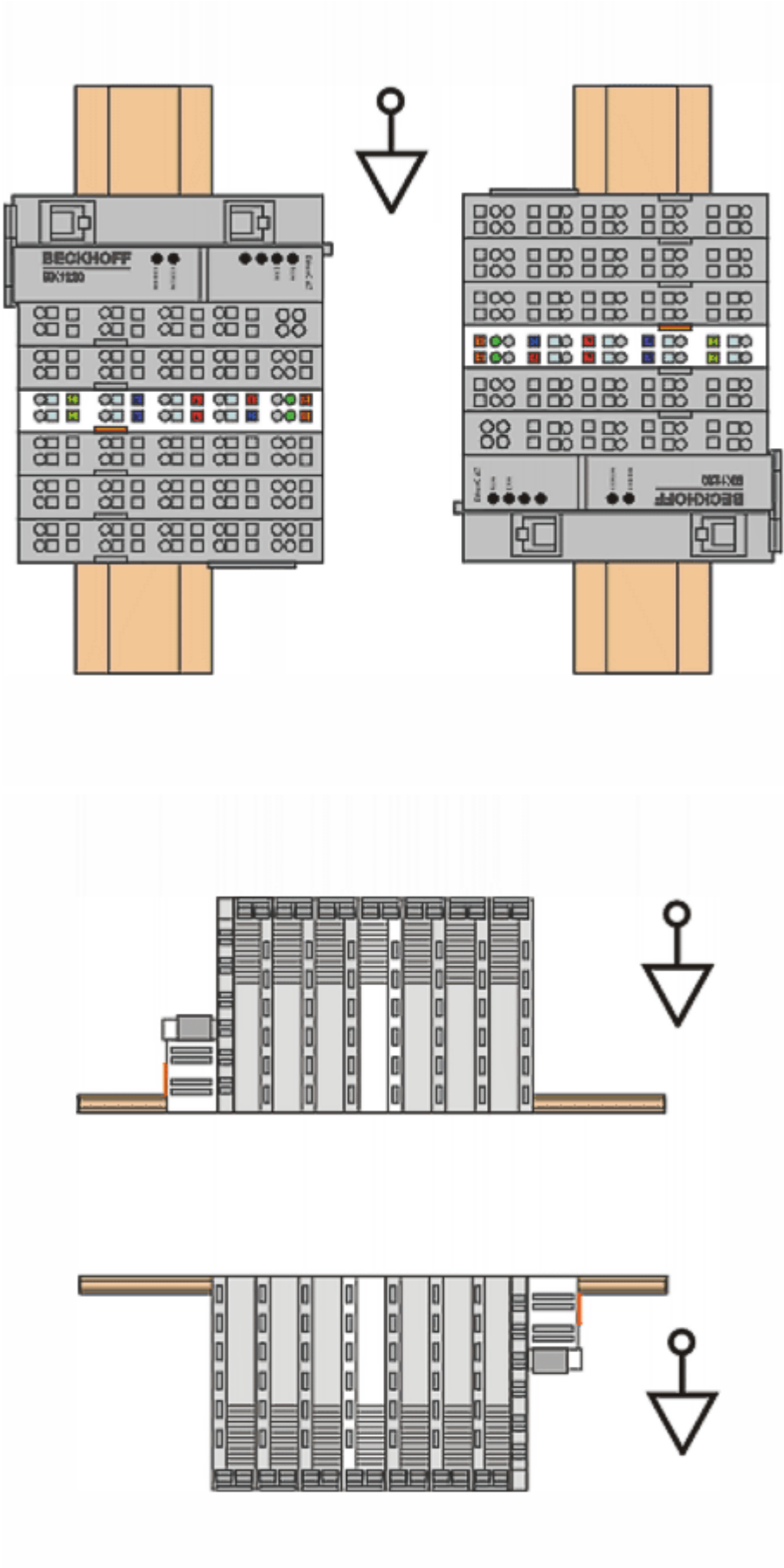The minimum distances to ambient specified above also apply to these installation positions.

Fig. 16: Other installation positions

4.5 Positioning of passive Terminals

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

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

Examples for positioning of passive terminals (highlighted)
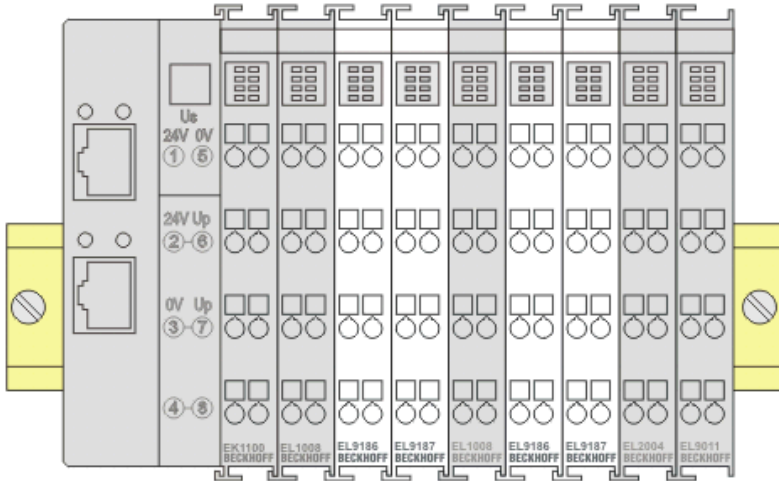
Fig. 17: Correct positioning
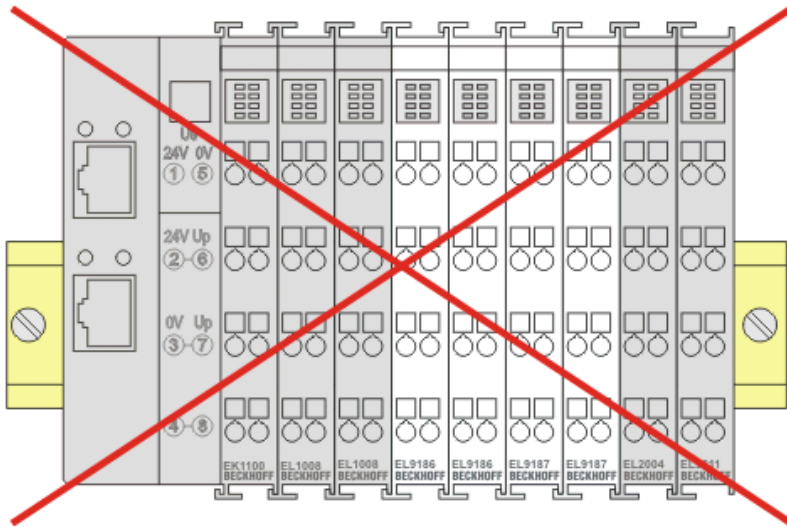
Fig. 18: Incorrect positioning

4.6   EM3701 - LED displays

Run LED ⟶
Overrange LED ⟶
Underrange LED ⟶

○ Run
● Overrange
● Underrange

X1 / P+ ⟶

X2 / P- ⟶

BECKHOFF EM3701
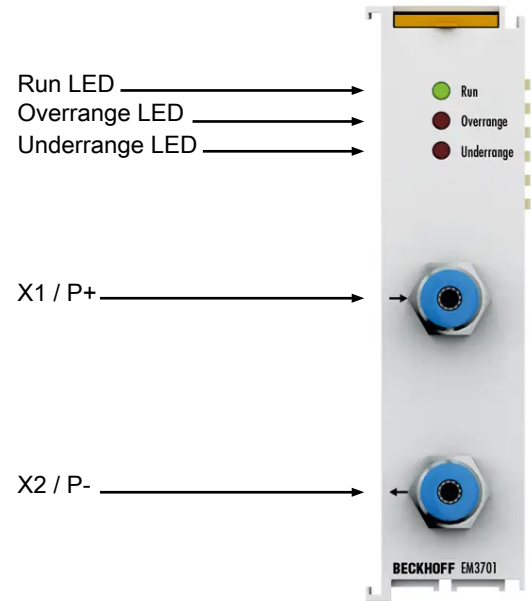
Fig. 19: EM3701 Top view

LED	Display	
Run (green)	off	Data communication on the E-Bus is not active
	on	Data transmission on the E-Bus is active
Overrange (red)	on	The differential pressure is above the permitted measuring range [▶ 17]
Underrange (red)	on	The differential pressure is below the permitted measuring range [▶ 17]

4.7 EM3702, EM3712 - LED displays

Run LED
Error X1 LED
Error X2 LED

X1

X2
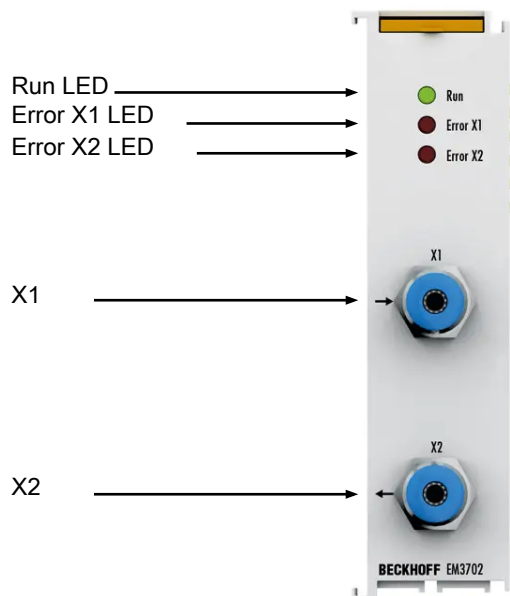
Fig. 20: LEDs, taking the EM3702 as an example

LED	Display	
Run (green)	off	Data transmission on the E-bus is not active
	on	Data transmission on the E-bus is active
Error X1 (red)	on	The pressure at connection X1 is below (underrange) or above (overrange) the permitted measuring range [▶ 17]
Error X2 (red)	on	The pressure at connection X2 is below (underrange) or above (overrange) the permitted measuring range [▶ 17]

4.8 EM37xx - Connection

The air hoses are connected to push-in fittings.

Technical Data	Threaded push-in fitting
Type	QSS-4-F
Outer hose diameter	4 mm
Nominal diameter	2.6 mm
Thread	M12 x 1
Width across flats	14 mm

4.9   Note - power supply

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

4.10   UL notice

⚠ CAUTION	
cULus	**Application**
	Beckhoff EtherCAT modules are intended for use with Beckhoff's UL Listed EtherCAT System only.

⚠ CAUTION	
cULus	**Examination**
	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).

⚠ CAUTION	
cULus	**For devices with Ethernet connectors**
	Not for connection to telecommunication circuits.

Basic principles

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

4.11 Disposal

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

5   Commissioning

5.1   TwinCAT Quick Start

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

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

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

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

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

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

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

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

The following relationship is envisaged between the user PC and individual control elements:
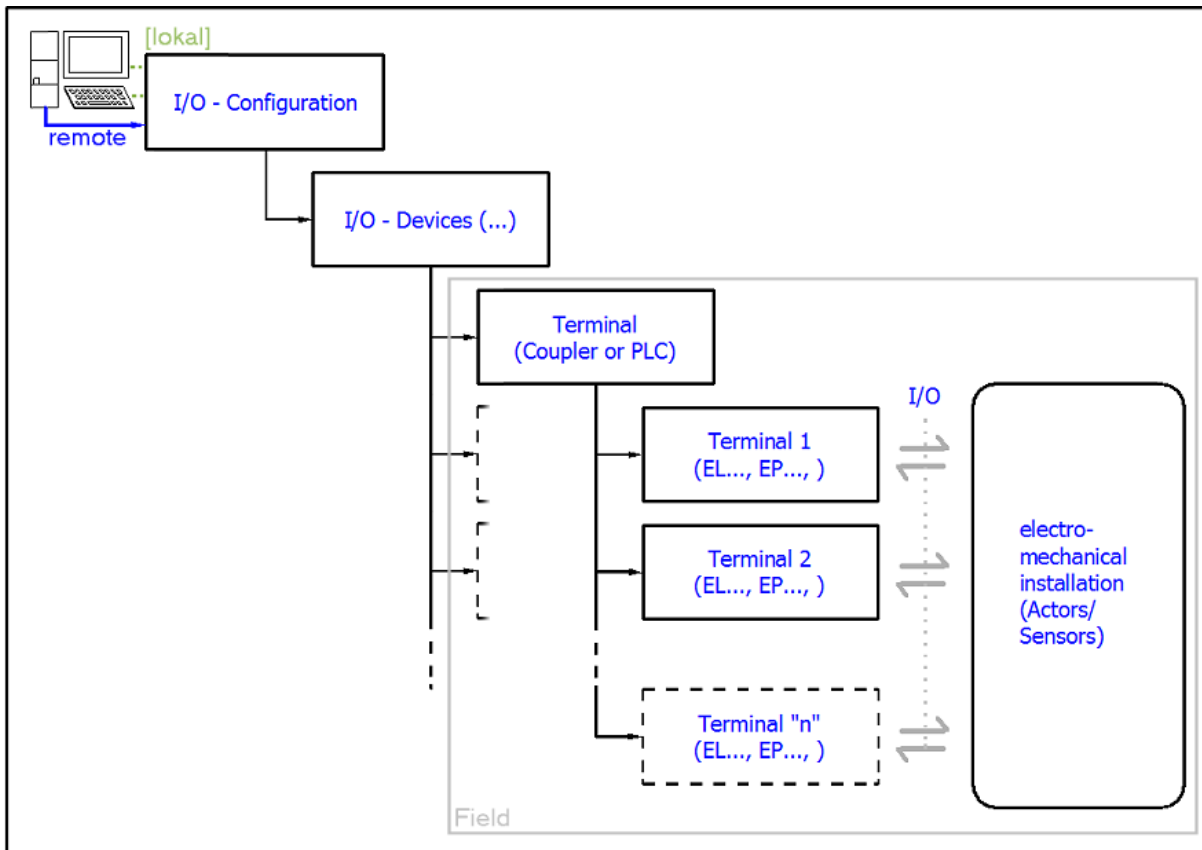
BECKHOFF

Fig. 21: Relationship between user side (commissioning) and installation

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

Example configuration (actual configuration)

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

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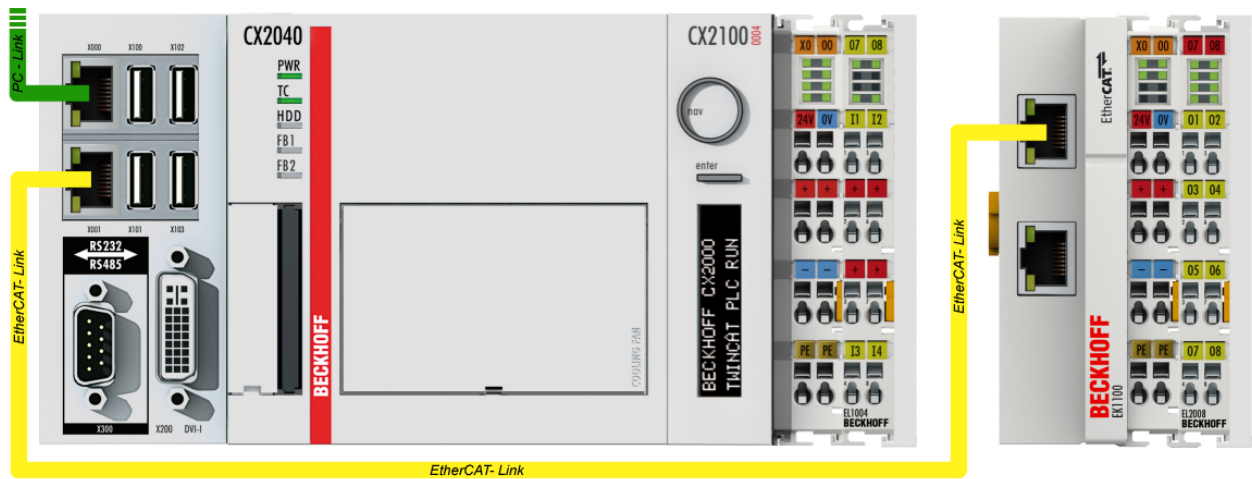- (Optional via X000: a link to an external PC for the user interface)

Fig. 22: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

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

5.1.1   TwinCAT 2

Startup

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

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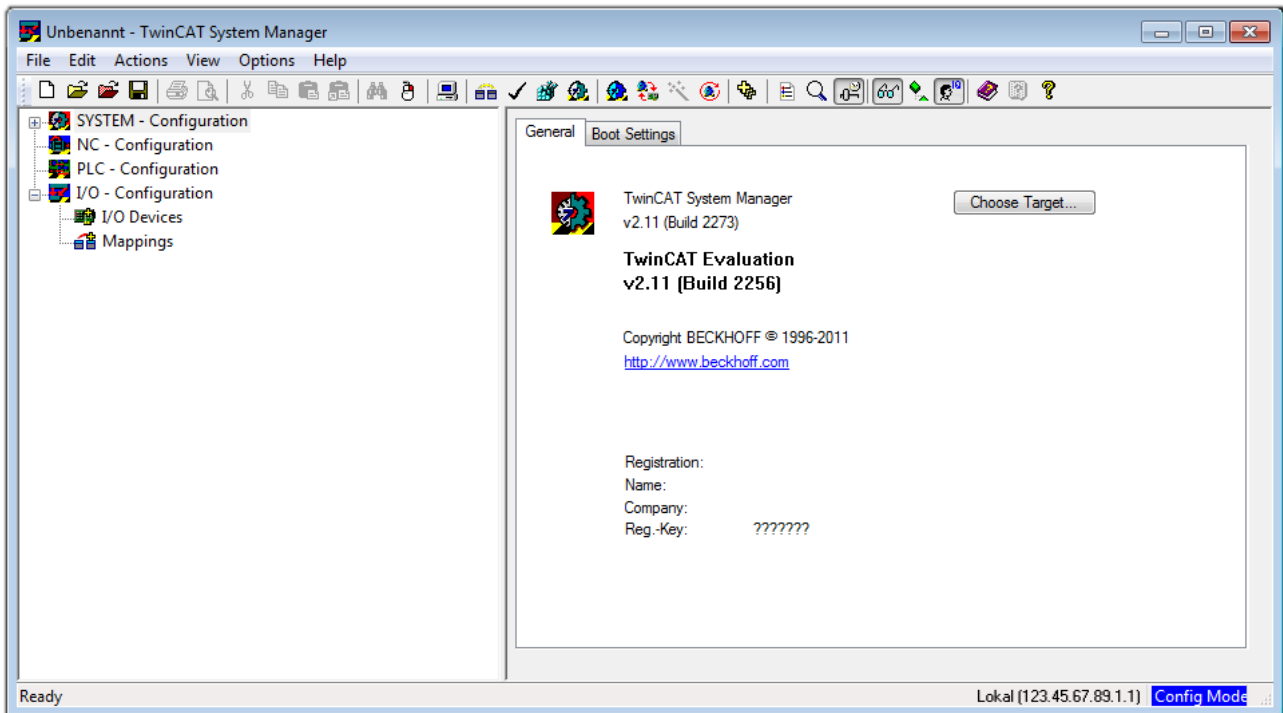

Fig. 23: Initial TwinCAT 2 user interface

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system, including the user interface (standard) is installed on the respective PLC, TwinCAT can be used in local mode and thus the next step is "Insert Device [▶ 44]".

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

"Actions" → "Choose Target System...", the following window is opened for this via the symbol " 🖥 " or the "F8" key:
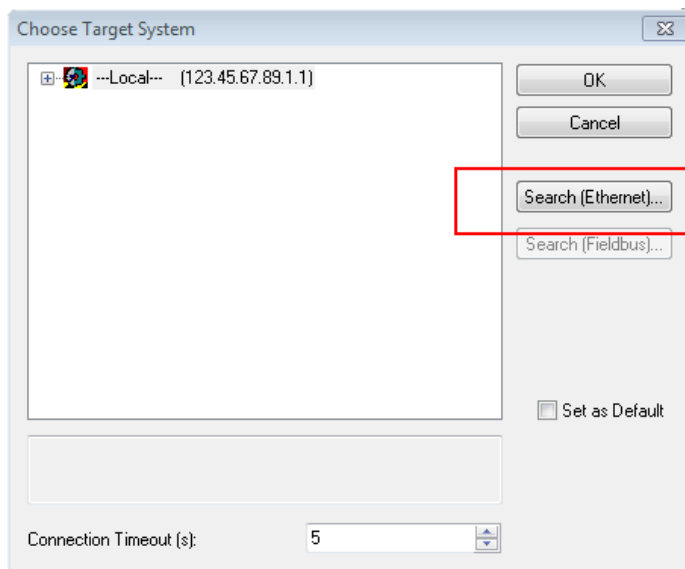
Fig. 24: Selection of the target system

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

- enter the known computer name after "Enter Host Name / IP:" (as shown in red)
- perform a "Broadcast Search" (if the exact computer name is not known)
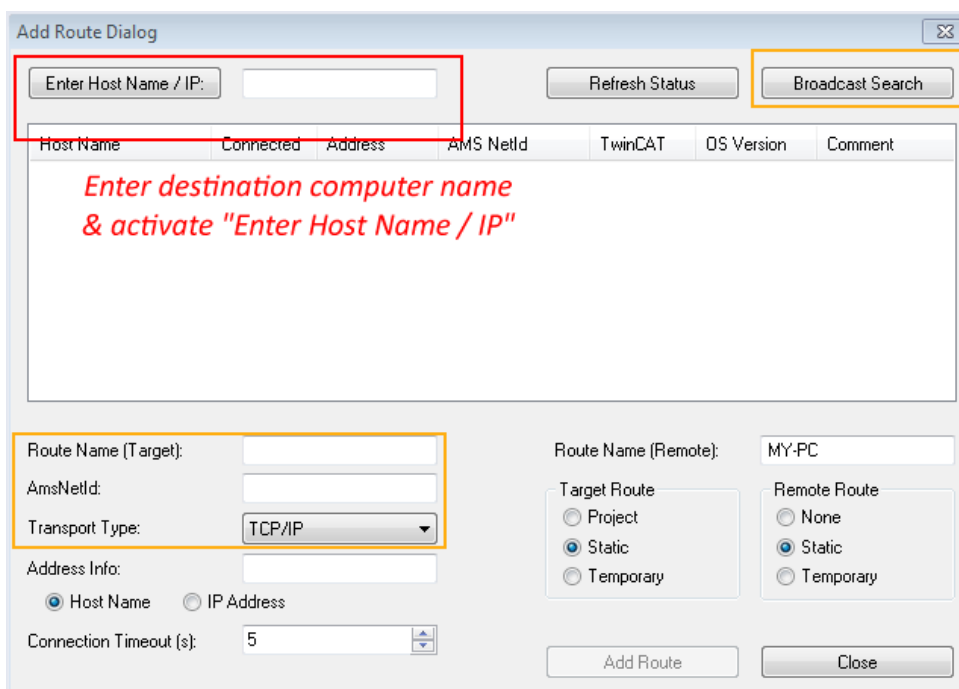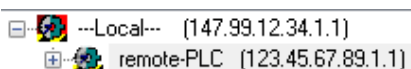- enter the known computer – IP or AmsNetID

Fig. 25: specify the PLC for access by the TwinCAT System Manager: selection of the target system

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

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

Adding devices

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

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

"Actions" → "Set/Reset TwinCAT to Config Mode…" (Shift + F4).
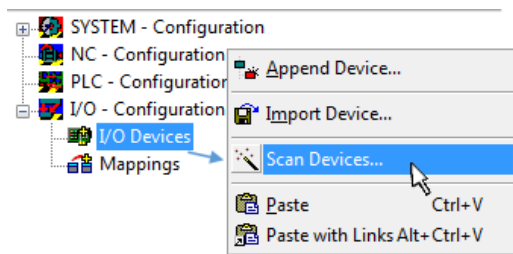
Fig. 26: Select "Scan Devices..."

Confirm the warning message, which follows, and select the "EtherCAT" devices in the dialog:
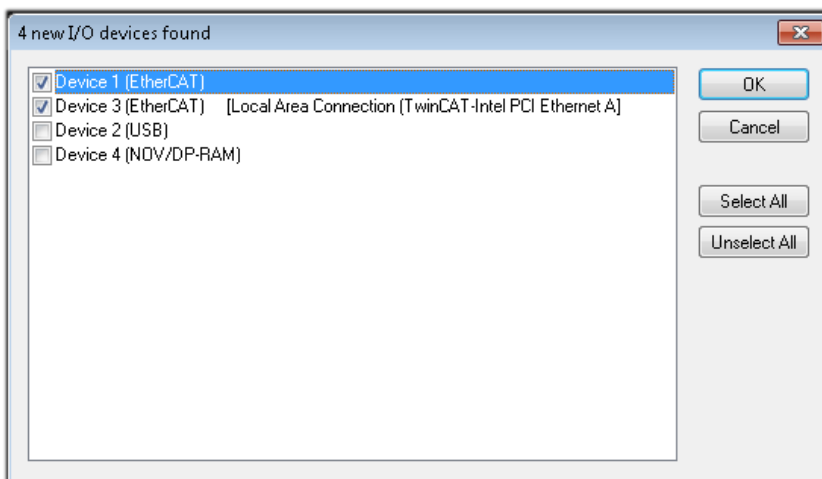
Fig. 27: Automatic detection of I/O devices: selection of the devices to be integrated

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

Based on the example configuration [▶ 40] described at the beginning of this section, the result is as follows:
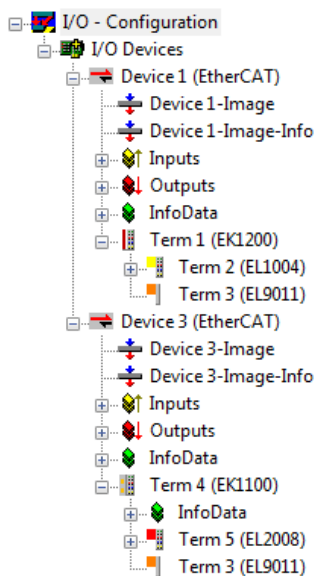
Fig. 28: Mapping of the configuration in the TwinCAT 2 System Manager

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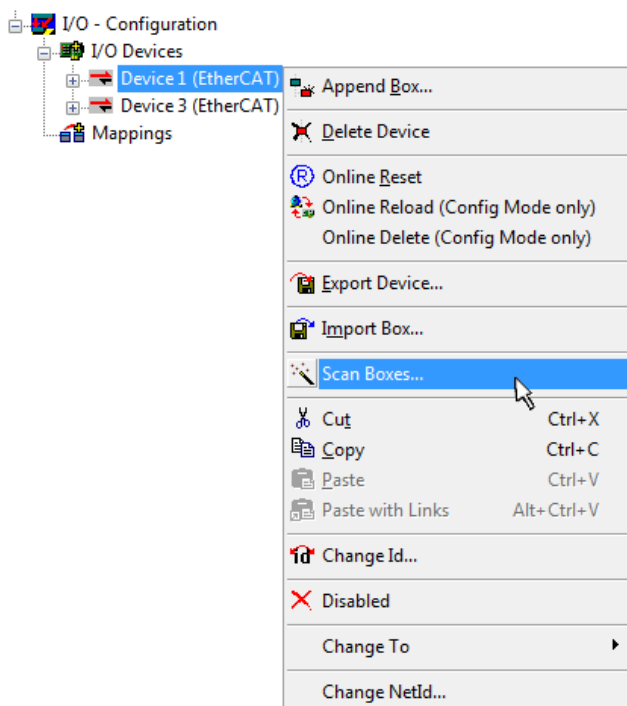

Fig. 29: Reading of individual terminals connected to a device

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

Programming and integrating the PLC

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

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

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

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

After starting TwinCAT PLC Control, the following user interface is shown for an initial project:
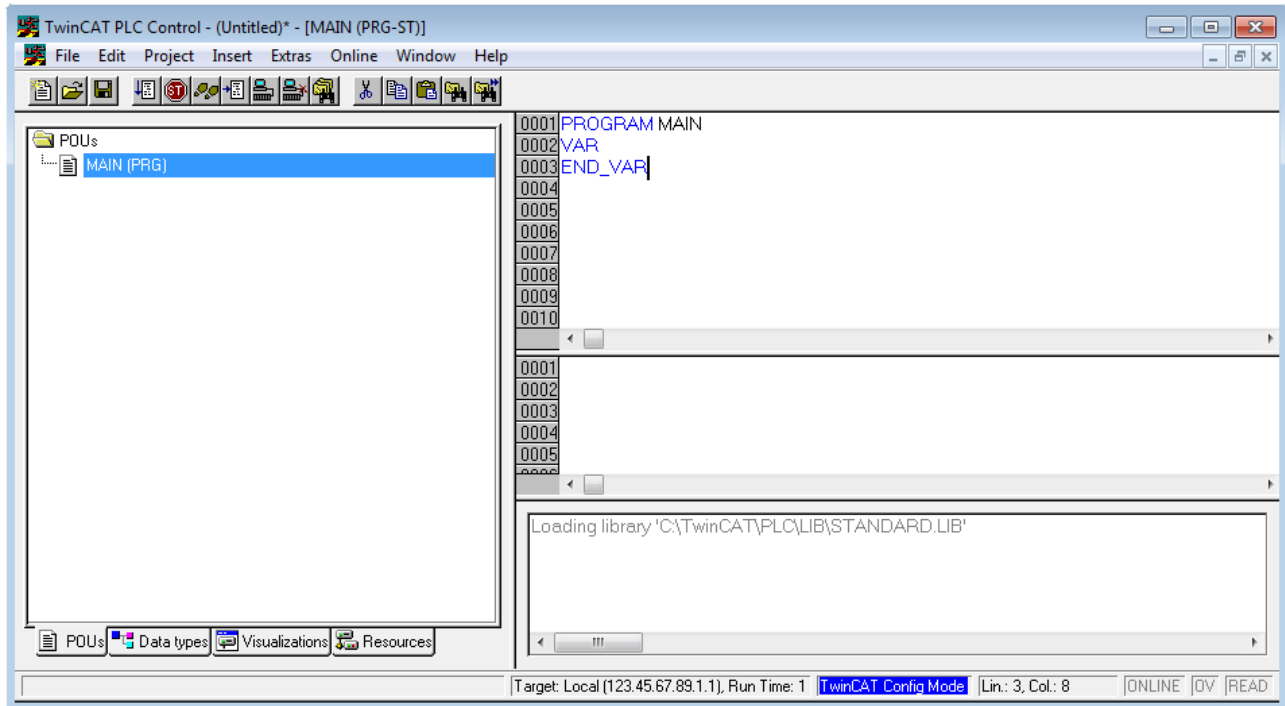
Fig. 30: TwinCAT PLC Control after startup

Example variables and an example program have been created and stored under the name "PLC_example.pro":
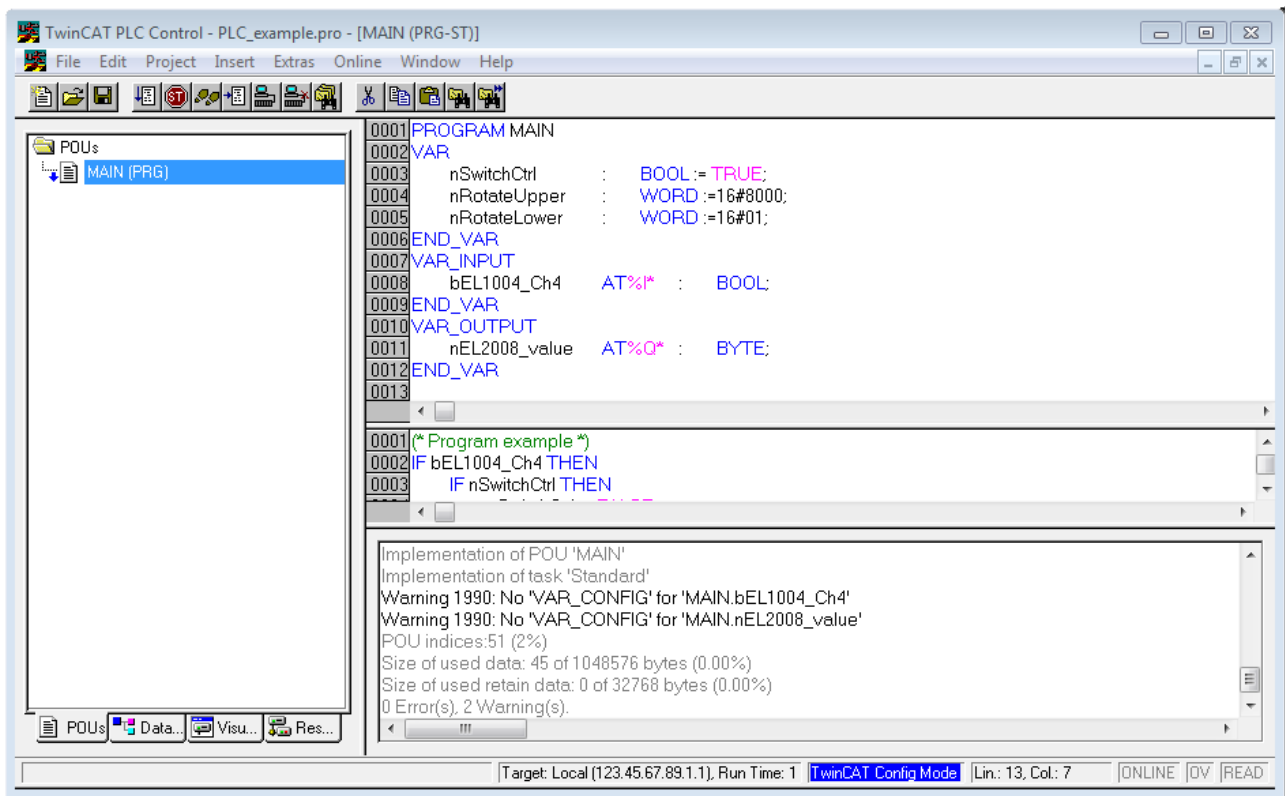
Fig. 31: Example program with variables after a compile process (without variable integration)

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

First, integrate the TwinCAT PLC Control project in the **System Manager**. This is performed via the context menu of the PLC configuration (right-click) and selecting "Append PLC Project…":
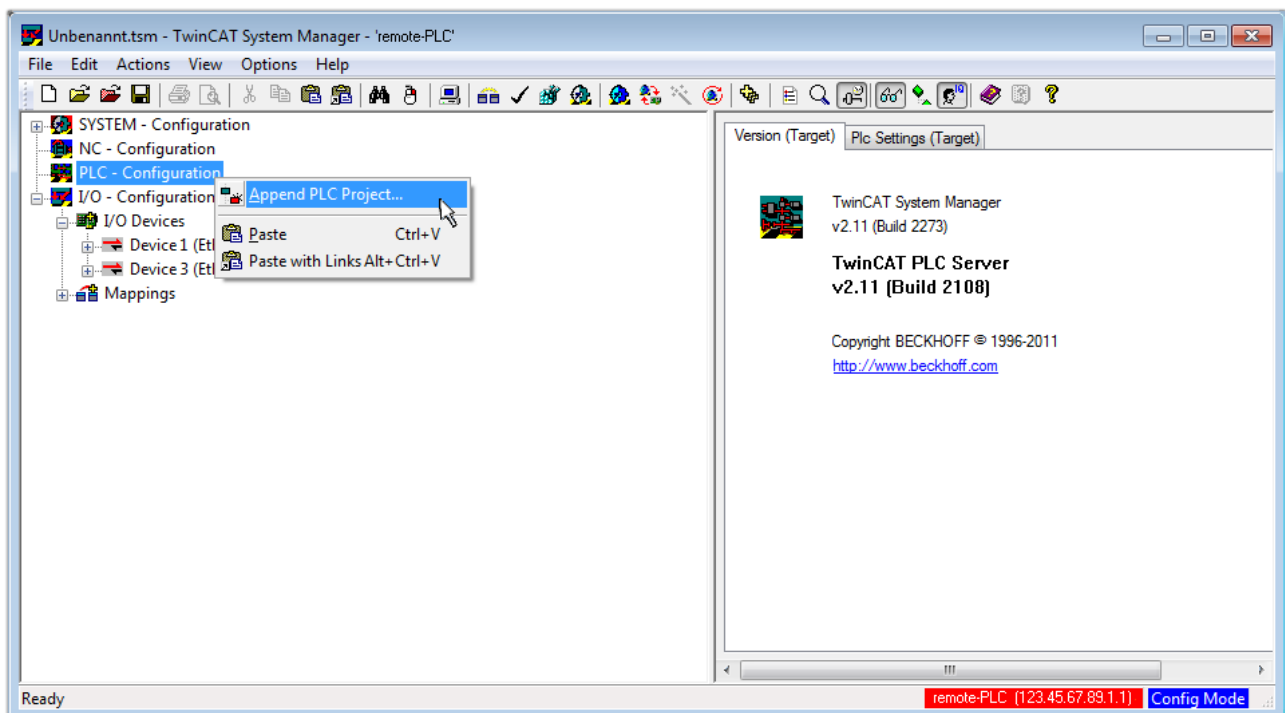
Fig. 32: Appending the TwinCAT PLC Control project

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:
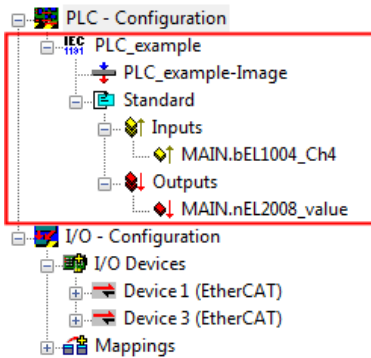
Fig. 33: PLC project integrated in the PLC configuration of the System Manager

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":
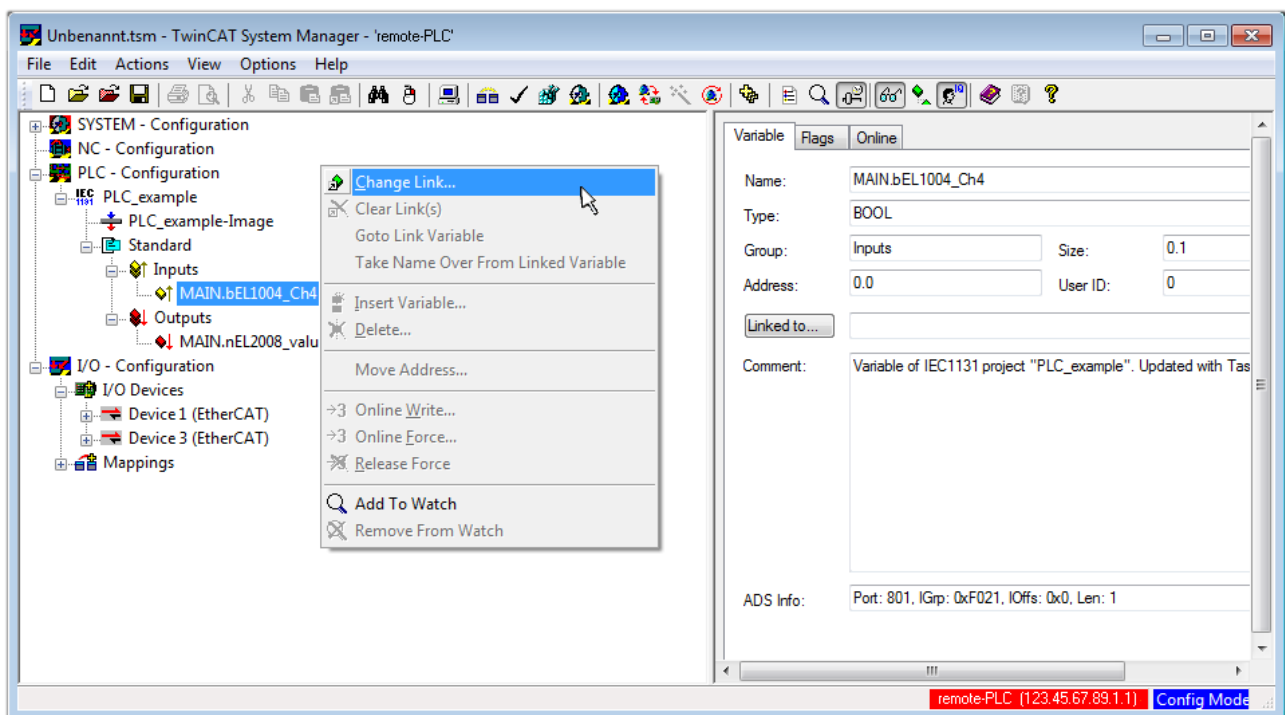
Fig. 34: Creating the links between PLC variables and process objects

In the window that opens, the process object for the "bEL1004_Ch4" BOOL-type variable can be selected from the PLC configuration tree:
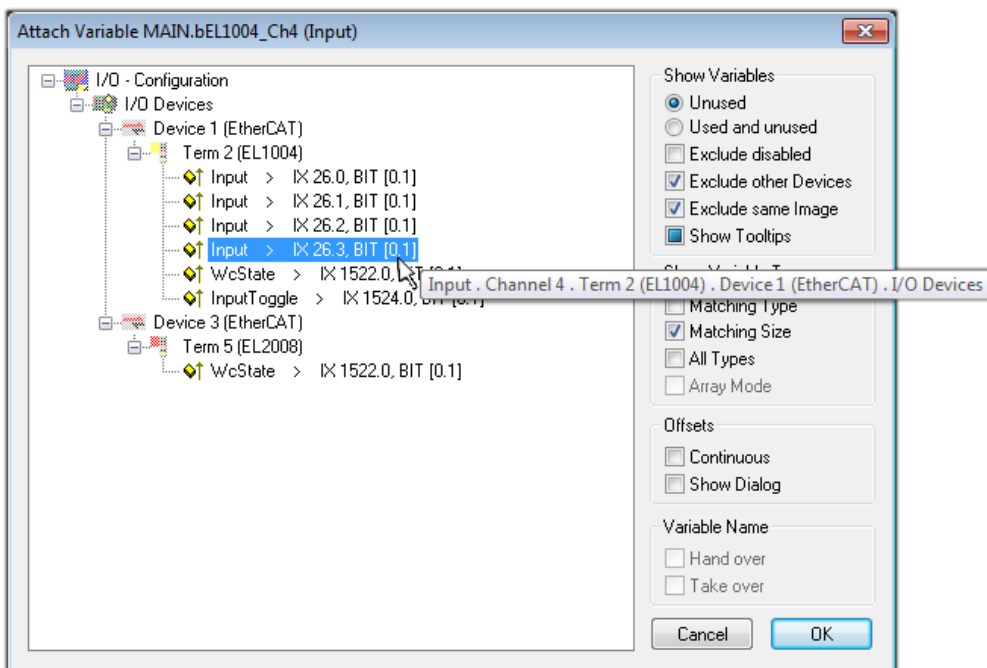
Fig. 35: Selecting BOOL-type PDO

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

Fig. 36: Selecting several PDOs simultaneously: activate "Continuous" and "All types"

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

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

Fig. 37: Application of a "Goto Link Variable", using "MAIN.bEL1004_Ch4" as an example

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

assignment", or via .

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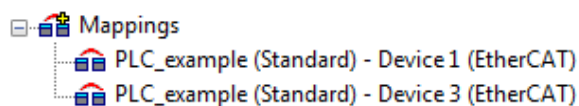
The process of creating links can also be performed in the opposite direction, i.e. starting with individual PDOs to a variable. However, in this example, it would not be possible to select all output bits for the EL2008, since the terminal only makes individual digital outputs available. If a terminal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

Activation of the configuration

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

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

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

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

Starting the controller

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

Online	
Login	F11
Logout	F12
Download	
Run	F5
Stop	Shift+F8
Reset	
Reset All	
Toggle Breakpoint	F9
Breakpoint Dialog	
Step over	F10
Step in	F8
Single Cycle	Ctrl+F5
Write Values	Ctrl+F7
Force Values	F7
Release Force	Shift+F7
Write/Force-Dialog	Ctrl+Shift+F7
Show Call Stack...	
Display Flow Control	Ctrl+F11
Simulation Mode	
Communication Parameters...	
Sourcecode download	
Choose Run-Time System...	
Create Bootproject	
Create Bootproject (offline)	
Delete Bootproject	

Choose Run-Time System

---Local---   (149.35.17.99.1.1)
   <Default>   (255.255.255.255.255.255)
   remote-PLC (123.45.67.89.1.1)
      Laufzeitsystem 1 (Port 801)

OK
Cancel
Version Info...

Fig. 38: Choose target system (remote)

In this example, "Runtime system 1 (port 801)" is selected and confirmed. Link the PLC with the real-time system via the menu option "Online" → "Login", the F11 key or by clicking on the symbol [illegible icon]. The control program can then be loaded for execution. This results in the message "No program on the controller! Should the new program be loaded?", which should be confirmed with "Yes". The runtime environment is ready for the program start:

BECKHOFF
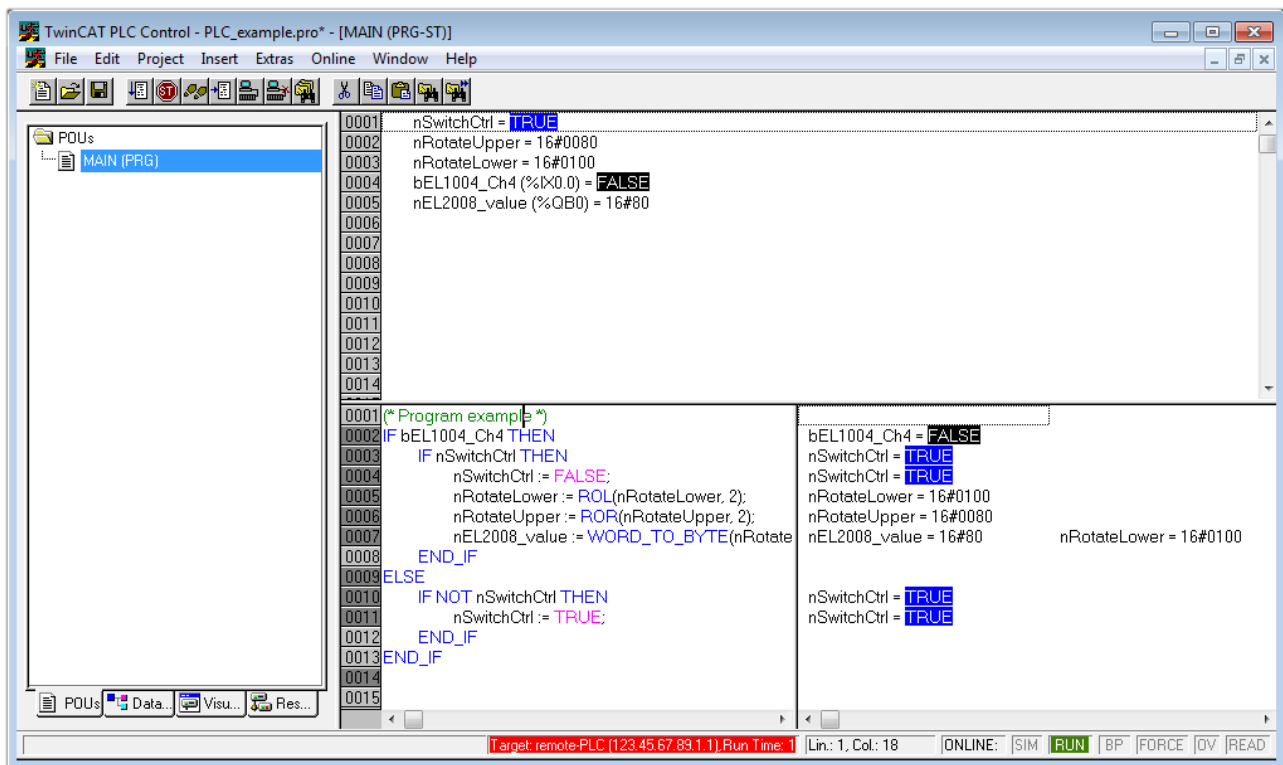
Fig. 39: PLC Control logged in, ready for program startup

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

5.1.2   TwinCAT 3

Startup

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

After successful installation of the TwinCAT system on the PC to be used for development, TwinCAT 3 (shell) displays the following user interface after startup:
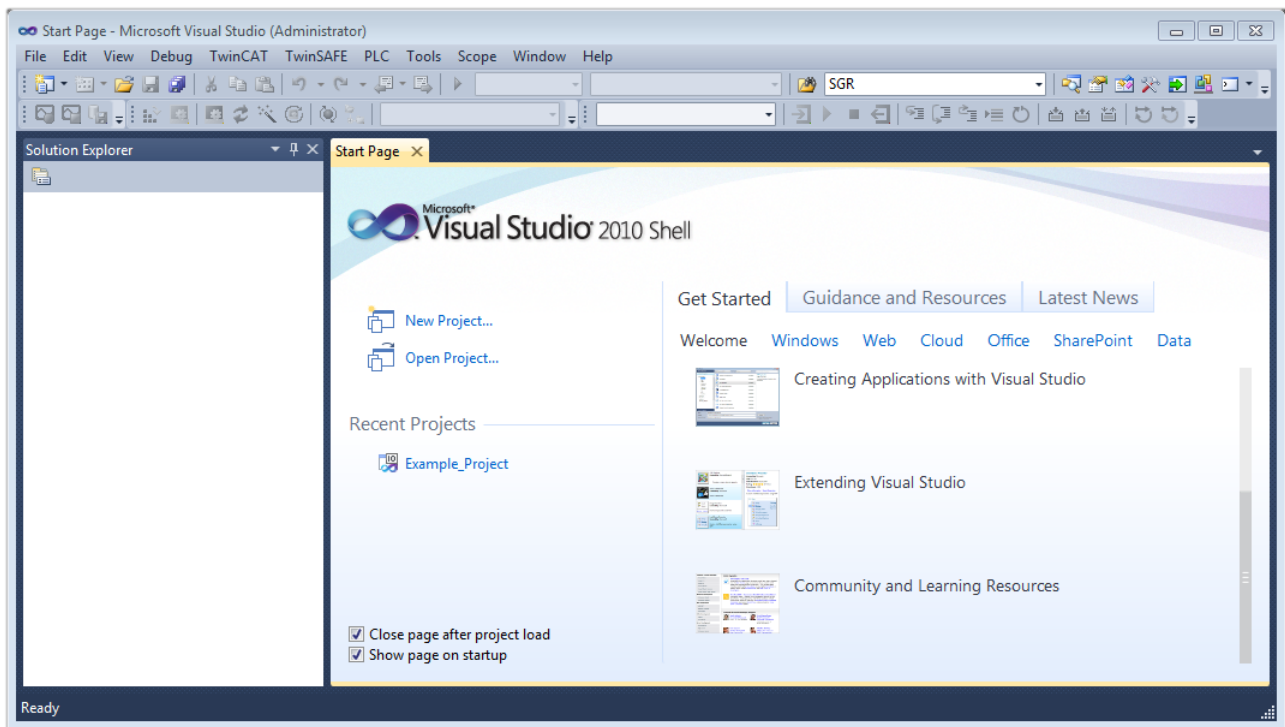
Fig. 40: Initial TwinCAT 3 user interface

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):
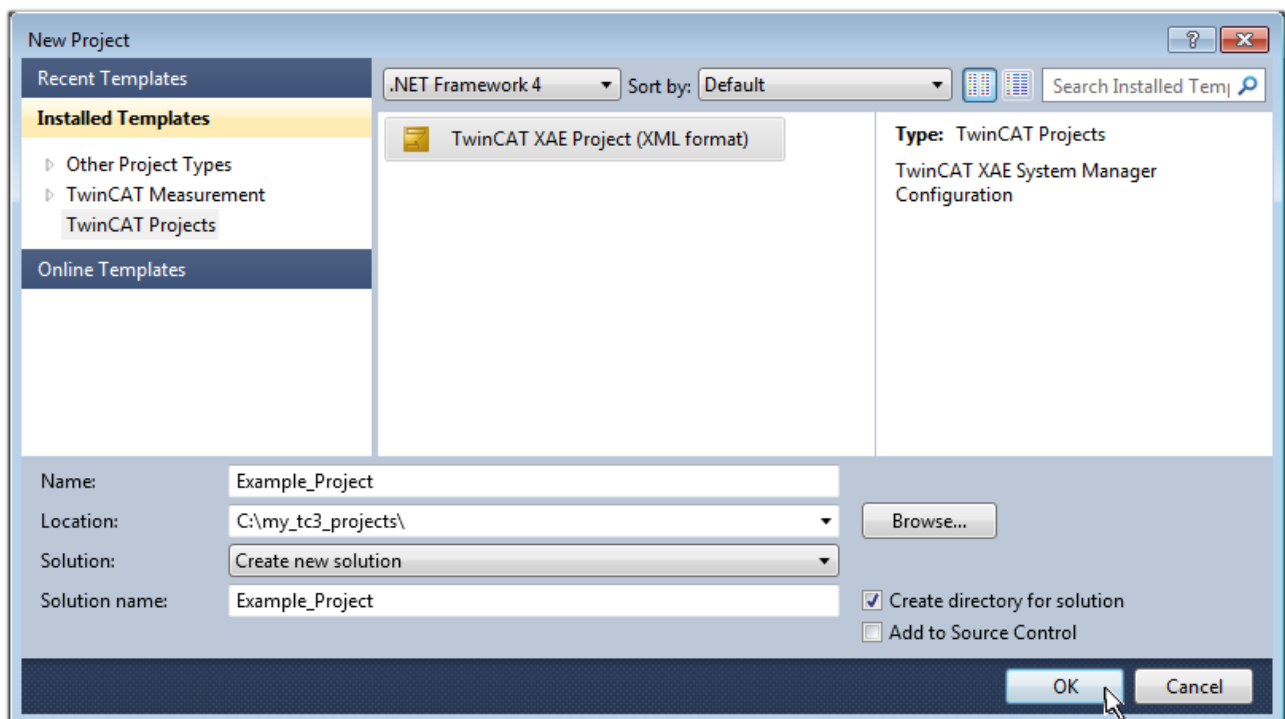
Fig. 41: Create new TwinCAT 3 project

The new project is then available in the project folder explorer:
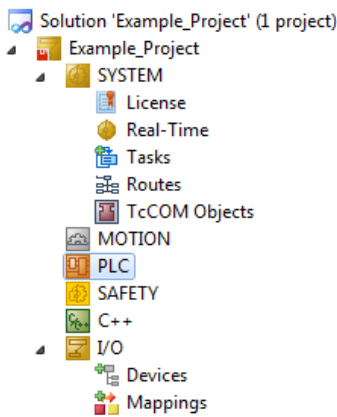
Fig. 42: New TwinCAT 3 project in the project folder explorer

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC (locally), TwinCAT can be used in local mode and the process can be continued with the next step, "Insert Device [▶ 55]".

If the intention is to address the TwinCAT runtime environment installed on a PLC remotely from another system used as a development environment, the target system must be made known first. Via the symbol in the menu bar:
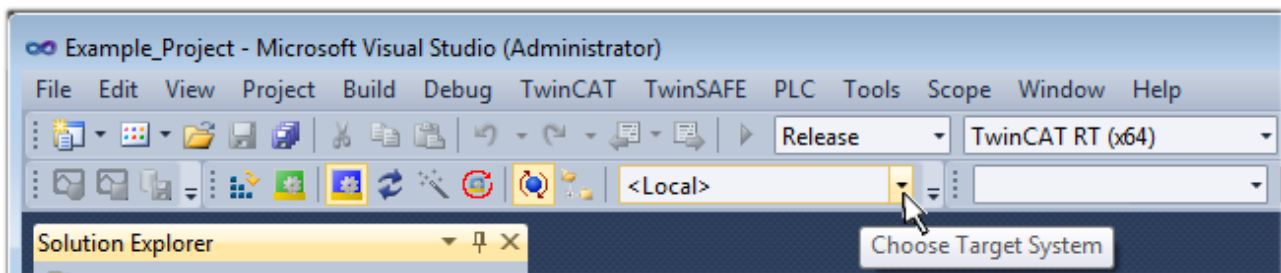
expand the pull-down menu:
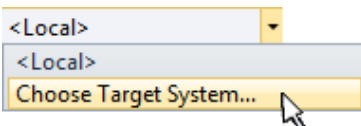
and open the following window:
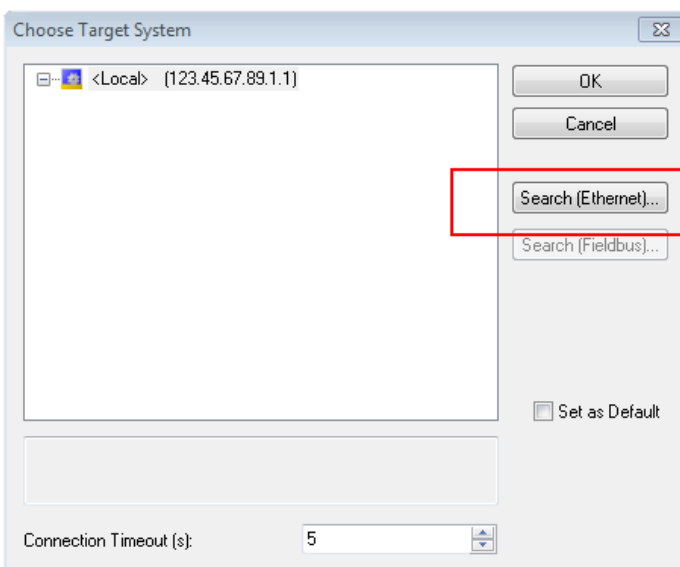
Fig. 43: Selection dialog: Choose the target system

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

- enter the known computer name after "Enter Host Name / IP:" (as shown in red)
- perform a "Broadcast Search" (if the exact computer name is not known)
- enter the known computer – IP or AmsNetID
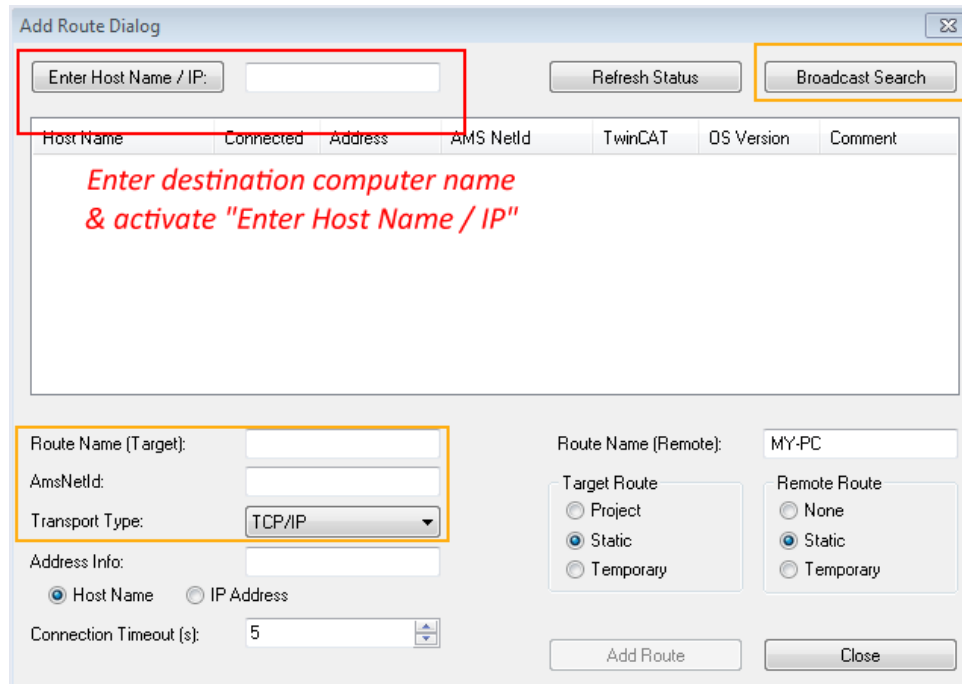
Fig. 44: specify the PLC for access by the TwinCAT System Manager: selection of the target system

Once the target system has been entered, it is available for selection as follows (the correct password may have to be entered beforehand):
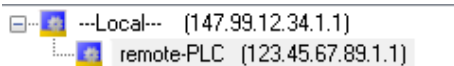
After confirmation with "OK" the target system can be accessed via the Visual Studio shell.

Adding devices

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

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

menu bar. The TwinCAT System Manager may first have to be set to "Config mode" via [image] or via the menu "TwinCAT" → "Restart TwinCAT (Config Mode)".
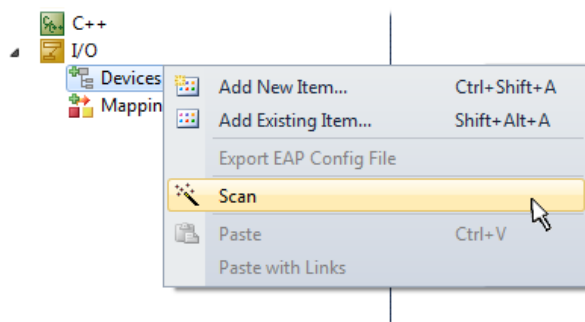
Fig. 45: Select "Scan"

Confirm the warning message, which follows, and select the "EtherCAT" devices in the dialog:
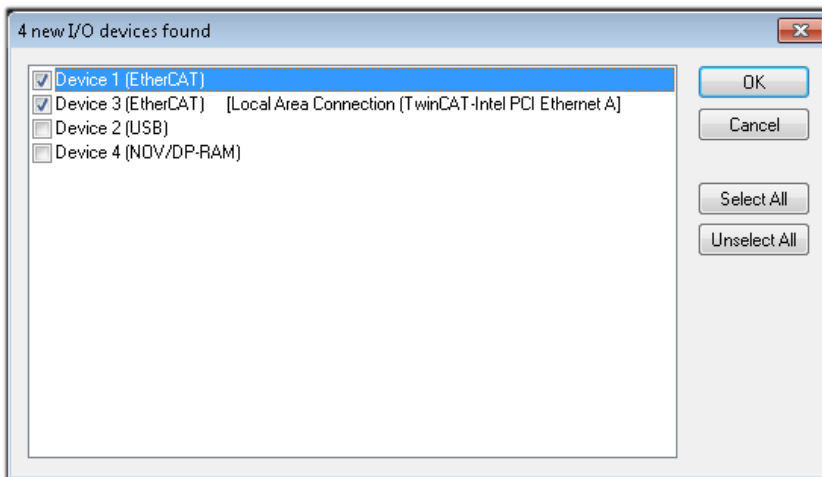
Fig. 46: Automatic detection of I/O devices: selection of the devices to be integrated

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

Based on the example configuration [▶ 40] described at the beginning of this section, the result is as follows:
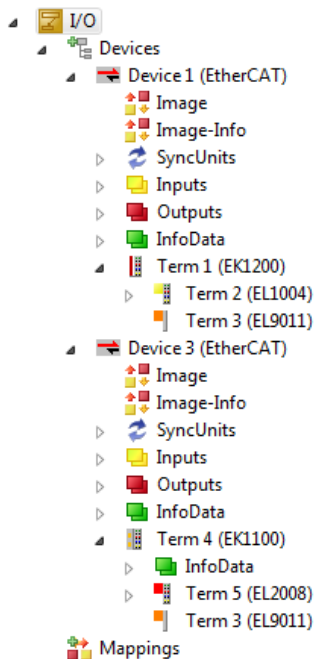
Fig. 47: Mapping of the configuration in VS shell of the TwinCAT 3 environment

The whole process consists of two stages, which can also be performed separately (first determine the devices, then determine the connected elements such as boxes, terminals, etc.). A scan (search function) can also be initiated by selecting "Device ..." from the context menu, which then only reads the elements below which are present in the configuration:
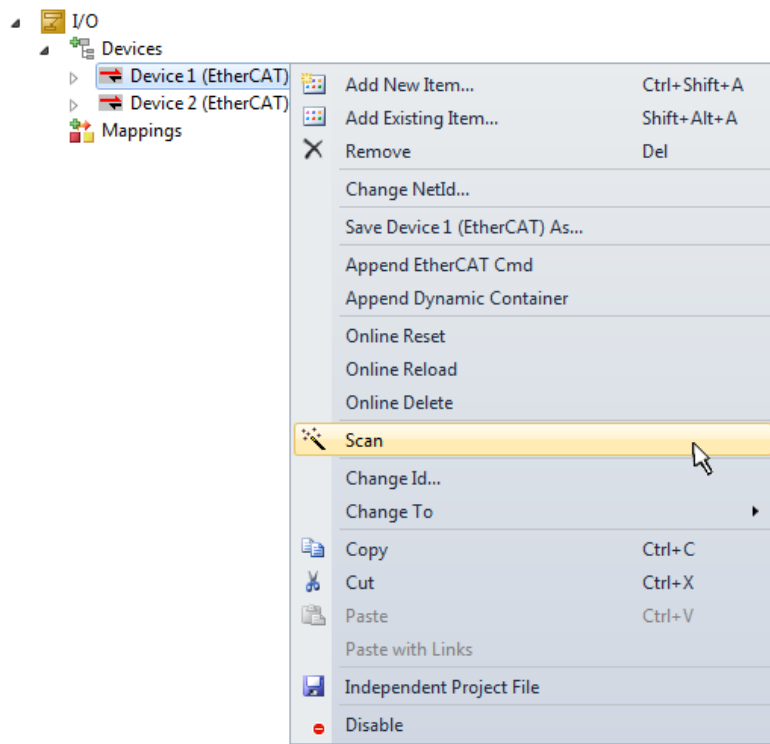
Fig. 48: Reading of individual terminals connected to a device

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

Programming the PLC

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

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

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

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

BECKHOFF
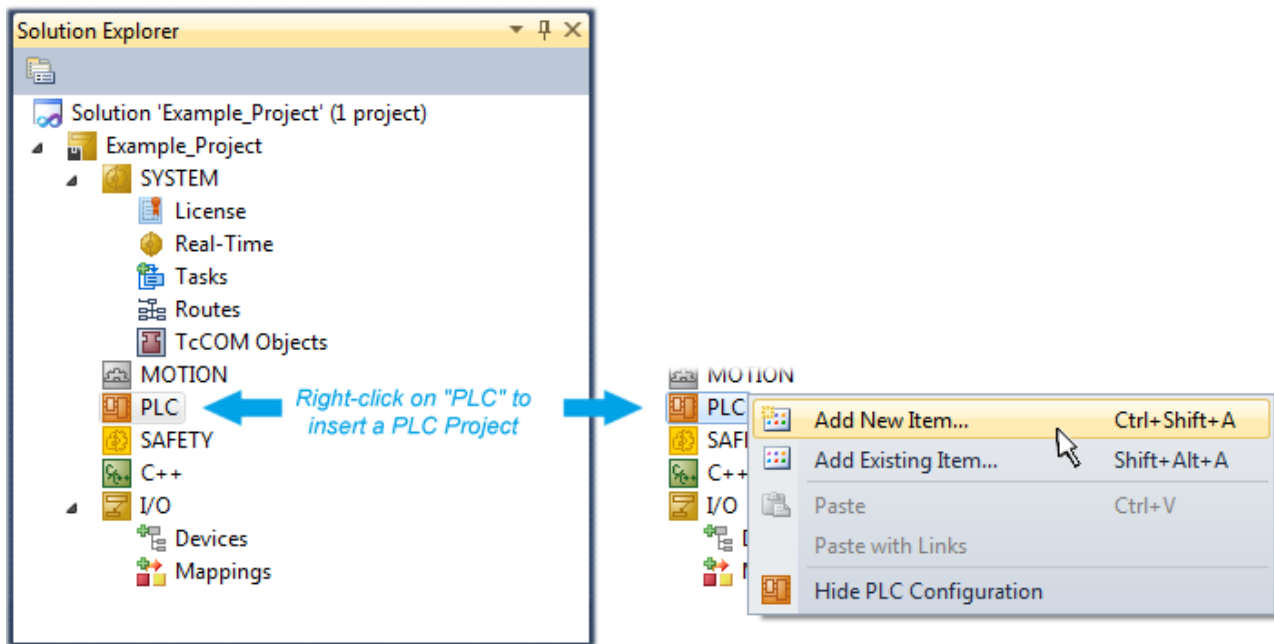
Fig. 49: Adding the programming environment in "PLC"

In the dialog that opens, select "Standard PLC project" and enter "PLC_example" as project name, for example, and select a corresponding directory:
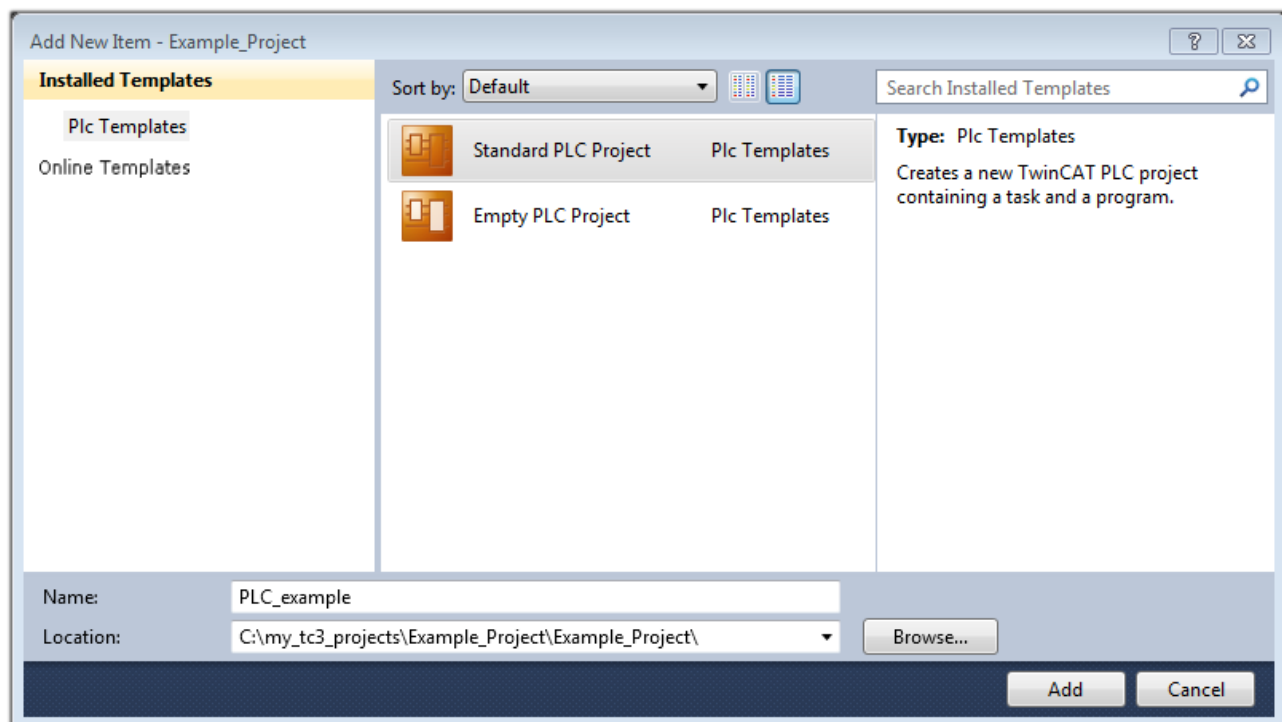
Fig. 50: Specifying the name and directory for the PLC programming environment

The "Main" program, which already exists due to selecting "Standard PLC project", can be opened by double-clicking on "PLC_example_project" in "POUs". The following user interface is shown for an initial project:
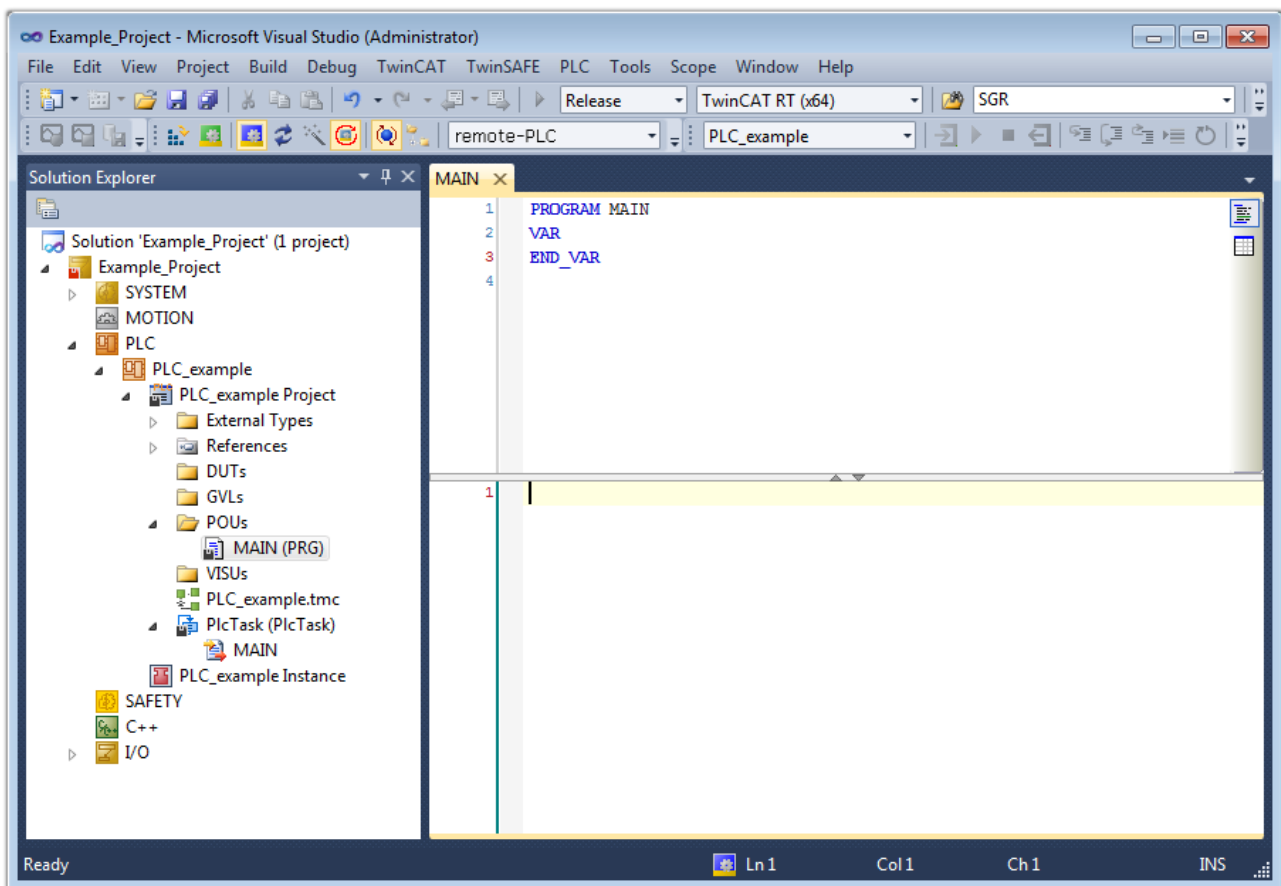
Fig. 51: Initial "Main" program for the standard PLC project

Now example variables and an example program have been created for the next stage of the process:
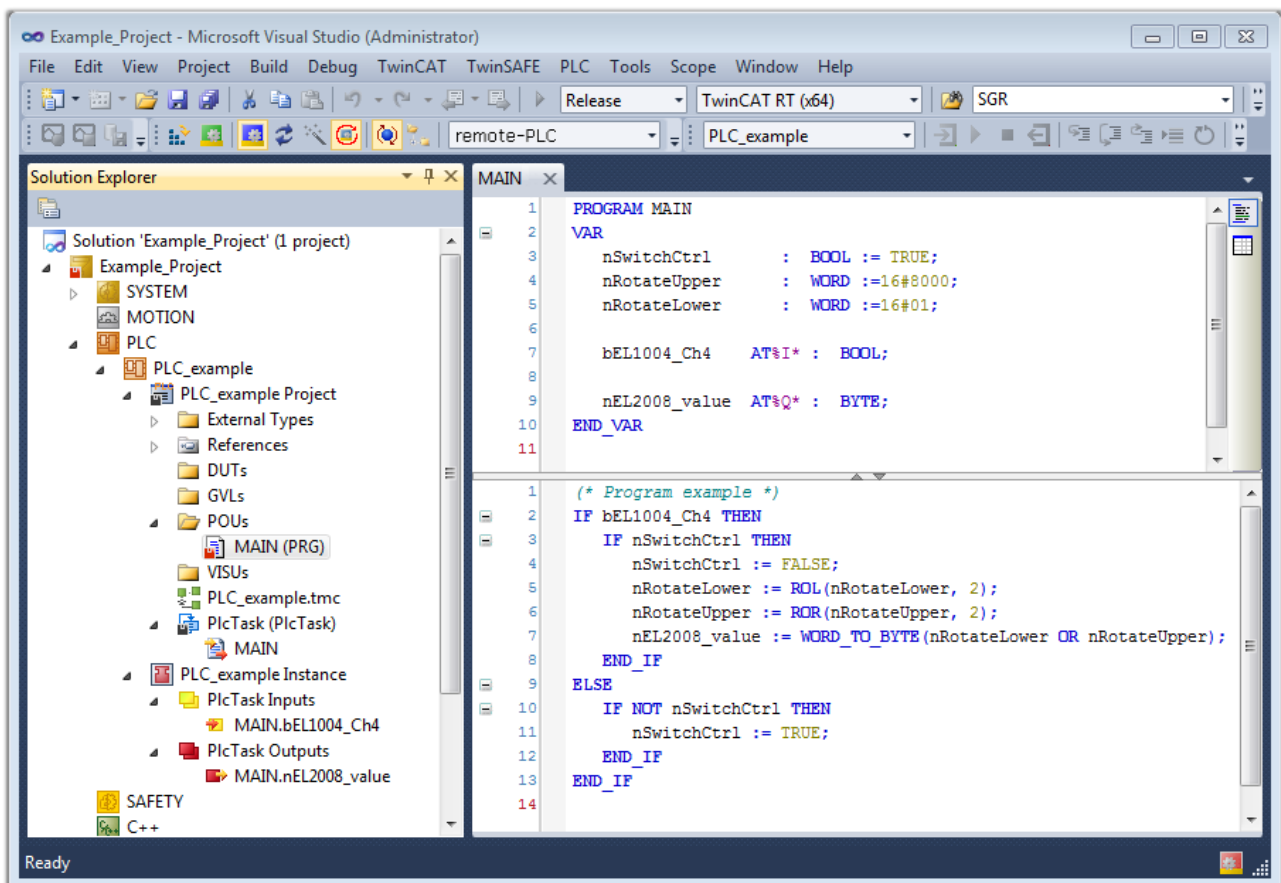
Fig. 52: Example program with variables after a compile process (without variable integration)

The control program is now created as a project folder, followed by the compile process:
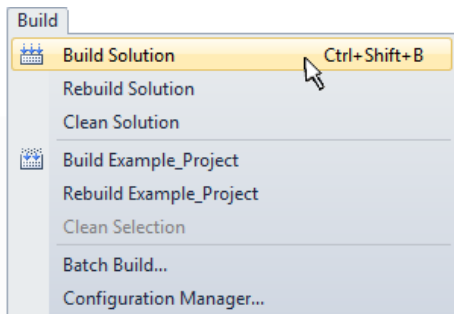
Fig. 53: Start program compilation

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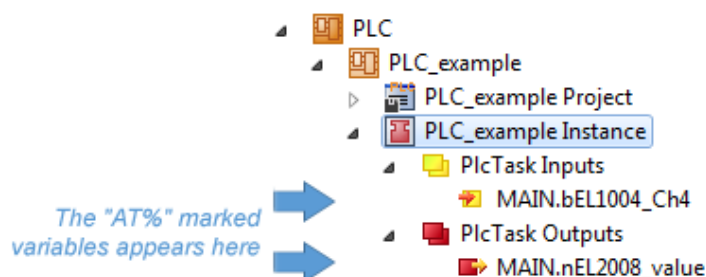
Assigning variables

Via the menu of an instance – variables in the "PLC" context, use the "Modify Link…" option to open a window to select a suitable process object (PDO) for linking:
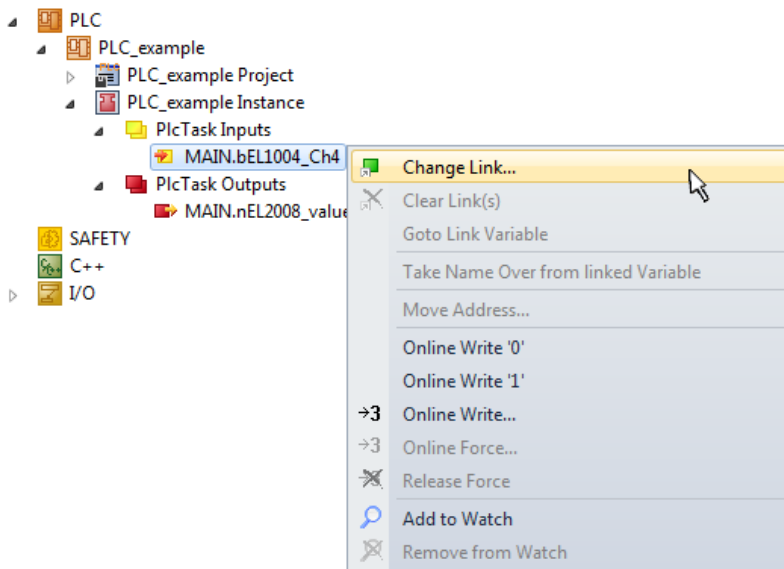
Fig. 54: Creating the links between PLC variables and process objects

In the window that opens, the process object for the "bEL1004_Ch4" BOOL-type variable can be selected from the PLC configuration tree:
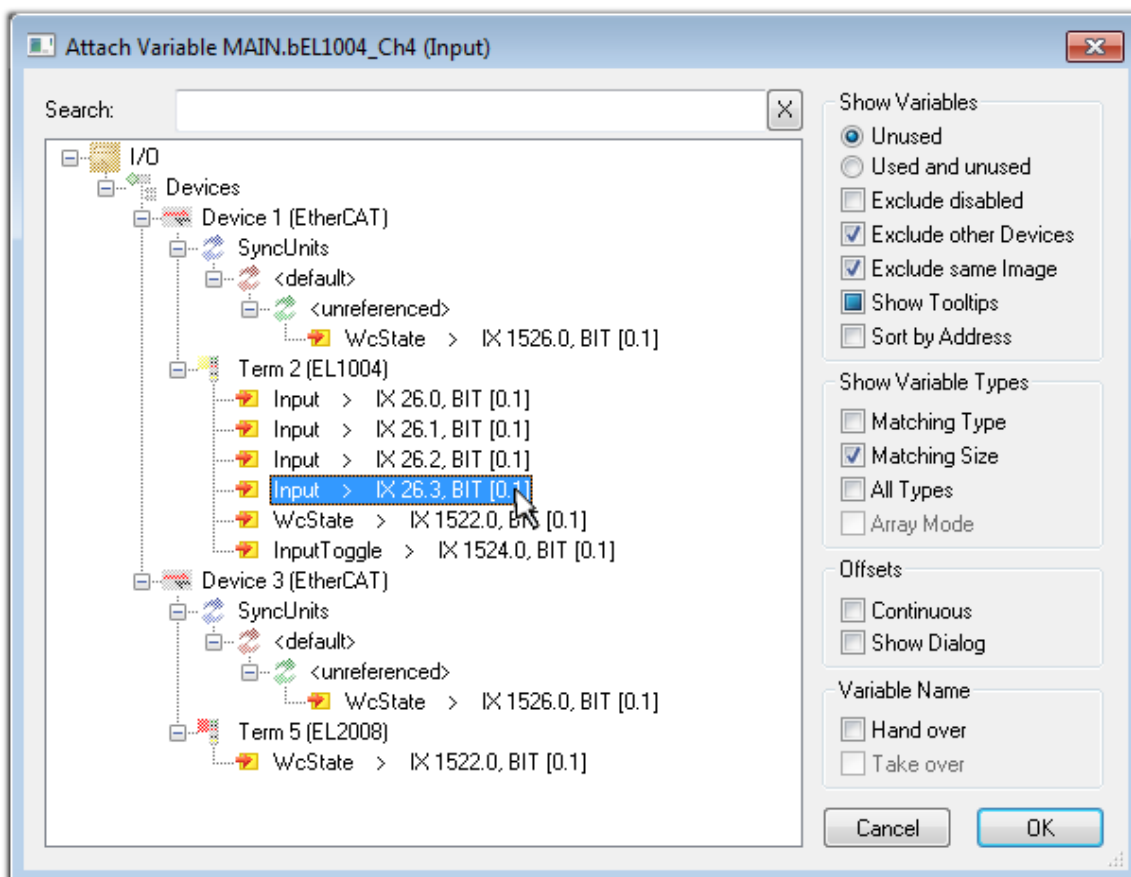
Fig. 55: Selecting BOOL-type PDO

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

Fig. 56: Selecting several PDOs simultaneously: activate "Continuous" and "All types"

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

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

Fig. 57: Application of a "Goto Link Variable", using "MAIN.bEL1004_Ch4" as an example

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

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

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

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

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

1. First, the required process data must be selected in the "Process data" tab in TwinCAT.
2. After that, the PLC data type must be generated in the "PLC" tab via the check box.
3. The data type in the "Data Type" field can then be copied using the "Copy" button.
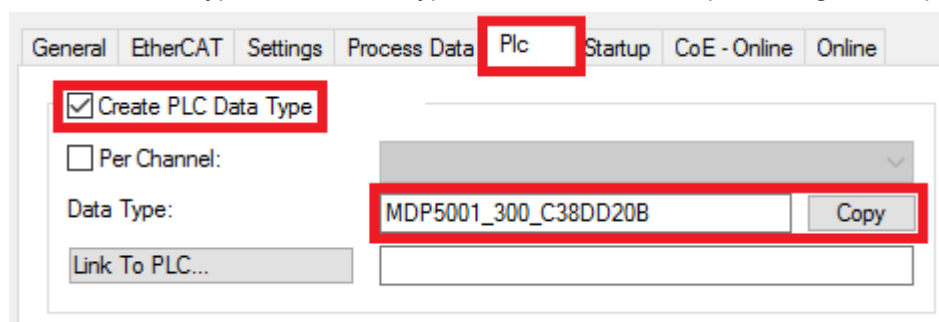
Fig. 58: Creating a PLC data type

4. An instance of the data structure of the copied data type must then be created in the PLC.
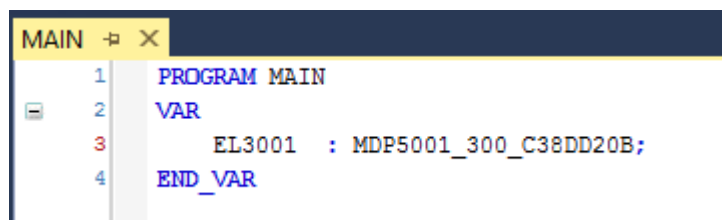
Fig. 59: Instance_of_struct

5. Then the project folder must be created. This can be done either via the key combination "CTRL + Shift + B" or via the "Build" tab in TwinCAT.
6. The structure in the "PLC" tab of the terminal must then be linked to the created instance.
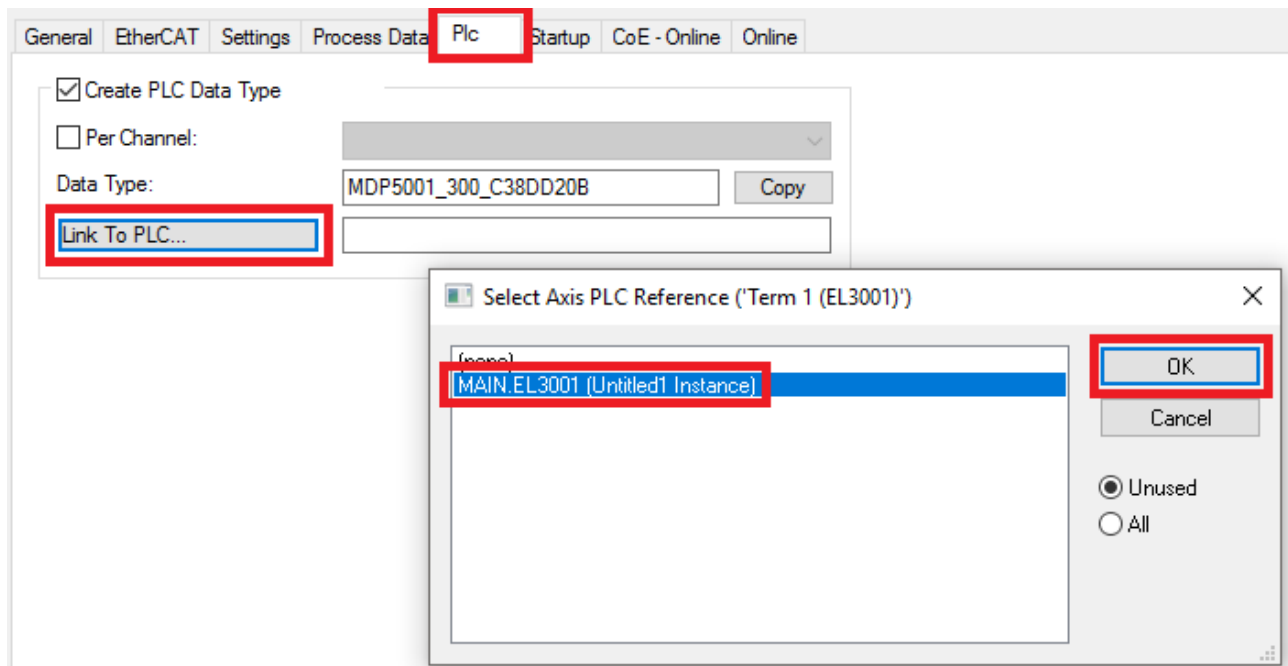
Fig. 60: Linking the structure

7. In the PLC, the process data can then be read or written via the structure in the program code.
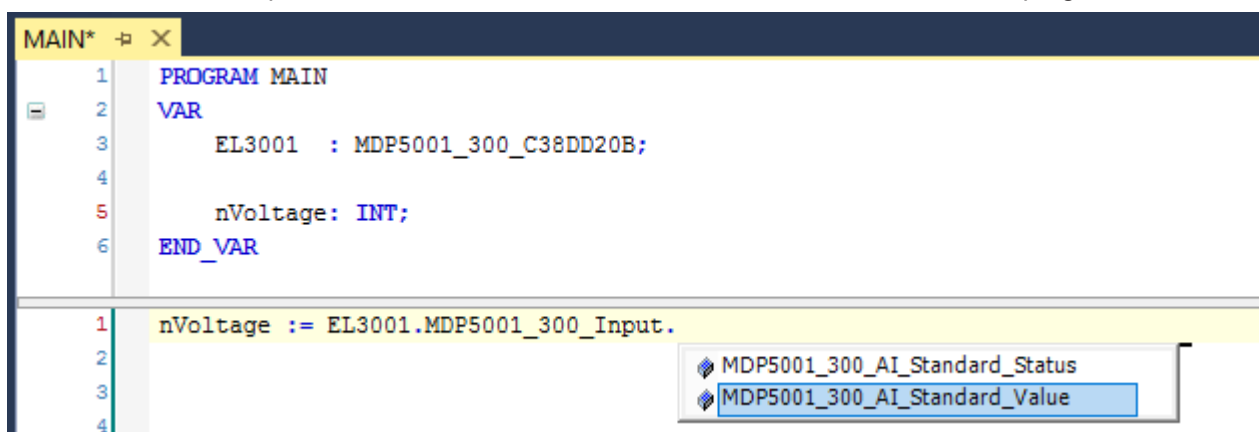
Fig. 61: Reading a variable from the structure of the process data

Activation of the configuration

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

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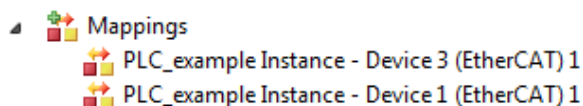
A few seconds later, the corresponding status of the Run mode is displayed in the form of a rotating symbol

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

Starting the controller

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

the program to be started by clicking on symbol ▶ , the "F5" key or via "PLC" in the menu, by selecting "Start". The started programming environment shows the runtime values of individual variables:
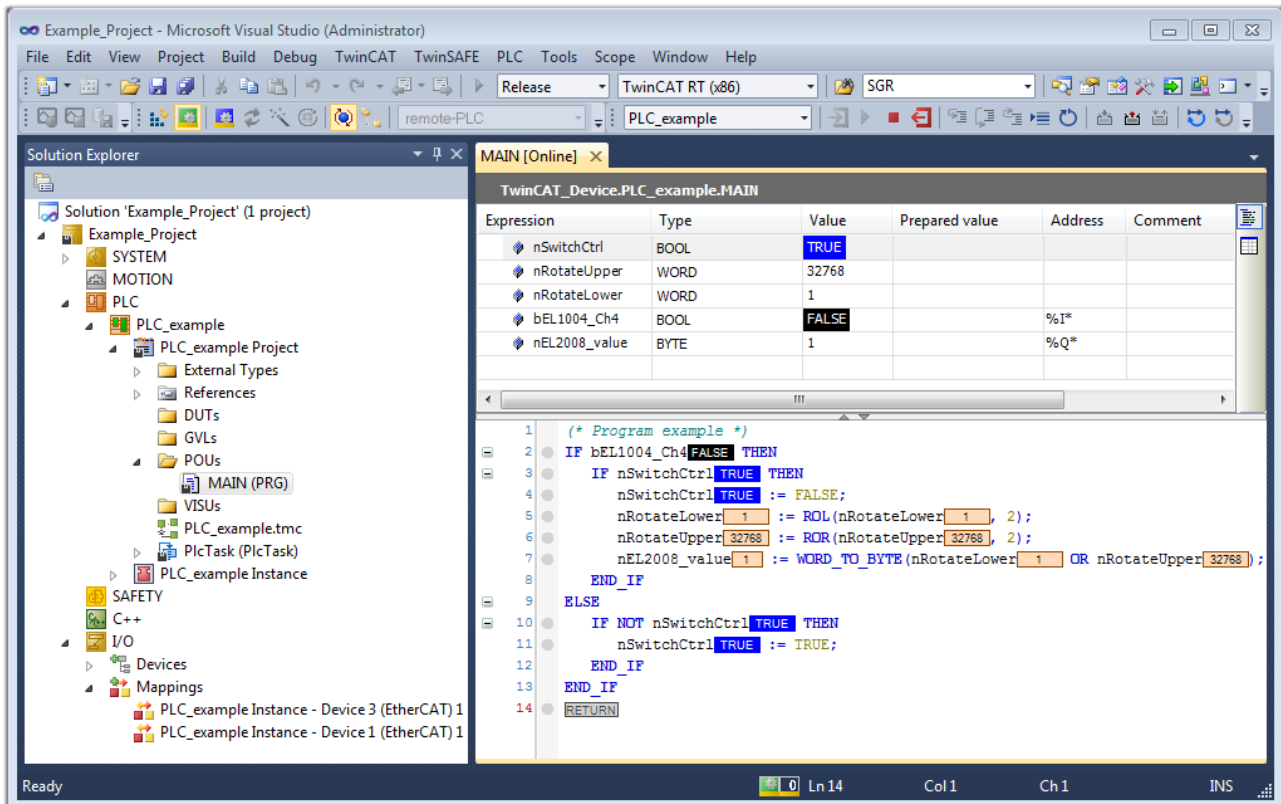
Fig. 62: TwinCAT 3 development environment (VS shell): logged-in, after program startup

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

5.2   **TwinCAT Development Environment**

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

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

Details:

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

- Datalink on NT - Programs by open Microsoft Standards (OLE, OCX, ActiveX, DCOM+, etc.)
- Integration of IEC 61131-3-Software-SPS, Software- NC and Software-CNC within Windows NT/ 2000/XP/Vista, Windows 7, NT/XP Embedded, CE
- Interconnection to all common fieldbusses
- More...

Additional features:

- **TwinCAT 3** (eXtended Automation)**:**
  - Visual Studio® integration
  - Choice of the programming language
  - Supports object orientated extension of IEC 61131-3
  - Usage of C/C++ as programming language for real time applications
  - Connection to MATLAB®/Simulink®
  - Open interface for expandability
  - Flexible run-time environment
  - Active support of multi-core- and 64 bit operating system
  - Automatic code generation and project creation with the TwinCAT Automation Interface
  - More...

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

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

5.2.1 Installation of the TwinCAT real-time driver

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

This can be done in several ways.

A: Via the TwinCAT Adapter dialog

In the System Manager call up the TwinCAT overview of the local network interfaces via Options → Show Real Time Ethernet Compatible Devices.
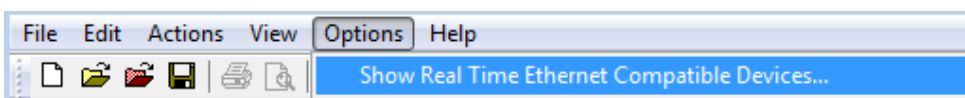
Fig. 63: System Manager "Options" (TwinCAT 2)

This have to be called up by the menu "TwinCAT" within the TwinCAT 3 environment:
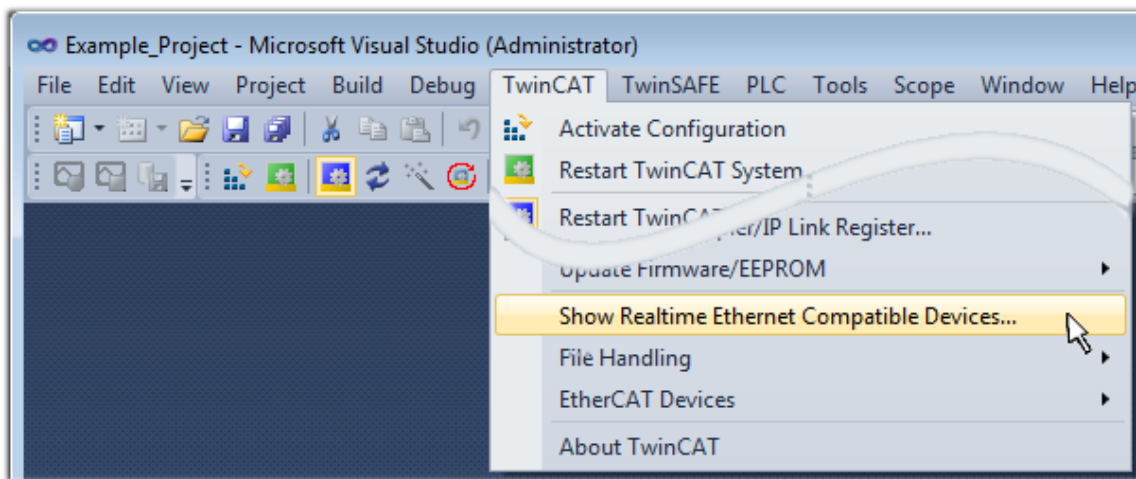
Fig. 64: Call up under VS Shell (TwinCAT 3)

B: Via TcRteInstall.exe in the TwinCAT directory
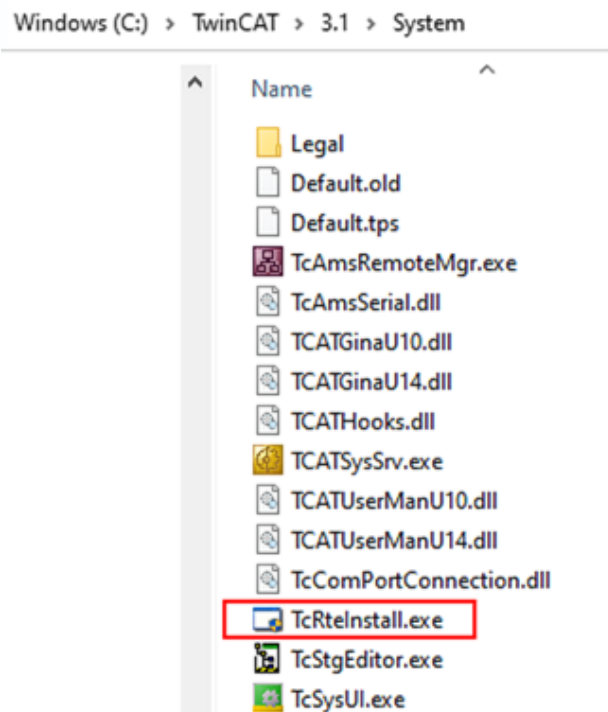
Fig. 65: TcRteInstall in the TwinCAT directory

In both cases, the following dialog appears:
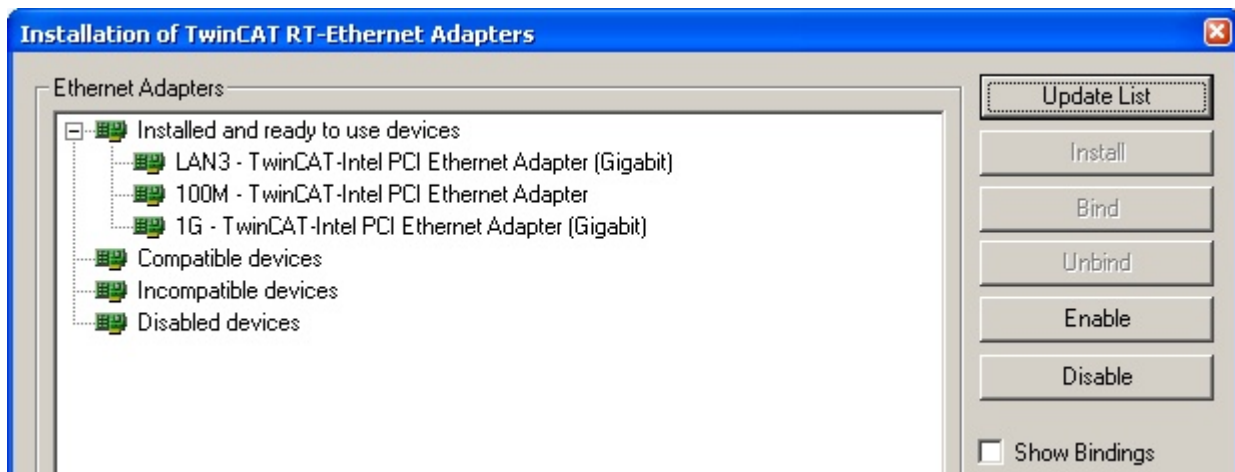
Fig. 66: Overview of network interfaces

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" [▶ 77] in order to view the compatible ethernet ports via its EtherCAT properties (tab "Adapter", button "Compatible Devices…"):

Fig. 67: 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":
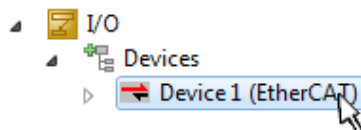
After the installation the driver appears activated in the Windows overview for the network interface (Windows Start → System Properties → Network)
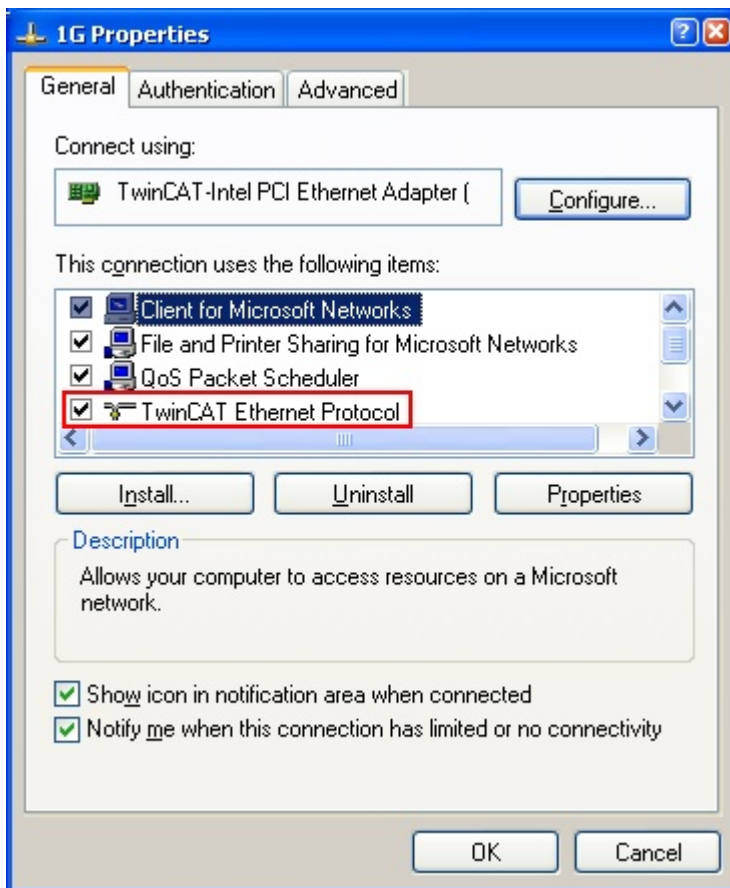
Fig. 68: Windows properties of the network interface

A correct setting of the driver could be:
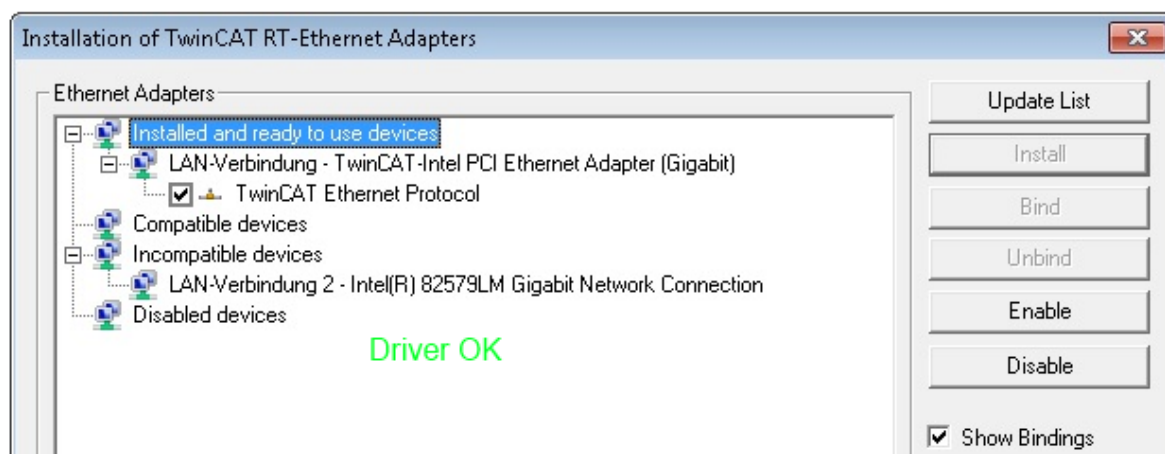
Fig. 69: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

BECKHOFF
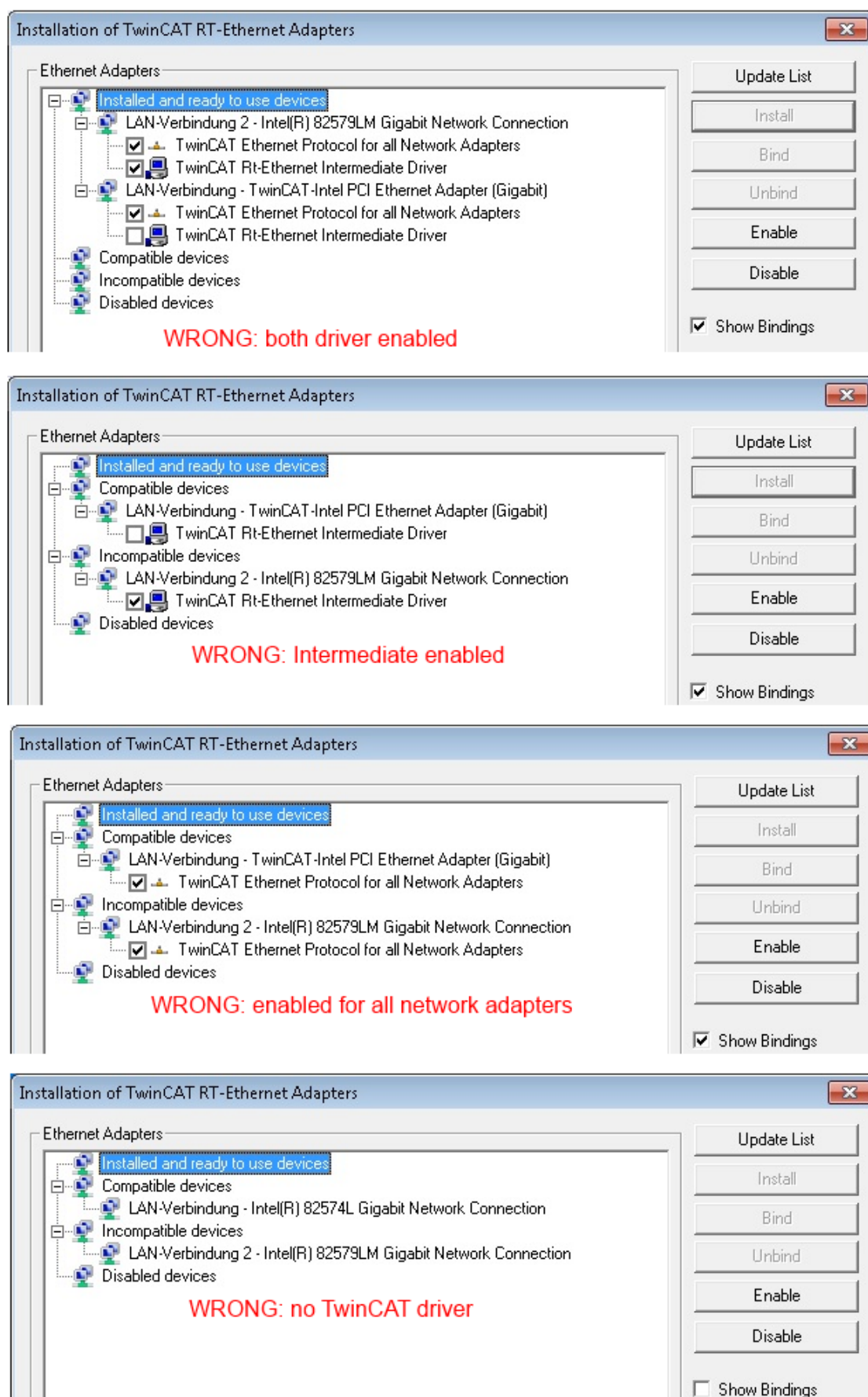
Fig. 70: Incorrect driver settings for the Ethernet port

IP address of the port used

● **IP address/DHCP**

ℹ️ 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.
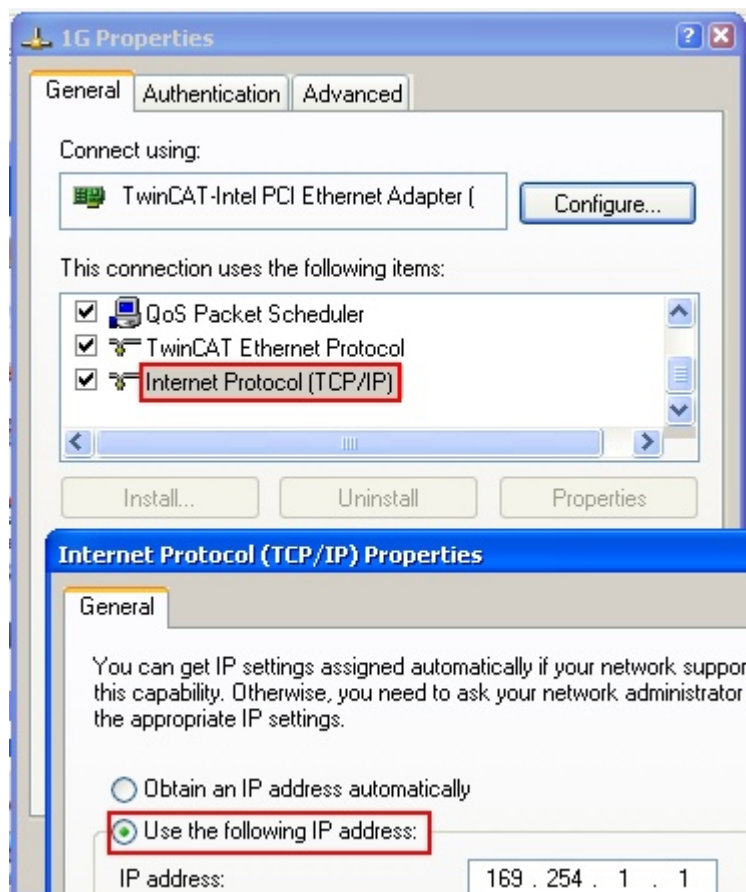
Fig. 71: TCP/IP setting for the Ethernet port

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 [▶ 76] is available for this purpose.

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

Device differentiation

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

- family key "EL"
- name "2521"
- type "0025"
- and revision "1018"

Fig. 72: Identifier structure

The order identifier consisting of name + type (here: EL2521-0025) describes the device function. The revision indicates the technical progress and is managed by Beckhoff. In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation. Each revision has its own ESI description. See further notes [▶ 9].

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.
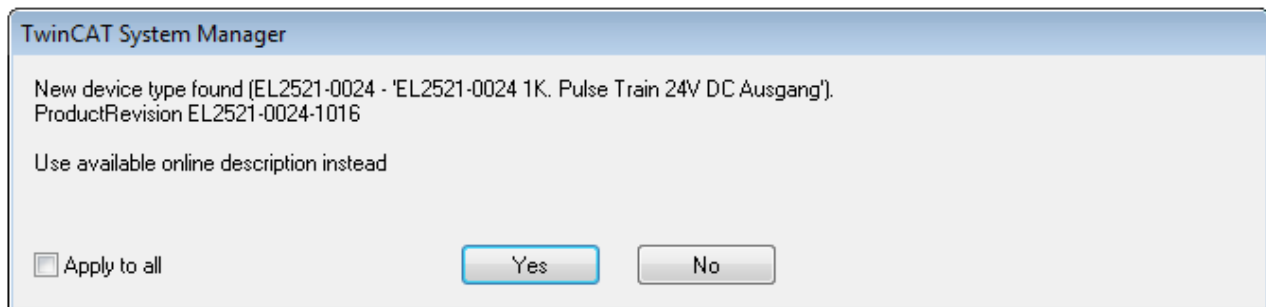
Fig. 73: OnlineDescription information window (TwinCAT 2)

In TwinCAT 3 a similar window appears, which also offers the Web update:
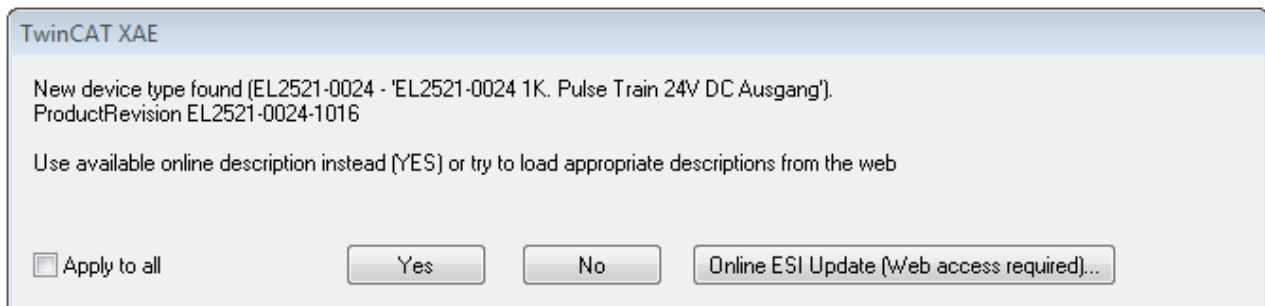
Fig. 74: Information window OnlineDescription (TwinCAT 3)

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

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

Refer in particular to the chapter "General notes on the use of Beckhoff EtherCAT IO components" and for manual configuration to the chapter "Offline configuration creation [▶ 77]".

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. 75: File OnlineDescription.xml created by the System Manager

Is a slave desired to be added manually to the configuration at a later stage, online created slaves are indicated by a prepended symbol ">" in the selection list (see Figure *Indication of an online recorded ESI of EL2521 as an example*).
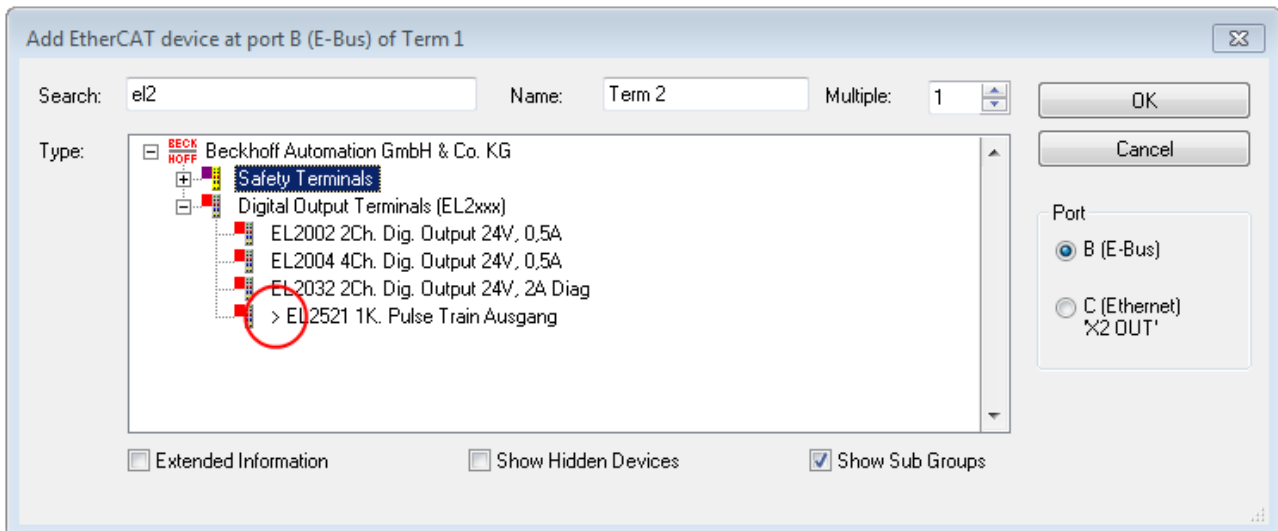
Fig. 76: Indication of an online recorded ESI of EL2521 as an example

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

i   **OnlineDescription for TwinCAT 3.x**

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

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

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

Faulty ESI file

If an ESI file is faulty and the System Manager is unable to read it, the System Manager brings up an information window.
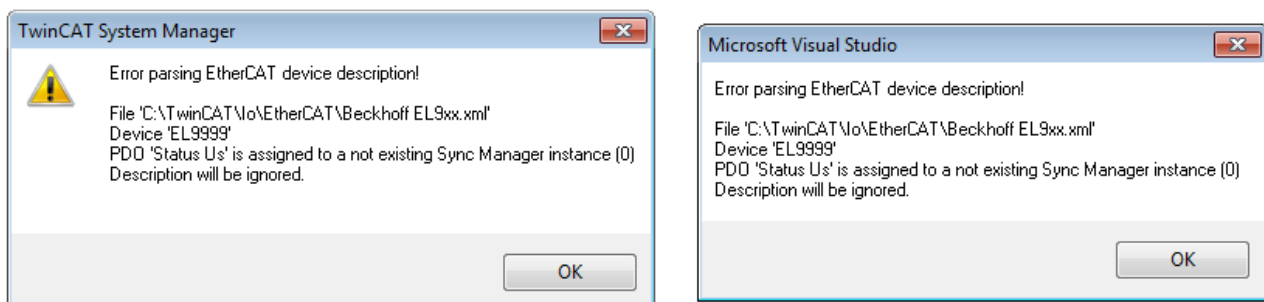
Fig. 77: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

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:
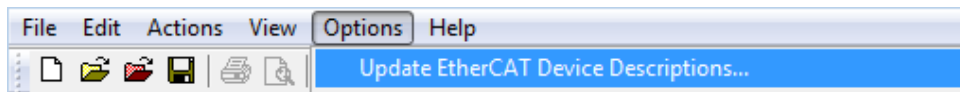
Fig. 78: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:
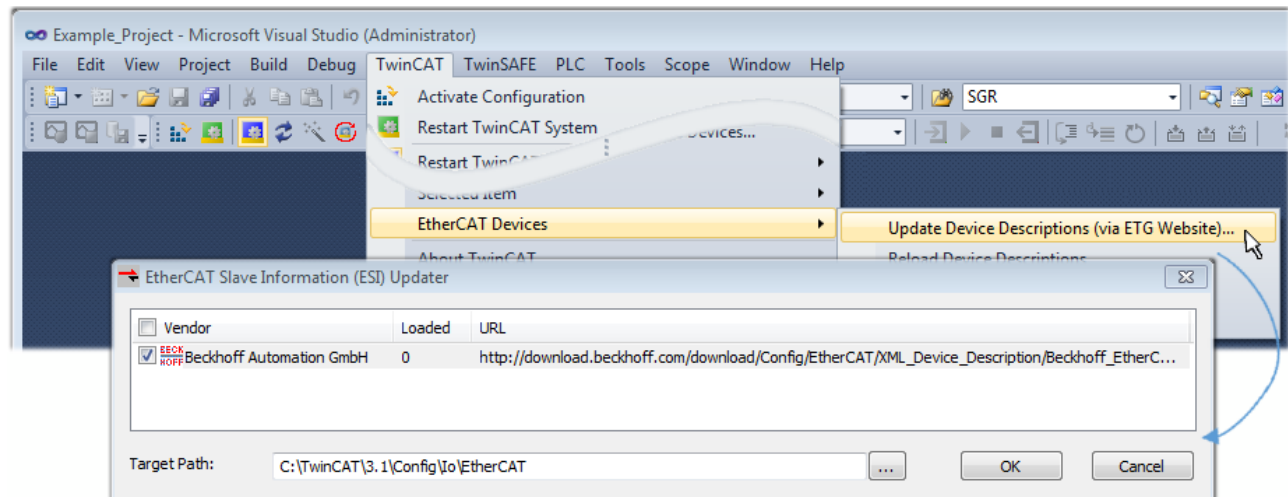
Fig. 79: Using the ESI Updater (TwinCAT 3)

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" [▶ 72].

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 [▶ 82] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 83]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 86]

The scan with existing configuration [▶ 87] 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.
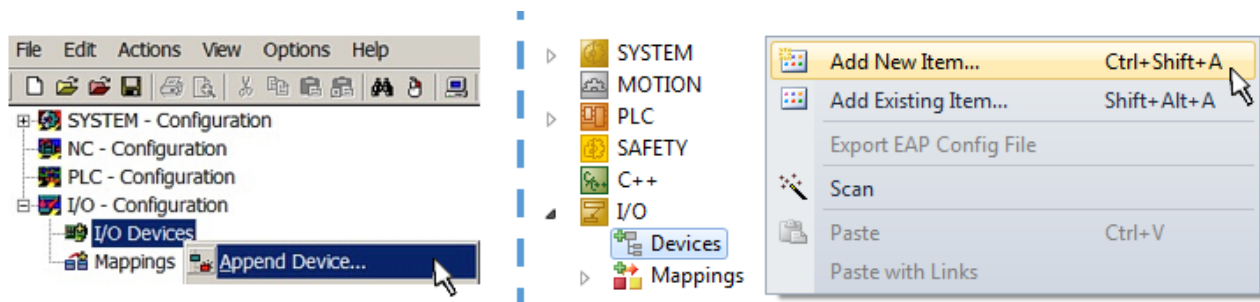
Fig. 80: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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".
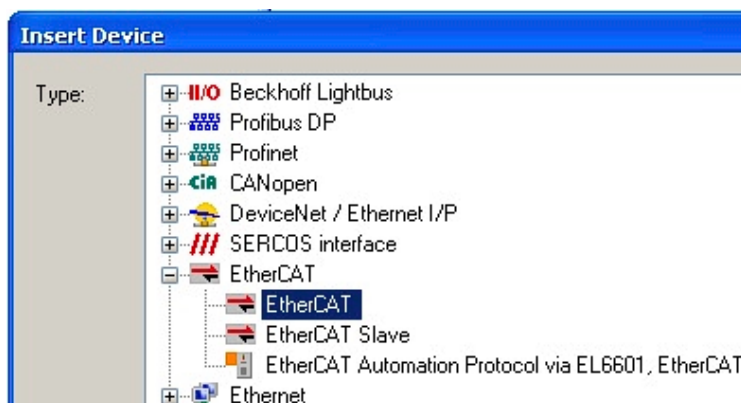
Fig. 81: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

Then assign a real Ethernet port to this virtual device in the runtime system.
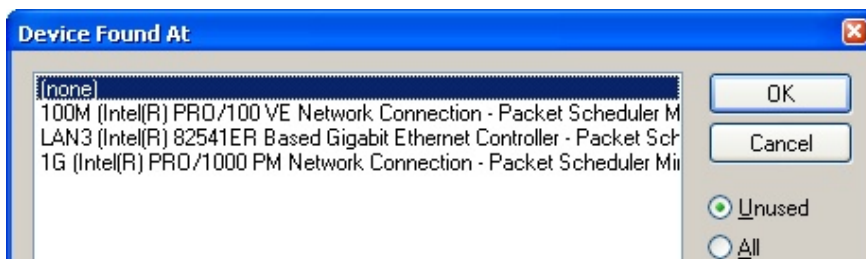
Fig. 82: Selecting the Ethernet port

BECKHOFF

This query may appear automatically when the EtherCAT device is created, or the assignment can be set/ modified later in the properties dialog; see Fig. "EtherCAT device properties (TwinCAT 2)".
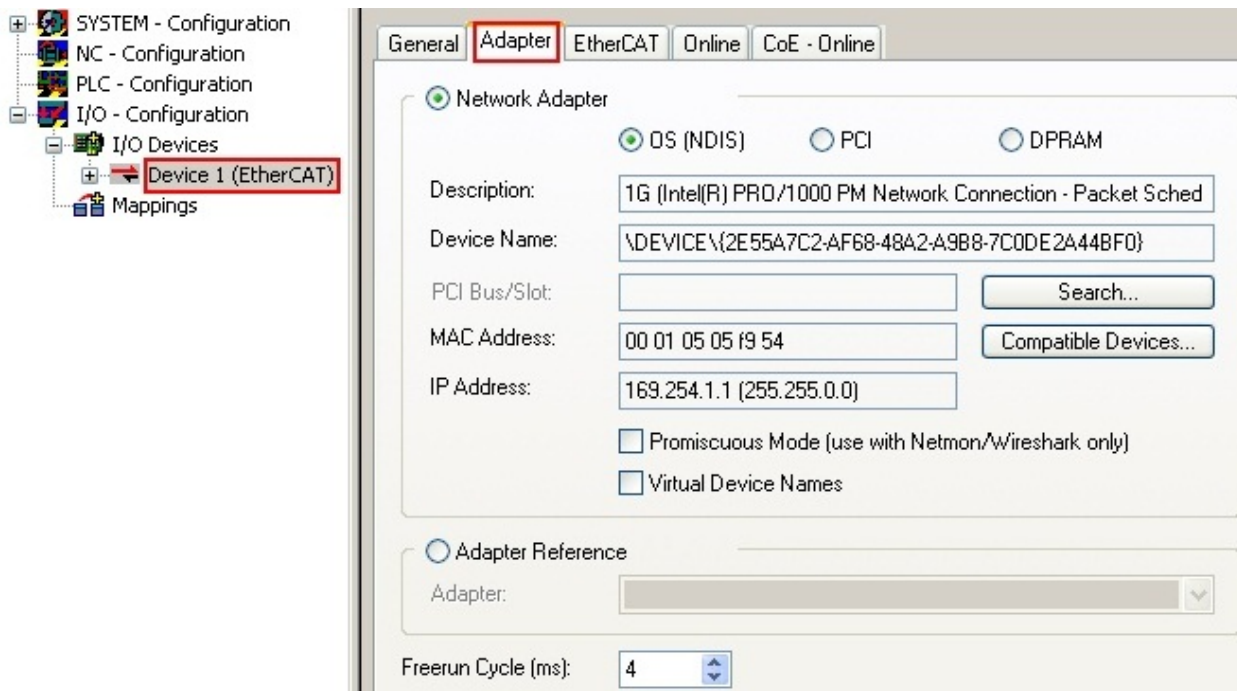
Fig. 83: EtherCAT device properties (TwinCAT 2)

TwinCAT 3: the properties of the EtherCAT device can be opened by double click on "Device .. (EtherCAT)" within the Solution Explorer under "I/O":
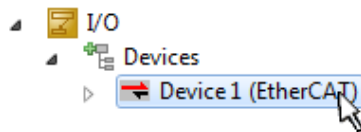
> **ℹ** **Selecting the Ethernet port**
>
> Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective installation page [▶ 66].

Defining EtherCAT slaves

Further devices can be appended by right-clicking on a device in the configuration tree.
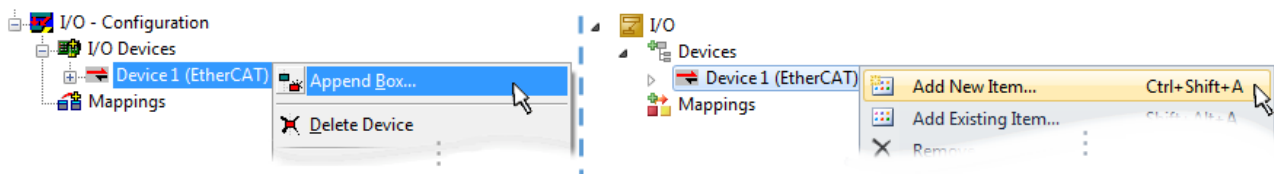
Fig. 84: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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

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

Overview of physical layer

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

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

The search field facilitates finding specific devices (since TwinCAT 2.11 or TwinCAT 3).
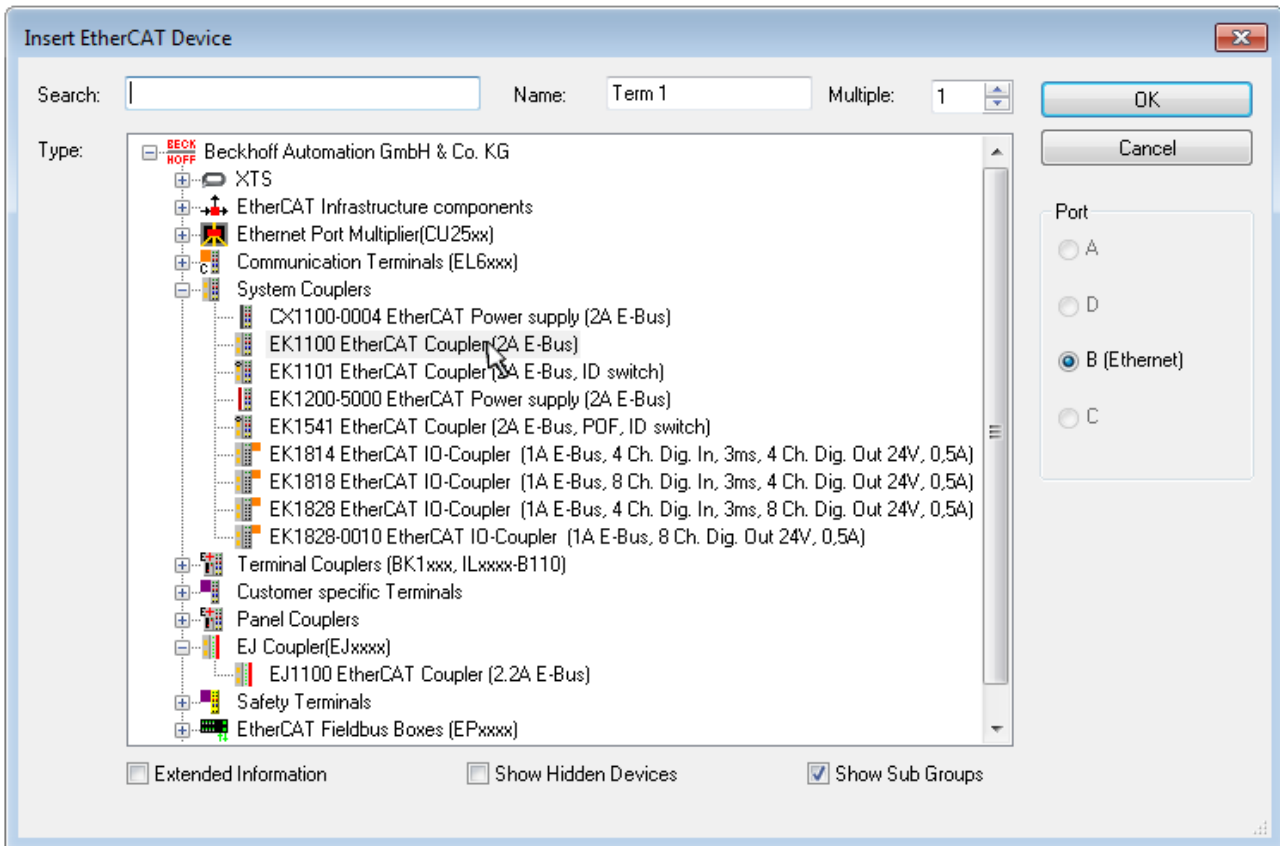
Fig. 85: Selection dialog for new EtherCAT device

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".
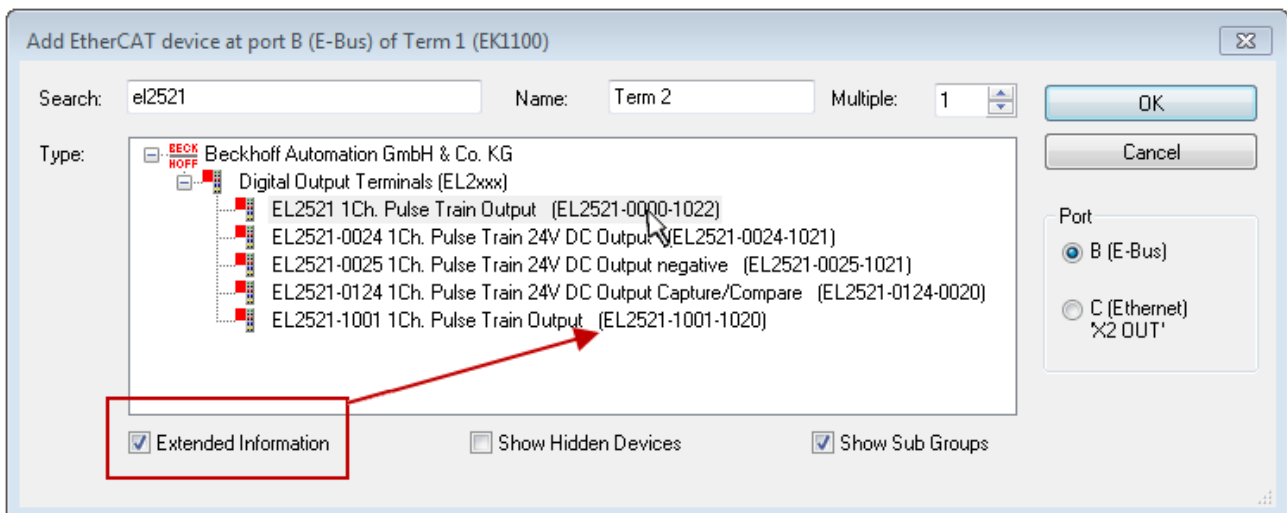
Fig. 86: Display of device revision

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. 87: Display of previous revisions

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

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

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

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

Example

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

Fig. 88: Name/revision of the terminal

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

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

Fig. 89: 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: `Config Mode` .

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

TwinCAT can be set into this mode:

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

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

ℹ Online scanning in Config mode

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

The TwinCAT 2 icon (🌐) or TwinCAT 3 icon (🧩) within the Windows-Taskbar always shows the TwinCAT mode of the local IPC. Compared to that, the System Manager window of TwinCAT 2 or the user interface of TwinCAT 3 indicates the state of the target system.
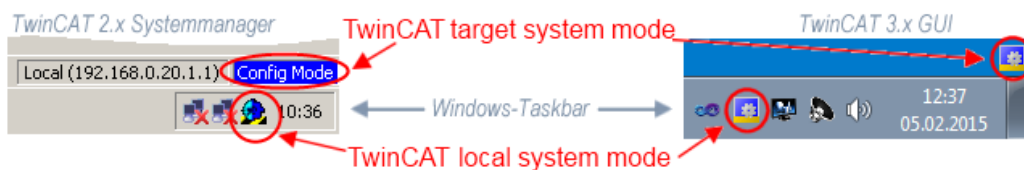
Fig. 90: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

Right-clicking on "I/O Devices" in the configuration tree opens the search dialog.
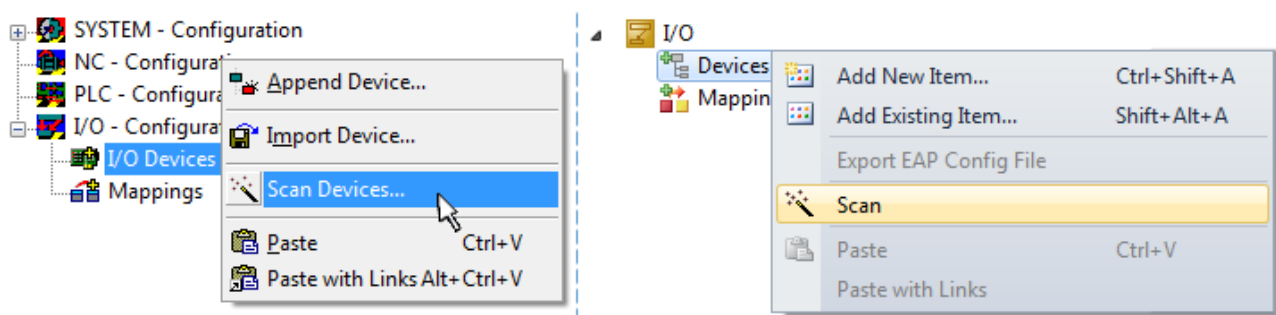
Fig. 91: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

This scan mode attempts to find not only EtherCAT devices (or Ethernet ports that are usable as such), but also NOVRAM, fieldbus cards, SMB etc. However, not all devices can be found automatically.
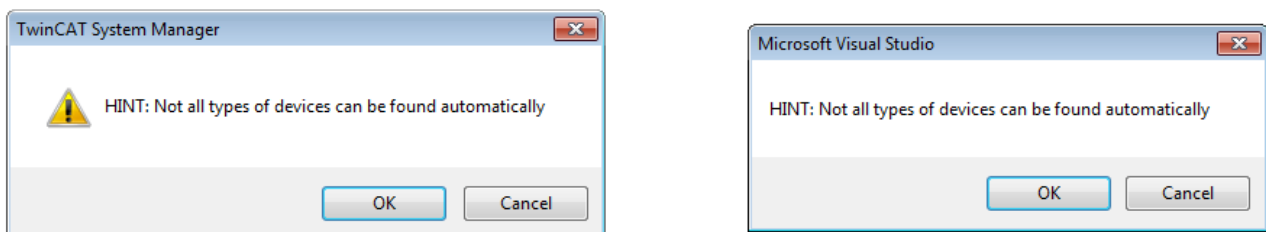
Fig. 92: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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" .
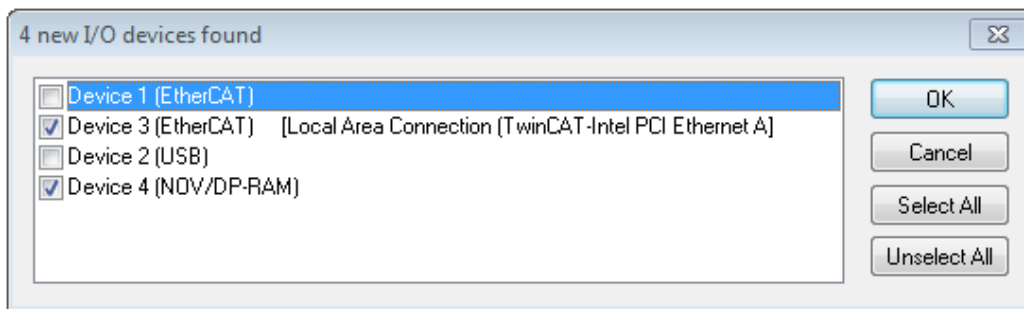
Fig. 93: Detected Ethernet devices

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".

ℹ Selecting the Ethernet port

Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective underlined installation page [▶ 66].

Detecting/Scanning the EtherCAT devices

ℹ Online scan functionality

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

Fig. 94: Example default state

NOTICE
Slave scanning in practice in series machine production
The scanning function should be used with care. It is a practical and fast tool for creating an initial configuration as a basis for commissioning. In series machine production or reproduction of the plant, however, the function should no longer be used for the creation of the configuration, but if necessary for comparison [▶ 87] with the defined initial configuration.Background: since Beckhoff occasionally increases the revision version of the delivered products for product maintenance reasons, a configuration can be created by such a scan which (with an identical machine construction) is identical according to the device list; however, the respective device revision may differ from the initial configuration.

Example:

Company A builds the prototype of a machine B, which is to be produced in series later on. To do this the prototype is built, a scan of the IO devices is performed in TwinCAT and the initial configuration "B.tsm" is created. The EL2521-0025 EtherCAT terminal with the revision 1018 is located somewhere. It is thus built into the TwinCAT configuration in this way:
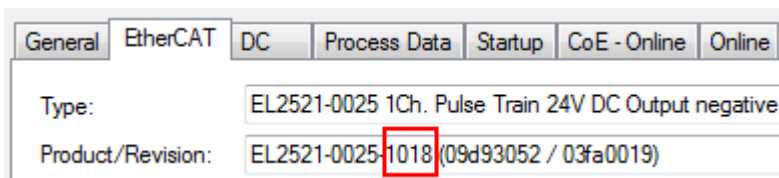
Fig. 95: Installing EthetCAT terminal with revision -1018

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

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

After some time Beckhoff extends the EL2521-0025 by a new feature C. Therefore the FW is changed, outwardly recognizable by a higher FW version and **a new revision -1019**. Nevertheless the new device naturally supports functions and interfaces of the predecessor version(s); an adaptation of "B.tsm" or even "B.pro" is therefore unnecessary. The series-produced machines can continue to be built with "B.tsm" and "B.pro"; it makes sense to perform a <u>comparative scan [▶ 87]</u> against the initial configuration "B.tsm" in order to check the built machine.

However, if the series machine production department now doesn't use "B.tsm", but instead carries out a scan to create the productive configuration, the revision **-1019** is automatically detected and built into the configuration:
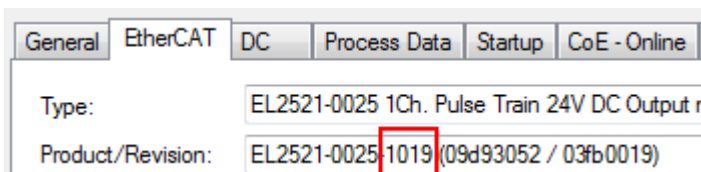
Fig. 96: Detection of EtherCAT terminal with revision -1019

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

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

If an EtherCAT device was created in the configuration (manually or through a scan), the I/O field can be scanned for devices/slaves.
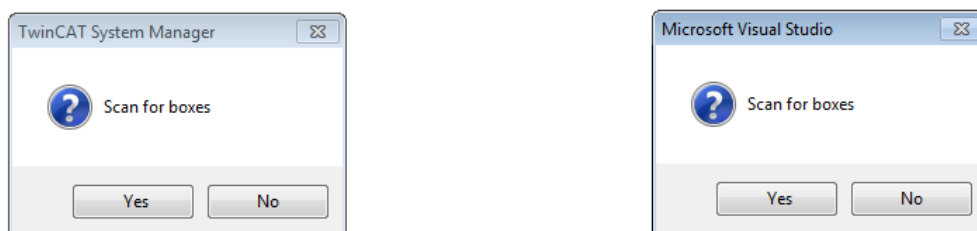
Fig. 97: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)
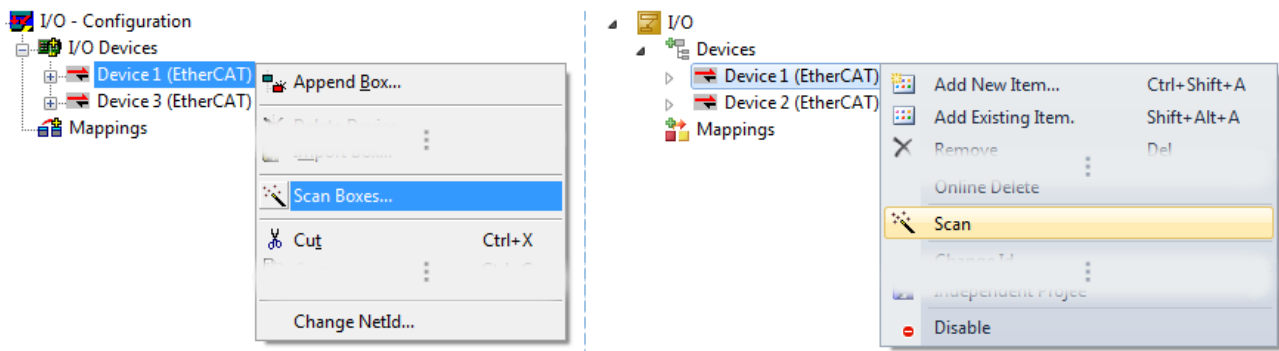
Fig. 98: Manual scanning for devices on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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

Fig. 99: *Scan progressexemplary by TwinCAT 2*

The configuration is established and can then be switched to online state (OPERATIONAL).
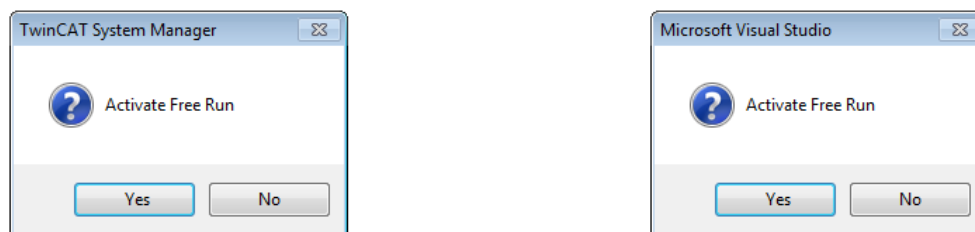
Fig. 100: Config/FreeRun query (left: TwinCAT 2; right: TwinCAT 3)

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

Fig. 102: 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*.
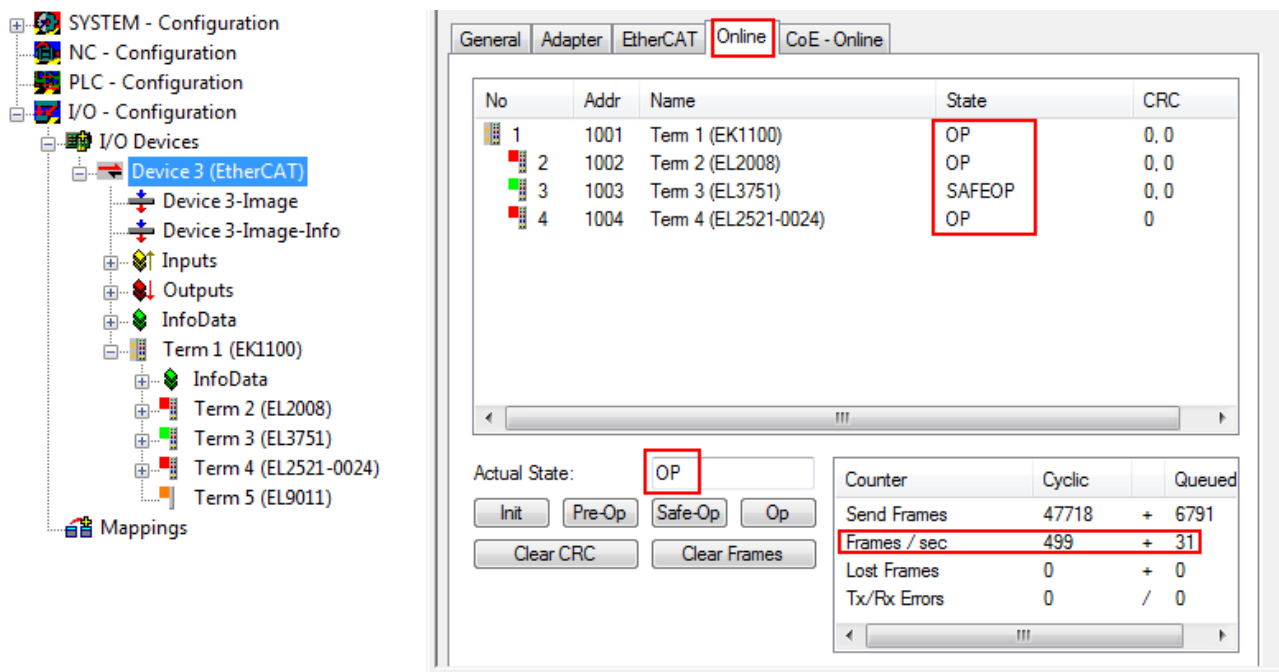
Fig. 103: Online display example

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 <u>manual procedure</u> [▶ 77].

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.
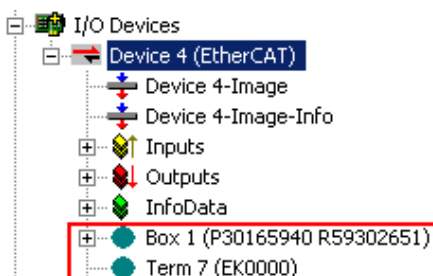
Fig. 104: Faulty identification

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

Scan over existing Configuration

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

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

Fig. 105: 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.
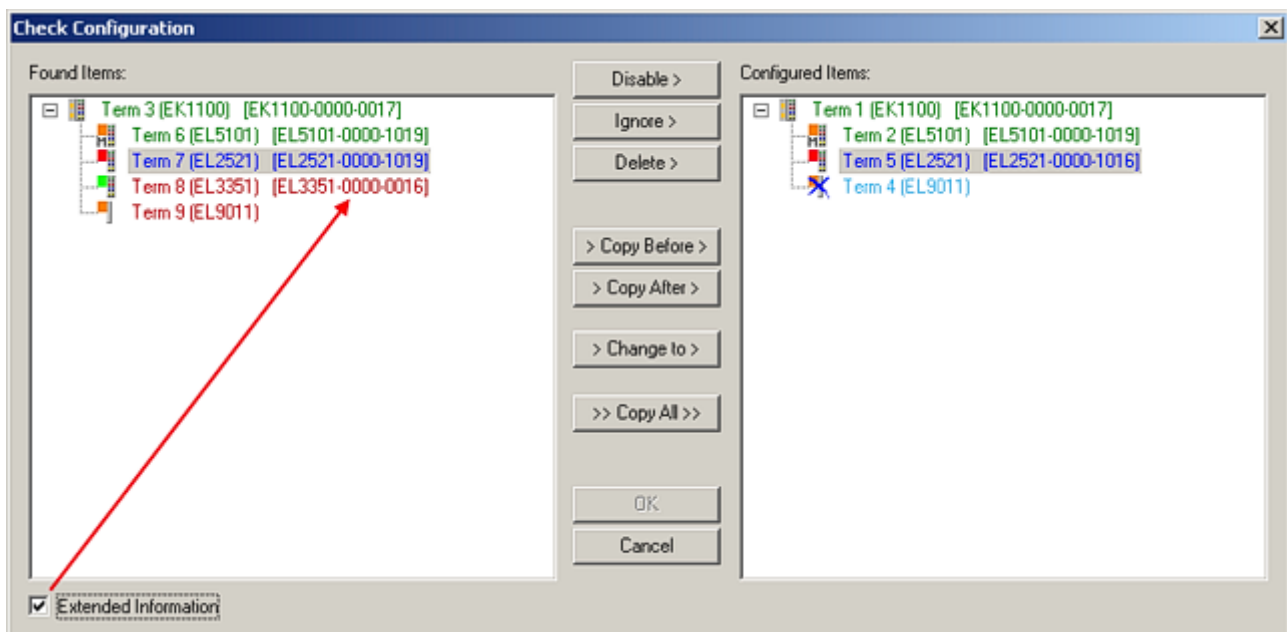
Fig. 106: Correction dialog

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

BECKHOFF

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

Device selection based on revision, compatibility

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

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

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

Example

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

Fig. 107: Name/revision of the terminal

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

In this case the process image of the device is shown in the configuration tree and can be parameterized as follows: linking with the task, CoE/DC settings, plug-in definition, startup settings, ...
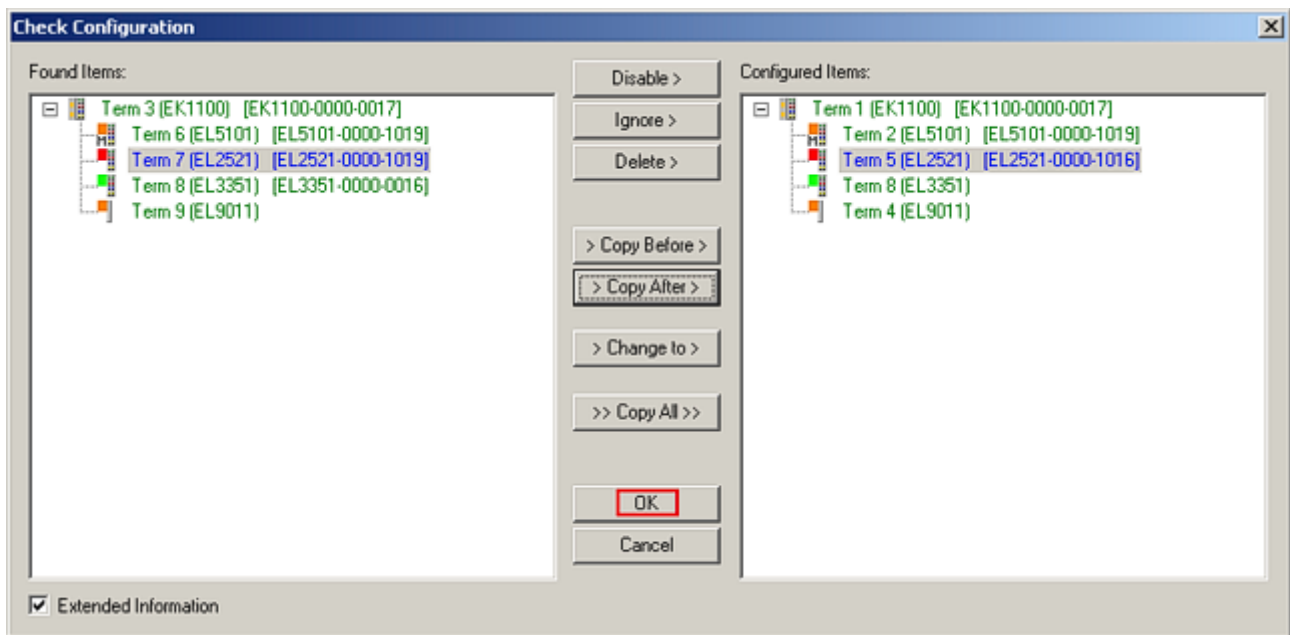
Fig. 108: Correction dialog with modifications

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

Change to Compatible Type

TwinCAT offers a function *Change to Compatible Type…* for the exchange of a device whilst retaining the links in the task.
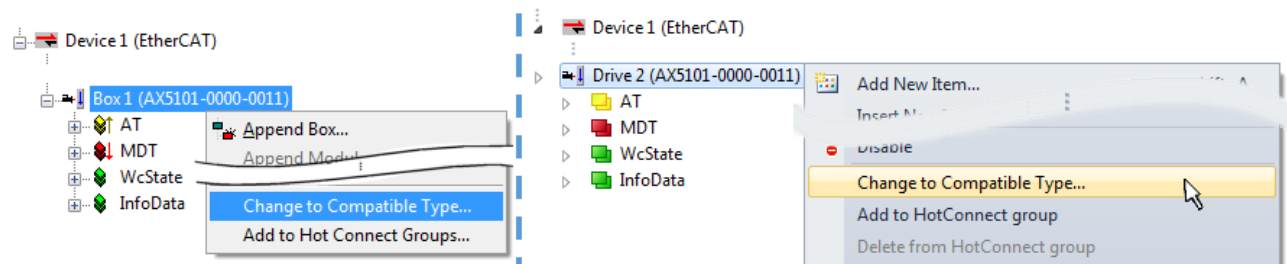
Fig. 109: Dialog "Change to Compatible Type…" (left: TwinCAT 2; right: TwinCAT 3)

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

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

- FMMU (additional ones are allowed)

- SyncManager (SM, additional ones are allowed)

- EoE (attributes MAC, IP)

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

- FoE

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

This function is preferably to be used on AX5000 devices.

Change to Alternative Type

The TwinCAT System Manager offers a function for the exchange of a device: Change to Alternative Type
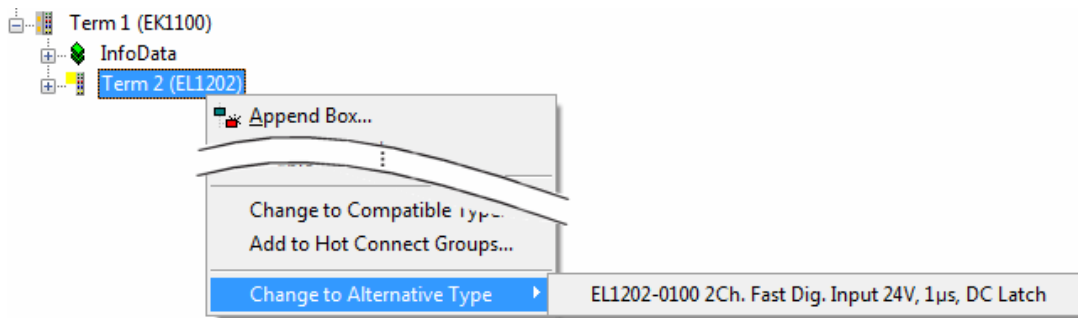
Fig. 110: TwinCAT 2 Dialog Change to Alternative Type

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

5.2.7 EtherCAT subscriber configuration

In the left-hand window of the TwinCAT 2 System Manager or the Solution Explorer of the TwinCAT 3 Development Environment respectively, click on the element of the terminal within the tree you wish to configure (in the example: EL3751 Terminal 3).
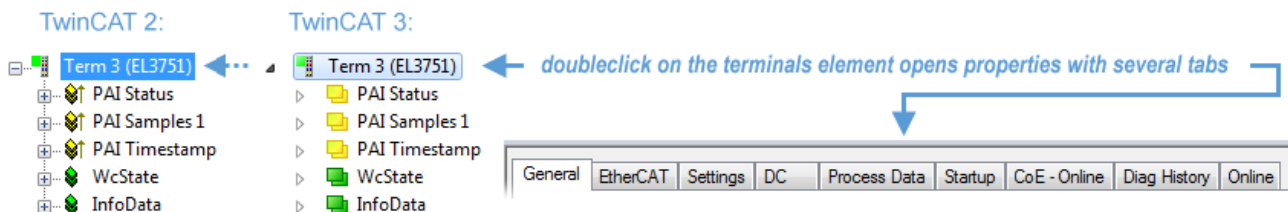
Fig. 111: Branch element as terminal EL3751

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
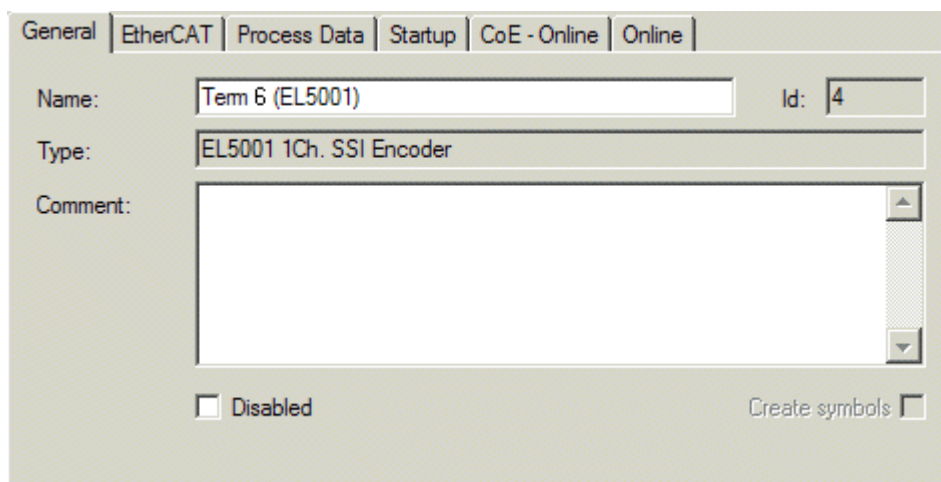
Fig. 112: "General" tab

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

"EtherCAT" tab

Fig. 113: "EtherCAT" tab

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

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

"Process Data" tab

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

Fig. 114: "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").
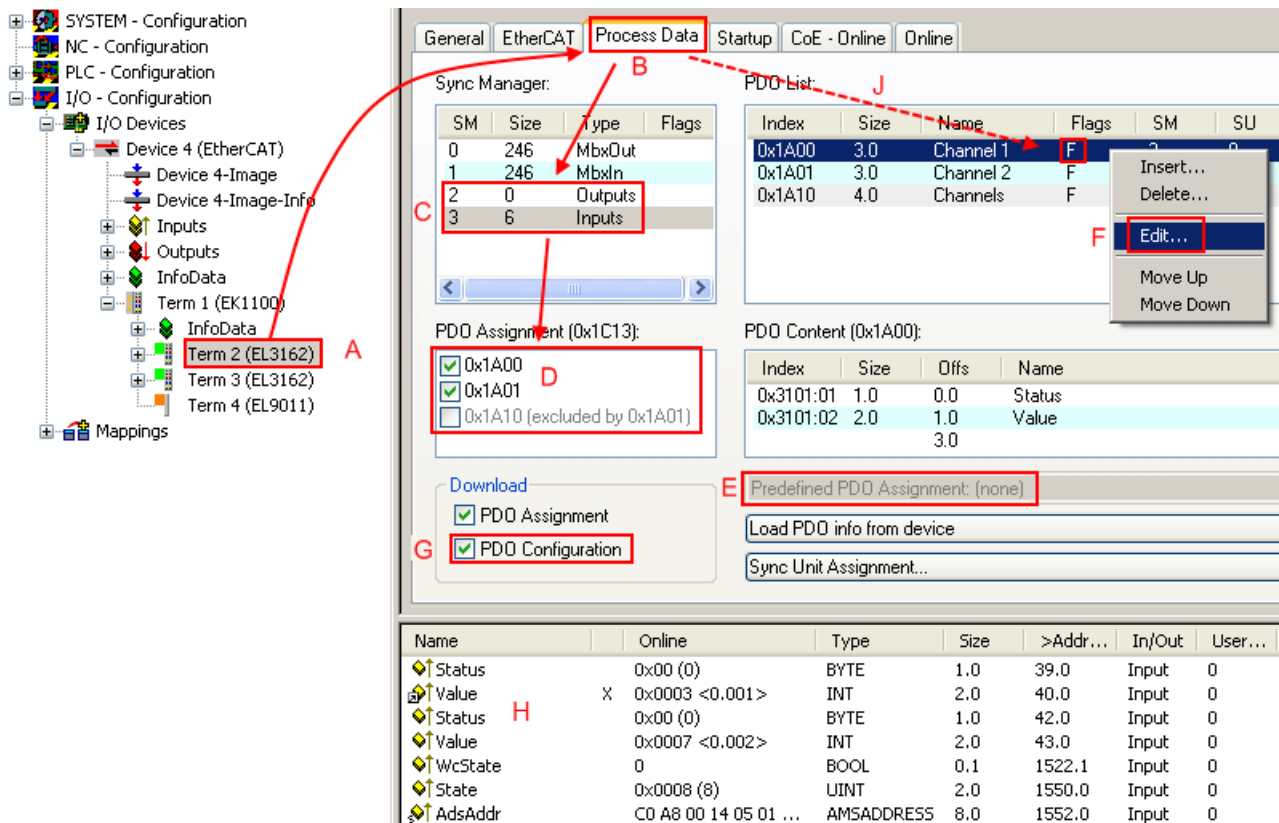
Fig. 115: Configuring the process data

> **ℹ** **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 [▶ 98] 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.

BECKHOFF
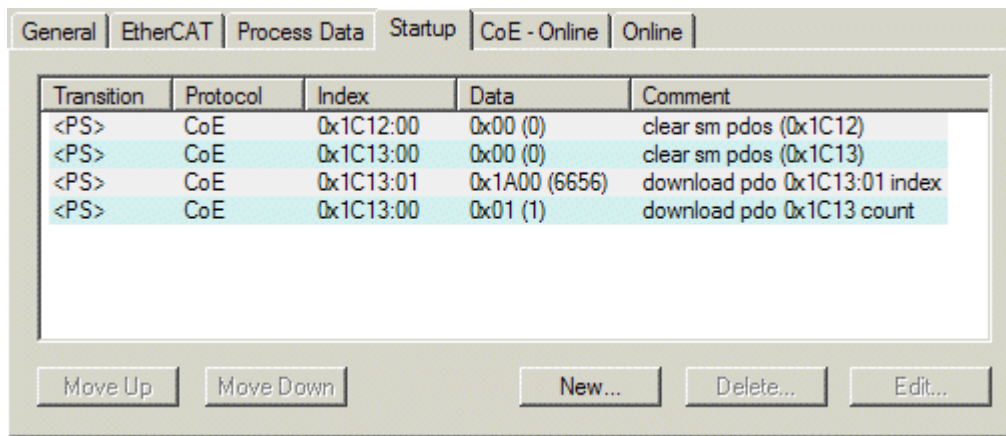
Fig. 116: "Startup" tab

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

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

"CoE - Online" tab

The additional *CoE - Online* tab is displayed if the EtherCAT slave supports the *CANopen over EtherCAT* (CoE) protocol. This dialog lists the content of the object list of the slave (SDO upload) and enables the user to modify the content of an object from this list. Details for the objects of the individual EtherCAT devices can be found in the device-specific object descriptions.
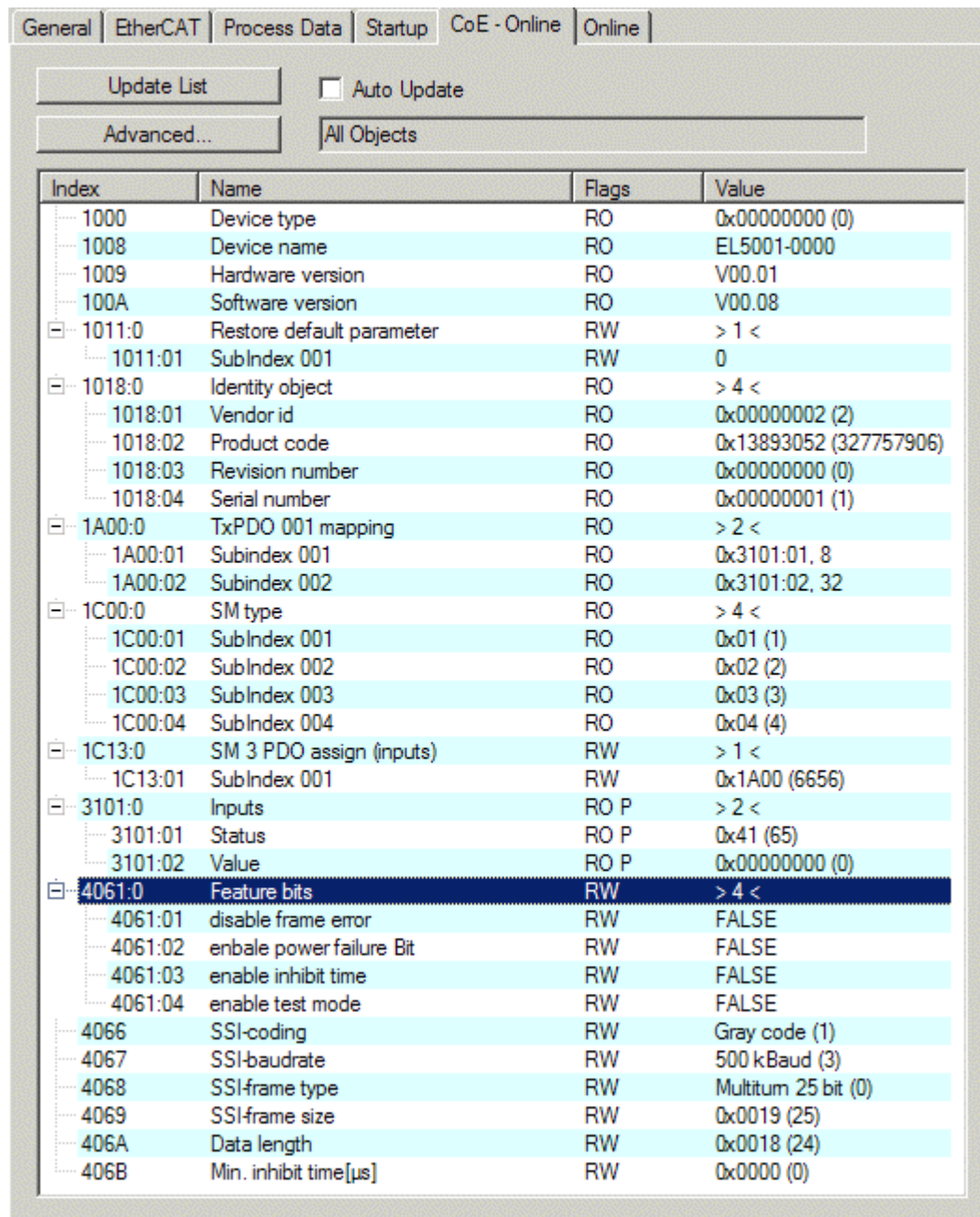
Fig. 117: "CoE - Online" tab

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.
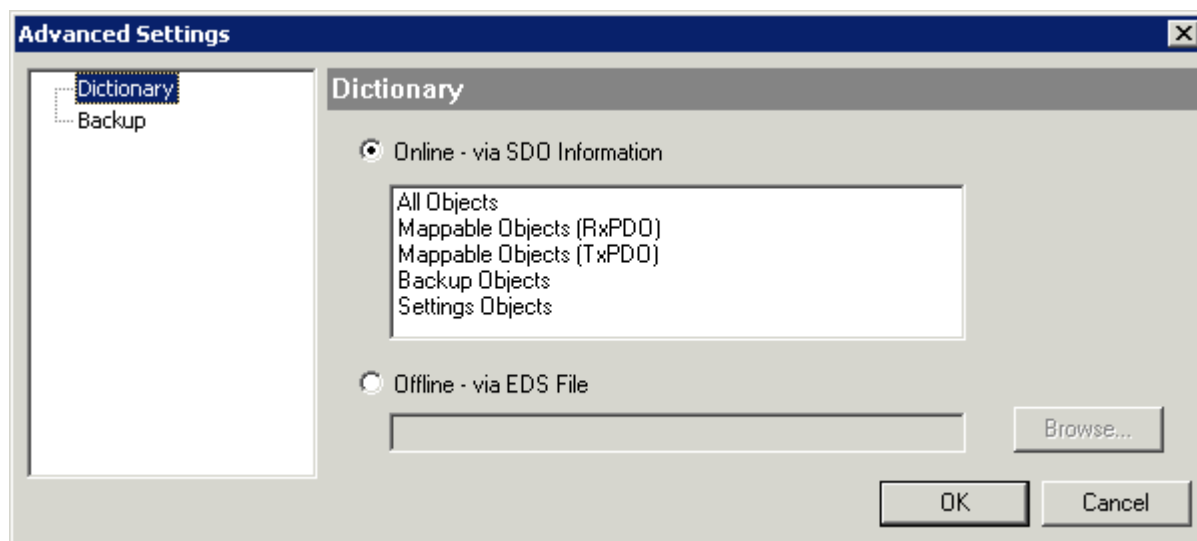
Fig. 118: Dialog "Advanced settings"

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

"Online" tab
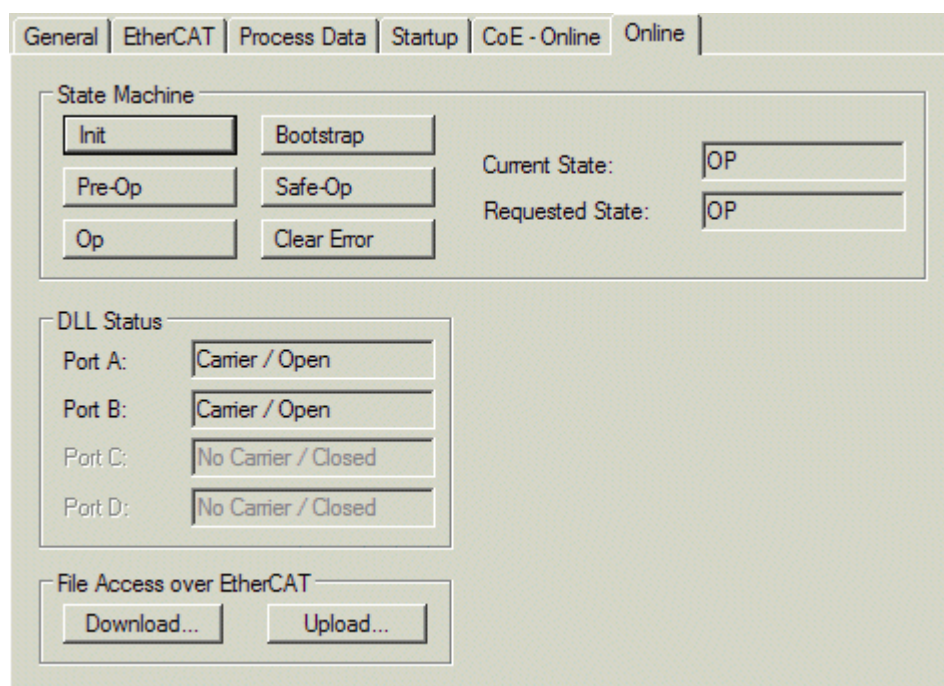
Fig. 119: "Online" tab

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)
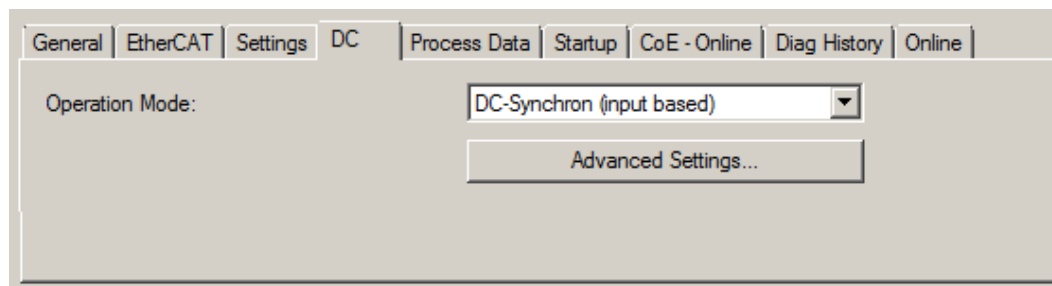
Fig. 120: "DC" tab (Distributed Clocks)

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

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

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

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.

ℹ **Activation of PDO assignment**

✓ If you have changed the PDO assignment, in order to activate the new PDO assignment,

a) the EtherCAT slave has to run through the PS status transition cycle (from pre-operational to safe-operational) once (see Online tab [▶ 96]),

b) and the System Manager has to reload the EtherCAT slaves

(🔀 button for TwinCAT 2 or ⟳ button for TwinCAT 3)

PDO list

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

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

PDO Content

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

Download

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

PDO Assignment

If this check box is selected, the PDO assignment that is configured in the PDO Assignment list is downloaded to the device on startup. The required commands to be sent to the device can be viewed in the <u>Startup [▶ 93]</u> tab.

PDO Configuration

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

5.2.8 Import/Export of EtherCAT devices with SCI and XTI

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

5.2.8.1 Basic principles

An EtherCAT slave is basically parameterized through the following elements:

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

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

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

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

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

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

TwinCAT offers two methods for this purpose:

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

An example is provided below for illustration purposes: an EL3702 terminal with standard setting is switched to 2-fold oversampling (blue) and the optional PDO "StartTimeNextLatch" is added (red):

The two methods for exporting and importing the modified terminal referred to above are demonstrated below.

5.2.8.2 Procedure within TwinCAT with xti files

Each IO device can be exported/saved individually:

The xti file can be stored:

and imported again in another TwinCAT system via "Insert Existing item":
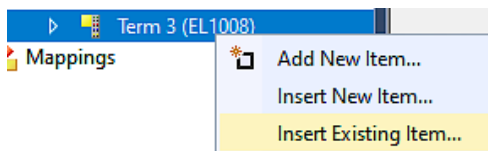
5.2.8.3 Procedure within and outside TwinCAT with sci file

Note regarding availability (2021/01)

The SCI method is available from TwinCAT 3.1 build 4024.14.

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

Export:

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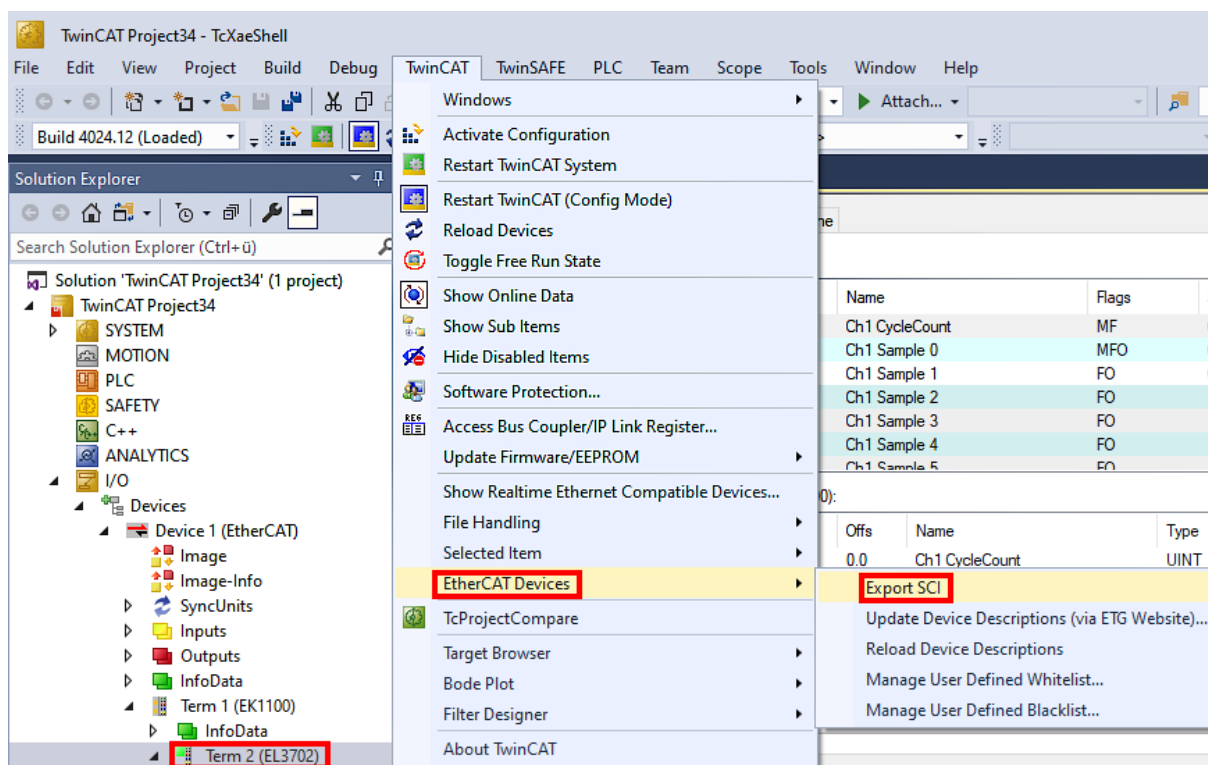
- If TwinCAT is offline (i.e. if there is no connection to an actual running controller) a warning message may appear, because after executing the function the system attempts to reload the EtherCAT segment. However, in this case this is not relevant for the result and can be acknowledged by clicking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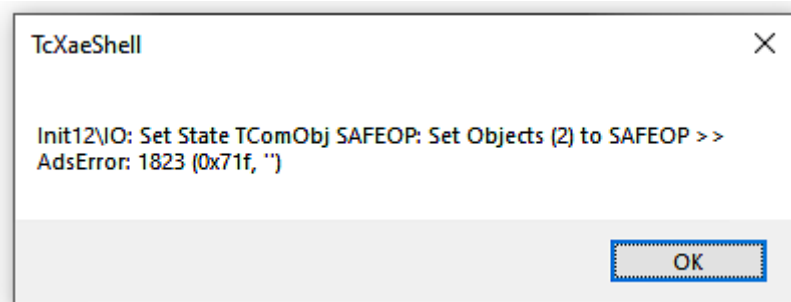
- A description may also be provided:

- Explanation of the dialog box:

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

- A list view is available for multiple selections (*Export multiple SCI files)*:

- Selection of the slaves to be exported:
  - All:
    All slaves are selected for export.

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

Dateiname:	EL3702 with added StartTimeNextLatch.sci
Dateityp:	SCI file (*.sci)

- The export takes place:

Export SCI | based on specification 1.0.12.3 (Draft)

Name   EL3702 with added StartTimeNextLatch

Description   just an example for a specific description

SCI Created   ✕

The SCI file 'C:\TwinCAT\3.1\Config\Io\EtherCAT\EL3702 with added StartTimeNextLatch.sci' was created

Open Folder   Close

☐ AoE | Set AmsNetId

☐ EoE | Set MAC and IP

☐ CoE | Set cycle time (0x1C3x.2)

Export
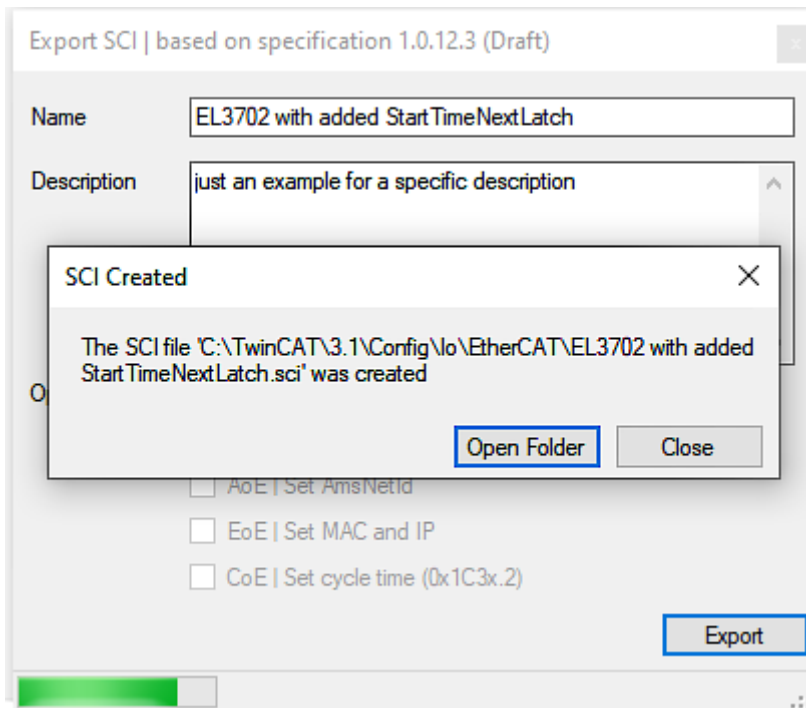
Import

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

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

- Open the selection dialog:

```
▲ 📊 Term 1 (EK1100)
  ▷ 🟩 InfoData
  ▷ 📊 Term 2 (EL3702)
  ▷ 📊 Term 3 (EL1008)
📍 Mappings        *📋 Add New Item...
                     Insert New Item...
```
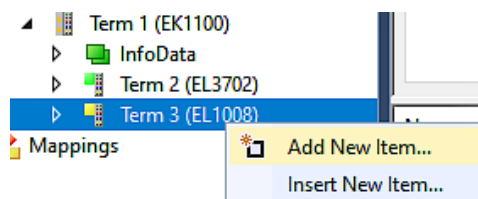
- Display SCI devices and select and insert the desired device:
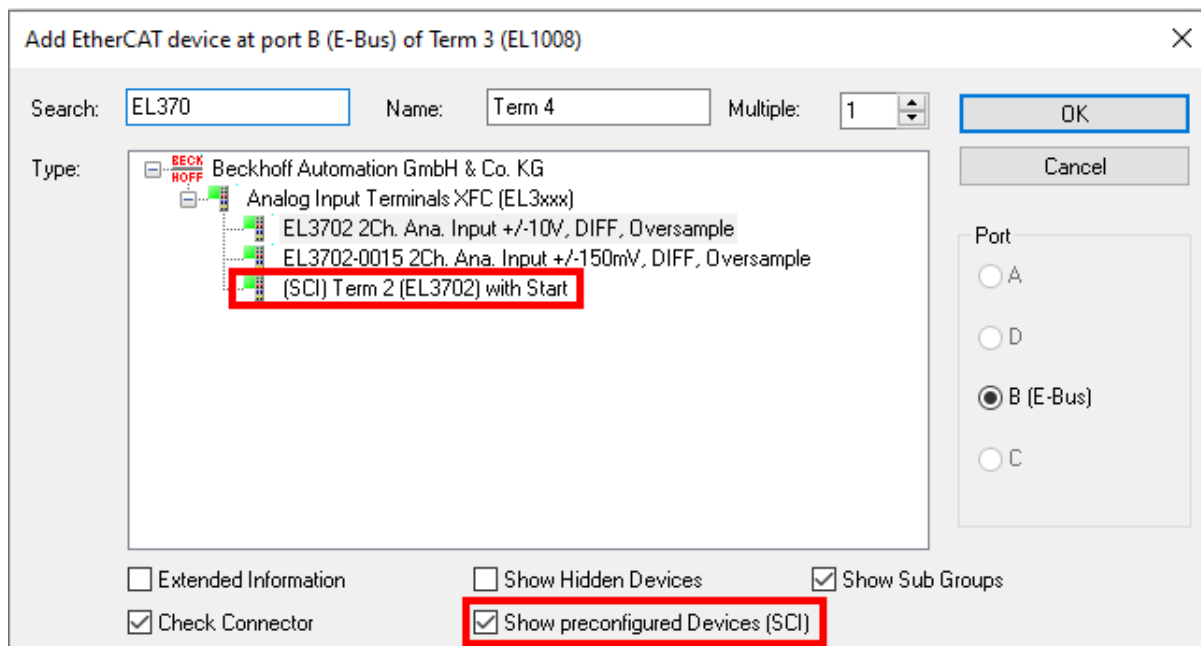
Additional Notes

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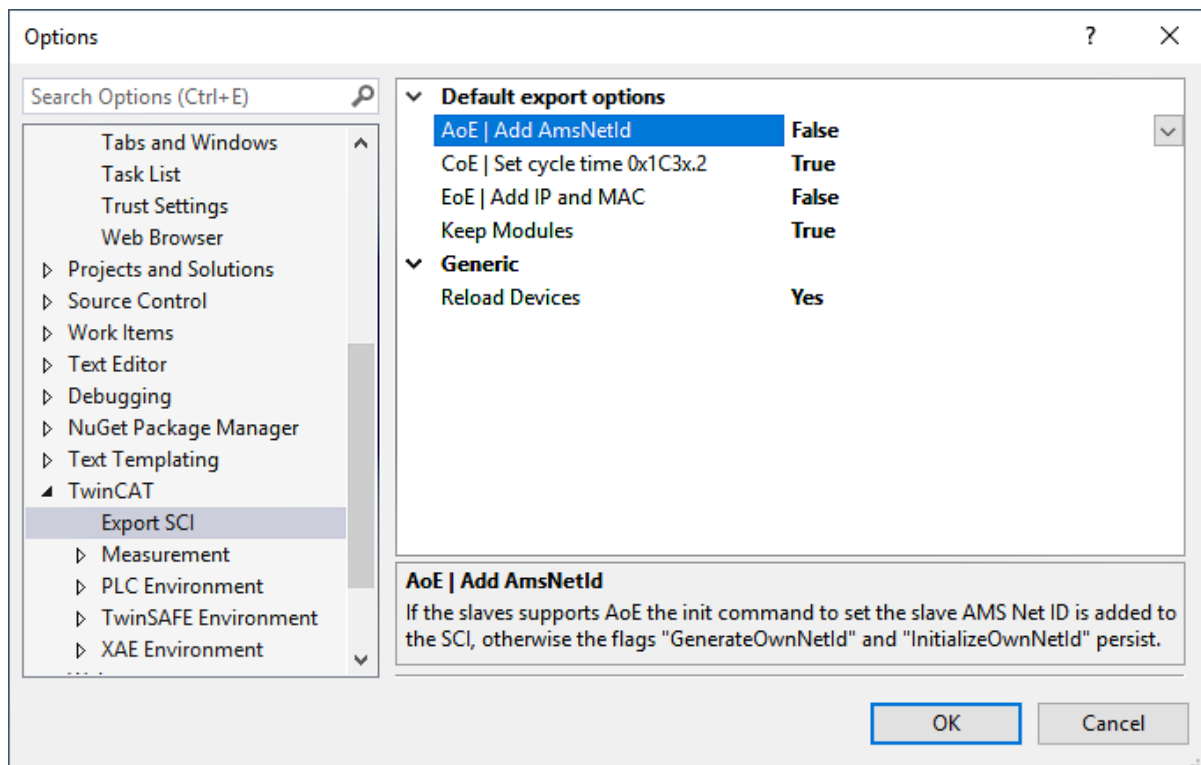
Explanation of the settings:

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

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

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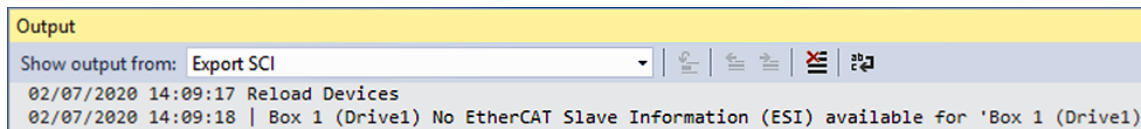
5.3   General Commissioning Instructions for an EtherCAT Slave

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

Diagnosis in real time: WorkingCounter, EtherCAT State and Status

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

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

Any application that relies on I/O data from a fieldbus being correct and up to date must make diagnostic access to the corresponding underlying layers. EtherCAT and the TwinCAT System Manager offer comprehensive diagnostic elements of this kind. Those diagnostic elements that are helpful to the controlling task for diagnosis that is accurate for the current cycle when in operation (not during commissioning) are discussed below.
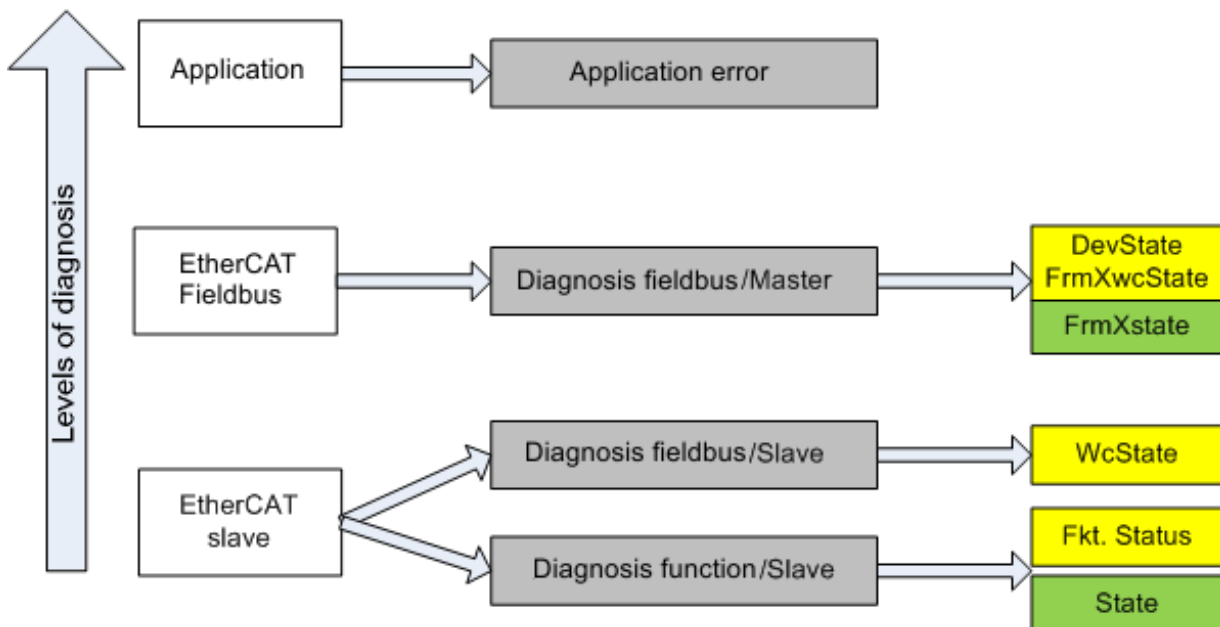
Fig. 121: Selection of the diagnostic information of an EtherCAT Slave

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.
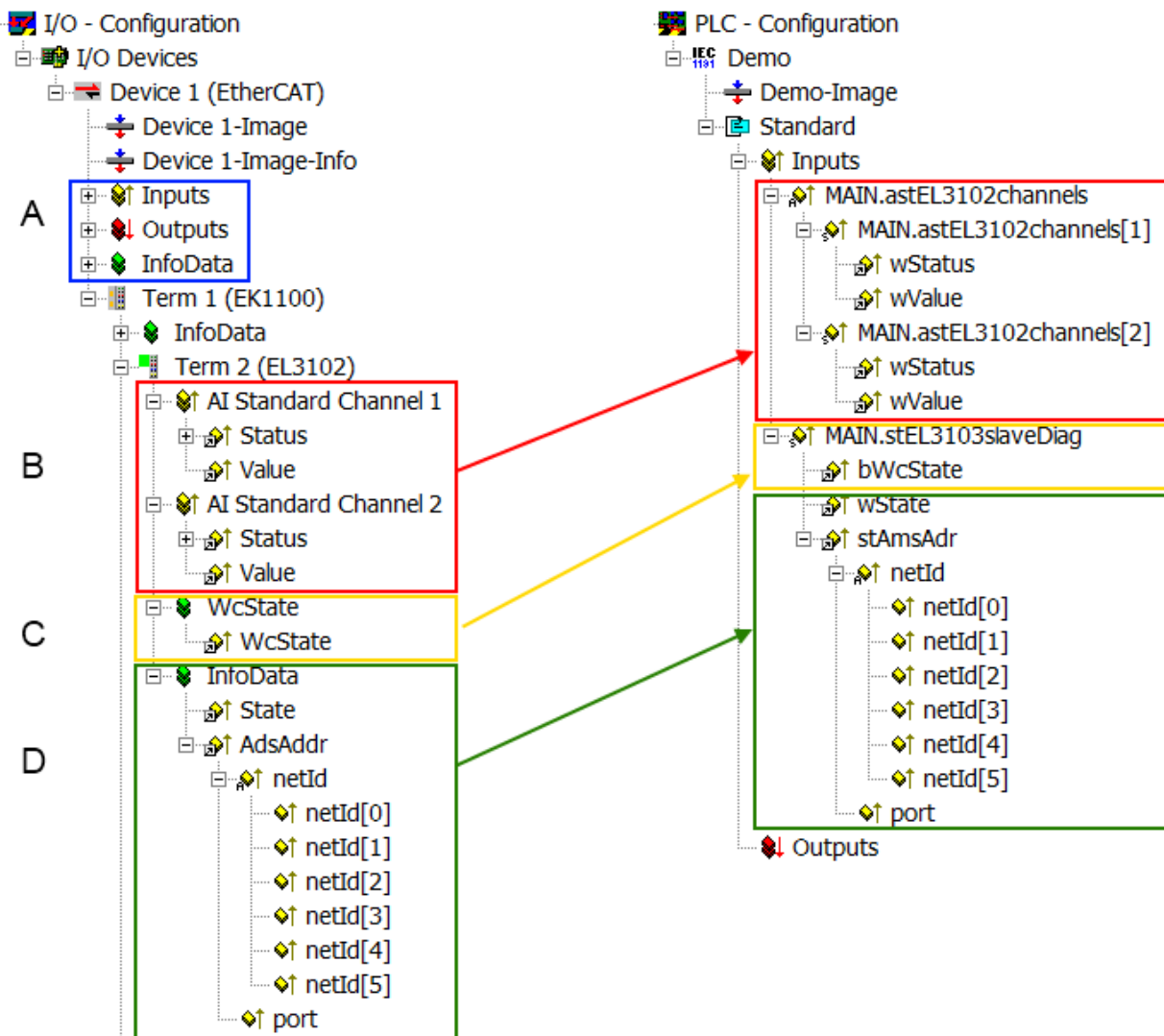
Fig. 122: Basic EtherCAT Slave Diagnosis in the PLC

The following aspects are covered here:

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

NOTICE

Diagnostic information

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

CoE Parameter Directory

The CoE parameter directory (CanOpen-over-EtherCAT) is used to manage the set values for the slave concerned. Changes may, in some circumstances, have to be made here when commissioning a relatively complex EtherCAT Slave. It can be accessed through the TwinCAT System Manager, see Fig. *EL3102, CoE directory*:
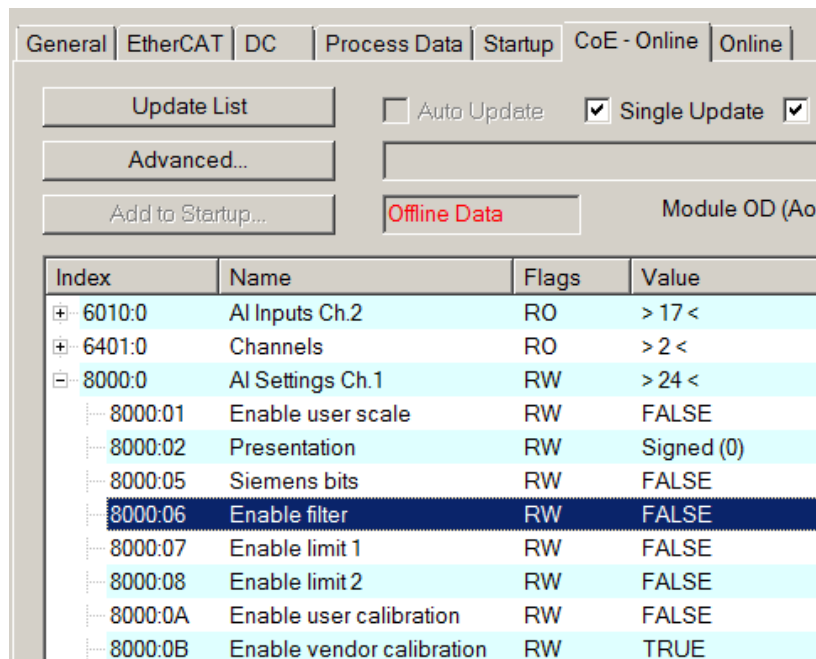
Fig. 123: EL3102, CoE directory

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

A few brief extracts:

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

Commissioning aid in the TwinCAT System Manager

Commissioning interfaces are being introduced as part of an ongoing process for EL/EP EtherCAT devices. These are available in TwinCAT System Managers from TwinCAT 2.11R2 and above. They are integrated into the System Manager through appropriately extended ESI configuration files.
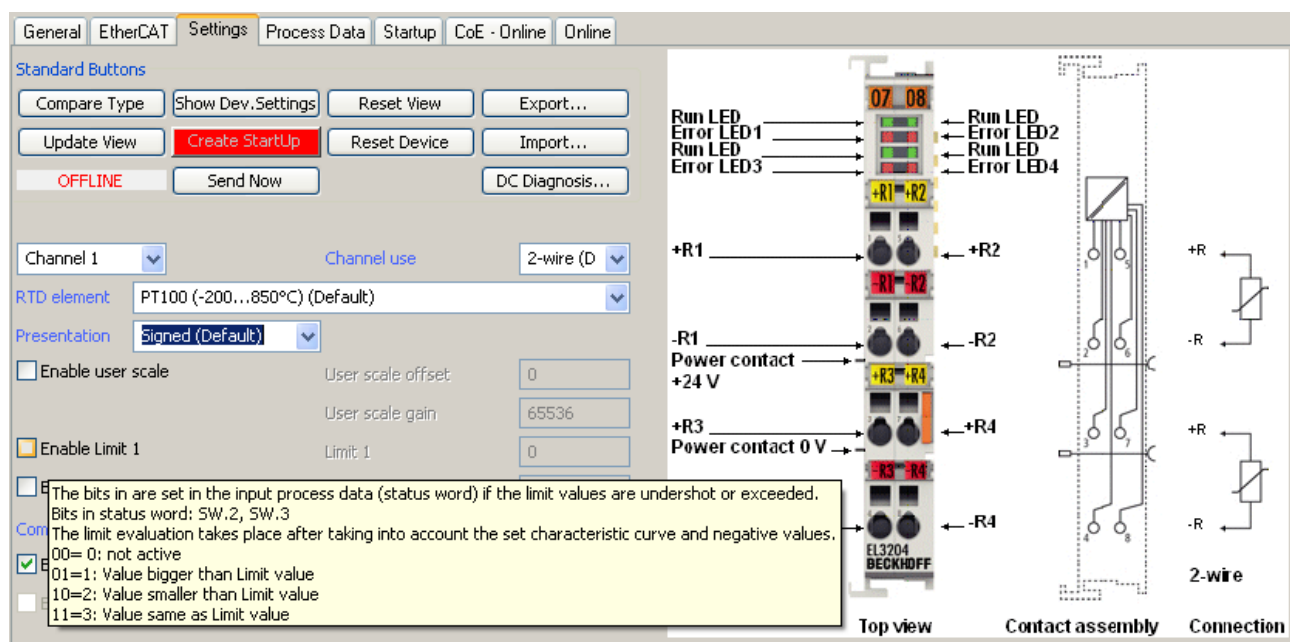
Fig. 124: Example of commissioning aid for a EL3204

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 [▶ 21]" 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.
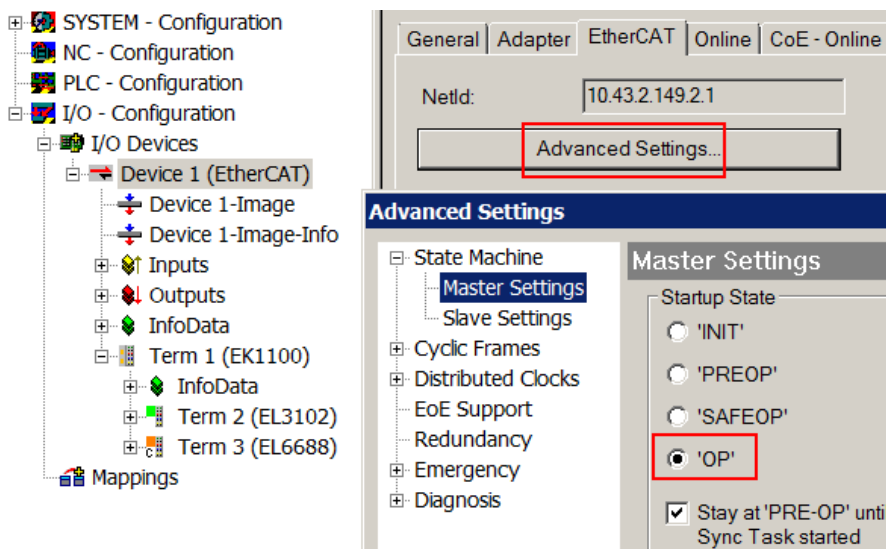
Fig. 125: Default behaviour of the System Manager

In addition, the target state of any particular Slave can be set in the "Advanced Settings" dialogue; the standard setting is again OP.
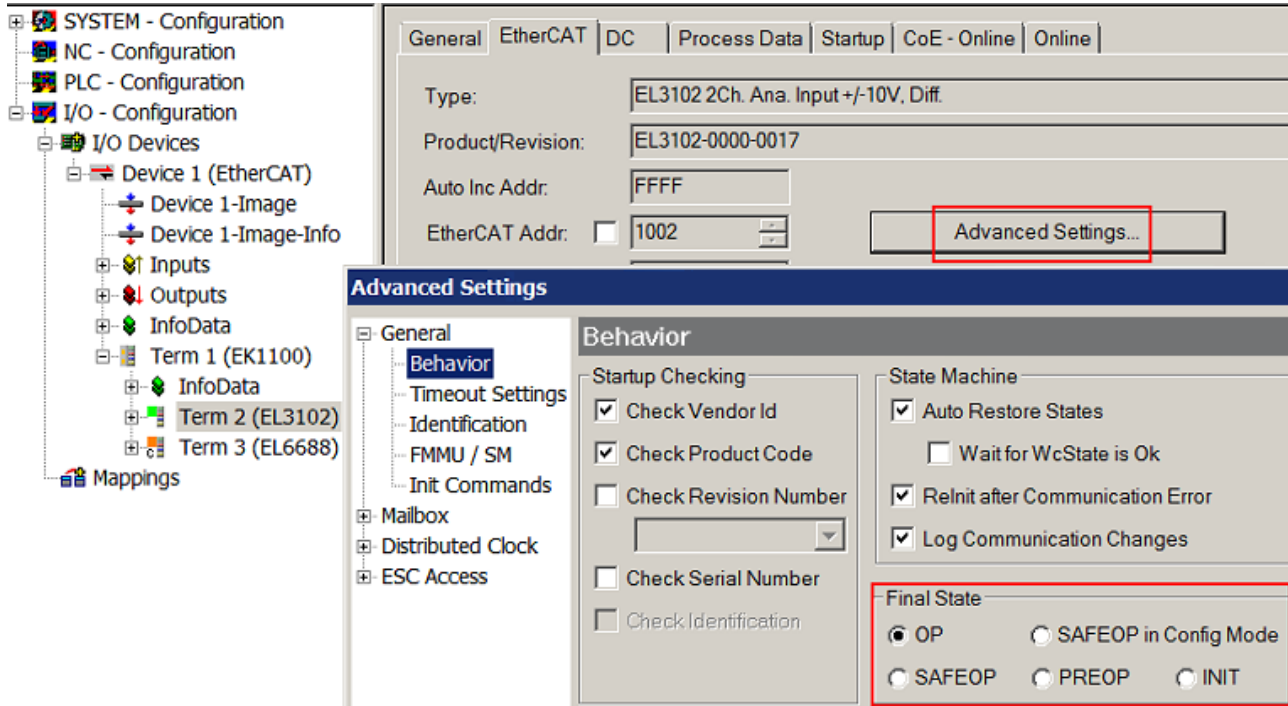
Fig. 126: Default target state in the Slave

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.
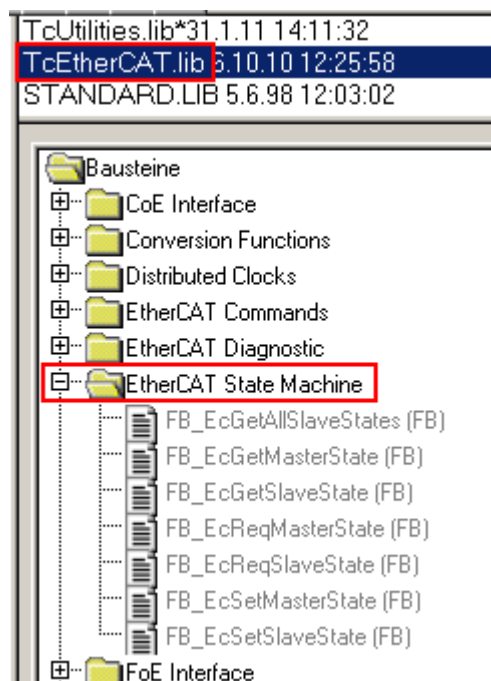
Fig. 127: PLC function blocks

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.

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. 128: Illegally exceeding the E-Bus current

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

Message

E-Bus Power of Terminal 'Term 3 (EL6688)' may to low (-240 mA) - please check!
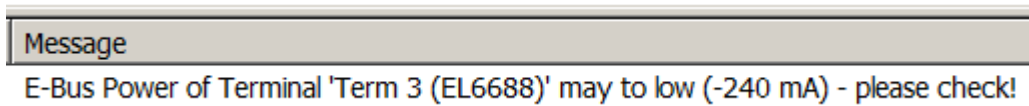
Fig. 129: Warning message for exceeding E-Bus current

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

6 EM37xx - Commissioning

6.1 Basic function principles

Fig. 130: EM37xx Modules

The pressure measuring modules EM3701, EM3702 and EM3712 record direct differential and relative pressures of non-aggressive gases.
As in electronic signal acquisition the pressure measurement takes place via a terminal module. The pressure measuring modules convert the measured pressure into an electrical signal and make it available to the higher-level control system with a resolution of 16 bits.
With the aid of user calibration, the measuring module can be adjusted to the ambient conditions.
A temperature compensation facility is integrated on the chip and increases the stability of the measurement result.
The status LEDs indicate proper function or errors such as range exceedance.

Pressure measuring modules measure the pressure relative to a reference value. A distinction is made between three physical measurement methods:

- **Absolute pressure measurement**: The pressure is measured relative to an absolute vacuum.
  Absolute pressure measurement is used in barometers, for example.

- **Relative pressure measurement**: The pressure is measured relative to a defined reference pressure.
  The ambient pressure is used as a reference pressure.
  The average atmospheric pressure at sea level is 1013.25 $hPa_{absolute}$ = 0 $hPa_{relative}$ (at 0 °C and 45° latitude). The ambient air pressure depends on the altitude, the temperature and the latitude of the measuring site.
  Values of lower ambient air pressure are referred to as negative pressure and have a negative sign.
  Values of greater ambient pressure are referred to as overpressure and have a positive sign.
  $P_{(relative)} = P_{(absolute)} - P_{(ambient)}$; see diagram: *Diagram showing relative and absolute pressure*
  The relative pressure measurement is applied e.g. for filling level measurements in aerated tanks.

- **Differential pressure measurement**: The difference between any two pressures is measured (note limit values). Differential pressure measurement is used for monitoring filters, for example.

Measuring method	Reference value	Value range (EM37xx)	Example
Relative pressure measurement (EM3702, EM3712)	Ambient air pressure $1013,25\ hPa_{Absolut} = 0\ hPa_{Relativ}$	EM3702: $0 \leq P_{relative} \leq 7500\ hPa_{relative}$ EM3712: $-1000\ hPa_{Relativ} \leq P_{Relativ} \leq 1000\ hPa_{Relativ}$	Filling level measurement of aerated tanks, pneumatic systems
Differential pressure measurement (EM3701)	$P_{difference} = P_1 - P_2$; $P_1 > P_2$	$-100\ hPa_{differential} \leq P_{difference} \leq 100\ hPa_{differential}$	Monitoring of filters
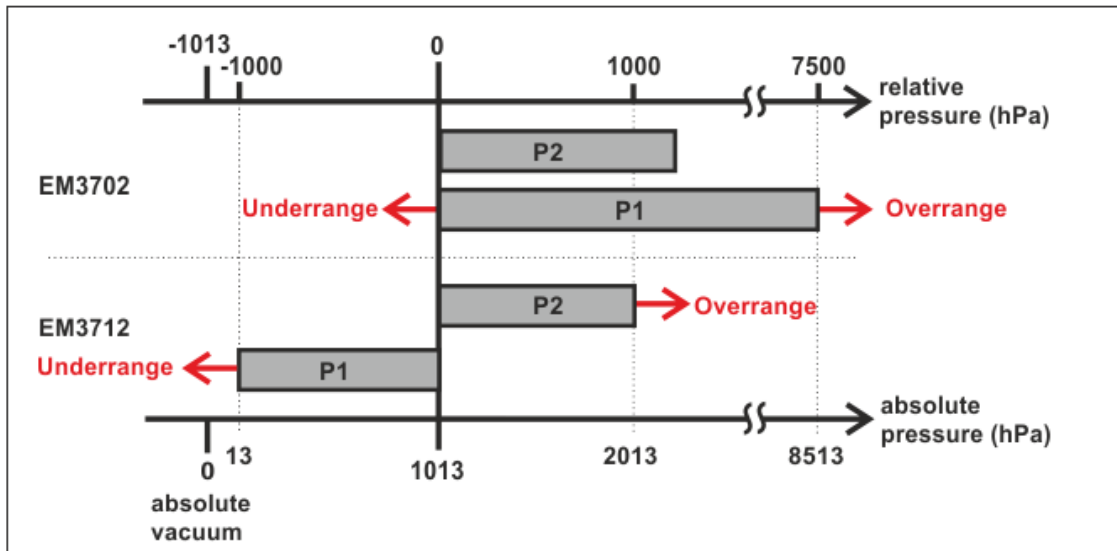
Fig. 131: Diagram showing relative and absolute pressure

EM3701 - single-channel differential pressure measuring module
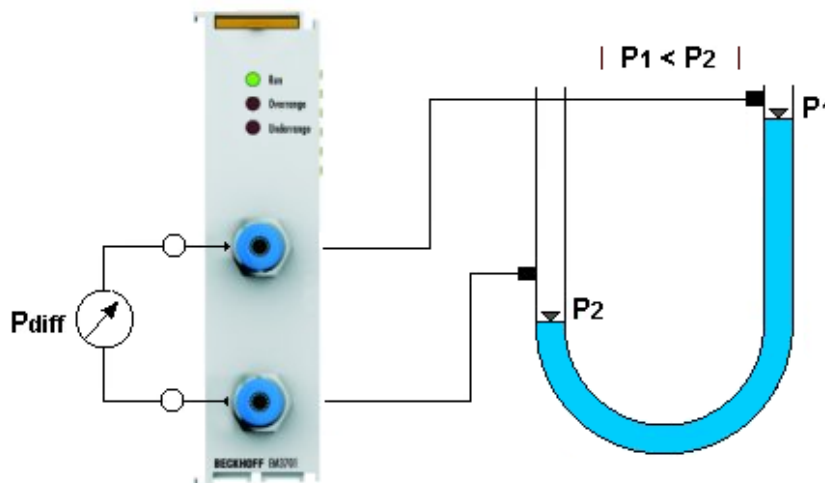
Fig. 132: EM3701 single-channel differential pressure measurement EM3701

The EM3701 terminal module can measure pressure differences between 0 and 100 hPa (0 and 100 mbar) between two hose connections.

EM3702 - two-channel relative pressure measuring module (0 hPa to 7500 hPa)
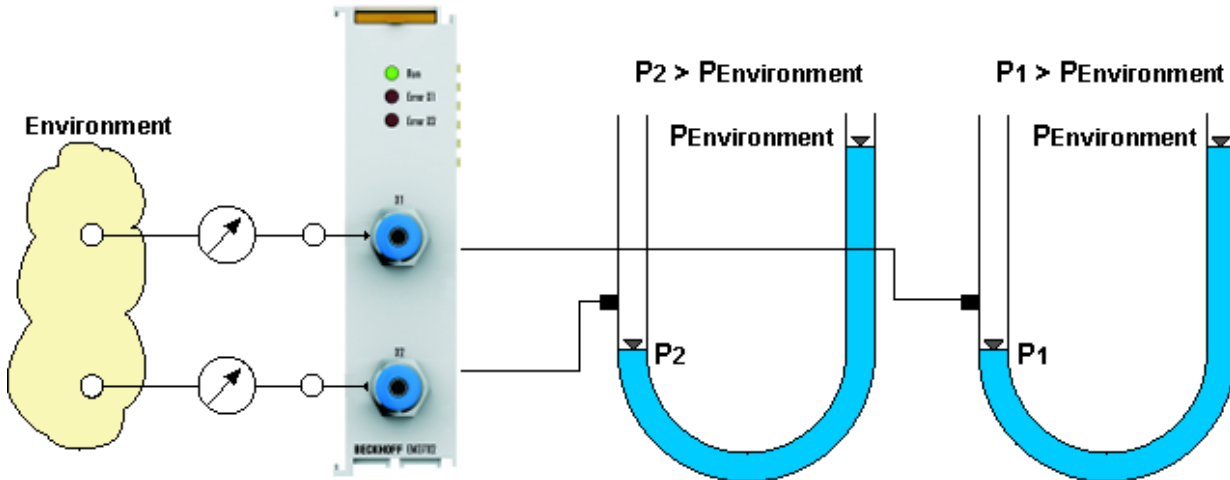
Fig. 133: EM3702 two-channel relative pressure measurement

The EM3702 terminal module can measure pressure values between 0 and 7500 hPa (0 and 7.5 bar) at each hose connection. The pressure is measured relative to the ambient air pressure. [▶ 114]

EM3712 - Dual channel relative pressure measuring module (-1000 hPa to +1000 hPa)
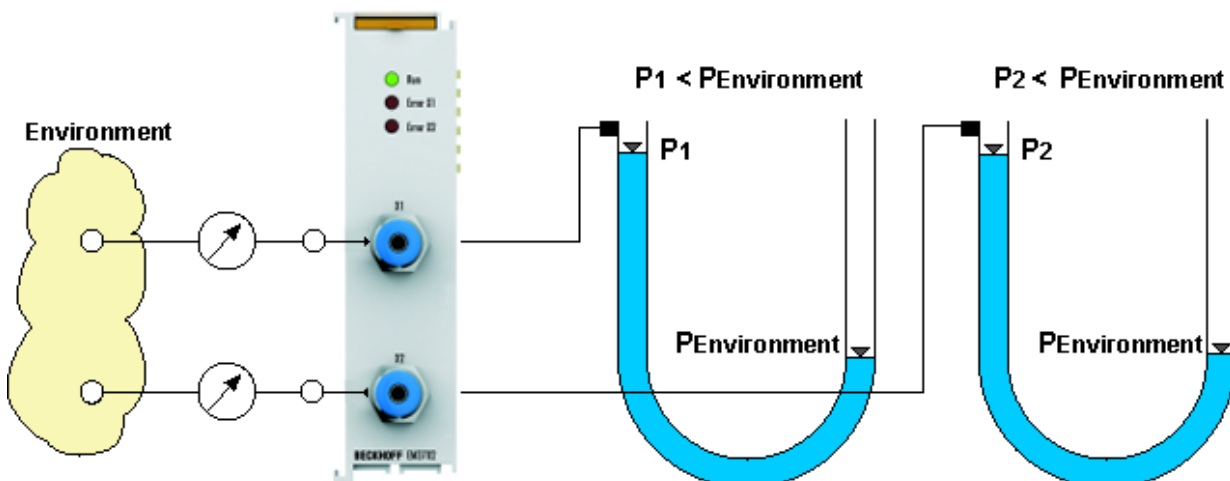
Fig. 134: EM3712 dual channel relative pressure measurement

The EM3712 terminal module can measure pressure values between -1000 and +1000 hPa (-1 and +1 bar) at each hose connection. The pressure is measured relative to the ambient air pressure. [▶ 114]

Installation and connection technique

The pressure measuring terminal samples the pressure values directly. Additional measuring instruments are unnecessary. This avoids connection systems and saves available space compared with the use of conventional measuring instruments. The installation of the pressure measuring terminals is simple and fast and can be carried out without any additional mounting tools. The measuring hoses are connected directly to the quick couplings of the pressure measuring terminal. Standard commercial plastic hoses can be used as measuring hoses. In terms of connections and installation space the pressure measuring terminals are designed like a "normal" 24 mm Bus Terminal and can be installed with minimum space requirement directly in the Bus Terminal system.

6.2 Application examples

The pressure measuring modules can be used in any applications requiring logging and monitoring of differential and dynamic pressures in non-aggressive gases, e.g. pressure vessels, pressure cabins or pneumatic, filter, suction, packaging and positioning systems.

Notice **Media compatibility: Use clean and dry gas!**

They can be used for measuring operating pressures, monitoring filters and screens, checking the tightness of containers, assisting in position verification of components and liquid level monitoring. If flow rates are calculated from the measured pressures, then the pressure measuring terminals can also be used for flow measurement. They can therefore be applied in areas such as process engineering, systems engineering, building services and heating, ventilating and air conditioning.

6.2.1 EM3701 - application examples

Tank systems
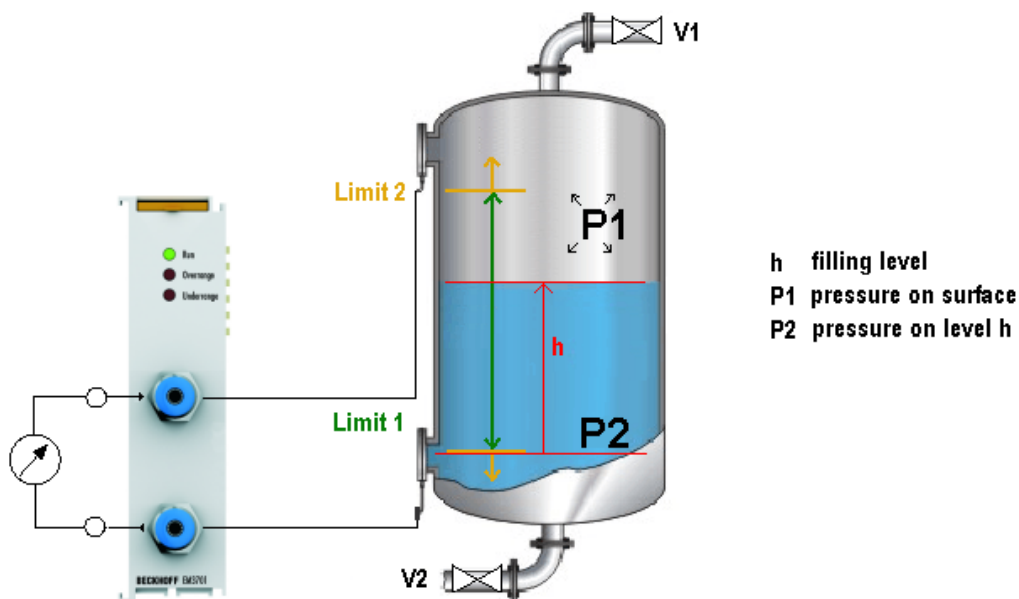
Fig. 135: EM3701 application example: filling level of tanks

The filling level of a tank system can be checked using a differential pressure measurement. In sealed tanks the gas phase exerts an additional pressure (P1) on the liquid in the tank. This effect is compensated by differential pressure measurement. Based on a tank linearization table, the filling level can be calculated in the PLC and displayed. The level can be controlled by introducing lower and upper limit values.

- Measuring the pressure difference as an indicator for the filling level
- Controlling the filling level of tanks and topping up if the level falls below a level defined by limit 1 (lower limit) or draining of a limit 2 (upper limit) is exceeded.
- Determining and displaying the filling level via PLC
- No additional pressure gauges, switches and associated connection equipment are required

Measurement result P = (P2 – P1)	Limit 1 (Index 0x6000:03)	Limit 2 (Index 0x6000:05)	Message	Meaning
Limit1 < P < Limit2	1	2	Status message (green)	Filling level OK
P < limit1 < limit2	2	2	Warning (yellow)	Level is below the minimum filling level (limit 1)
P > limit2 > limit1	1	1	Warning (yellow)	Level is above the maximum filling level (limit 2)

Filter systems, pipe constrictions
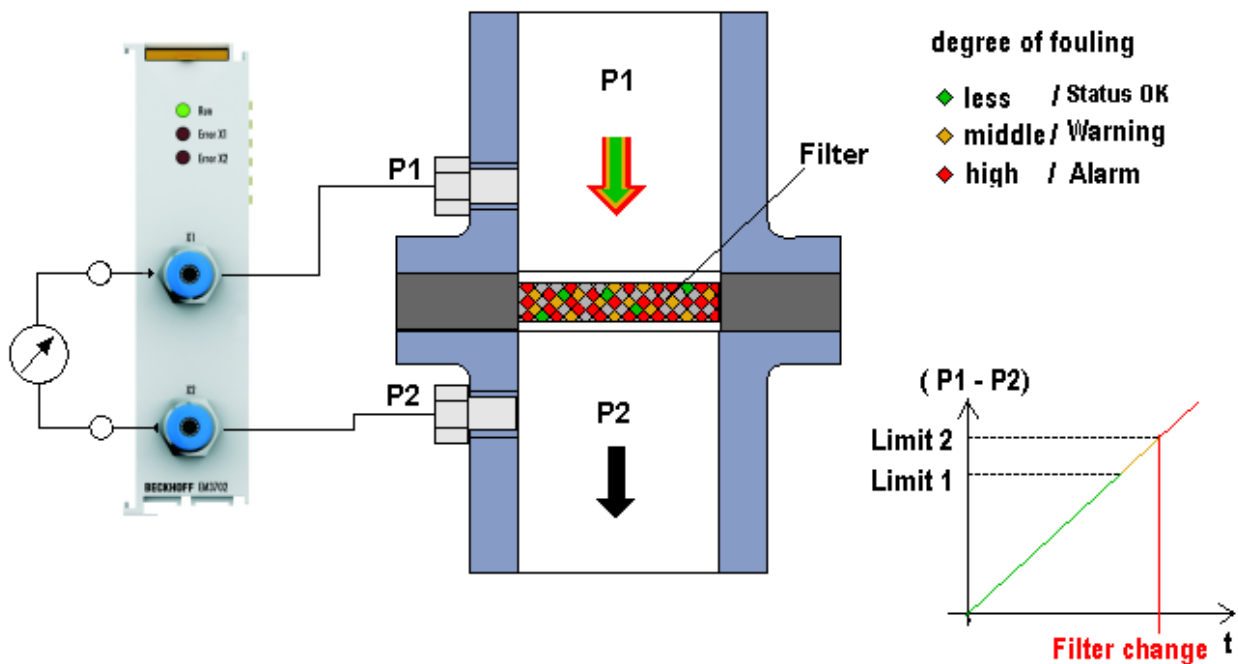
Fig. 136: EM3701 application example: filter systems, pipe restrictions

The pressure difference of the transport medium is the driving force in a filtration. The difference between the inlet pressure P1 and the outlet pressure P2 increases with increasing soiling of the filter. Increased soiling results in increasing energy costs and increasing probability of failure. The filter condition should therefore be clearly displayed at all times.

- Measuring the pressure difference as an indicator for the degree of soiling
- Assessing the degree of soiling via the PLC
- Monitoring the filter status via limit 1 (warning) and limit 2 (alarm)
- Issuing of messages

Measurement result P = (P2 - P1)	Limit 1 (Index 0x6000:03)	Limit 2 (Index 0x6000:05)	Message	Meaning
P < limit 1 < limit 2	2	2	Status message (green)	Degree of filter soiling OK
Limit 1 < P < limit 2	1	2	Warning (yellow)	Increased degree of soiling, filter should be changed.
P > limit 2 > limit 1	1	1	Alarm (red)	Change filter

6.2.2   EM3702 - application example: pneumatic systems
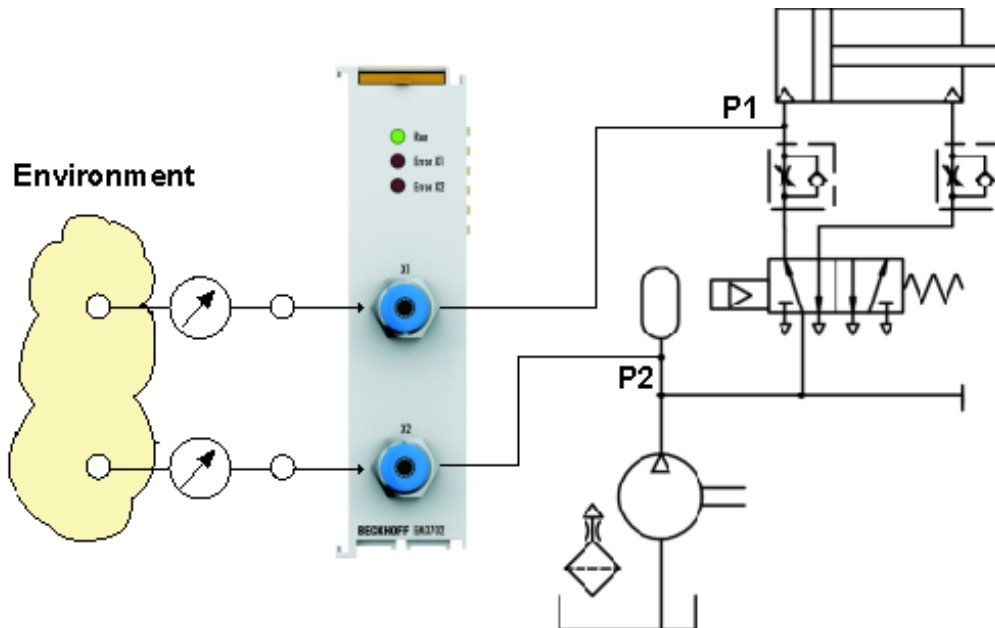
Fig. 137: EM3702 application example: pneumatic systems

- Checking the filling level of stores
- Monitoring the operating pressure of equipment
- enables monitoring and avoidance of overpressure

6.2.3   EM3712 - Application example Packaging plant for eggs
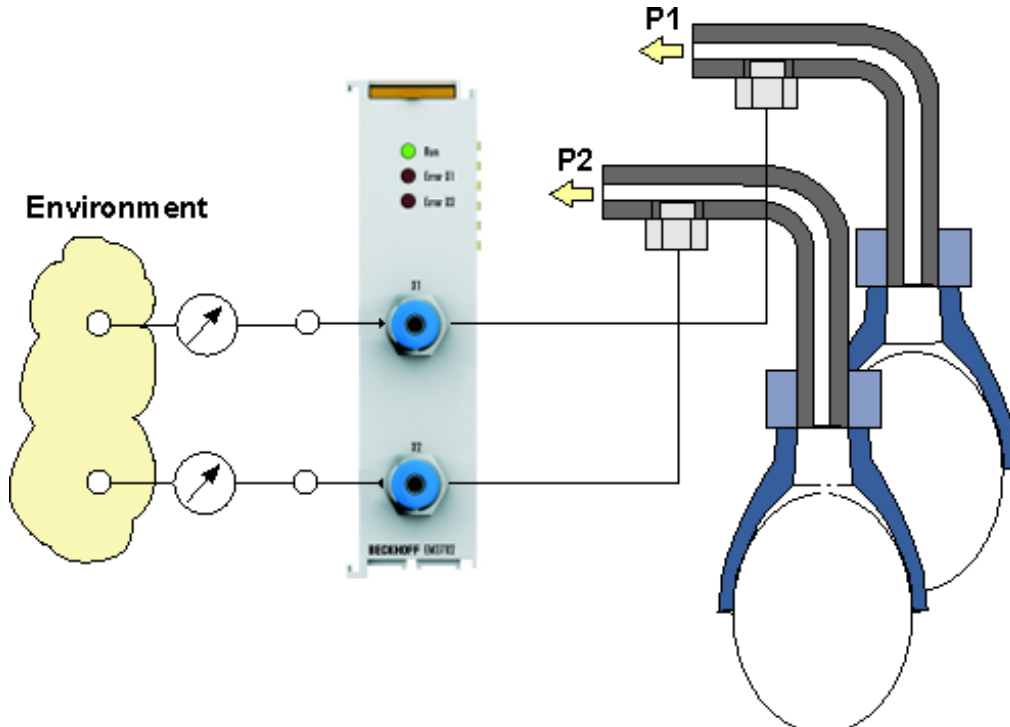
Fig. 138: EM3712 Application example Packaging plant for eggs

- Monitoring of suction
- Pressure deviations indicate leakages or positioning inaccuracies

6.3 Process data and settings

6.3.1 Parameterization

In the TwinCAT System Manager an EM37xx is parameterized via two dialog windows. Dialog box (A) shows the process data that can be parameterized based on the CoE directory (B).
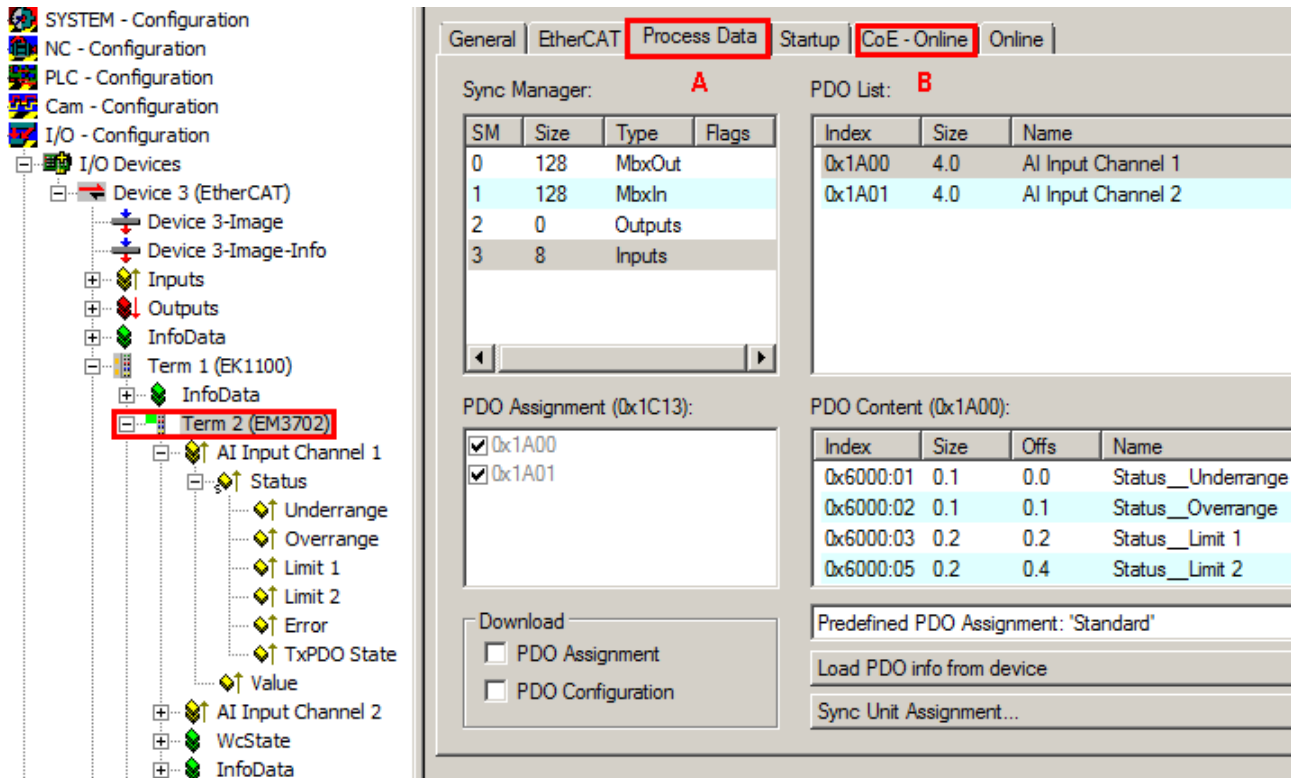
Fig. 139: Parameterization EM37xx

- Changes to the process data-specific settings are generally only effective after a restart of the EtherCAT master: Restart TwinCAT in RUN or CONFIG mode; RELOAD in CONFIG mode
- Changes to the online CoE directory
  - are in general immediately effective
  - are in general stored non-volatile only in the terminal/in the slave and should therefore be entered in the CoE StartUp list. This list is processed at each EtherCAT start and the settings are loaded into the slave.

6.3.2 Process data

The EM37xx terminals offer the following process data per analog channel for transmission:

Value	16 (bit) value, after passing through the process data chain resolution: EM3701: 0.1 mbar EM3702, EM3712: 1 mbar
Status	16 (bit) status information

Using the EM3702 (2 channels: 0 – 7.5 bar) as an example, the settings are explained below. The data apply to TwinCAT 2.11 from build 1544 onward and XML revision from EM37xx-0000-0016 onward.

Interpretation of value & status variable

Fig. 140: EM37xx default process data

The plain text display of the bit meanings of the Status word is particularly helpful not only in commissioning but also for linking to the PLC program.
By right-clicking on the Status variable in the configuration tree (A), the structure can be opened for linking (B).

In order to be able to read the it meanings in plain text in the online display (C), use the button *"Show Sub Variables"* (D) to display all subvariables and the structure content of the status word.

Display of the analog values for EM3701

The terminal module displays the analog input values as follows:

Differential pressure	Decimal [0.1 mbar]	Hexadecimal [0.1 mbar]
-100 mbar	-1000	0xFC18
0 mbar	0	0x0000
+100 mbar	+1000	0x03E8

Display of the analog values for EM3702

The terminal module displays the analog input values as follows:

Print	Decimal [1 mbar]	Hexadecimal [1 mbar]
0 mbar	0	0x0000
7500 mbar	7500	0x1D4C

Display of the analog values for EM3712

The terminal module displays the analog input values as follows:

Print	Decimal [1 mbar]	Hexadecimal [1 mbar]
-1000 mbar	-1000	0xFC18
0 mbar	0	0x0000
1000 mbar	1000	0x03E8

Left-clicking on Value (A) results in graphic display of the online value (C) and the corresponding unit (B) in the Online tab.
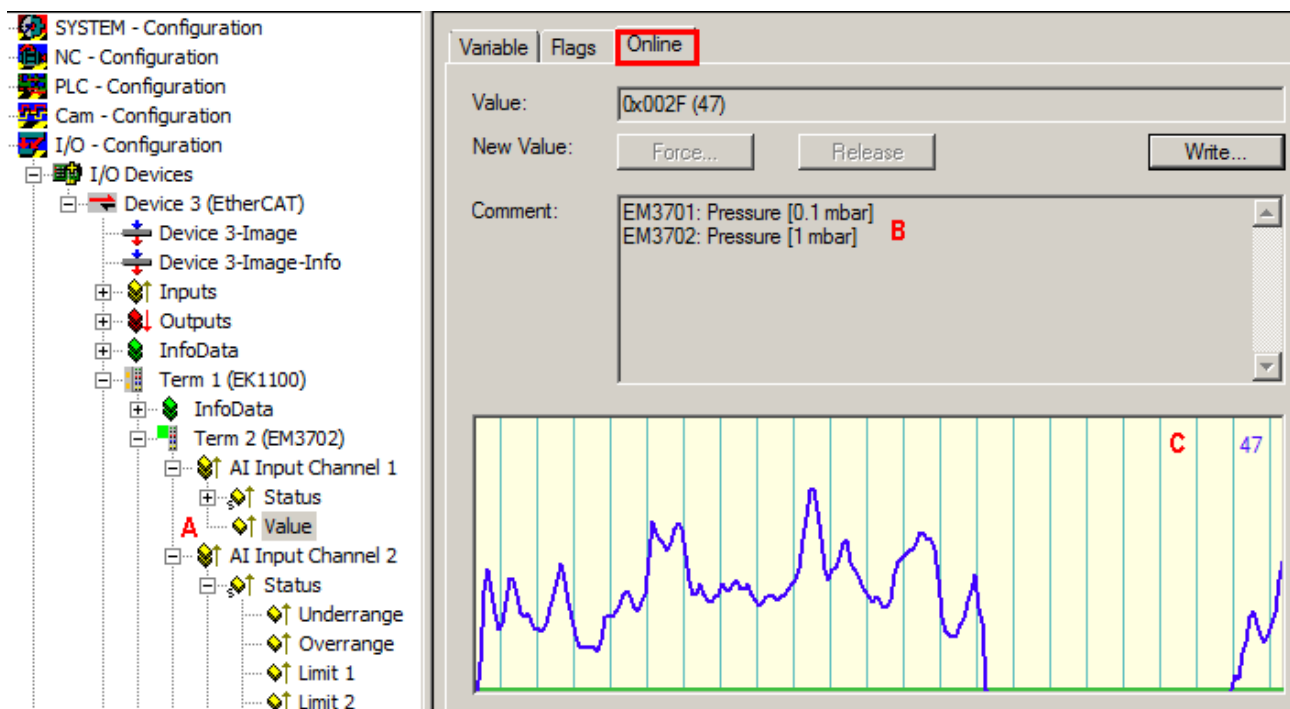
Fig. 141: Display of the value for EM3702 Ch1

6.3.3   Data stream

The flow chart below illustrates the data stream of the EM37xx.

The raw value of the sensor can be read in object 0x80nE:01 [▶ 130].

After completion of the process chain, the "value" index: 0x60n0:11 [▶ 130] is output to the controller.
The individual objects of the process are described below.
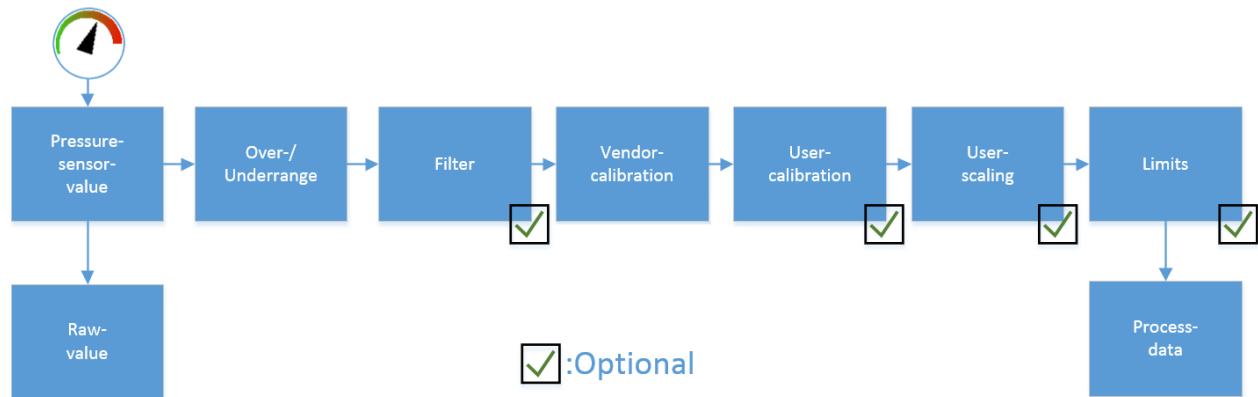
Fig. 142: EM37xx data stream

Undershoot and overshoot of the measuring range (underrange, overrange), index 0x60n0:01, 0x60n0:02 [▶ 130]

The limits refer to the raw value of the sensor.

Terminal module	Overrange / Underrange		PDO index	Error bit index	LED
EM3701	Nominal pressure range	-100 mbar < (P1-P2) < 100 mbar	(0x6000:01 and 0x6000:02): 0	0x60n0:07: 0	Overrange and underrange (red): off
	Underrange	(P1-P2) < -100 mbar	0x6000:01: 1	0x60n0:07: 1	Overrange (red): on
	Overrange	(P1-P2) > 100 mbar	0x6000:02: 1	0x60n0:07: 1	Underrange (red): on
EM3702	Nominal pressure range	0 mbar < Pn < 7500 mbar	(0x60n0:01 and 0x60n0:02): 0	0x60n0:07: 0	Error Xn (red): off
	Underrange	Pn < -100 mbar	0x60n0:01: 1	0x60n0:07: 1	Error Xn (red): on
	Overrange	Pn > 7500 mbar	0x60n0:02: 1	0x60n0:07: 1	Error Xn (red): on
EM3712	Nominal pressure range	-1000 mbar < Pn < +1000 mbar	(0x60n0:01 and 0x60n0:02): 0	0x60n0:07: 0	Error Xn (red): off
	Underrange	Pn < -1000 mbar	0x60n0:01: 1	0x60n0:07: 1	Error Xn (red): on
	Overrange	Pn > +1000 mbar	0x60n0:02: 1	0x60n0:07: 1	Error Xn (red): on

---

NOTICE

Measuring range exceeded

Pressure values below or above the permitted measuring range may lead to destruction of the pressure measuring unit. The limits must be observed.

The error LED lights up if the error bit is set.

Error bit (index 0x60n0:07)

The error bit indicates an overrange or underrange. Terminal EM3702 (0 to 7500 mbar) shows an overrange or an underrange of -100 mbar.

6.3.4   Settings

6.3.4.1   IIR Filter

The EM37xx terminals are equipped with a digital filter that has the characteristics of an infinite impulse response filter (IIR filter).

The filter of IIR characteristics is a time-discrete, linear, time-invariant filter that can be set to 8 levels via index 0x8000:15 (level 1 = slightly recursive filter, up to level 8 = strongly recursive filter).

The IIR can be understood to be a moving average value calculation after a low-pass filter.

Due to the FreeRun synchronization mode, the IIR filter operates with an internal cycle time of 500 µs.

The EM37xx filter is activated or deactivated via CoE index 0x8000:06 [▶ 129] (default: 1).

● **The filter characteristics are set via index 0x8000:15 [▶ 129]**

ⅰ The filter frequencies are set for *all* channels of the EM37xx terminals centrally via index 0x8000:15 (channel 1).

The corresponding indices 0x80n0:15 of the other channels have no parameterization function!
The latest firmware version (see status table [▶ 136]) returns an EtherCAT-compliant error message, if the filter characteristics of other channels (index 0x80n0:06, 0x80n0:15) are set.

Filter data for IIR filter

IIR filter	Input value	Limit frequency with internal terminal cycle time 1 ms (-3 dB)
IIR1	2	168 Hz
IIR2	3	88 Hz
IIR3	4	43 Hz
IIR4	5	21 Hz
IIR5	6	10.5 Hz
IIR6	7	5.2 Hz
IIR7	8	2.5 Hz
IIR8	9	1.2 Hz

6.3.4.2   Calibration and scaling

Vendor calibration, <u>index 0x80n0:0B [▶ 129]</u>

The vendor calibration is enabled via index 0x80n0:0B. The parameterization takes place via the indices:

Index (hex)	Meaning
<u>0x80nF:01 [▶ 129]</u>	Offset (vendor calibration)
<u>0x80nF:02 [▶ 129]</u>	Gain (vendor calibration)

● **Vendor calibration**

i The vendor reserves the authority for the basic calibration of the terminals. Therefore, the vendor calibration cannot be changed.

User calibration, <u>index 0x80n0:17, 0x80n0:18 [▶ 129]</u>

With the aid of user calibration, the measuring module can be adjusted to the ambient conditions of the measuring site. The calculation formulas can be found in section: <u>Calculation of process data [▶ 127]</u>.

The user calibration is enabled via <u>index 0x80n0:0A [▶ 129]</u>.

The parameterization takes place via the indices <u>0x80n0:17 and 0x80n0:18 [▶ 129]</u>.

Index (hex)	Meaning
<u>0x80n0:17 [▶ 129]</u>	Offset (user calibration)
<u>0x80n0:18 [▶ 129]</u>	Gain (user calibration)

User scaling, <u>index 0x80n0:01 [▶ 129]</u>

The scaling establishes the relationship between the input value and the display value of the sensor. The calculation formulas can be found in section: <u>Calculation of process data [▶ 127]</u>.

- The user scaling is enabled via index 0x80n0:01.
- The scaling can be adjusted to measurement application by parameterizing indices <u>0x80n0:11 and 0x80n0:12 [▶ 129]</u>.

Index (hex)	Meaning
<u>0x80n0:11 [▶ 129]</u>	Offset (user scaling)
<u>0x80n0:12 [▶ 129]</u>	Gain (user scaling)

6.3.4.3 Limit value monitoring (Limit 1 and Limit 2)

Limit 1 and limit 2, <u>index 0x80n0:13, index 0x80n0:14 [▶ 129]</u>

For each channel two limits available for limit value monitoring within the measuring range.
In this way it is possible to generate different warning levels or a traffic light function (<u>see application examples [▶ 117]</u>).

- Activation of limit value monitoring via indices <u>0x80n0:07 [▶ 129]</u> (Limit 1) or <u>0x80n0:08 [▶ 129]</u> (Limit 2).

- Entering the values for limit 1 <u>(index 0x80n0:13) [▶ 129]</u> and limit 2 <u>(index 0x80n0:14). [▶ 129]</u>

- The bits in indices <u>0x60n0:03 [▶ 130]</u> (limit 1) and <u>0x60n0:05 [▶ 130]</u> (limit 2) are set when the value exceeds or falls below the defined limits.
  Output Limit n (2-bit):
  - 0: not active
  - 1: Value is smaller than the limit value
  - 2: Value is larger than the limit value
  - 3: Value is equal to the limit value

6.3.5 Calculation of process data

The terminal records measured values continuously and places the raw value of its A/D converter into the ADC raw value object 0x80nE:01. The calculation of the correction with the vendor calibration values takes place after each acquisition of the analog signal. This is followed by (optional) user calibration, user scaling and evaluation of the limits (see diagram *Calculation of process data*)
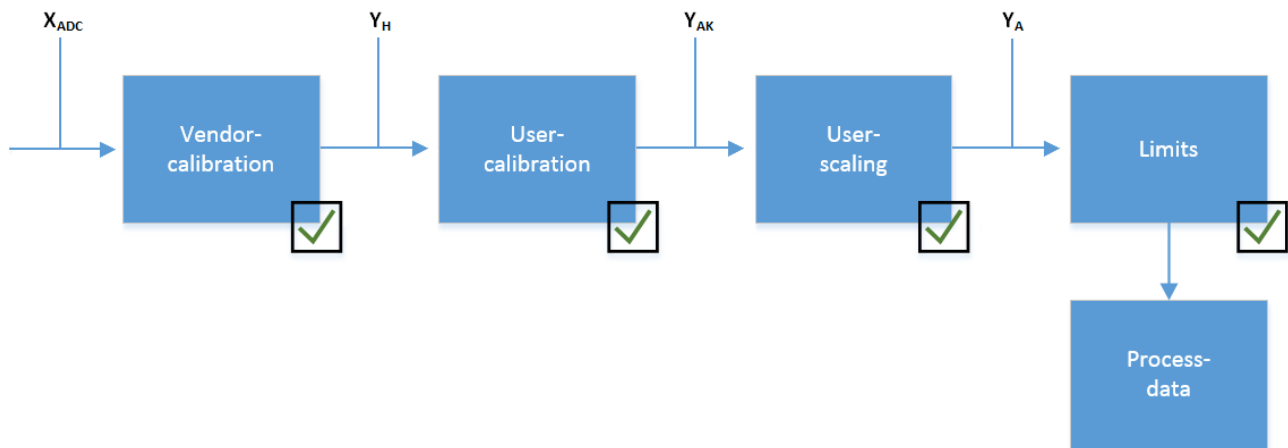
Fig. 143: Calculation of process data

Calculation	Designation
X_{ADC}	Output of the A/D converter
$Y_H = (X_{ADC} - B_K) \times A_K \times 2^{-14}$ $Y_H = (X_{ADC} - 0) \times 2^{14} \times 2^{-14}$ $Y_H = X_{ADC}$	Measured value after vendor calibration (corresponds to the raw value of the A/D converter)
$Y_{AK} = (Y_H - B_{AK}) \times A_{AK} \times 2^{-14}$ $Y_{AK} = (X_{ADC} - B_{AK}) \times 2^{14} \times 2^{-14}$ $Y_{AK} = X_{ADC,}$ if index 0x80n0:0A (enable user calibration)	Measured value after vendor and user calibration
$Y_A = Y_{AK} \times A_W \times 2^{-16} + B_W$ $Y_A = Y_{AK}$, if index 0x80n0:01 (enable user scale) inactive $Y_A = X_{ADC,}$ if index 0x80n0:01 and index 0x80n0:0A inactive	Measured value following user scaling

Legend

Name	Designation	Index
X_{ADC}	Output value of the A/D converter	0x80nE:01 [▶ 130]
B_K	Vendor calibration offset (cannot be changed)	0x80nF:01 [▶ 129]
A_K	Vendor calibration gain (cannot be changed)	0x80nF:02 [▶ 129]
B_{AK}	User calibration offset (can be activated via index 0x80n0:0A)	0x80n0:17 [▶ 129]
A_{AK}	User calibration gain (can be activated via index 0x80n0:0A)	0x80n:018 [▶ 129]
B_W	User scaling offset (can be activated via index 0x80n0:01)	0x80n0:11 [▶ 129]
A_W	User scaling gain (can be activated via index 0x80n0:01)	0x80n0:12 [▶ 129]
Y_A	Process data for controller	-

6.4 Object description and parameterization

ℹ **EtherCAT XML Device Description**

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

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

The EtherCAT device is parameterized via the CoE-Online tab [▶ 94] (double-click on the respective object) or via the Process Data tab [▶ 91](allocation of PDOs). Please note the following general CoE notes [▶ 23] when using/manipulating the CoE parameters:

• Keep a startup list if components have to be replaced

• Differentiation between online/offline dictionary, existence of current XML description

• use "CoE reload [▶ 149]" for resetting changes

6.4.1 Restore object

Index 1011 Restore default parameters

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

6.4.2 Configuration data

Index 80n0 AI settings (for Ch.1: n = 0; Ch.2: n = 1)

Index (hex)	Name	Meaning	Format	Flags	Default value
80n0:0	AI Settings	Maximum subindex	UINT8	RO	0x18 (24_{dec})
80n0:01	Enable user scale	Activates scaling	BOOLEAN	RW	0x00 (0_{dec})
80n0:06	Enable filter	Activates filter (0x80n0:15), eliminates the need for data exchange synchronized with PLC cycle	BOOLEAN	RW	0x01 (1_{dec})
80n0:07	Enable limit 1	Activates limit 1	BOOLEAN	RW	0x00 (0_{dec})
80n0:08	Enable limit 2	Activates limit 2	BOOLEAN	RW	0x00 (0_{dec})
80n0:0A	Enable user calibration	Enables user calibration	BOOLEAN	RW	0x00 (0_{dec})
80n0:0B	Enable vendor calibration	Enable manufacturer calibration	BOOLEAN	RW	0x01 (1_{dec})
80n0:11	User scale offset	User scaling offset	INT16	RW	0x00 (0_{dec})
80n0:12	User scale gain	User scaling gain. The gain is represented in fixed-point format, with the factor 2^{-16}. The value 1 corresponds to 65535_{dec} ($0x00010000_{hex}$) and is limited to +/-0x7FFFF	INT32	RW	0x00010000 (65536_{dec})
80n0:13	Limit 1	First limit value for setting the status bit	INT16	RW	0x00 (0_{dec})
80n0:14	Limit 2	Second limit value for setting the status bit	INT16	RW	0x00 (0_{dec})
80n0:15	Filter settings	Permitted values: 2 — IIR1 3 — IIR2 4 — IIR3 5 — IIR4 6 — IIR5 7 — IIR6 8 — IIR7 9 — IIR8	UINT16	RW	0x0002 (2_{dec})
80n0:17	User calibration offset	User offset calibration	INT16	RW	0x0000 (0_{dec})
80n0:18	User calibration gain	User gain compensation	INT16	RW	0x4000 (16384_{dec})

6.4.3 Configuration data (vendor-specific)

Index 80nF AI Vendor data

Index (hex)	Name	Meaning	Format	Flags	Default value
80nF:0	AI vendor data	Maximum subindex	UINT8	RO	0x02 (2_{dec})
80nF:01	Calibration offset	Offset (vendor calibration)	INT16	RW	0x0000 (0_{dec})
80nF:02	Calibration gain	Gain (vendor calibration)	INT16	RW	0x4000 (16384_{dec})

6.4.4 Input data

Index 60n0 AI inputs (for Ch.1: n = 0; Ch.2: n = 1)

Index (hex)	Name	Meaning	Data type	Flags	Default
60n0:0	AI inputs	Maximum subindex	UINT8	RO	0x11 (17$_{dec}$)
60n0:01	Underrange	Measurement is below range	BOOLEAN	RO	0x00 (0$_{dec}$)
60n0:02	Overrange	Measuring range exceeded	BOOLEAN	RO	0x00 (0$_{dec}$)
60n0:03	Limit 1	Limit value monitoring Limit 1 0: not active 1: Value is smaller than Limit Value 1 2: Value is larger than Limit Value 1 3: Value is equal to limit value 1	BIT2	RO	0x00 (0$_{dec}$)
60n0:05	Limit 2	Limit value monitoring Limit 2 0: not active 1: Value is smaller than Limit Value 2 2: Value is larger than Limit Value 2 3: Value is equal to limit value 2	BIT2	RO	0x00 (0$_{dec}$)
60n0:07	Error	The error bit is set if the data is invalid (over-range, under-range)	BOOLEAN	RO	0x00 (0$_{dec}$)
60n0:0F	TxPDO State	Validity of the data of the associated TxPDO (0 = valid, 1 = invalid).	BOOLEAN	RO	0x00 (0$_{dec}$)
60n0:11	Value	Pressure value EM3701: Pressure in 0.1 mbar EM3702: Pressure in 1 mbar EM3712: Pressure in 1 mbar			

6.4.5 Command object

Index FB00 command

Index (hex)	Name	Meaning	Data type	Flags	Default
FB00:0	Command	Max. subindex	UINT8	RO	0x03 (3$_{dec}$)
FB00:01	Request	Commands can be sent to the terminal via the request object	OCTET-STRING[2]	RW	{0}
FB00:02	Status	Status of the command currently being executed 0: Command terminated without error and without response 1: Command terminated without error with response 255: Command is being executed	UINT8	RO	0x00 (0$_{dec}$)
FB00:03	Response	Return value, if 0xFB00:02=1	OCTET-STRING[6]	RO	{0}

6.4.6 Information / diagnosis data

Index 80nE AI internal data

Index (hex)	Name	Meaning	Format	Flags	Default value
80nE:0	AI internal data	Maximum subindex	UINT8	RO	0x01 (1$_{dec}$)
80nE:01	ADC raw value	ADC raw value	INT16	RO	0x0000 (0$_{dec}$)

6.4.7 Standard objects

Index 1000 Device type

Index (hex)	Name	Meaning	Format	Flags	Default value
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	0x012C1389 (19665801$_{dec}$)

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EM3701 EM3702 EM3712

Index 1009 Hardware version

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

Index 100A Software version

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

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Information for identifying the slave	UINT8	RO	0x04 (4_{dec})
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2_{dec})
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	EM3701: 0x0E753452 (242562130_{dec}) EM3702: 0x0E763452 (242627666_{dec}) EM3712: 0x0E803452 (243283026_{dec})
1018:03	Revision	Revision number of the EtherCAT slave: • The low word (bit 0-15) indicates the special terminal number. • The high word (bit 16-31) refers to the device description	UINT32	RO	0x00000000 (0_{dec})

1018:04	Serial number	Serial number of the EtherCAT slave:
		• Low word
		  ◦ The low byte (bit 0-7) of the low word contains the year of production.
		  ◦ The high byte (bit 8-15) of the low word contains the week of production.
		• The high word (bit 16-31) is 0
			UINT32	RO	0x00000000 (0_{dec})

Index 10F0 Backup parameter handling

Index (hex)	Name	Meaning	Data type	Flags	Default
10F0:0	Backup parameter handling	Information for standardized loading and saving of backup entries	UINT8	RO	0x01 (1_{dec})
10F0:01	Checksum	Checksum across all backup entries of the EtherCAT slave	UINT32	RO	0x00000000 (0_{dec})

Index 1A0n AI TxPDO-Map Inputs (for Ch.1, n = 0; Ch.2, n = 1)

Index (hex)	Name	Meaning	Format	Flags	Default value
1A0n:0	AI TxPDO-Map InputsCh.1; Ch2	PDO Mapping TxPDO 1; PDO Mapping TxPDO 2	UINT8	RO	0x09 (9$_{dec}$)
1A0n:01	SubIndex 001	1. PDO Mapping entry (object 0x60n0 (AI Inputs), entry 0x01 (Underrange))	UINT32	RO	0x60n0:01, 1
1A0n:02	SubIndex 002	2. PDO Mapping entry (object 0x60n0 (AI Inputs), entry 0x02 (Overrange))	UINT32	RO	0x60n0:02, 1
1A0n:03	SubIndex 003	3. PDO Mapping entry (object 0x60n0 (AI Inputs), entry 0x03 (Limit 1))	UINT32	RO	0x60n0:03, 2
1A0n:04	SubIndex 004	4. PDO Mapping entry (object 0x60n0 (AI Inputs), entry 0x05 (Limit 2))	UINT32	RO	0x60n0:05, 2
1A0n:05	SubIndex 005	5. PDO Mapping entry (object 0x60n0 (AI Inputs), entry 0x01 (Underrange))	UINT32	RO	0x60n0:07, 1
1A0n:06	SubIndex 006	6. PDO Mapping entry (7 bits align)	UINT32	RO	0x0000:00, 7
1A0n:07	SubIndex 007	7. PDO Mapping entry (object 0x60n0 (AI Inputs), entry 0x0F (TxPDO State))	UINT32	RO	0x60n0:0F, 1
1A0n:08	SubIndex 008	8. PDO Mapping entry (1 bit align)	UINT32	RO	0x0000:00, 1
1A0n:09	SubIndex 009	1. PDO Mapping entry (object 0x60n0 (AI Inputs), entry 0x11 (Value))	UINT32	RO	0x60n0:11, 16

Index 1C00 Sync manager type

Index (hex)	Name	Meaning	Format	Flags	Default value
1C00:0	Sync manager type	Using the sync managers	UINT8	RO	0x04 (4$_{dec}$)
1C00:01	SubIndex 001	Sync-Manager Type Channel 1: Mailbox Write	UINT8	RO	0x01 (1$_{dec}$)
1C00:02	SubIndex 002	Sync-Manager Type Channel 2: Mailbox Read	UINT8	RO	0x02 (2$_{dec}$)
1C00:03	SubIndex 003	Sync-Manager Type Channel 3: Process Data Write (Outputs)	UINT8	RO	0x03 (3$_{dec}$)
1C00:04	SubIndex 004	Sync-Manager Type Channel 4: Process Data Read (Inputs)	UINT8	RO	0x04 (4$_{dec}$)

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Format	Flags	Default value
1C12:0	RxPDO assign	PDO assign outputs	UINT8	RO	0x00 (0$_{dec}$)

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Format	Flags	Default value
1C13:0	TxPDO assign	PDO assign inputs	UINT8	RO	0x02 (2$_{dec}$)
1C13:0	SubIndex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RO	0x1A00 (6656$_{dec}$)
1C13:0	SubIndex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RO	0x1A01 (6657$_{dec}$)

Index 1C33 SM input parameter

Index (hex)	Name	Meaning	Format	Flags	Default value
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32_{dec})
1C33:01	Sync mode	Current synchronization mode: 0: Free Run	UINT16	RW	0x0000 (0_{dec})
1C33:02	Cycle time	Cycle time (in ns): Free Run: Cycle time of the local timer Synchronous with SM 2 event: Master cycle time DC-Mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x003D0900 (4000000_{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000384 (900_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: Bit 0: free run is supported	UINT16	RO	0x0001 (1_{dec})
1C33:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x00002710 (10000_{dec})
1C33:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:07	Minimum delay time		UINT32	RO	0x00000384 (900_{dec})
1C33:08	Command	With this entry the real required process data provision time can be measured. 0: Measurement of the local cycle time is stopped 1: Measurement of the local cycle time is started The entry 0x1C33:05 is updated with the maximum measured value. If another measurement takes place, the measured value is reset	UINT16	RW	0x0000 (0_{dec})
1C33:09	Maximum delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000384 (900_{dec})
1C33:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C33:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	0x00 (0_{dec})

Index F000 Modular device profile

Index (hex)	Name	Meaning	Data type	Flags	Default
F000:0	Modular device profile	General information for the modular device profile	UINT8	RO	0x02 (2_{dec})
F000:01	Module index distance	Index distance of the objects of the individual channels	UINT16	RO	0x0010 (16_{dec})
F000:02	Maximum number of modules	Number of channels	UINT16	RO	EM3701: 0x0001 (1_{dec}) EM3702, EM3712: 0x0002 (2_{dec})

Index F008 Code word

Index (hex)	Name	Meaning	Format	Flags	Default value
F008:0	Code word	reserved	UINT32	RW	0x00000000 (0_{dec})

Index F010 Module list

Index (hex)	Name	Meaning	Format	Flags	Default value
F010:0	Module list	Maximum subindex	UINT8	RW	0x02 (2_{dec})
F010:01	SubIndex 001	Analog input profile (300)	UINT32	RW	0x0000012C (300_{dec})
F010:02	SubIndex 002	Analog input profile (300)	UINT32	RW	0x0000012C (300_{dec})

6.5 Notices on analog specifications

Beckhoff I/O devices (terminals, box modules, modules) with analog inputs and outputs are characterized by a number of technical characteristic data; refer to the technical data in the respective documents.

Some explanations are given below for the correct interpretation of these characteristic data.

Unless otherwise stated, the explanations apply accordingly to input and output signals.

6.5.1 Full scale value (FSV), output end value

An I/O device with analog input measures over a nominal measuring range, which is limited by an upper and a lower limit (start value and end value), which can usually already be taken from the device designation. The range between both limits is called measuring span and corresponds to the formula (end value - start value). Analogous to pointing devices this is the measuring scale (see IEC 61131) or also the dynamic range.

For analog I/O devices from Beckhoff, the full scale value (FSV) of the respective product (also: reference value) is selected as the largest limit of the nominal measuring range and is given a positive sign. This applies to both symmetrical and asymmetrical measuring spans.
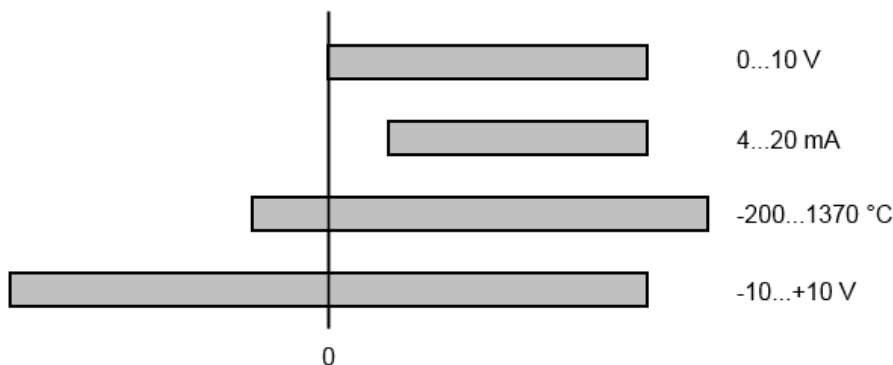

Fig. 144: Full scale value, measuring span

For the above **examples** this means:

- Measuring range 0...10 V: asymmetric unipolar, FSV = 10 V, measuring span = 10 V
- Measuring range 4...20 mA: asymmetric unipolar, FSV = 20 mA, measuring span = 16 mA
- Measuring range -200...1370 °C: asymmetric bipolar, FSV = 1370 °C, measuring span = 1570 °C
- Measuring range -10...+10 V: symmetric bipolar, FSV = 10 V, measuring span = 20 V

Depending on the functionality, an analog input channel may have a technical measuring range that exceeds the nominal measuring range, e.g. to gain more diagnostic information about the signal.
The case-by-case information in the device documentation on the behavior outside the nominal measuring range (measurement uncertainty, display value) must be observed.

The above thoughts are correspondingly applicable to analog output devices:

- The full scale value (FSV) becomes the output end value
- Here, too, there can be a (larger) technical output range in addition to the nominal output range

6.5.2 Measurement error/measurement deviation/measurement uncertainty, output uncertainty

● **Analog output**

The following information also applies analogously to the output end value of analog output devices.

The relative measuring error as a specification value of a Beckhoff analog device is specified in % of the nominal FSV (output end value) and calculated as the quotient of the numerically largest probable deviation from the true measured value (output value) with respect to the FSV (output end value):

$$\text{Measuring error} = \frac{\left| \text{max. deviation} \right|}{\text{full scale value}}$$

It should be noted here that the "true measured value" cannot be determined with infinite accuracy either, but can only be determined via reference devices with a higher expenditure of technology and measuring time and thus a significantly lower measurement uncertainty.

The value therefore describes the result window in which the measured value determined by the device under consideration (Beckhoff analog device) lies with a very high probability in relation to the "true value". Thus, colloquially, this is a "typical" value (typ.); this expresses that the vast statistical majority of values will be within the specification window, but in rare cases there may/will be deviations outside the window.

For this reason, the term "measurement uncertainty" has become established for this window, since "error" is now used to refer to known disturbance effects that can generally be systematically eliminated.

The uncertainty of measurement must always be considered in relation to potential environmental influences:

- invariable electrical channel properties such as temperature sensitivity,
- variable settings of the channel (noise via filters, sampling rate, ...).

Measurement uncertainty specifications without further operational limitation (also called "service error limit") can be assumed as a value "over everything": entire permissible operating temperature range, default setting, etc.

The window is always to be understood as a positive/negative span with "±", even if occasionally indicated as a "half" window without "±".

The maximum deviation can also be specified directly.

Example: measuring range 0...10 V (FSV = 10 V) and measurement uncertainty < ± 0.3%$_{\text{FSV}}$ → the expected maximum usual deviation is ± 30 mV in the permissible operating temperature range.

● **Lower measurement uncertainty possible**

If this specification also includes the temperature drift, a significantly lower measuring error can usually be assumed in case of a constant ambient temperature of the device and thermal stabilization after a user calibration.

7   Appendix

7.1   EtherCAT AL Status Codes

For detailed information please refer to the <u>EtherCAT system description</u>.

7.2   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.

<table>
<tr><td align="center">NOTICE</td></tr>
<tr><td>Risk of damage to the device!
Pay attention to the instructions for firmware updates on the <u>separate page [▶ 137]</u>.
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!</td></tr>
</table>

EM3701			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Release date**
00-02	01	EM3701-0000-0016	2014/12
03-06*	02*		2016/07

EM3702			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Release date**
00-05*	01*	EM3702-0000-0016	2014/12

EM3712			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Release date**
00-02*	01*	EM3712-0000-0016	2016/06

*) 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 <u>documentation</u> is available.

7.3 Firmware Update EL/ES/EM/ELM/EP/EPP/ERPxxxx

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

NOTICE

Only use TwinCAT 3 software!

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

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

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

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

Storage locations

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

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

NOTICE

Application-specific writing of the ESI-EEPROM

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

- Depending on functionality and performance EtherCAT slaves have one or several local controllers for processing I/O data. The corresponding program is the so-called **firmware** in *.efw format.
- In some EtherCAT slaves the EtherCAT communication may also be integrated in these controllers. In this case the controller is usually a so-called **FPGA** chip with *.rbf firmware.

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

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

Simplified update by bundle firmware

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

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

---

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

Following the update, its success should be verified

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

NOTICE

Risk of damage to the device!

✓ Note the following when downloading new device files

a) Firmware downloads to an EtherCAT device must not be interrupted

b) Flawless EtherCAT communication must be ensured. CRC errors or LostFrames must be avoided.

c) The power supply must adequately dimensioned. The signal level must meet the specification.

⇨ In the event of malfunctions during the update process the EtherCAT device may become unusable and require re-commissioning by the manufacturer.

7.3.1 Device description ESI file/XML

NOTICE

Attention regarding update of the ESI description/EEPROM

Some slaves have stored calibration and configuration data from the production in the EEPROM. These are irretrievably overwritten during an update.

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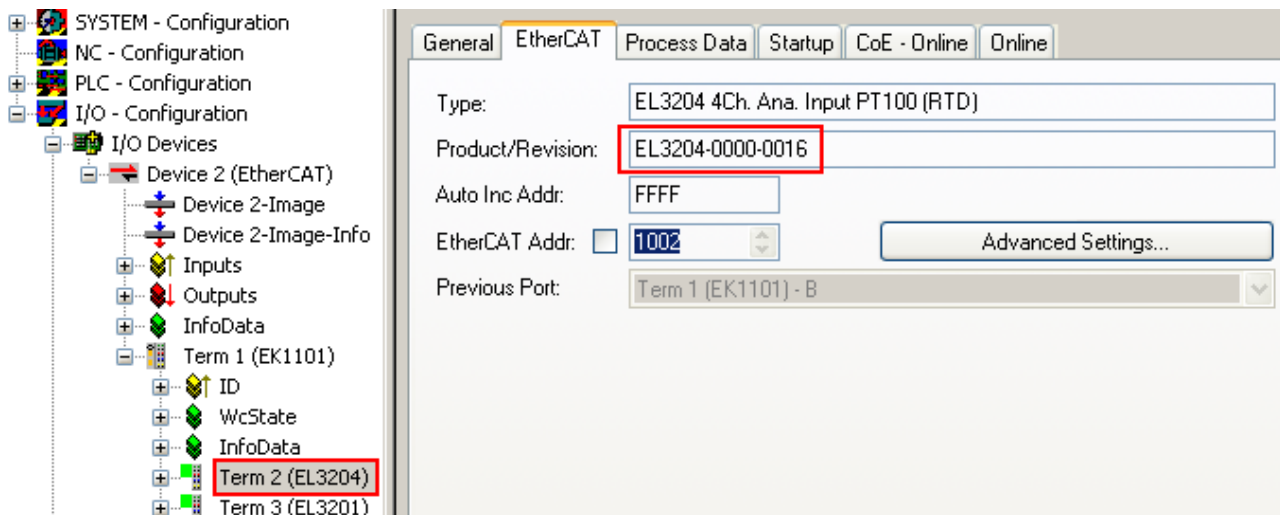

Fig. 145: Device identifier consisting of name EL3204-0000 and revision -0016

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.

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

Display of ESI slave identifier

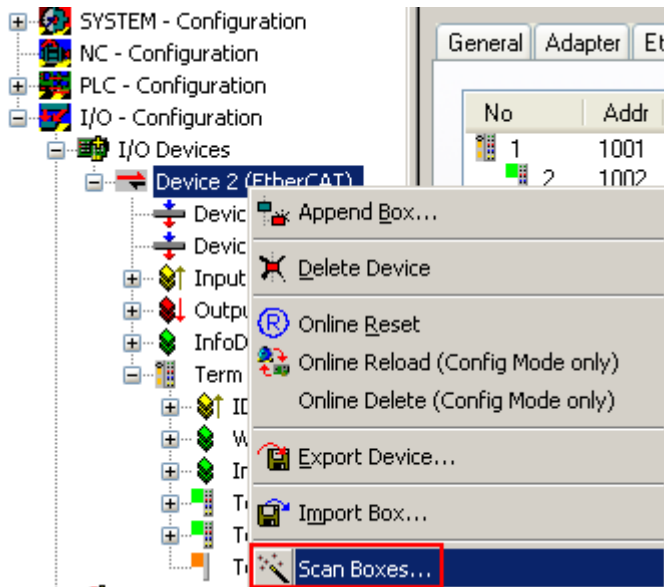The simplest way to ascertain compliance of configured and actual device description is to scan the EtherCAT boxes in TwinCAT mode Config/FreeRun:

Fig. 146: Scan the subordinate field by right-clicking on the EtherCAT device

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

Fig. 147: Configuration is identical

otherwise a change dialog appears for entering the actual data in the configuration.
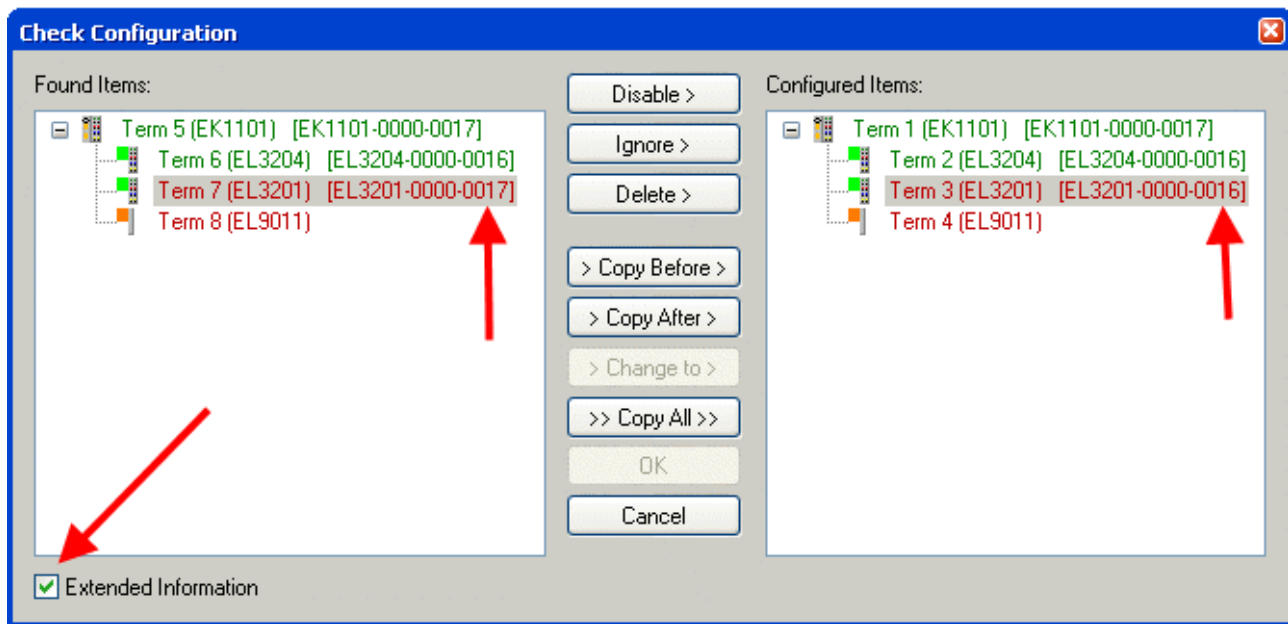
Fig. 148: Change dialog

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*
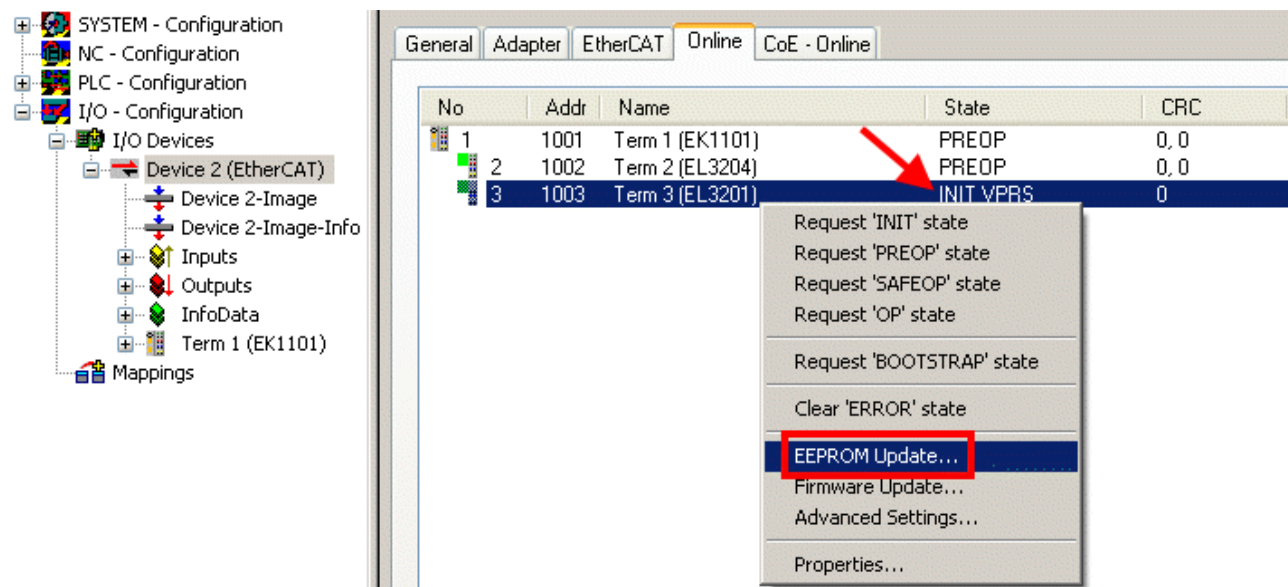
Fig. 149: EEPROM Update

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.
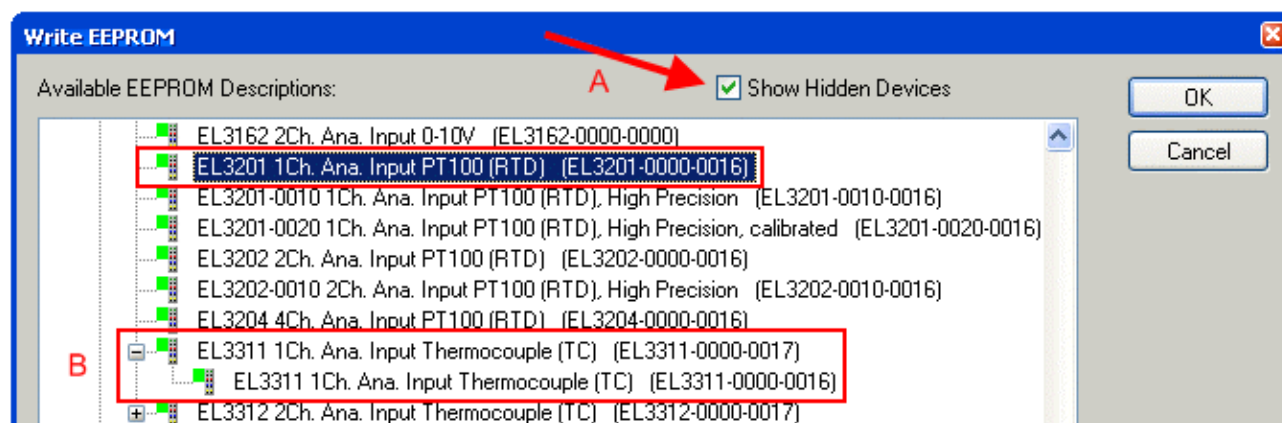
Fig. 150: Selecting the new ESI

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

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

7.3.2   Firmware explanation

Determining the firmware version

Determining the version via the TwinCAT 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).

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

In Fig. *Display of EL3204 firmware version* the firmware version of the selected EL3204 is shown as 03 in CoE entry 0x100A.
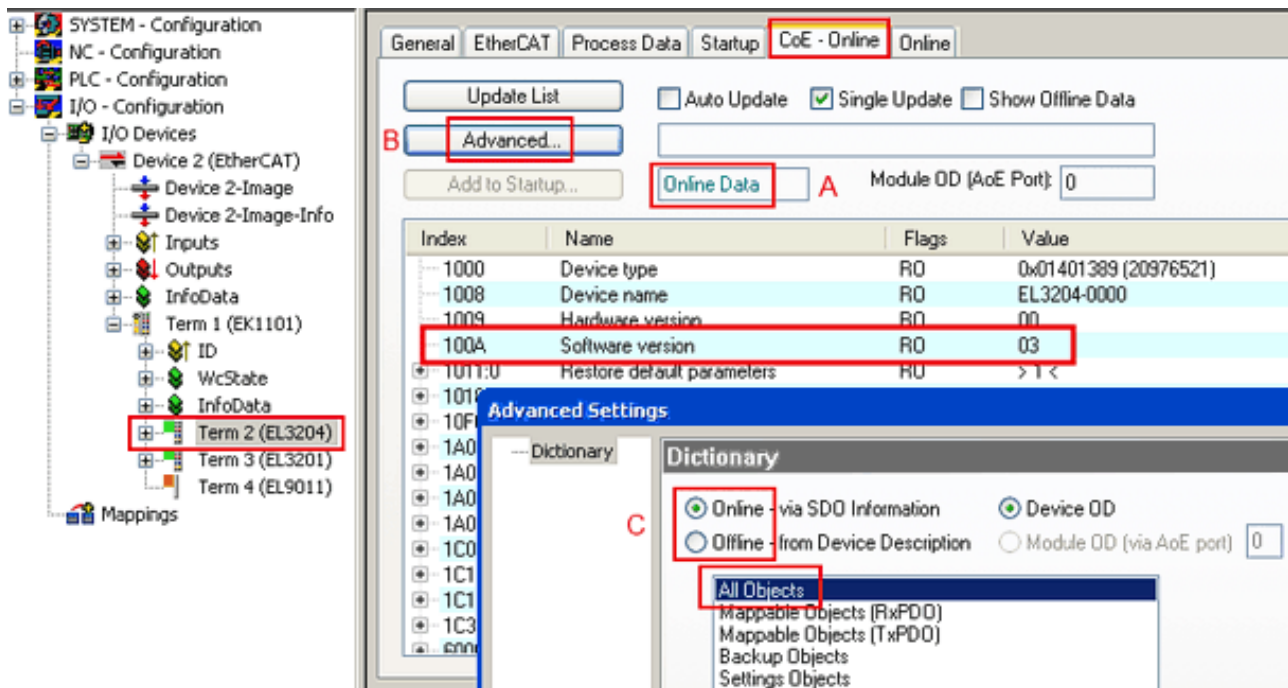
Fig. 151: Display of EL3204 firmware version

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*.

7.3.3   Updating controller firmware *.efw

● CoE directory

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

Switch to the *Online* tab to update the controller firmware of a slave, see Fig. *Firmware Update.*
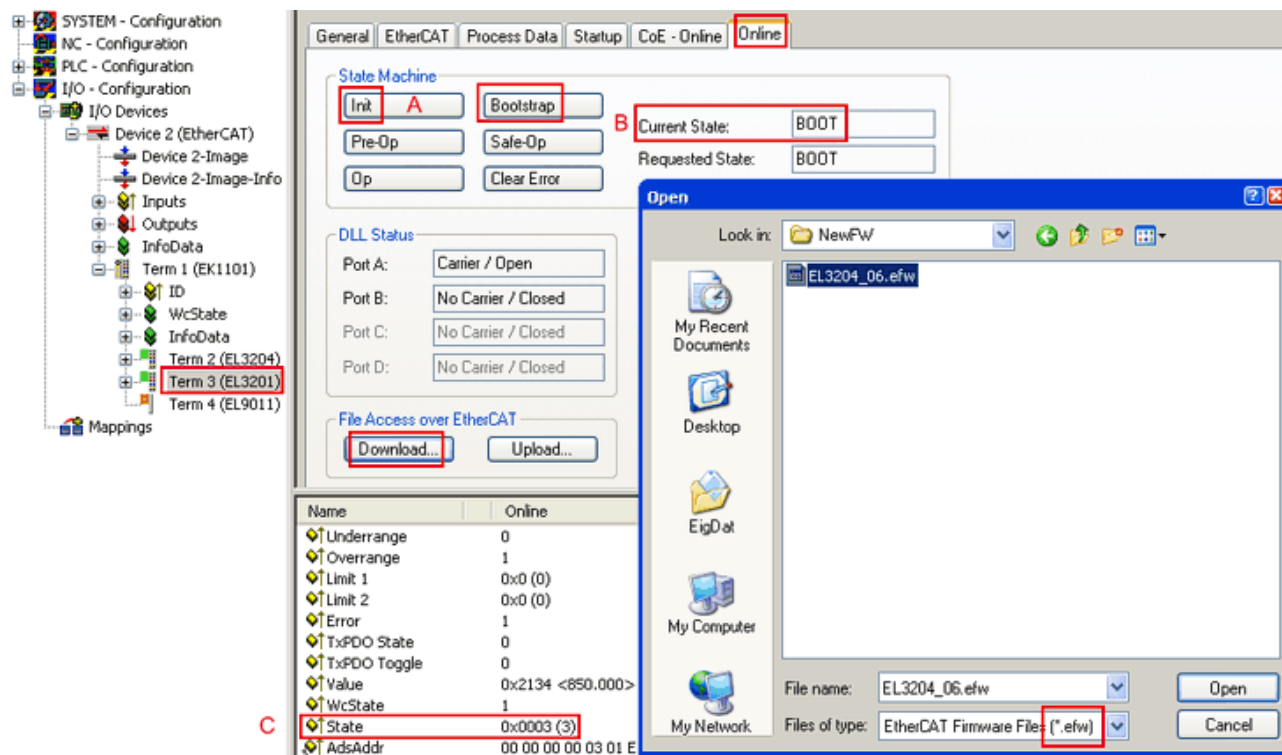
Fig. 152: Firmware Update

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

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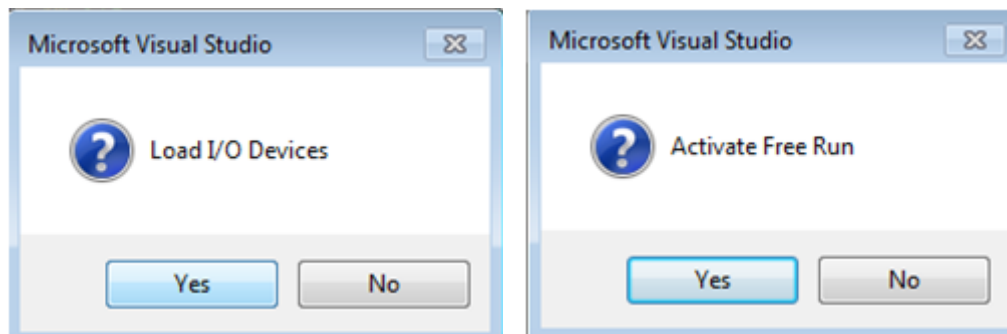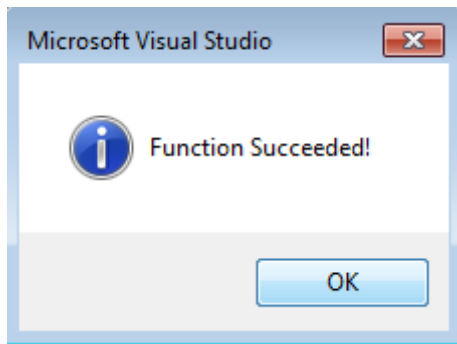

- Switch EtherCAT Master to PreOP
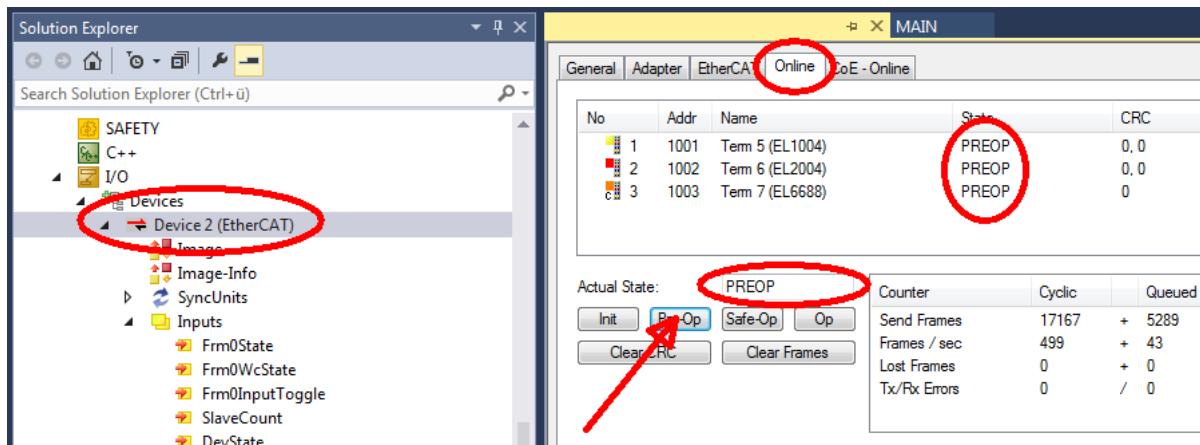
- Switch slave to INIT (A)
- Switch slave to BOOTSTRAP

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

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

7.3.4   FPGA firmware *.rbf

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

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

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

Determining the version via the TwinCAT 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.
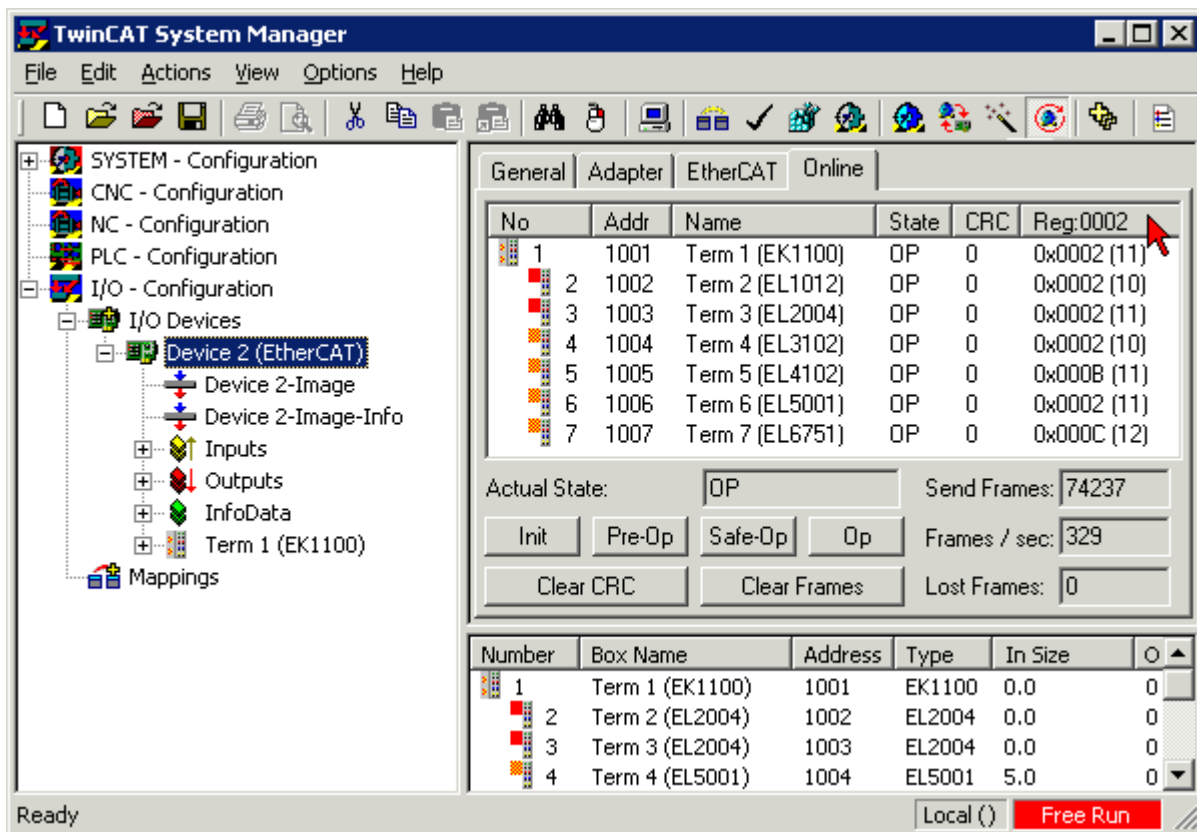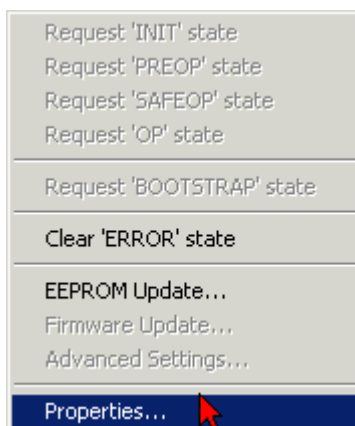
Fig. 153: FPGA firmware version definition

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

Fig. 154: Context menu *Properties*

The *Advanced Settings* dialog appears where the columns to be displayed can be selected. Under *Diagnosis/***Online View** select the *'0002 ETxxxx Build'* check box in order to activate the FPGA firmware version display.
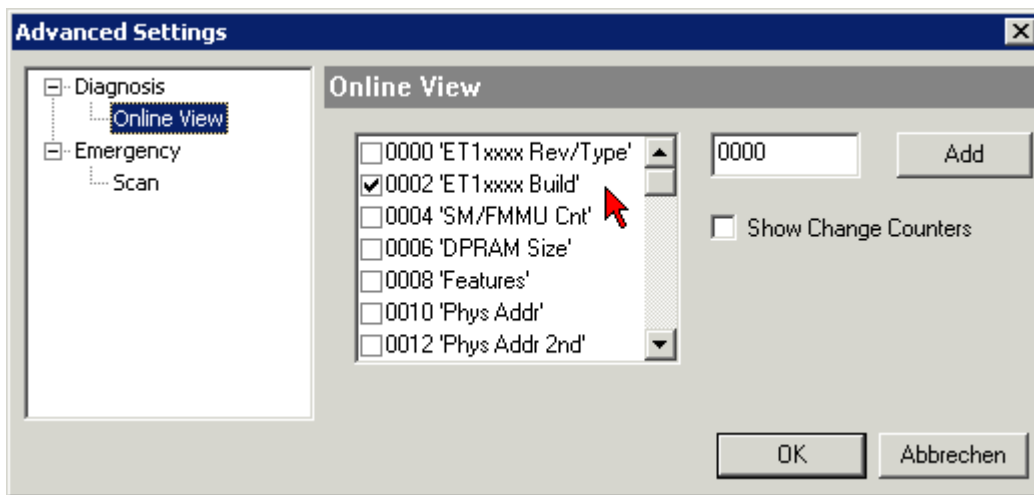
Fig. 155: Dialog *Advanced Settings*

Update

For updating the FPGA firmware

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

Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

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

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

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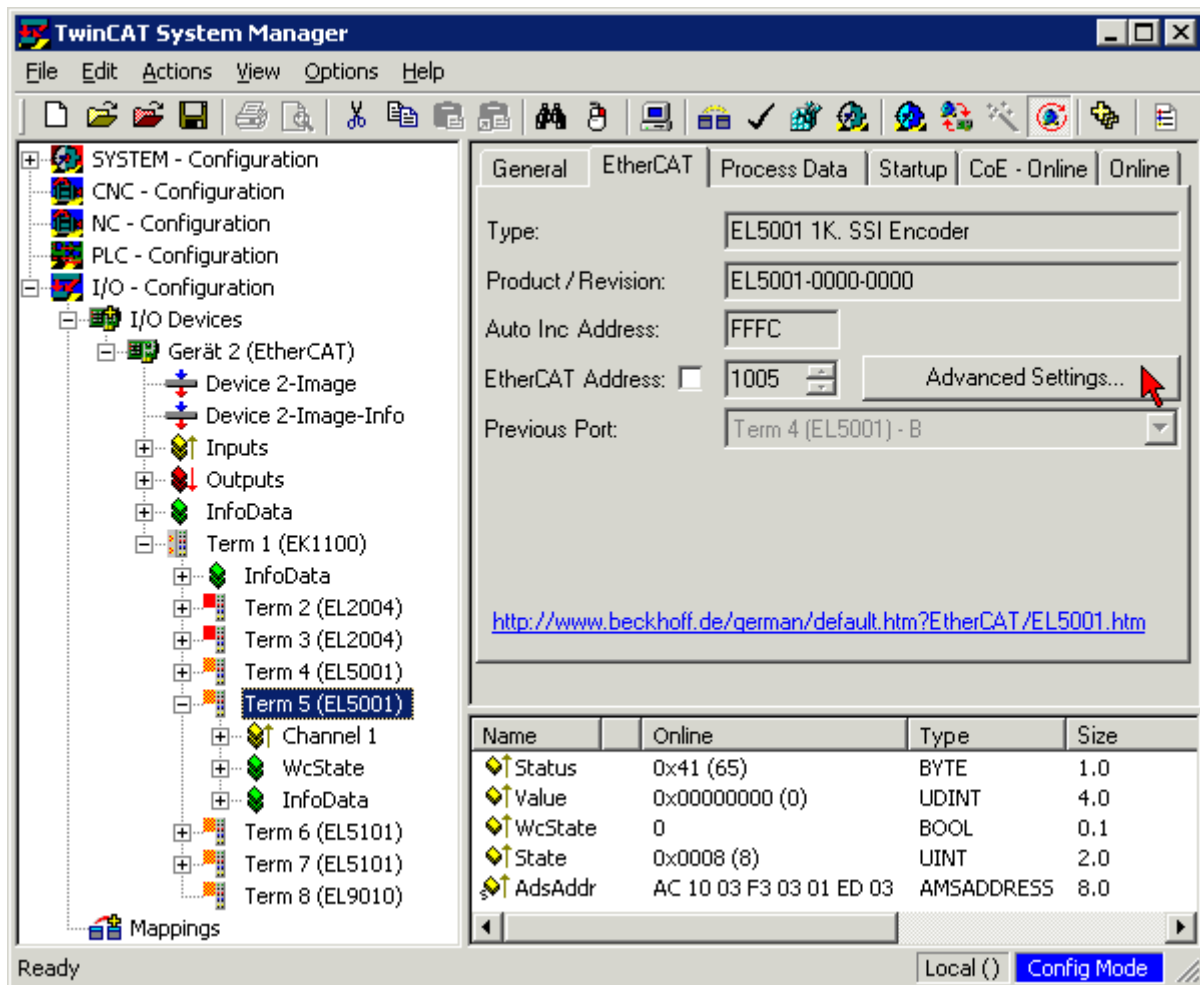
- The *Advanced Settings* dialog appears. Under *ESC Access/E²PROM*/FPGA click on *Write FPGA* button:
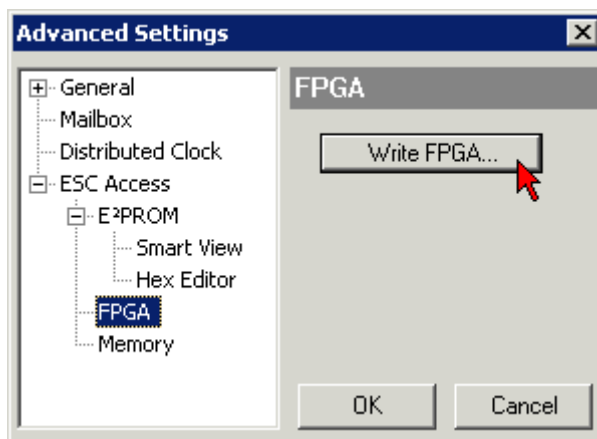
- Select the file (*.rbf) with the new FPGA firmware, and transfer it to the EtherCAT device:
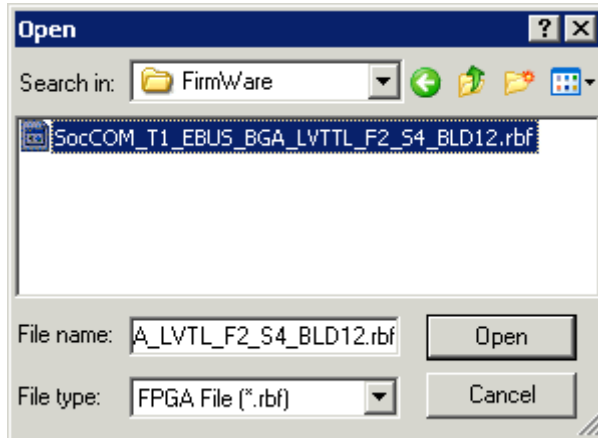
- Wait until download ends
- Switch slave current less for a short time (don't pull under voltage!). In order to activate the new FPGA firmware a restart (switching the power supply off and on again) of the EtherCAT device is required.
- Check the new FPGA status

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

7.3.5   Simultaneous updating of several EtherCAT devices

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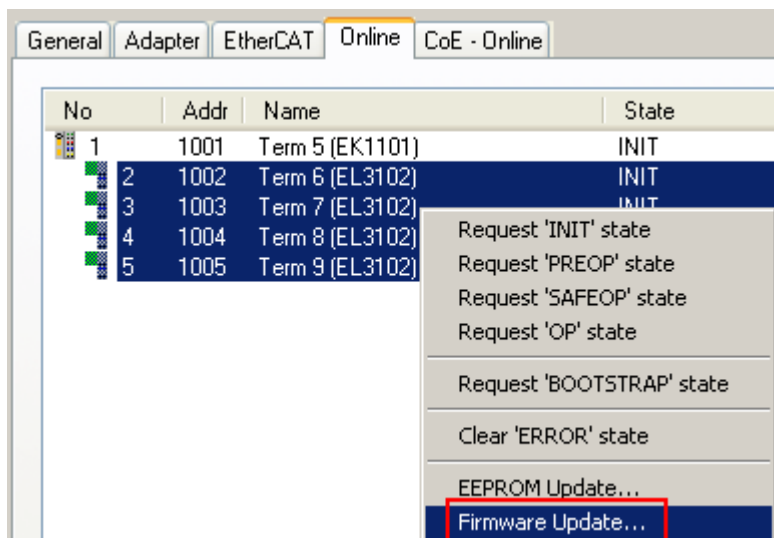

Fig. 156: Multiple selection and firmware update

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

7.4   Restoring the delivery state

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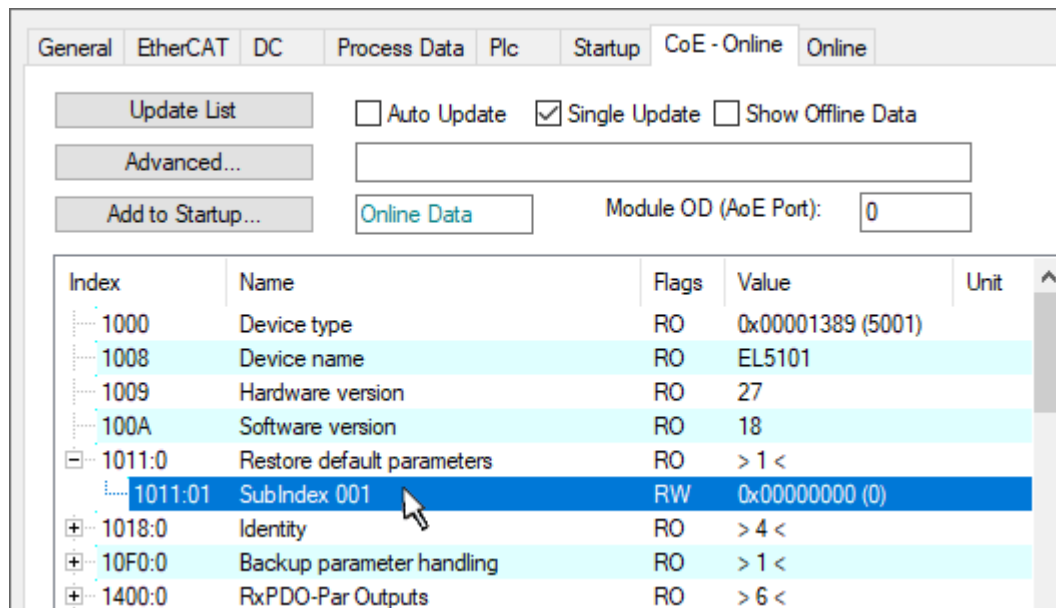

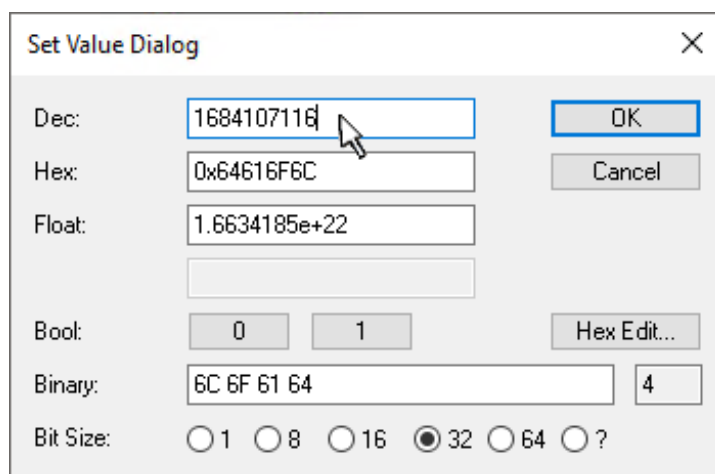Fig. 157: Selecting the *Restore default parameters* PDO

Fig. 158: Entering a restore value in the Set Value dialog

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

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

All backup objects are reset to the delivery state.

i **Alternative restore value**

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

An incorrect entry for the restore value has no effect.

7.5 Support and Service

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

Beckhoff's branch offices and representatives

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

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

You will also find further documentation for Beckhoff components there.

Support

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

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

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

Service

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

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

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

Headquarters Germany

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Germany

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

Trademark statements

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

Third-party trademark statements

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

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

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

More Information:
www.beckhoff.com/EM3701