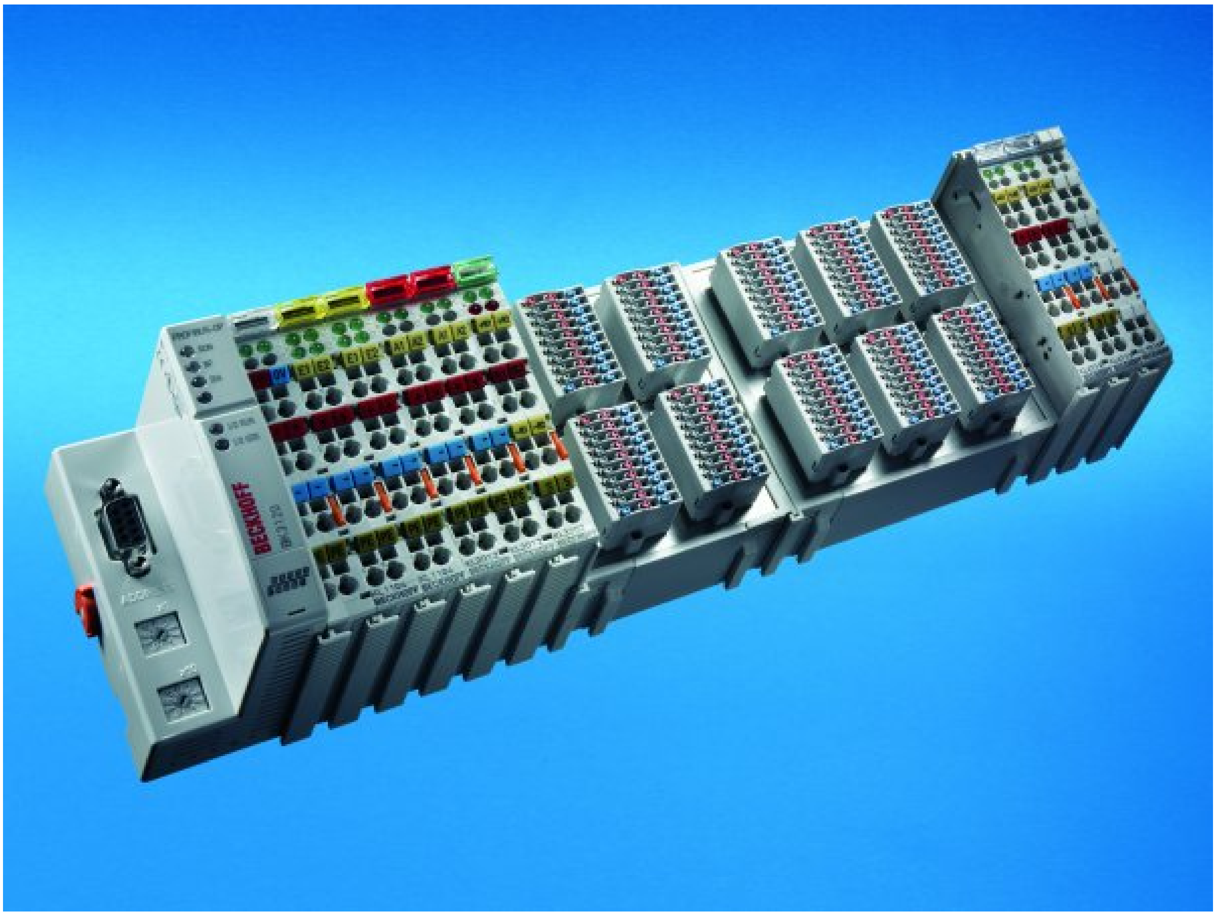


Documentation

EM7004

4 Axis Interface, 16 Digital Inputs, 16 Digital Outputs, 4 Analog Outputs, 4 Encoder Inputs

Version: 2.0
Date: 2016-09-21

BECKHOFF

Table of contents

1 Foreword

1.1 Notes on the documentation

Intended audience

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

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. For that reason the documentation is not in every case checked for consistency with performance data, standards or other characteristics. In the event that it contains technical or editorial errors, we retain the right to make alterations at any time and without warning. No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

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

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents: EP1590927, EP1789857, DE102004044764, DE102007017835 with corresponding applications or registrations in various other countries.

The TwinCAT Technology is covered, including but not limited to the following patent applications and patents: EP0851348, US6167425 with corresponding applications or registrations in various other countries.

EtherCAT®

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

Copyright

1.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Description of symbols

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

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

1.3 Documentation issue status

Version	Comment
2.0	• Migration • Update structure
1.9	• Update chapter "Basic function principles"
1.8	• Update chapter "Wiring"
1.7	• Update chapter "Firmware"
1.6	• New safety and trademark notes added
1.5	• Object description added
1.4	• Technical data added
1.3	• Correction wiring description
1.2	• Update description added
1.1	• Technical data added
1.0	• Object description added, PLS function added
0.2	• Object description added
0.1	• First preliminary documentation for EM7004

1.4 Version identification of EtherCAT devices

Designation

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

- family key
- type
- version
- revision

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

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 website. From 2014/01 the revision is shown on the outside of the IP20 terminals, see Fig. *"EL5021 EL terminal, standard IP20 IO device with batch number and revision ID (since 2014/01)"*.

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

Identification number

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

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

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

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

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

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

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

Syntax: D ww yy x y z u

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

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

Unique serial number/ID, ID number

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

See also the further documentation in the area

- IP67: EtherCAT Box
- Safety: TwinSafe
- Terminals with factory calibration certificate and other measuring terminals

Examples of markings:

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

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

Fig. 3: *CU2016 switch with batch number*

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

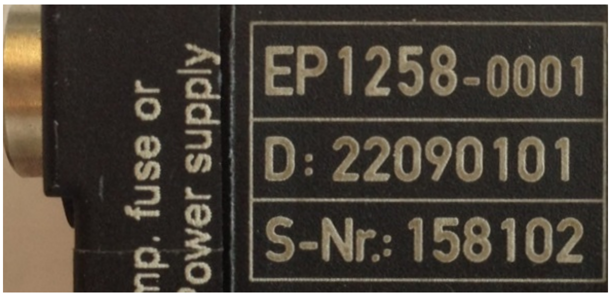


Fig. 5: *EP1258-00001 IP67 EtherCAT Box with batch number 22090101 and unique serial number 158102*

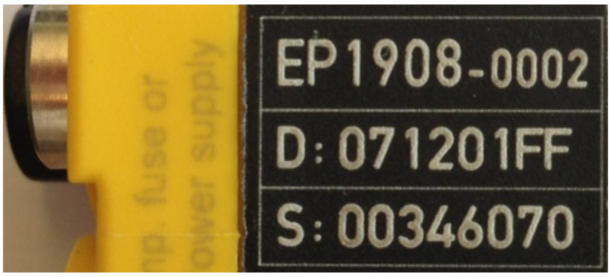


Fig. 6: *EP1908-0002 IP76 EtherCAT Safety Box with batch number 071201FF and unique serial number 00346070*

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

2 Product overview

2.1 Terminal Modules – System Overview

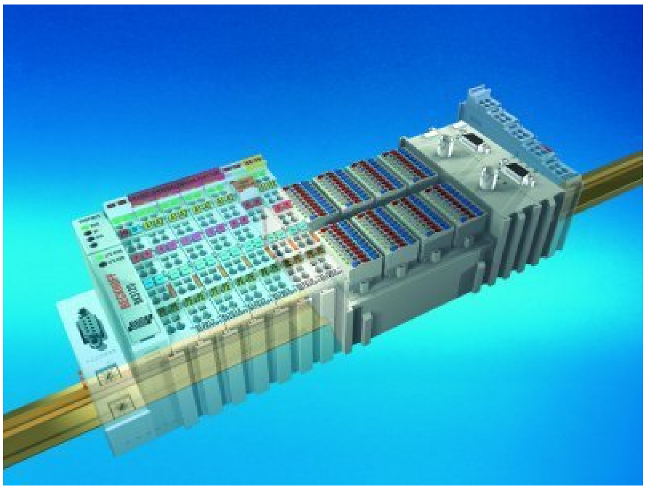


Fig. 8: Terminal Modules – System Overview

Better sensor and actuator functionality makes machines and systems more and more powerful. The Bus Terminal reliably meets increased requirements for I/O signals through its modularity and compact design. The existing Beckhoff Bus Terminal system is complemented by the new version of the EMxxxx / KMxxxx Terminal Modules with increased packing density. In many areas of application, cost benefits can be realized through lower overall installed size and application-specific signal mix.

The new Terminal Modules are fully system-compatible. Like the Bus Terminals, they are bus-neutral and can therefore be operated with any Beckhoff Bus Coupler and Bus Terminal Controller. Like the standard Bus Terminals, the EM / KM modules are integrated in the I/O system and connected with the internal terminal bus (K-bus). Bus Terminals and terminal modules can be combined without restriction.

Connector

Like for the Bus Terminals, no tools are required for the wiring. Spring-loaded technology is used, however the connection layer is pluggable (fixed wiring).

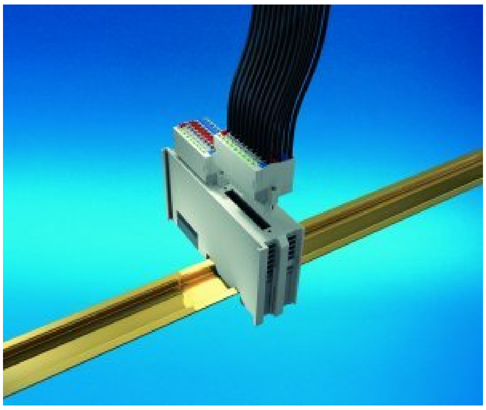


Fig. 9: Terminal modules - pluggable connection

Connection technology

Plug connectors are available for single and triple conductor connection methods.

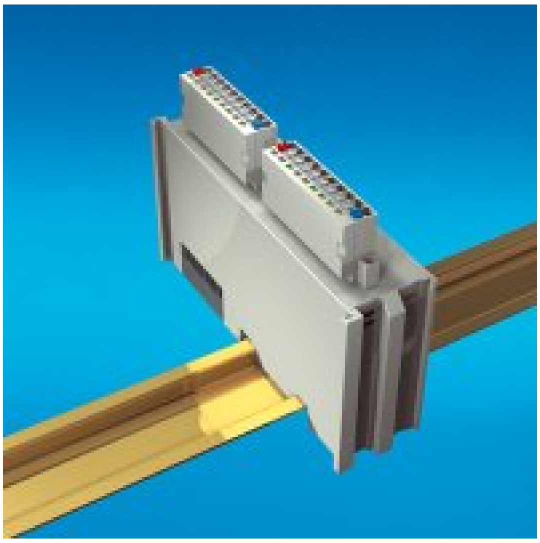


Fig. 10: Terminal module with plug connector for single conductor connection method (ZS2001-0002)

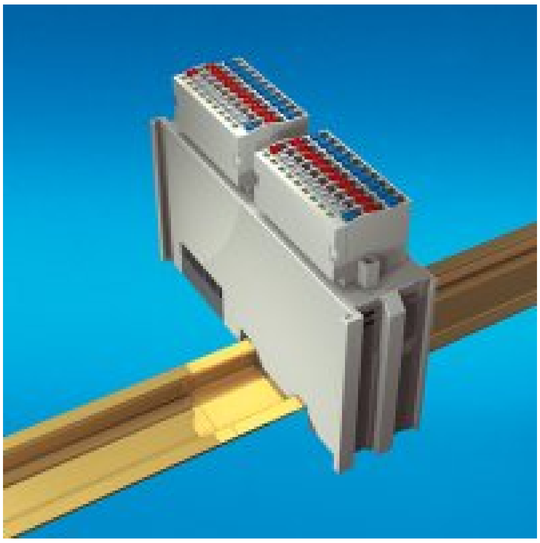


Fig. 11: Terminal module with connector for three-wire connection (ZS2001-0004)

Packing density

The Terminal Modules combine 16, 32 or 64 digital inputs or outputs on a very small area. This compact and slimline design enables very high packing densities, leading to smaller control cabinets and terminal boxes.

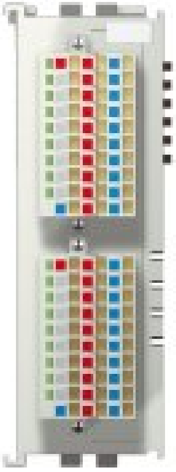


Fig. 12: Terminal module with 16 channels

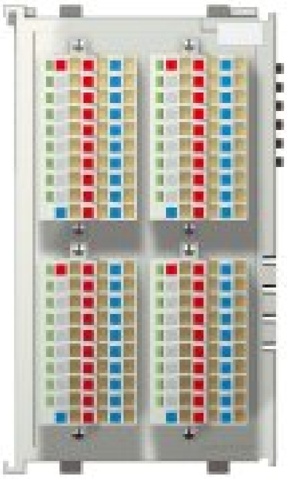


Fig. 13: Terminal module with 32 channels

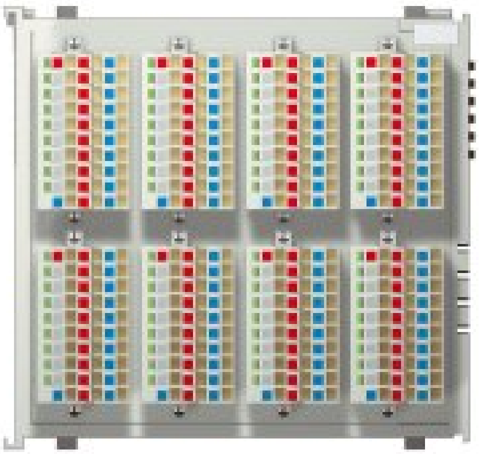


Fig. 14: Terminal module with 64 channels

2.2 Introduction

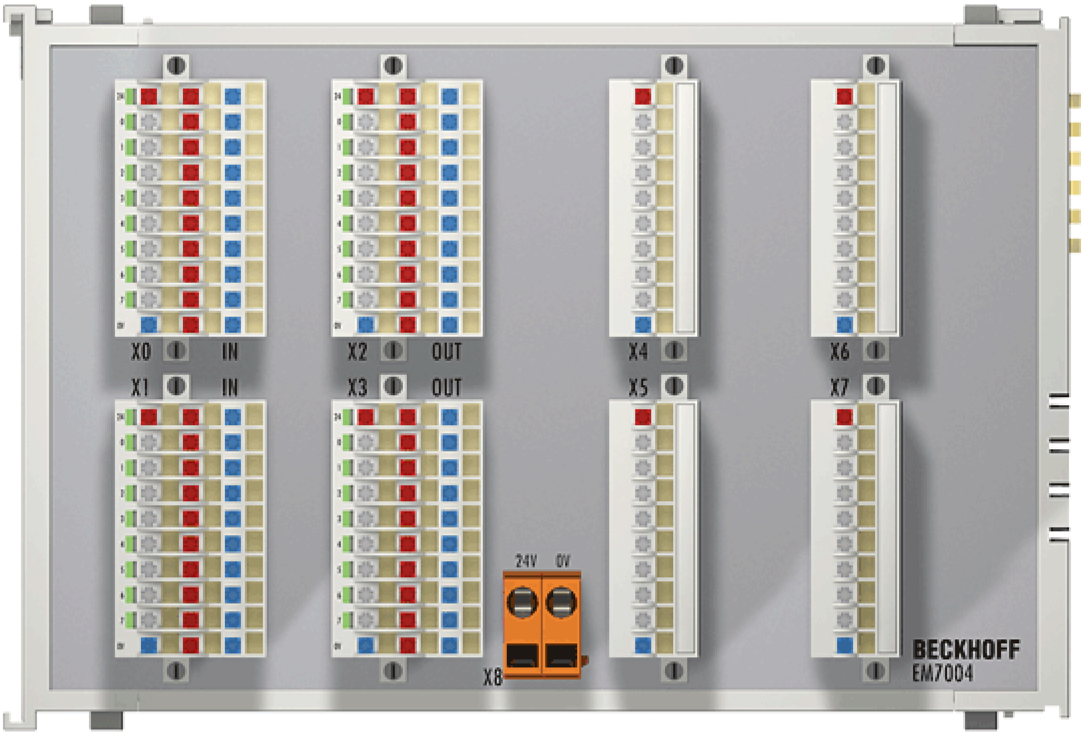


Fig. 15: EM7004

Terminal module axis interface

The EtherCAT module EM7004 is an interface that is optimized for direct connection of 4 servo drives, whose "encoder simulation" feeds the module with RS485 signals. The compact module features 4 integrated incremental encoders, 16 digital 24 V_{DC} inputs and outputs, and 4 analog ±10 V outputs. For fast preprocessing the digital outputs can be connected directly via the 4 encoders (PLS). This function is parameterizable. All inputs and outputs operate with a 24 V supply. Connectors X4, X5, X6 and X7 each feature an encoder input and an analog output. The connectors are galvanically isolated from each other and from the supply voltage. Connectors X0 and X1 with 16 digital inputs and X2 and X3 with 16 digital outputs enable 3-wire connection.

See also section Ordering information for EM/KM connector [▶ 113].

2.3 Technical data

Technical data	EM7004
Digital inputs	16, 24 V_{DC}
Digital outputs	16, 24 V_{DC}
Max. output current Digital outputs	X2.0 - X2.3: 0.5 A X2.4 - X2.7: 1.5 A X3.0 - X3.3: 0.5 A X3.4 - X3.7: 1.5 A
Max. sum current (24 V supply voltage)	10 A
Analog outputs	4 x ±10 V (2 mA)
Encoder inputs	4 x (A, /A, B, /B, gate, latch, ground); A B – insulated RS485 inputs (RS422); 4 x 16 bit, quadrature encoder; < 400 kHz PLS function (Programmable Limit Switch) [▶ 14]
Minimum cycle time	1 ms
Power supply for the electronics	via the E-bus
Current consumption via E-bus	typ. 280 mA
Electrical isolation	500 V (E-bus/signal voltage)
Dimensions with connector (W x H x D)	approx. 147 mm x 100 mm x 55 mm (width aligned: 145 mm), see dimensional drawing [▶ 19]
Weight (without connector)	approx. 260 g
Weight of a connector ZS2001-0004 (three-pole)	approx. 20 g
Weight of a connector ZS2001-0005 (single-pole)	approx. 10 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
Mounting [▶ 19]	on 35 mm mounting rail conforms to EN 60715
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approval	CE

2.4 Basic function principles

The 4-axis interface terminal module EM7004 integrates 4 Incremental encoder, 4 analog outputs with +/-10 V and 16 digital 24 V inputs and outputs. The digital inputs and outputs can be connected in single, two or three wire connection mode.
For the incremental encoder a 16-bit counter with quadrature decoder and a 16-bit latch can be read, set or enabled. In addition to encoder inputs A, /A, B, /B, an additional latch input L (24 V) and a gate input G (24 V) for disabling the counter are available. The "Value" input value represents a 16-bit "position counter".
The PLS function (Programmable Limit Switch) can be used to control digital outputs automatically depending on the encoder counter value (up to 75 entries).

Reference

An object description and an overview of adjustable encoder parameters can be found in chapter "TwinCAT System Manager - Object overview [▶ 96]".

2.4.1 Analog process data

In the delivery state the process data are shown in two's complement form (-1_{dez} corresponds to 0xFFFF). The feature object 0x8020:02 [▶ 97] (channel 1), 0x8030:02 [▶ 97] (channel 2), 0x8040:02 [▶ 97] (channel 3), 0x8050:02 [▶ 97] (channel 4) can be used to select other presentation types (e.g. magnitude sign format, absolute value).

Output value		Output voltage
hexadecimal	decimal	
0x8001	-32769	-10 V
0xC001	-16383	-5 V
0x0000	0	0 V
0x3FFF	+16383	+5 V
0x7FFF	+32767	+10 V

2.4.2 Process data equations

Calculation

The process data, which are transferred to the Bus Terminal, are calculated based on the following equations:

$Y_H = X \times A_K + B_K$	Output value after manufacturer calibration (the feature object user scaling [0x8020:01 [▶ 97] (channel 1), 0x8030:01 [▶ 97] (channel 2), 0x8040:01 [▶ 97] (channel 3), 0x8050:01 [▶ 97] (channel 4) is inactive]
$Y_A = Y_H \times A_W \times 2^{-16} + B_W$	Output value following user scaling

Legend

Name	Name	Object index
Y_H	Process data to D/A converter after manufacturer calibration	-
Y_A	Process data to D/A converter after user scaling	-
X	Controller process data	-
B_K	Manufacturer calibration offset (can only be changed if the producer code word (object 0xF008 [▶ 112]) is set)	0x802F:01 [▶ 98], 0x803F:01 [▶ 98], 0x804F:01 [▶ 98], 0x805F:01 [▶ 98]
A_K	Manufacturer calibration gain (can only be changed if the Producer codeword (object 0xF008 [▶ 112]) is set)	0x802F:02 [▶ 98], 0x803F:02 [▶ 98], 0x804F:02 [▶ 98], 0x805F:02 [▶ 98]
B_W	User scaling offset (can be activated via feature object user scaling [0x8020:01 [▶ 97] (channel 1), 0x8030:01 [▶ 97] (channel 2), 0x8040:01 [▶ 97] (channel 3), 0x8050:01 [▶ 97] (channel 4)]	0x8020:11 [▶ 97], 0x8030:11 [▶ 97], 0x8040:11 [▶ 97], 0x8050:11 [▶ 97]
A_W	User scaling gain (can be activated via feature object user scaling [0x8020:01 [▶ 97] (channel 1), 0x8030:01 [▶ 97] (channel 2), 0x8040:01 [▶ 97] (channel 3), 0x8050:01 [▶ 97] (channel 4)]	0x8020:12 [▶ 97], 0x8030:12 [▶ 97], 0x8040:12 [▶ 97], 0x8050:12 [▶ 97]

Sample: Limitation of the output range from -5 V to +5 V, calculation of the user scaling GAIN factor

$Y_A = 16383_{dec}$ corresponds to the desired upper limit value of +5 V

$Y_H = 32767_{dec}$ corresponds to the upper limit value of +10 V
$B_W = 0_{dec}$ corresponds to the offset of the user scaling

$Y_A = Y_H \times A_W \times 2^{-16} + B_W$ Calculation of the GAIN value for an upper limit value of +5 V (corresponds to a lower limit value of -5 V through cancelling the sign)

Û $(Y_A - B_W) / (Y_H \times 2^{-16}) = A_W$

$(16383 - 0) / 32767 \times 2^{-16}$ corresponds to the user scaling gain factor (factor 0.5)

$= A_W = 32767$

Sample: Shifting of the output range from -3 V to +10 V, calculation of the user scaling OFFSET value

$Y_A = (-9831_{dec})$ corresponds to the desired lower limit value of -3 V
$Y_H = (-32769_{dec})$ corresponds to the lower limit value of -10 V
$A_W = 65536_{dec}$ corresponds to the gain factor of the user scaling (factor 1)

$Y_A = Y_H \times A_W \times 2^{-16} + B_W$ Calculation of the OFFSET value for the shifting of the lower limit value to -3 V

Û $Y_A - Y_H \times A_W \times 2^{-16} = B_W$

$(-9831) - (-32769 \times 65536 \times 2^{-16})$ corresponds to the offset value of the user scaling of +7 V

$= B_W = 22938$

Note	**OFFSET value** Shifting of the output value through the OFFSET is linear up to the lower (- 10 V) or upper limit value (+ 10 V).

User-specific output value (user default output)

The analog output value can, e.g. in the case of a failure of communication with the controller, be set to a user-specific value. The objects 0x8020:06 [▸ 97] (channel 1), 0x8030:06 [▸ 97] (channel 2), 0x8040:06 [▸ 97] (channel 3), 0x8050:06 [▸ 97] (channel 4) activate this option (value: TRUE); the output values are determined with the objects 0x8020:13 [▸ 97] (channel 1), 0x8030:13 [▸ 97] (channel 2), 0x8040:13 [▸ 97] (channel 3), 0x8050:13 [▸ 97] (channel 4). If this function is disabled (value of object 0x8020:06 [▸ 97] (channel 1), 0x8030:06 [▸ 97] (channel 2), 0x8040:06 [▸ 97] (channel 3), 0x8050:06 [▸ 97] (channel 4) is FALSE), the manufacturer default value (0V) is output.
If the watchdog timer is deactivated (value of object 0x8020:05 [▸ 97] (channel 1), 0x8030:05 [▸ 97] (channel 2), 0x8040:05 [▸ 97] (channel 3), 0x8050:05 [▸ 97] (channel 4) is TRUE), no default value is output.

PLS function (Programmable Limit Switch)

The PLS function enables a certain value ("PLS output data", object 0x80A2 [▸ 111]) to be assigned to the digital outputs when the assigned encoder ("Encoder as PLS source", object 0x80A0:01 [▸ 98]) has reached a particular switch value ("PLS switch value", object 0x80A1 [▸ 111]). Object 0x80A0:11 [▸ 98] ("output mask") is used to specify which of the digital outputs are allocated to the PLS function.

Sample

Value in object 0x80A0:11 [▸ 98] = 0x00FF, outputs 0-7 are assigned to the PLS function, outputs 8-15 are not linked to the function.

If no valid table entry is available (counter reading >=0, but less than the first table entry), the value in object 0x80A0:12 [▸ 98] ("default output") is output.
Object 0x80A1 [▸ 111] contains up to 75 switch values ("PLS switch values"); object 0x80A2 [▸ 111] contains up to 75 output data ("PLS output data"). The PLS function can only be activated if the subindex :0 (number of following subindices) for objects 0x80A1 [▸ 111] and 0x80A2 [▸ 111] is identical.

Note	**PLS function: Response time for triggering the outputs** The internal processing time of the terminal from reaching the PLS switch value to switching of the digital outputs is specified with less than 365 µs.

Code Word

Note	**Code word** The vendor reserves the authority for the basic calibration of the terminals. The code word is therefore at present reserved.

3 Mounting and wiring

3.1 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:

- DIN Rail TH 35-7.5 with 1 mm material thickness (according to EN 60715)
- DIN Rail TH 35-15 with 1,5 mm material thickness

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

3.2 Dimensions

EM7004

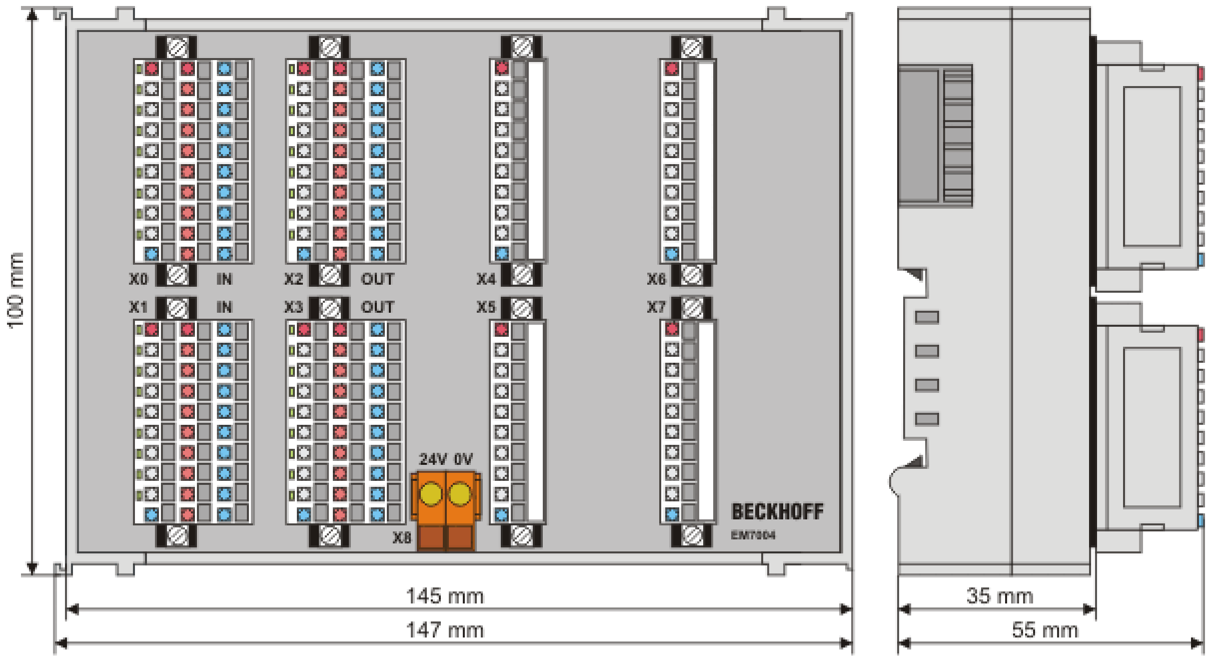


Fig. 16: EM7004 dimensions

3.3 Mounting and demounting - terminals with traction lever unlocking

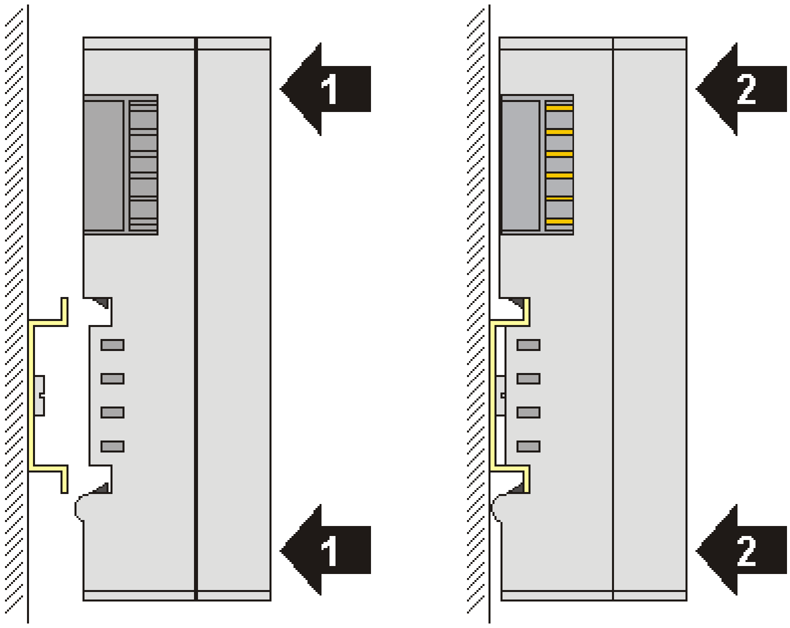
The terminal modules are fastened to the assembly surface with the aid of a 35 mm mounting rail (e.g. mounting rail TH 35-15).

Note	**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.

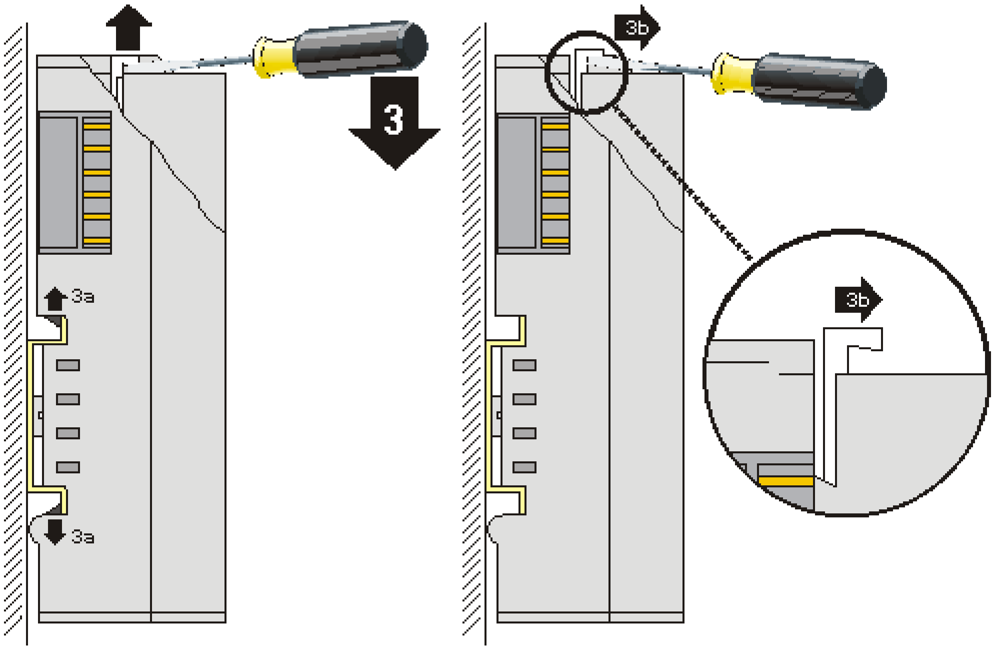


and press (1) the terminal module against the mounting rail until it latches in place on the mounting rail (2).

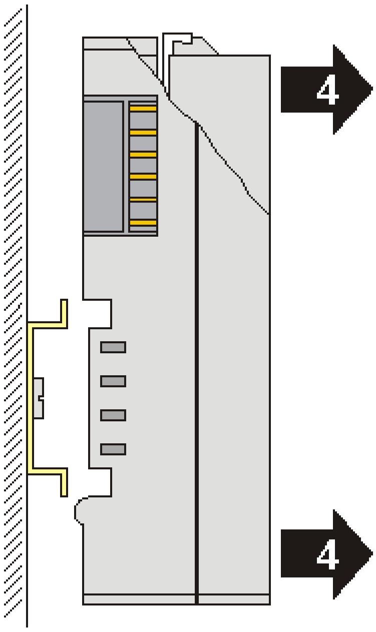
- Attach the cables.

Demounting

- Remove all the cables. Thanks to the KM/EM connector, it is not necessary to remove all the cables separately for this, but for each KM/EM connector simply undo 2 screws so that you can pull them off (fixed wiring)!
- Lever the unlatching hook on the left-hand side of the terminal module upwards with a screwdriver (3). As you do this
    - an internal mechanism pulls the two latching lugs (3a) from the top hat rail back into the terminal module,
    - the unlatching hook moves forwards (3b) and engages



- In the case 32 and 64 channel terminal modules (KMxxx4 and KMxxx8 or EMxxx4 and EMxxx8) you now lever the second unlatching hook on the right-hand side of the terminal module upwards in the same way.
- Pull (4) the terminal module away from the mounting surface.



3.4 Installation positions

Attention	**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.

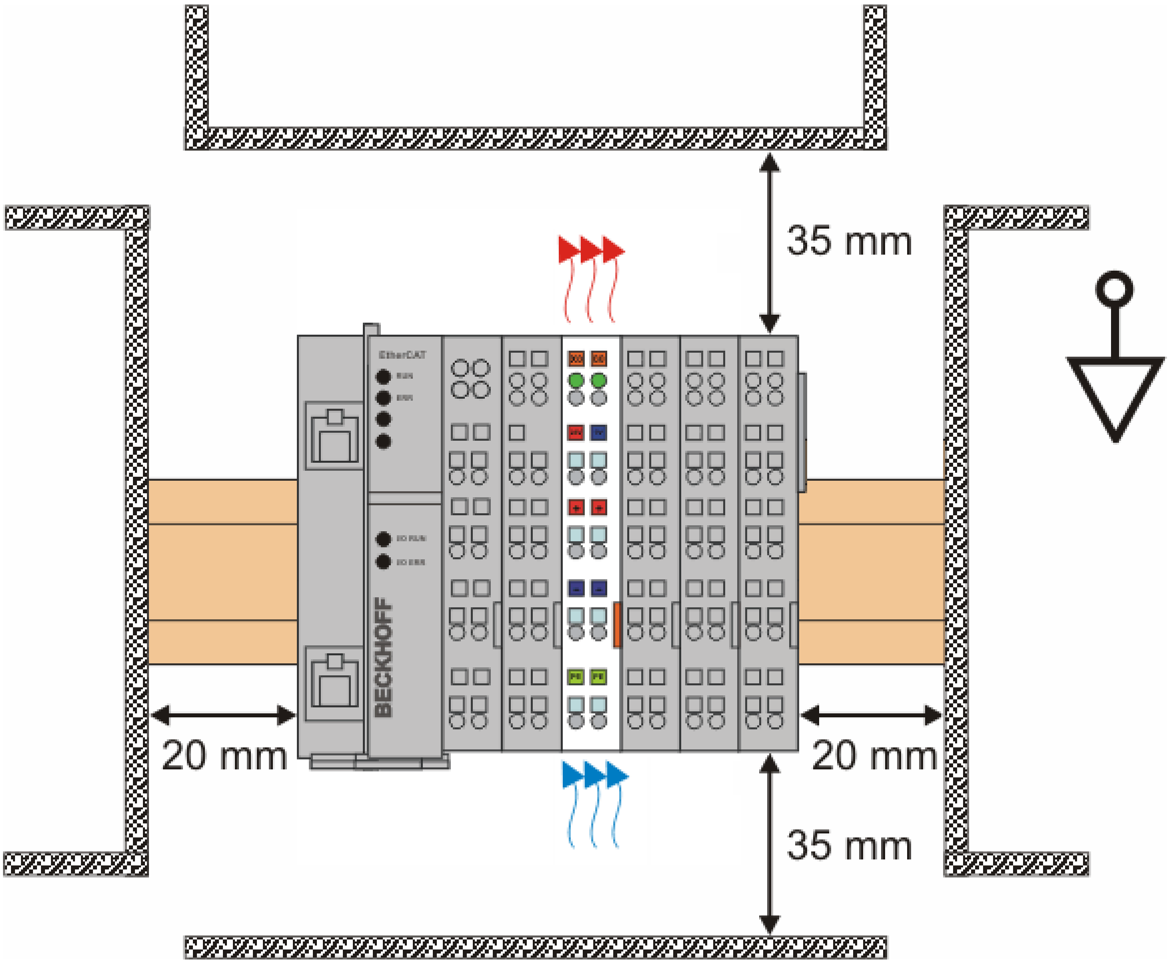


Fig. 17: *Recommended distances for standard installation position*

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.

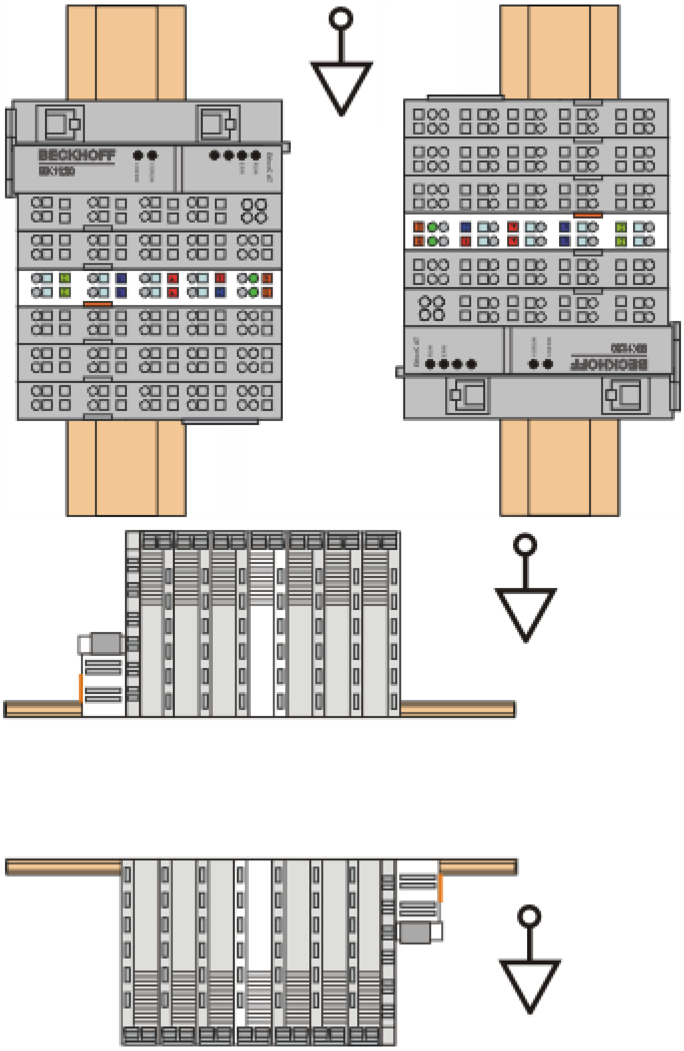


Fig. 18: *Other installation positions*

3.5 Wiring

! Attention	**Otherwise the device may be damaged** Bring the bus system into a safe, de-energized state before starting installation, disassembly or wiring of the terminal modules!

Connection of the supply voltages and pin assignment

Version 1: 24 V supply voltage via clamped joint X8

For power supply of the EM module, connect the clamped joint X8 (see Fig. *Clamped joint for power supply*) with the 24 V supply voltage.

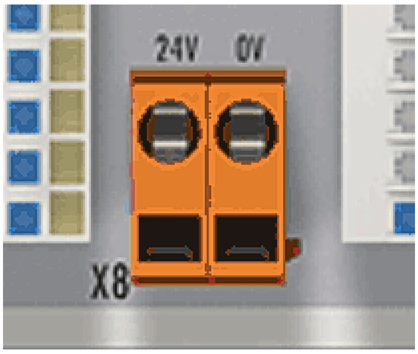


Fig. 19: Clamped joint for power supply

! Attention	**Risk of damage to the module; note sum current!** Regardless of the maximum output current of the individual module channels, the maximum sum current flowing via the clamped joint X8 must not exceed 10 A (see Technical data [▶ 14])!

Version 2: Alternative 24 V module supply

The diagram shows the alternative connection of the supply voltage for the digital inputs and outputs, and the further connection of the module.

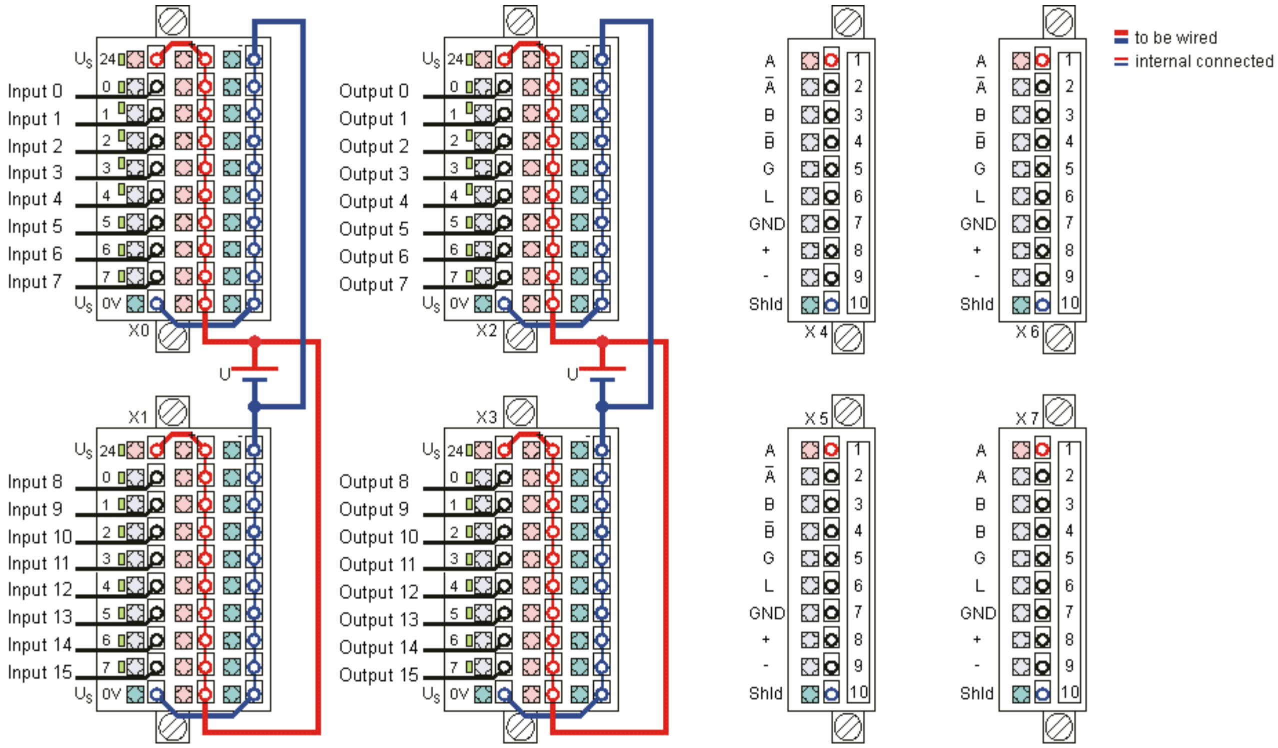


Fig. 20: Connection of the module with alternative power supply

Note	**Alternative 24 V power supply of the EM module**
	For alternative supply (U_S) of the EM module, connect connectors X0-X3 (see Fig. *Connection of the module with alternative power supply*) • the positive supply voltage to terminal location +24 V • the negative supply voltage to terminal location 0 V

Connection for clamped joints X0 - X1

Digital inputs, channel 0 - 15 (16 channels)

(with connector ZS2001-0002 or ZS2001-0004)

Connection	Terminal point	on EM/KM plug connector
24 V	24	X0
Input 0	0	X0
Input 1	1	X0
Input 2	2	X0
Input 3	3	X0
Input 4	4	X0
Input 5	5	X0
Input 6	6	X0
Input 7	7	X0
0 V	0 V	X0
24 V	24	X1
Input 8	0	X1
Input 9	1	X1
Input 10	2	X1
Input 11	3	X1
Input 12	4	X1
Input 13	5	X1
Input 14	6	X1
Input 15	7	X1
0 V	0 V	X1

Connection for clamped joints X2 – X3

Digital outputs, channel 0 - 15 (16 channels)

(with connector ZS2001-0002 or ZS2001-0004)

Connection	Terminal point	on EM/KM plug connector
24 V	24	X2
Output 0	0	X2
Output 1	1	X2
Output 2	2	X2
Output 3	3	X2
Output 4	4	X2
Output 5	5	X2
Output 6	6	X2
Output 7	7	X2
0 V	0 V	X2
24 V	24	X3
Output 8	8	X3
Output 9	9	X3
Output 10	10	X3
Output 11	11	X3
Output 12	12	X3
Output 13	13	X3
Output 14	14	X3
Output 15	15	X3
0 V	0 V	X3

Connection for clamped joints X4 – X7

Incremental encoder and analog outputs

(with connector ZS2001-0005)

Connection	Terminal point	on EM/KM plug connector
Encoder 1, input A	1	X4
Encoder 1, input /A	2	X4
Encoder 1, input B	3	X4
Encoder 1, input /B	4	X4
Encoder 1, Gate	5	X4
Encoder 1, Latch	6	X4
Ground	7	X4
Analog output 1, +10 V	8	X4
Analog output 1, -10 V	9	X4
Shield	10	X4
Encoder 2, input A	1	X5
Encoder 2, input /A	2	X5
Encoder 2, input B	3	X5
Encoder 2, input /B	4	X5
Encoder 2, Gate	5	X5
Encoder 2, Latch	6	X5
Ground	7	X5
Analog output 2, +10 V	8	X5
Analog output 2, -10 V	9	X5
Shield	10	X5
Encoder 3, input A	1	X6
Encoder 3, input /A	2	X6
Encoder 3, input B	3	X6
Encoder 3, input /B	4	X6
Encoder 3, Gate	5	X6
Encoder 3, Latch	6	X6
Ground	7	X6
Analog output 3, +10 V	8	X6
Analog output 3, -10 V	9	X6
Shield	10	X6
Encoder 4, input A	1	X7
Encoder 4, input /A	2	X7
Encoder 4, input B	3	X7
Encoder 4, input /B	4	X7
Encoder 4, Gate	5	X7
Encoder 4, Latch	6	X7
Ground	7	X7
Analog output 4, +10 V	8	X7
Analog output 4, -10 V	9	X7
Shield	10	X7

3.6 Connection technology

The digital inputs and outputs can be connected in

- single-conductor (see example, terminal point 0),
- two-conductor (see example, terminal point 3), or
- three-conductor mode (see example, terminal point 6)

.

Input circuits

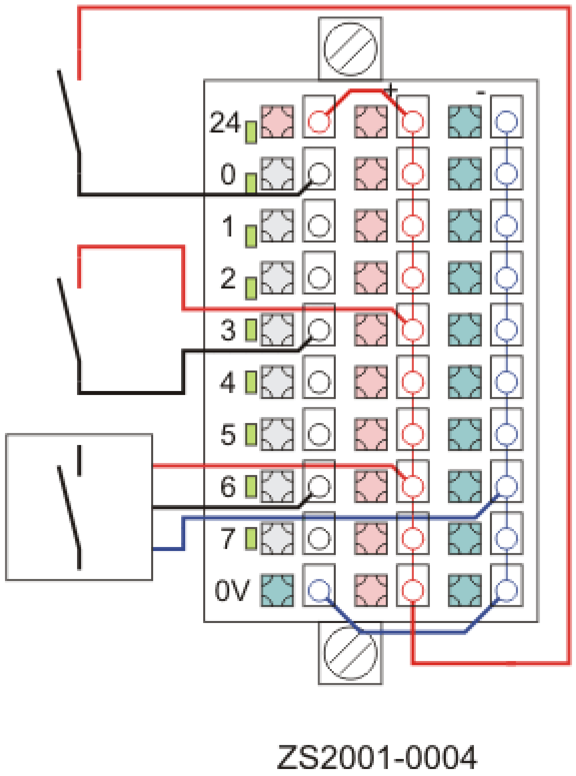


Fig. 21: Input circuits single-, two- and three-conductor mode

Output circuits

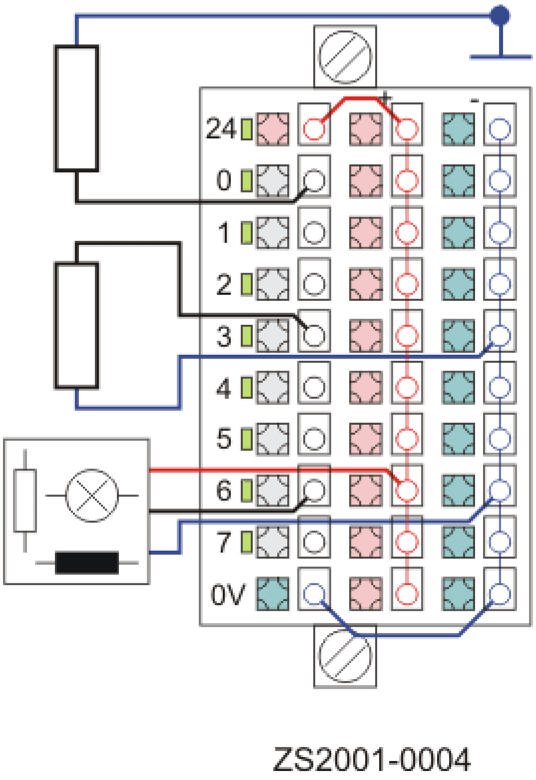


Fig. 22: Output circuits single-, two- and three-conductor mode

4 Commissioning

4.1 TwinCAT Quick Start

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

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

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

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

- **"offline"**: The configuration can be customized by adding and positioning individual components. These can be selected from a directory and configured.
  - The procedure for offline mode can be found under http://infosys.beckhoff.com:
    TwinCAT 2 → TwinCAT System Manager → IO - Configuration → Adding an I/O Device
- **"online"**: The existing hardware configuration is read
  - See also http://infosys.beckhoff.com:
    Fieldbus components → Fieldbus cards and switches → FC900x – PCI Cards for Ethernet → Installation → Searching for devices

The following relationship is envisaged from user PC to the individual control elements:

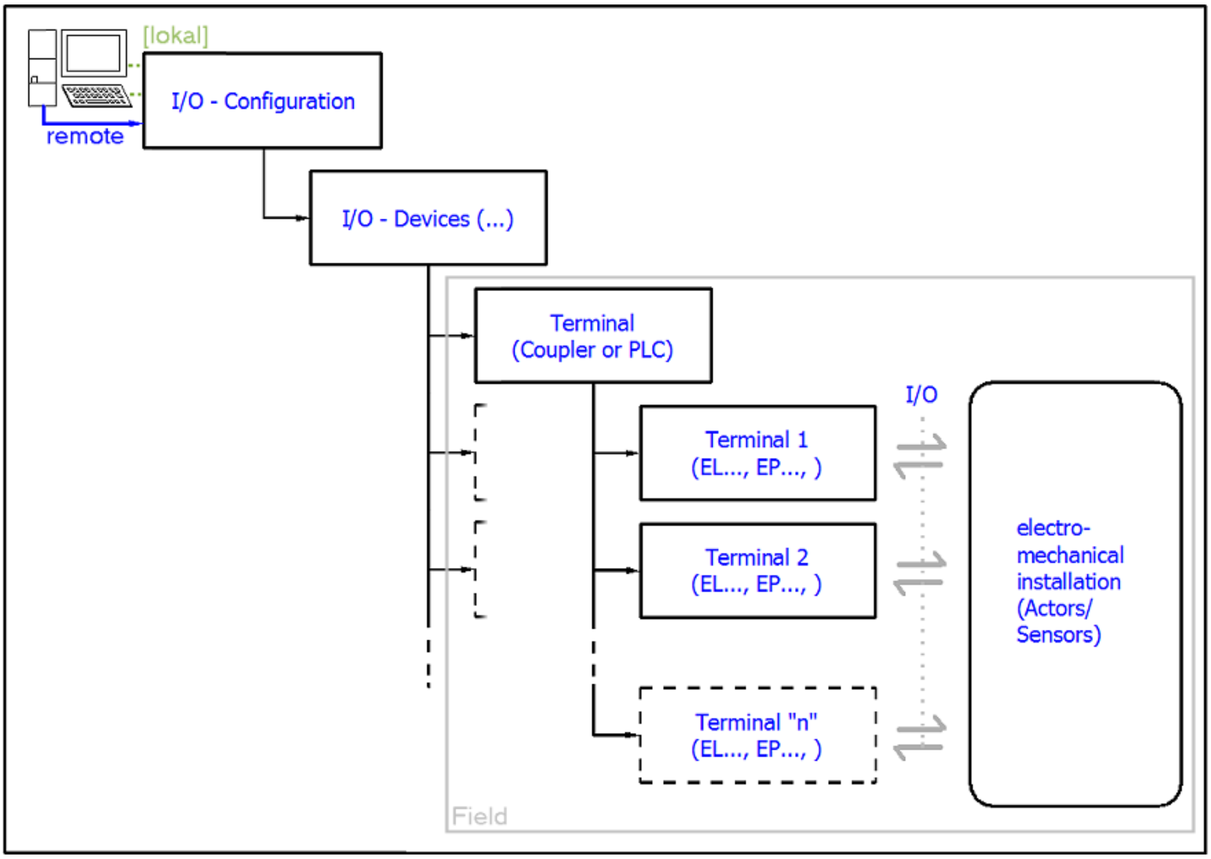


Fig. 23: Relationship between user side (commissioning) and installation

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

Sample configuration (actual configuration)

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

- Control system (PLC) **CX2040** including **CX2100-0004** power supply unit
- Connected to the CX2040 on the right (E-bus):
  EL1004 (4-channel analog input terminal -10…+10 V)
- Linked via the X001 port (RJ-45): **EK1100** EtherCAT Coupler
- Connected to the EK1100 EtherCAT coupler on the right (E-bus):
  EL2008 (8-channel digital output terminal 24 V DC; 0.5 A)
- (Optional via X000: a link to an external PC for the user interface)

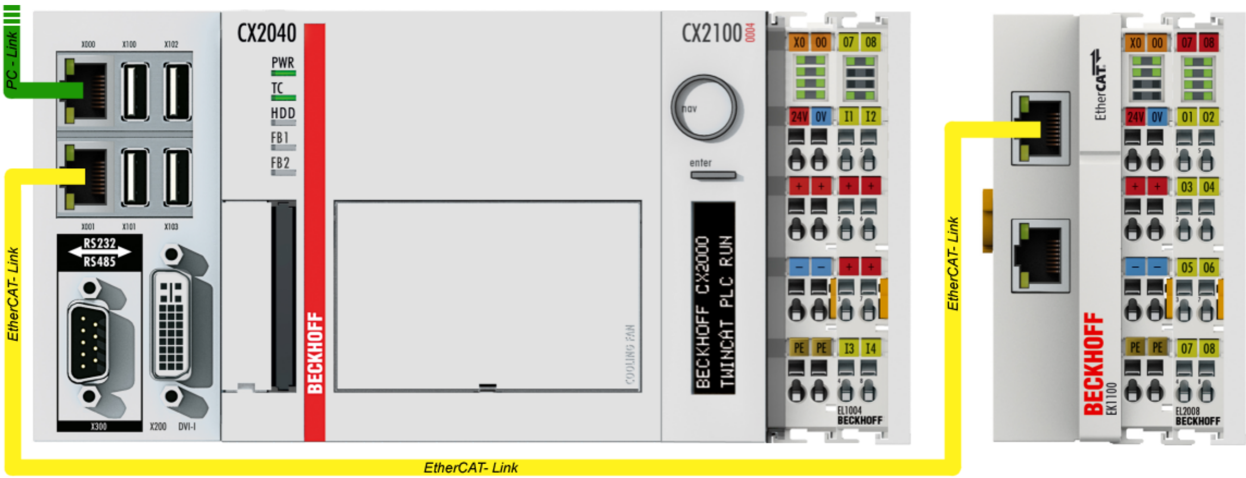


Fig. 24: Control configuration with an Embedded PC and input (EL1004) and output (EL2008)

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.

4.1.1 TwinCAT 2

Startup

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

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

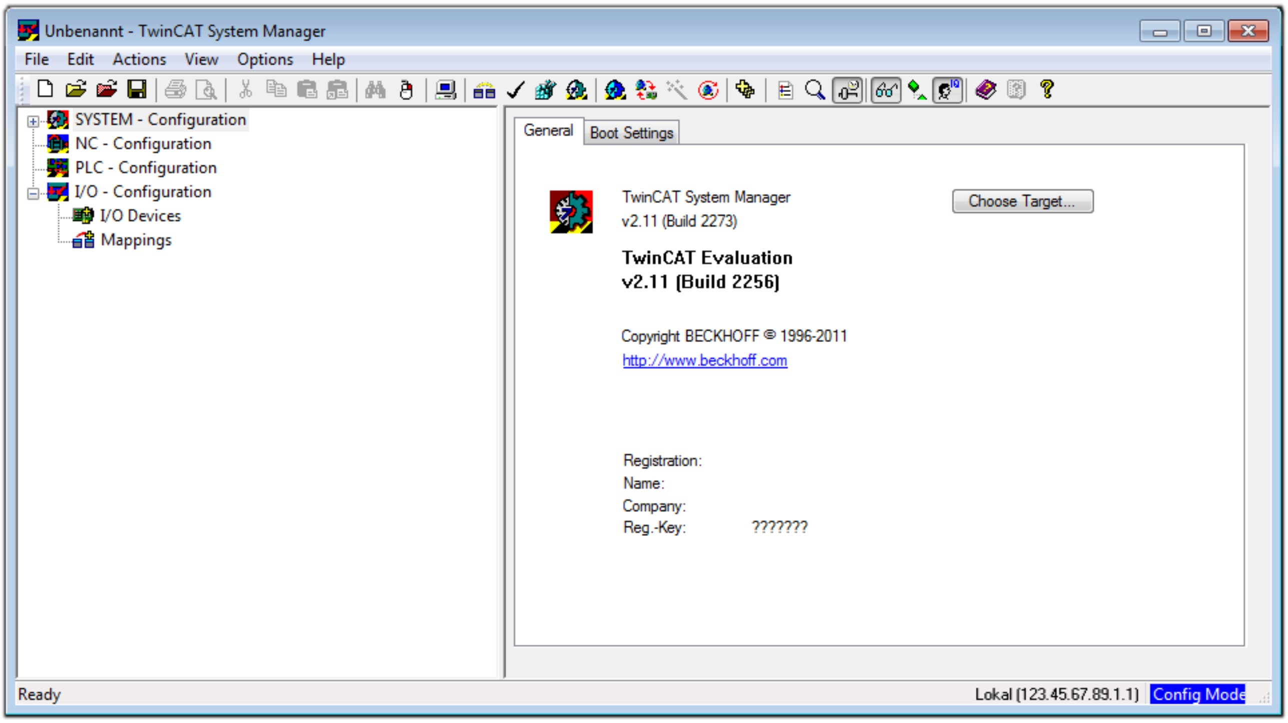


Fig. 25: Initial TwinCAT 2 user interface

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC, TwinCAT can be used in local mode and thereby the next step is "Insert Device [▶ 34]".

If the intention is to address the TwinCAT runtime environment installed on a PLC as development environment remotely from another system, the target system must be made known first. In the menu under "Actions" → "Choose Target System...", via the symbol "  " or the "F8" key, open the following window:

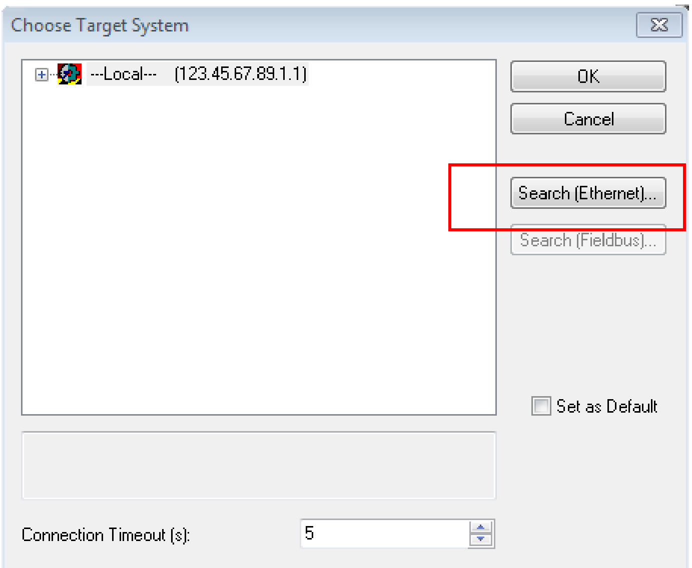


Fig. 26: Selection of the target system

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

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

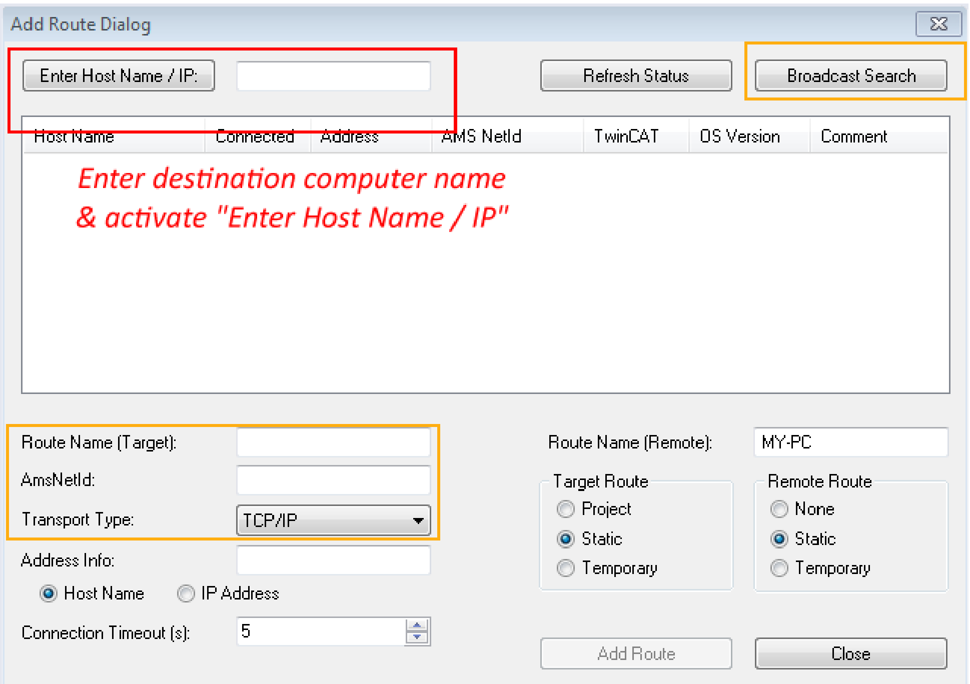
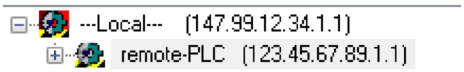


Fig. 27: Specify the PLC for access by the TwinCAT System Manager: selection of the target system

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



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

Adding devices

In the configuration tree of the TwinCAT 2 System Manager user interface on the left, select "I/O Devices” and then right-click to open a context menu and select "Scan Devices…", or start the action in the menu bar via  . The TwinCAT System Manager may first have to be set to "Config mode" via  or via menu “Actions" → "Set/Reset TwinCAT to Config Mode…" (Shift + F4).

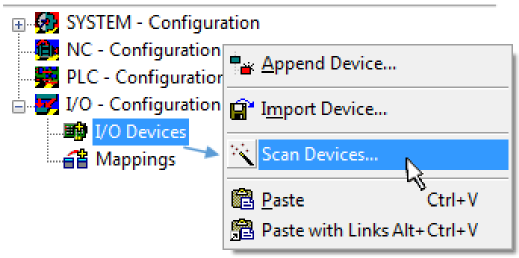


Fig. 28: Select "Scan Devices..."

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

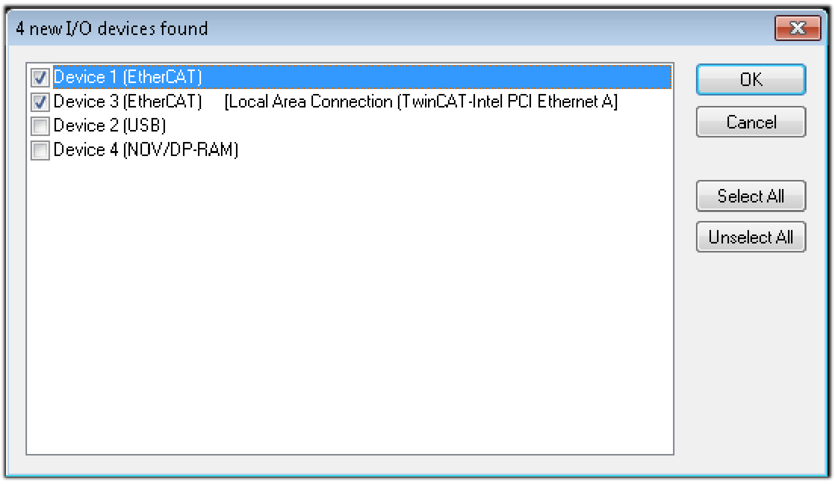


Fig. 29: Automatic detection of I/O devices: selection the devices to be integrated

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

Based on the sample configuration [▶ 31] described at the beginning of this section, the result is as follows:

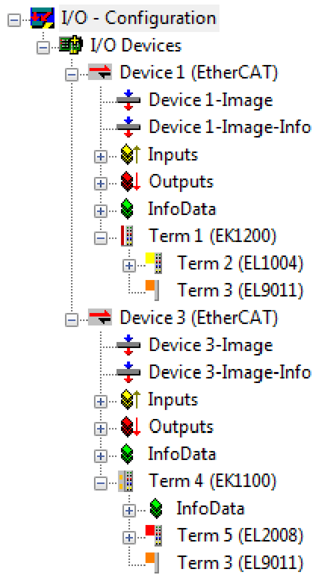


Fig. 30: Mapping of the configuration in the TwinCAT 2 System Manager

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

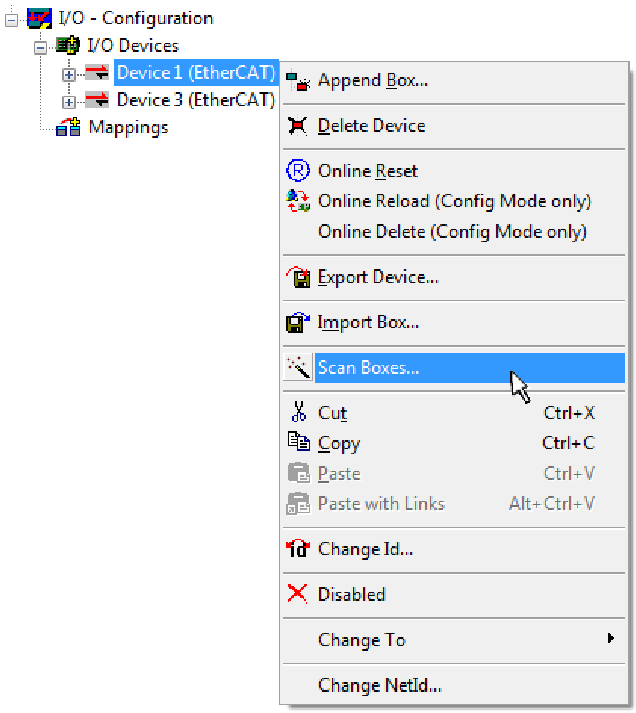


Fig. 31: Reading of individual terminals connected to a device

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

Programming and integrating the PLC

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

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

- Structured Text (ST)

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

The following section refers to Structured Text (ST).

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

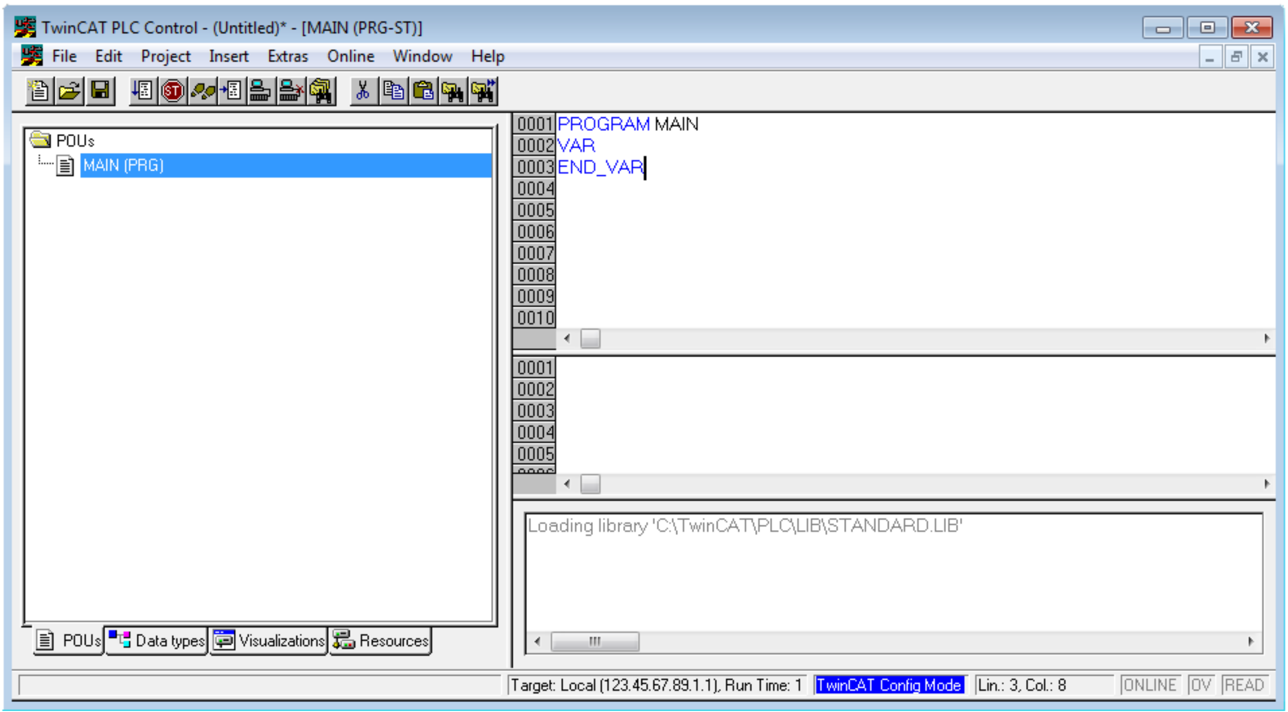


Fig. 32: TwinCAT PLC Control after startup

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

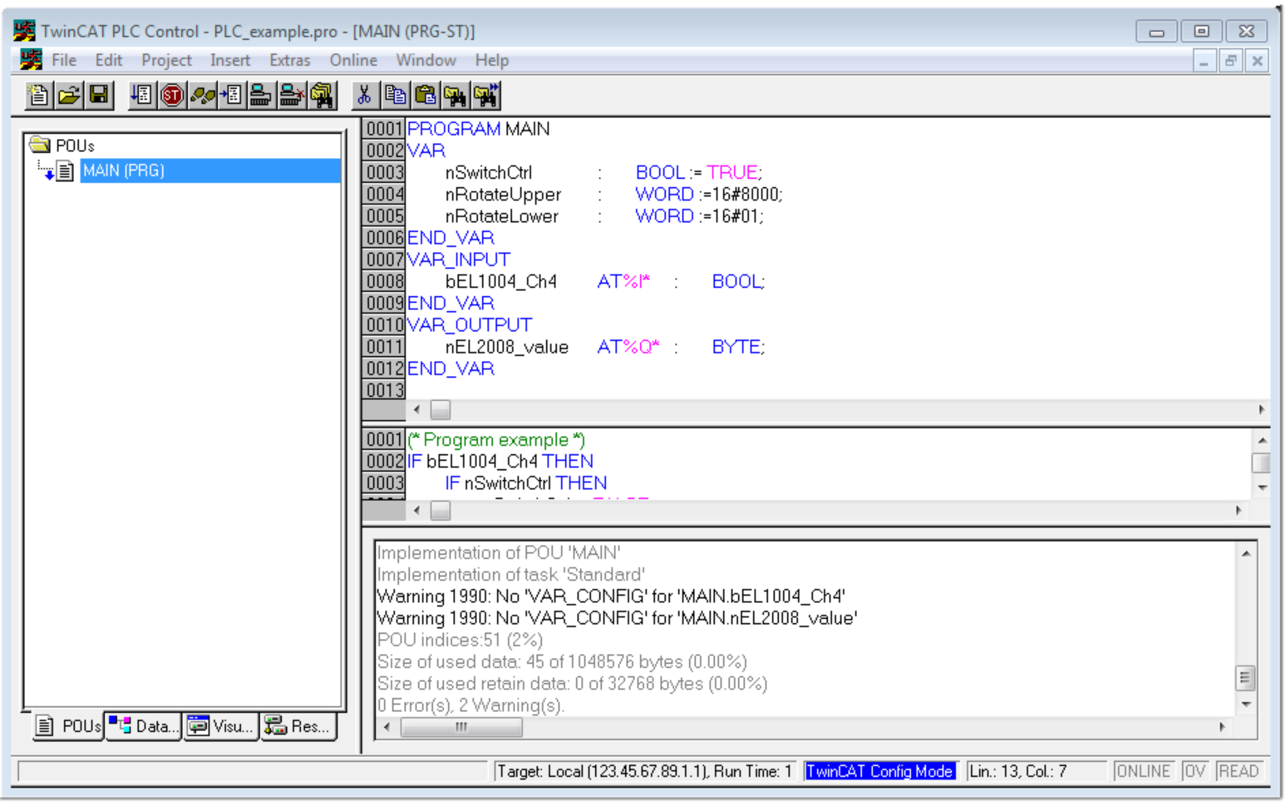


Fig. 33: Sample program with variables after a compile process (without variable integration)

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

First, integrate the TwinCAT PLC Control project in the **System Manager** via the context menu of the PLC configuration; right-click and select "Append PLC Project…":

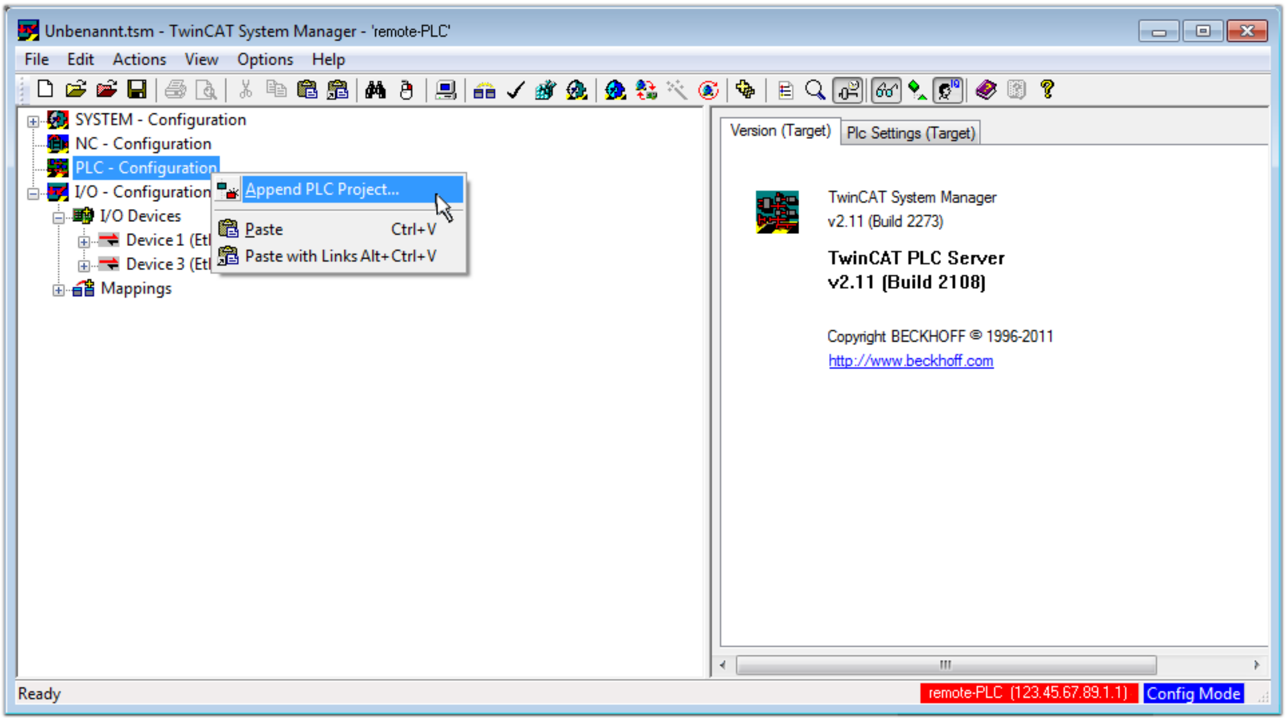


Fig. 34: Appending the TwinCAT PLC Control project

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

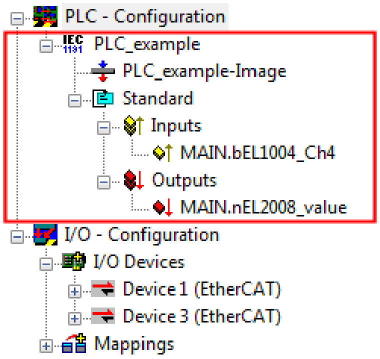


Fig. 35: PLC project integrated in the PLC configuration of the System Manager

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

Assigning variables

Open a window for selecting a suitable process object (PDO) via the context menu of a variable of the integrated project "PLC_example" and via "Modify Link..." "Standard":

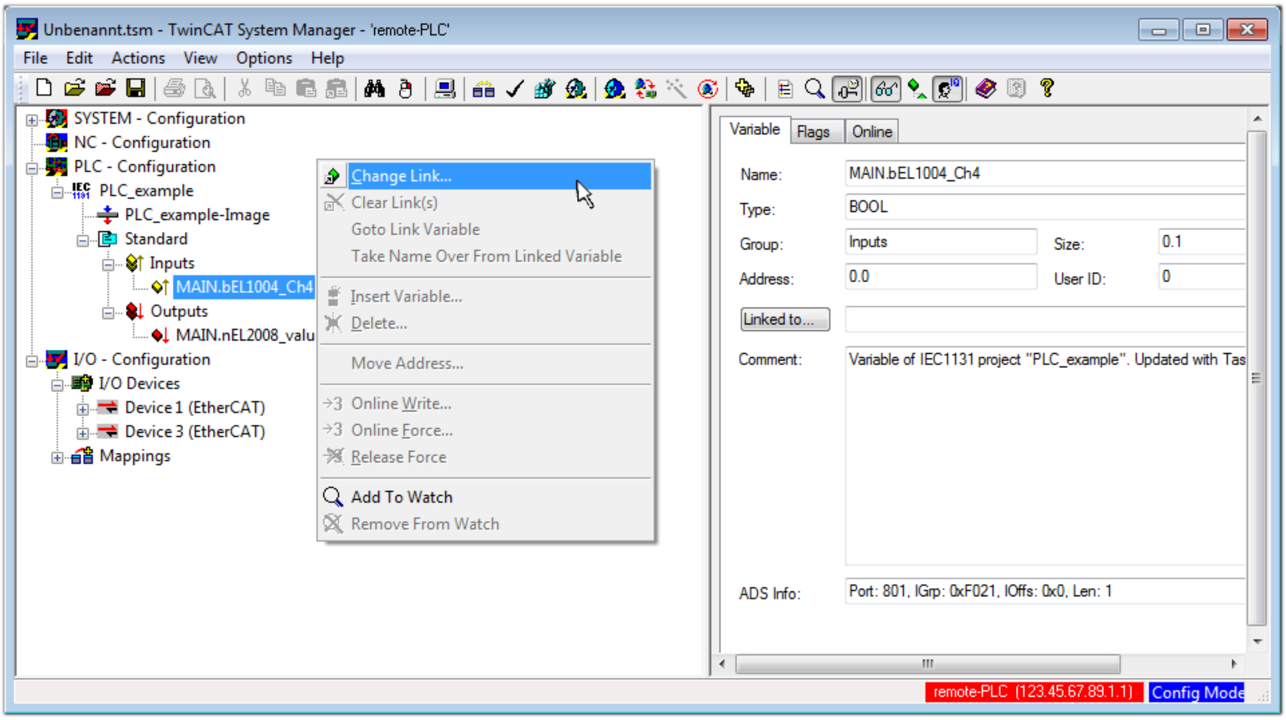


Fig. 36: Creating the links between PLC variables and process objects

In the window that opens, the process object for the variable “bEL1004_Ch4” of type BOOL can be selected from the PLC configuration tree:

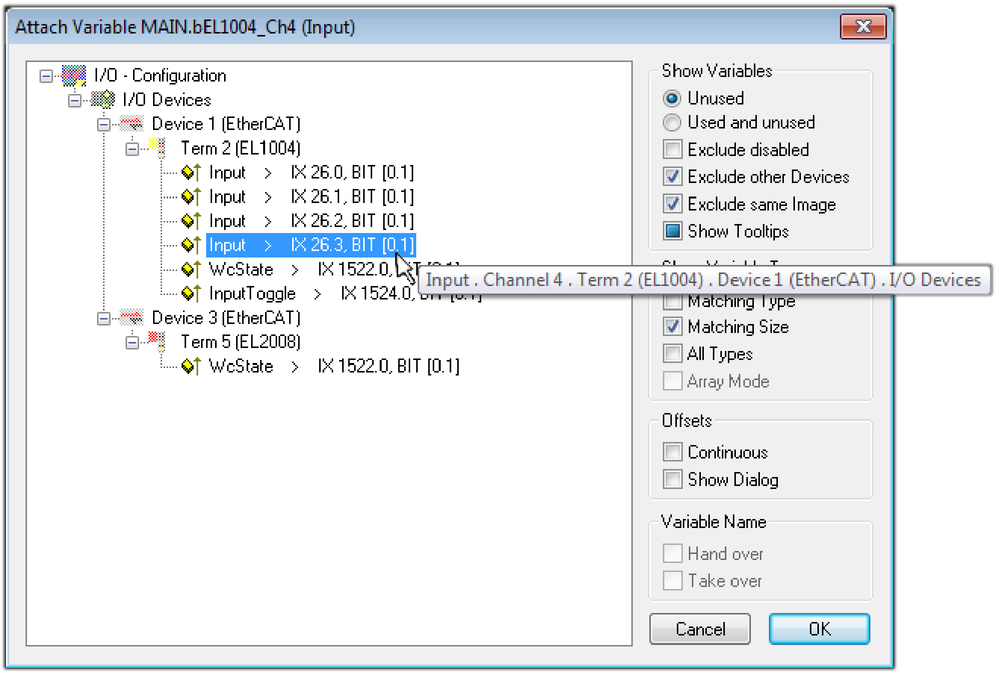


Fig. 37: Selecting PDO of type BOOL

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

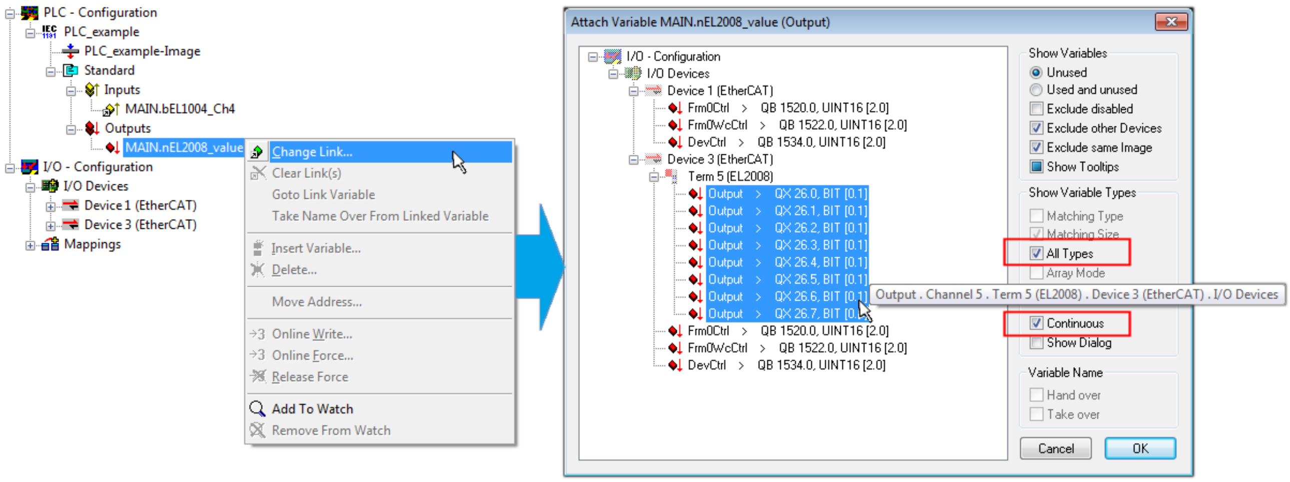


Fig. 38: Selecting several PDOs simultaneously: activate "Continuous" and "All types"

Note that the "Continuous" checkbox was also activated. This is designed to allocate the bits contained in the byte of the variable "nEL2008_value" sequentially to all eight selected output bits of the EL2008 terminal. In this way it is possible to subsequently address all eight outputs of the terminal in the program with a byte corresponding to bit 0 for channel 1 to bit 7 for channel 8 of the PLC. A special symbol (◪) at the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting a "Goto Link Variable" from the context menu of a variable. The object opposite, in this case the PDO, is automatically selected:

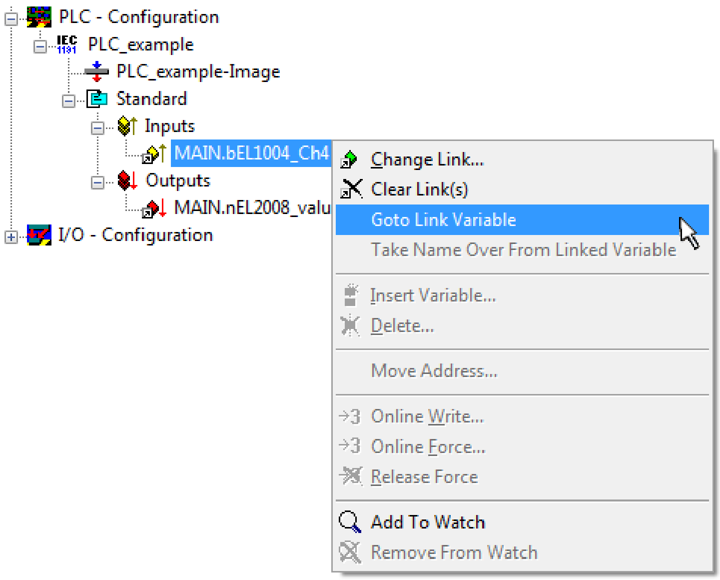
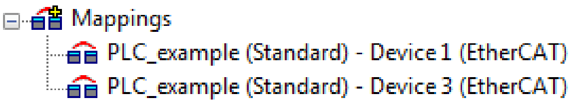


Fig. 39: Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample

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

This can be visualized in the configuration:



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

Activation of the configuration

The allocation of PDO to PLC variables has now established the connection from the controller to the inputs and outputs of the terminals. The configuration can now be activated. First, the configuration can be verified via  (or via "Actions" → "Check Configuration"). If no error is present, the configuration can be activated via  (or via "Actions" → "Activate Configuration…") to transfer the System Manager settings to the runtime system. Confirm the messages "Old configurations are overwritten!" and "Restart TwinCAT system in Run mode" with "OK".

A few seconds later the real-time status 

is displayed at the bottom right in the System Manager. The PLC system can then be started as described below.

Starting the controller

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

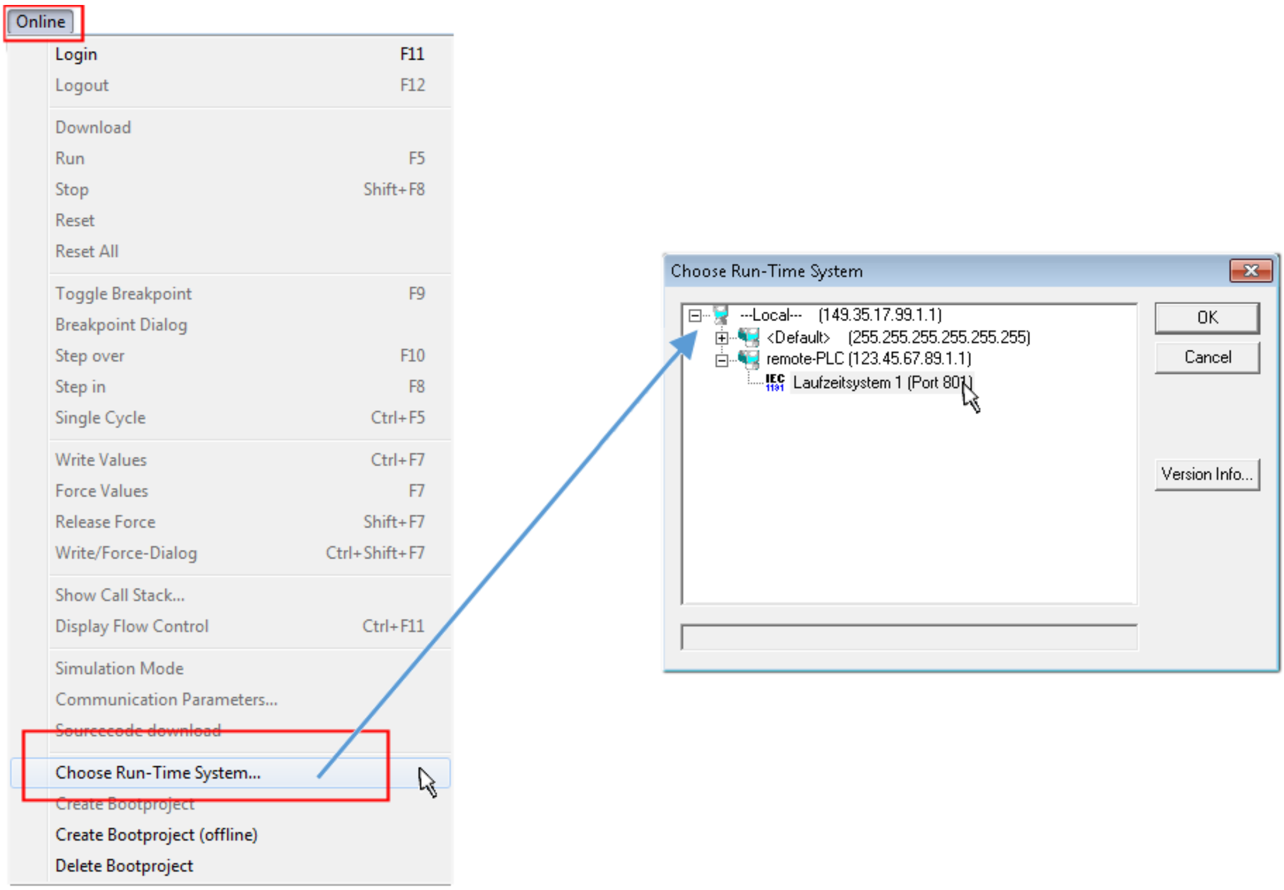
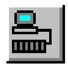


Fig. 40: Choose target system (remote)

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

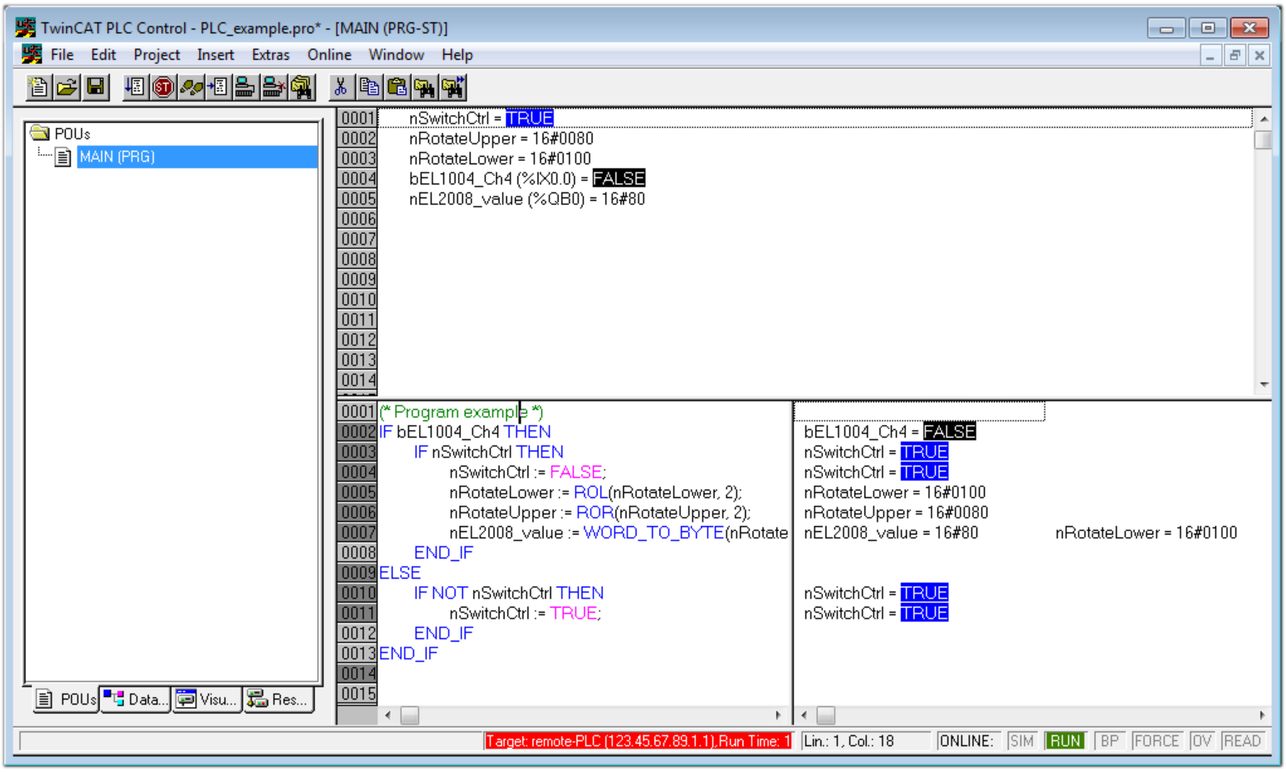


Fig. 41: PLC Control logged in, ready for program startup

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

4.1.2 TwinCAT 3

Startup

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

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

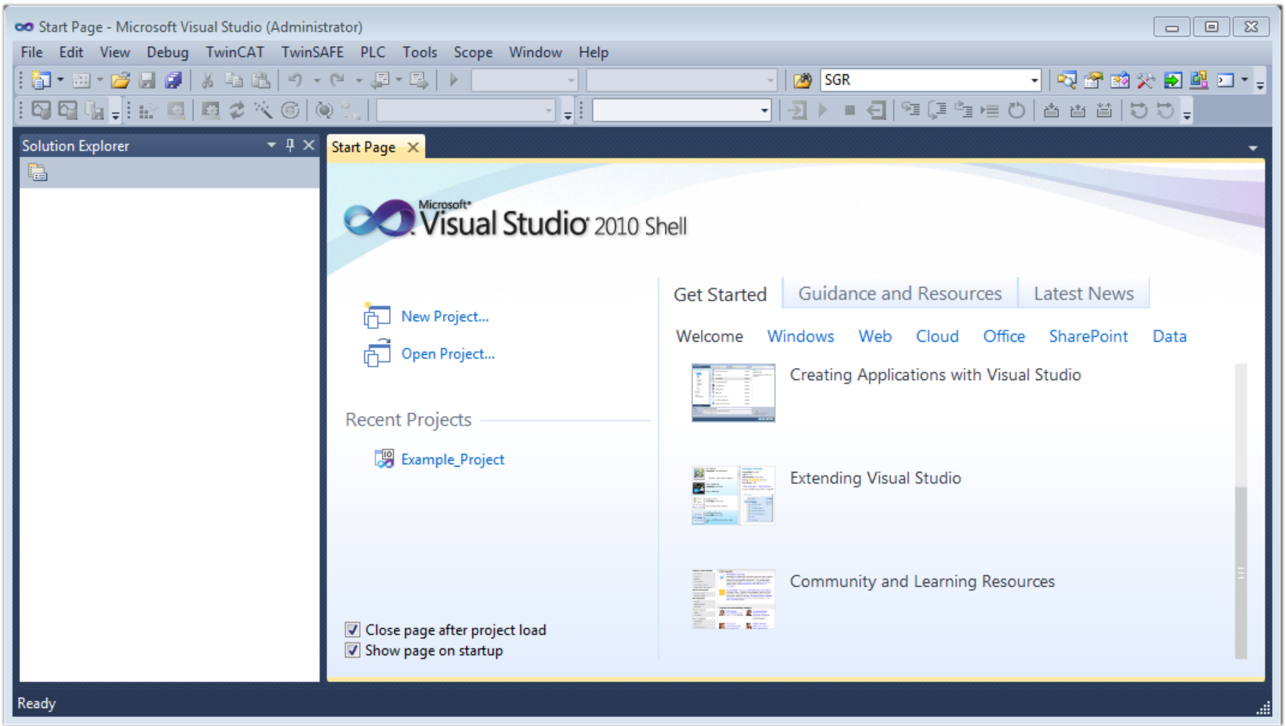


Fig. 42: Initial TwinCAT 3 user interface

First create a new project via 

(or under "File"→“New”→ "Project…"). In the following dialog make the corresponding entries as required (as shown in the diagram):

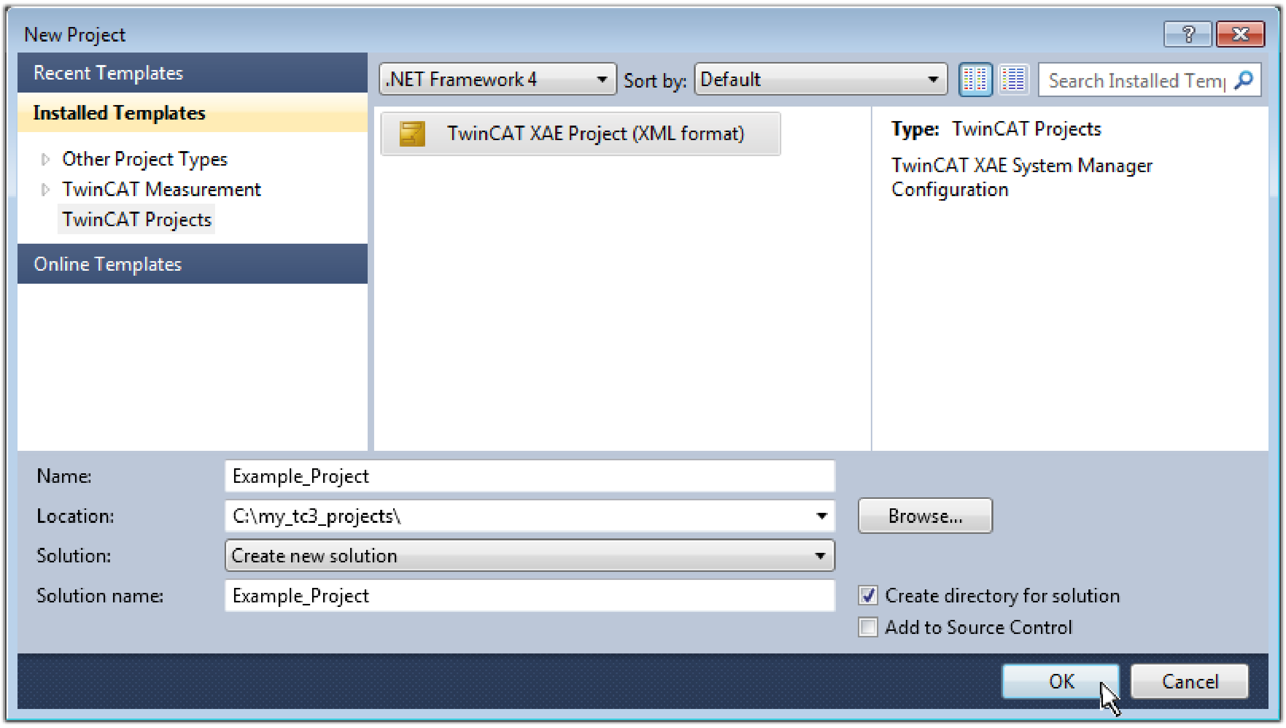


Fig. 43: Create new TwinCAT project

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

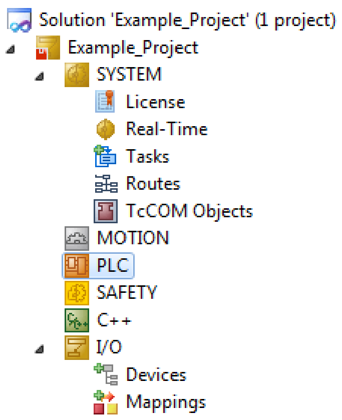
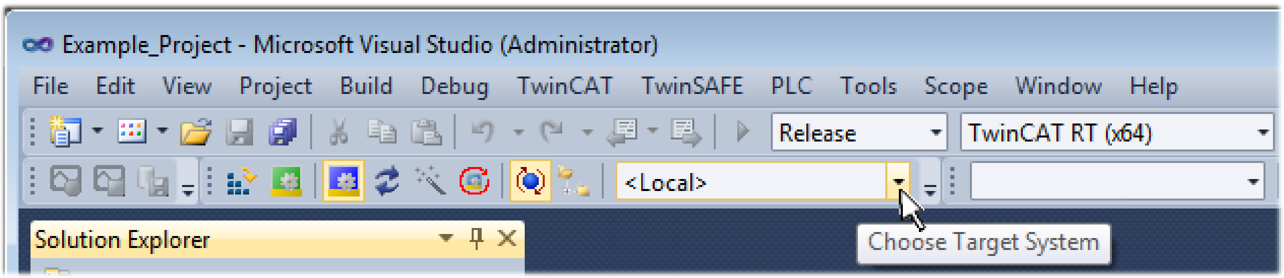


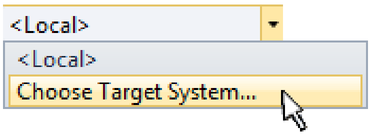
Fig. 44: New TwinCAT3 project in the project folder explorer

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC, TwinCAT can be used in local mode and thereby the next step is "Insert Device [▸ 45]".

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



expand the pull-down menu:



and open the following window:

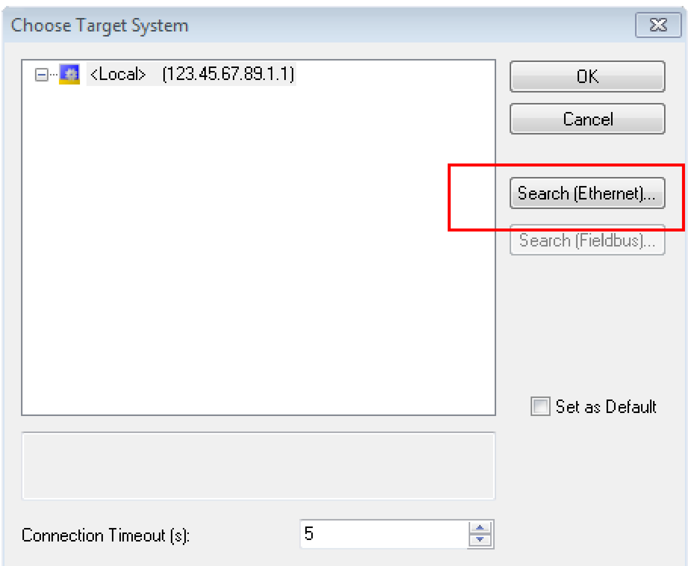


Fig. 45: Selection dialog: Choose the target system

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

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

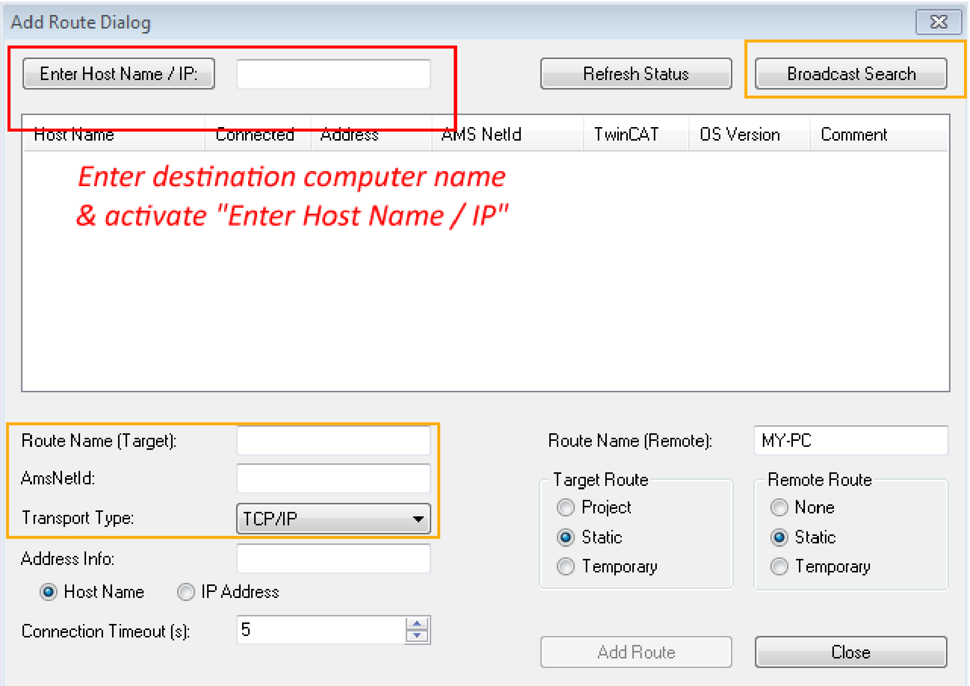
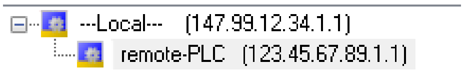


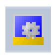
Fig. 46: Specify the PLC for access by the TwinCAT System Manager: selection of the target system

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



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

Adding devices

In the project folder explorer of the Visual Studio shell user interface on the left, select "Devices" within element “I/O”, then right-click to open a context menu and select "Scan" or start the action via  in the menu bar. The TwinCAT System Manager may first have to be set to "Config mode" via  or via the menu "TwinCAT" → "Restart TwinCAT (Config mode)".

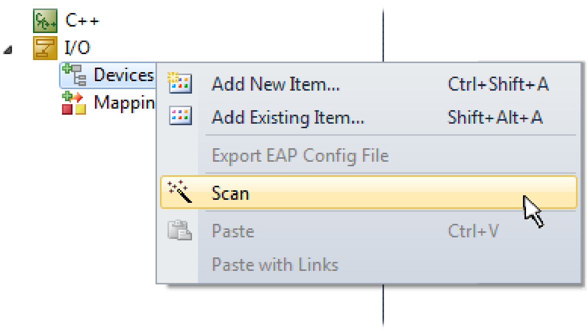


Fig. 47: Select "Scan"

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

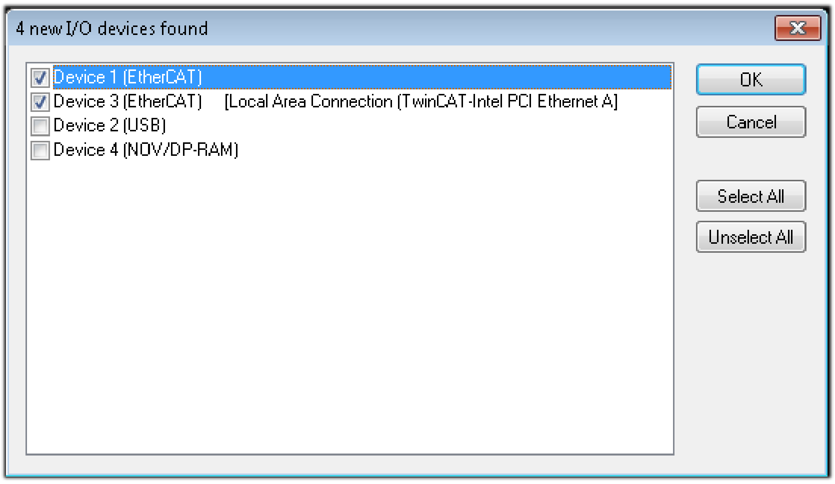


Fig. 48: Automatic detection of I/O devices: selection the devices to be integrated

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

Based on the sample configuration [▶ 31] described at the beginning of this section, the result is as follows:

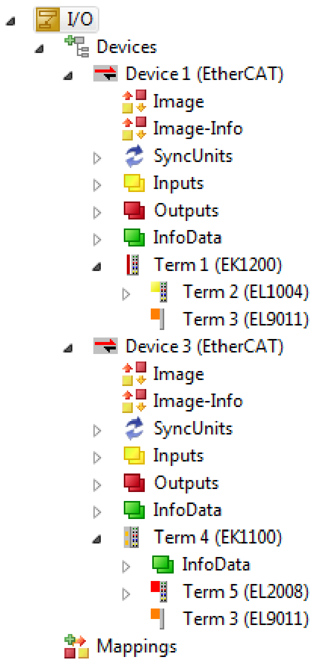


Fig. 49: Mapping of the configuration in VS shell of the TwinCAT3 environment

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

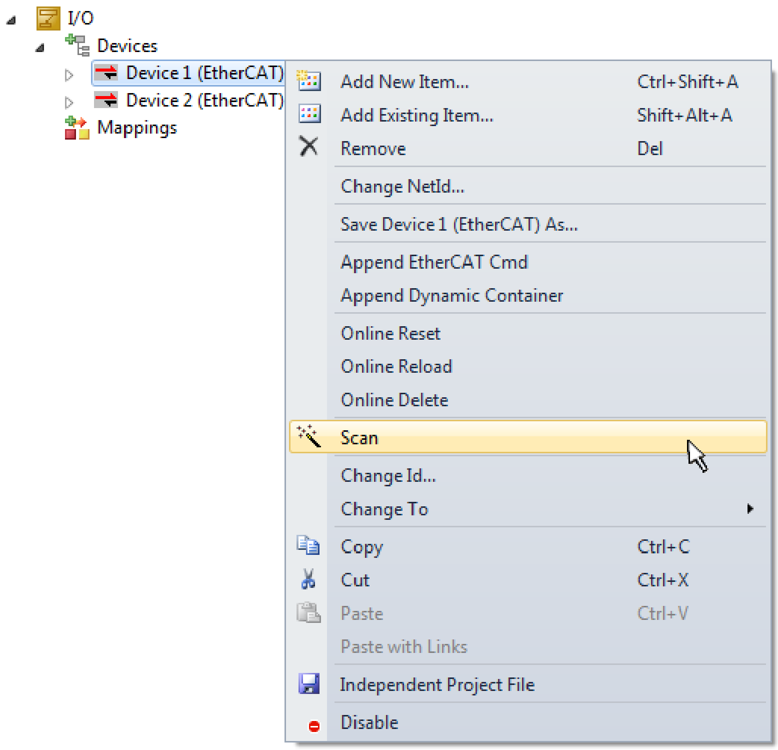


Fig. 50: Reading of individual terminals connected to a device

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

Programming the PLC

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

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

The following section refers to Structured Text (ST).

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

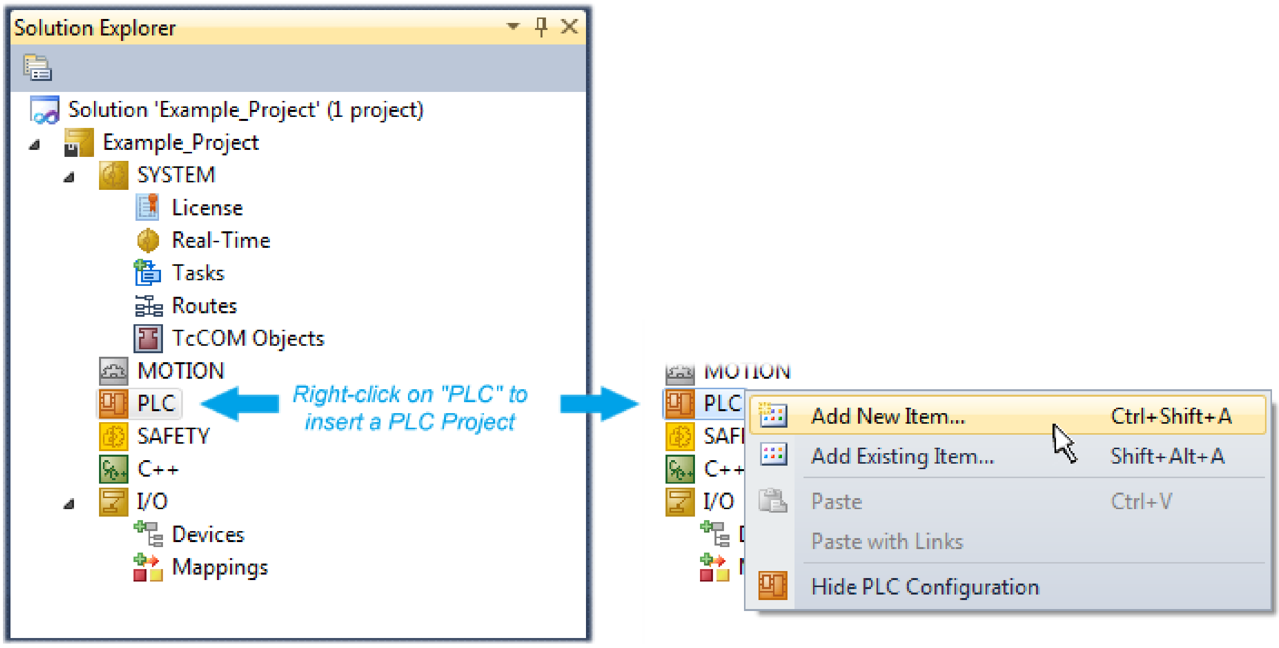


Fig. 51: Adding the programming environment in "PLC"

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

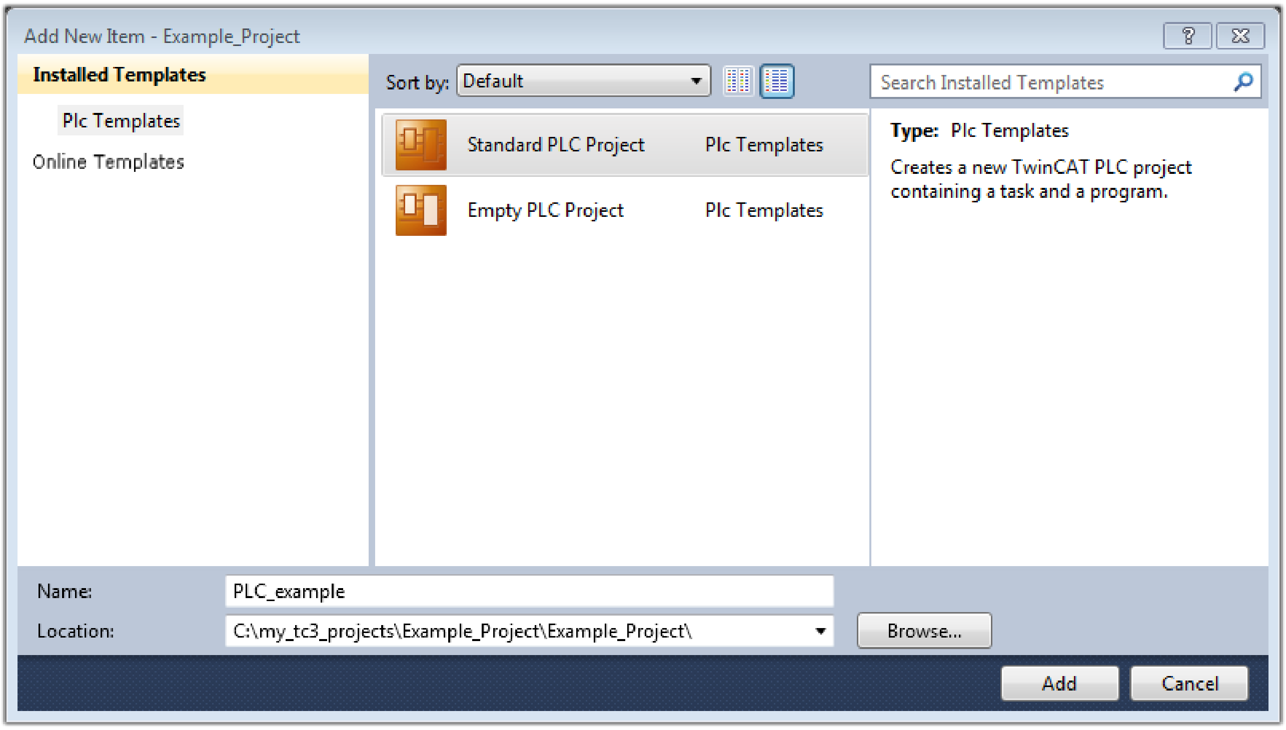


Fig. 52: Specifying the name and directory for the PLC programming environment

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

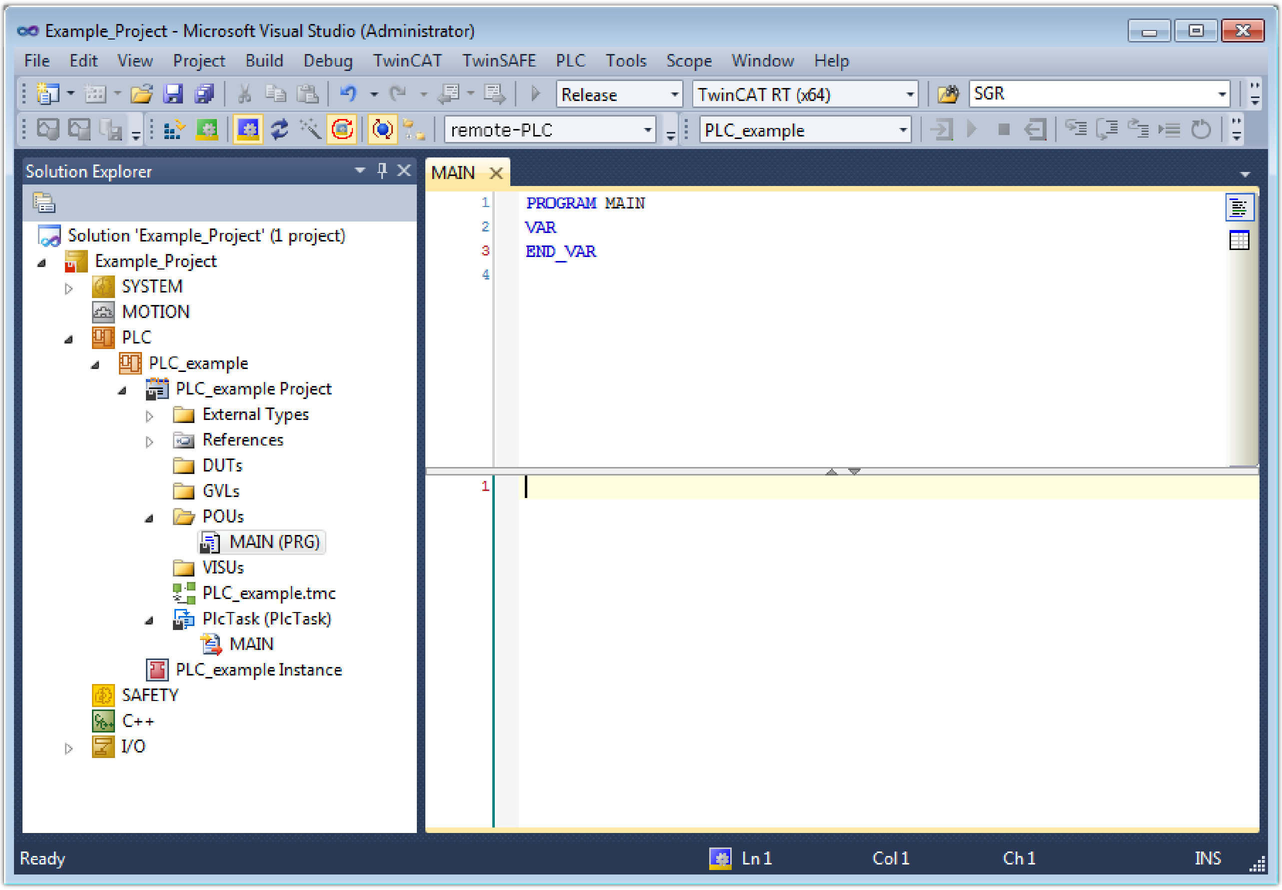


Fig. 53: Initial "Main" program of the standard PLC project

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

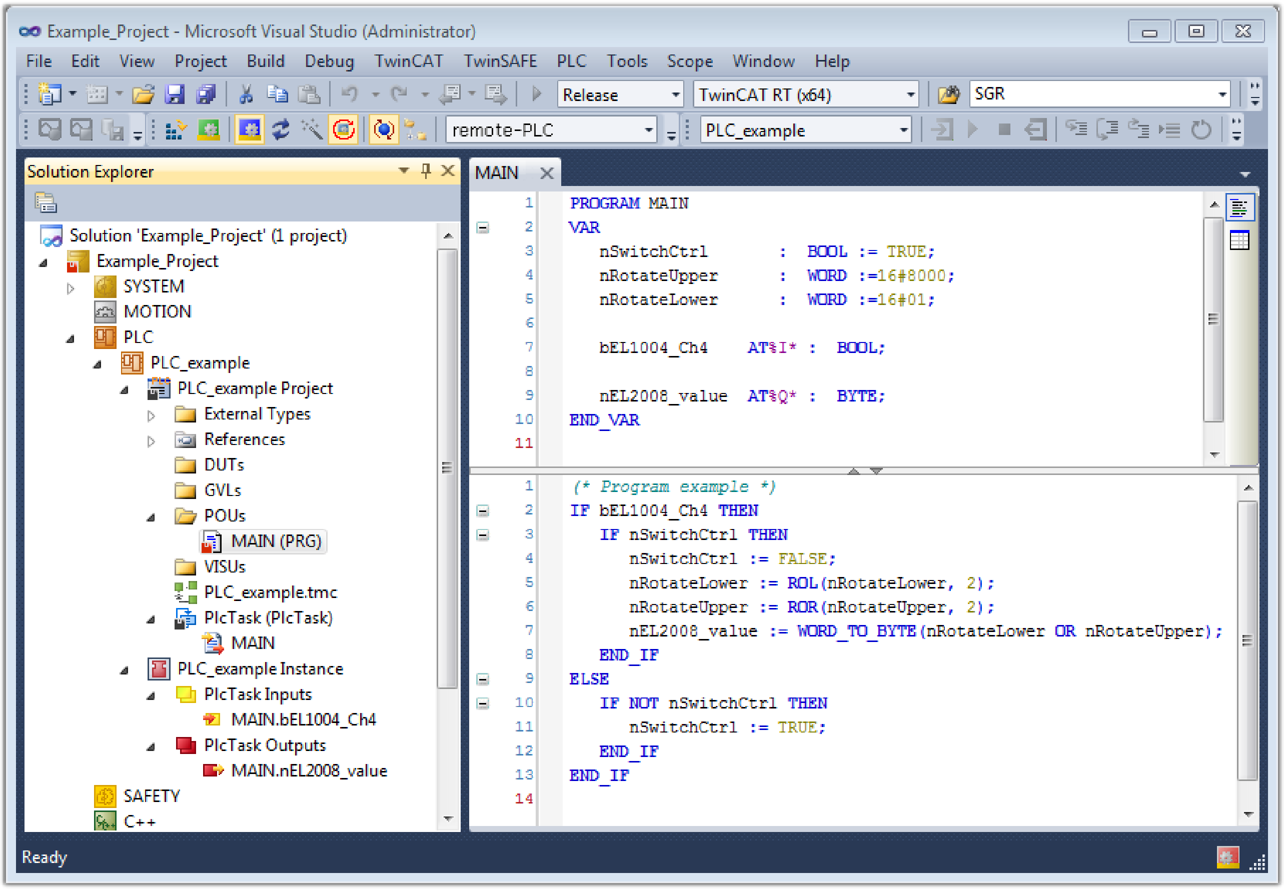


Fig. 54: Sample program with variables after a compile process (without variable integration)

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

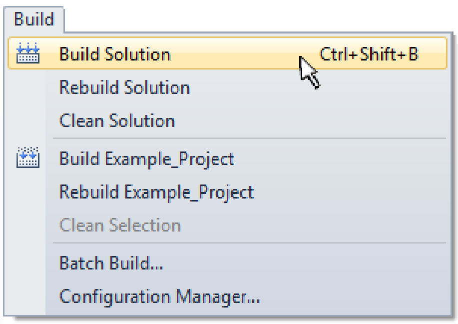
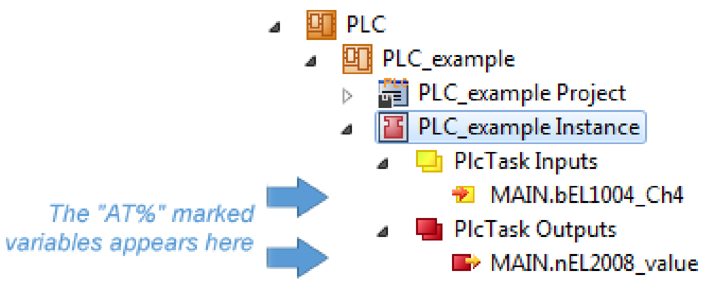


Fig. 55: Start program compilation

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



Assigning variables

Via the menu of an instance - variables in the "PLC" context, use the "Modify Link…" option to open a window for selecting a suitable process object (PDO) for linking:

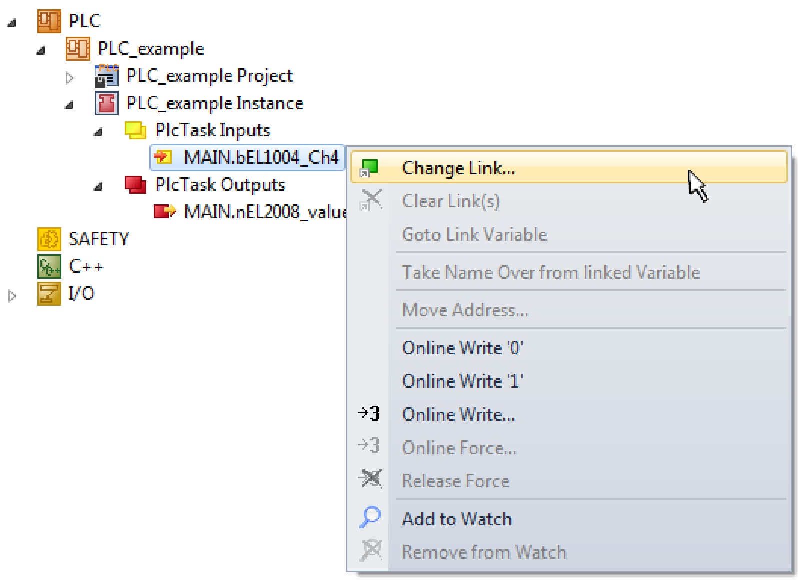


Fig. 56: Creating the links between PLC variables and process objects

In the window that opens, the process object for the variable "bEL1004_Ch4" of type BOOL can be selected from the PLC configuration tree:

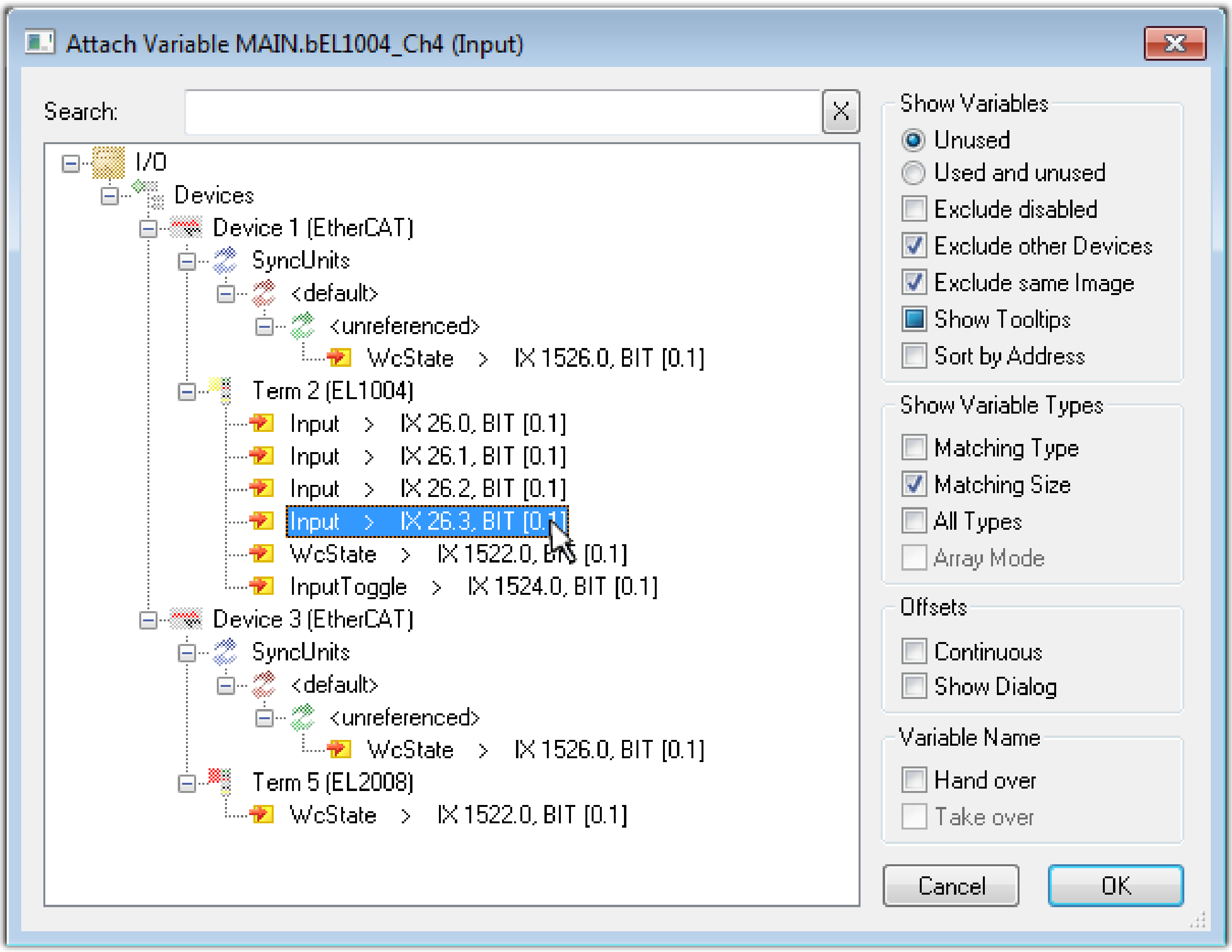


Fig. 57: Selecting PDO of type BOOL

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

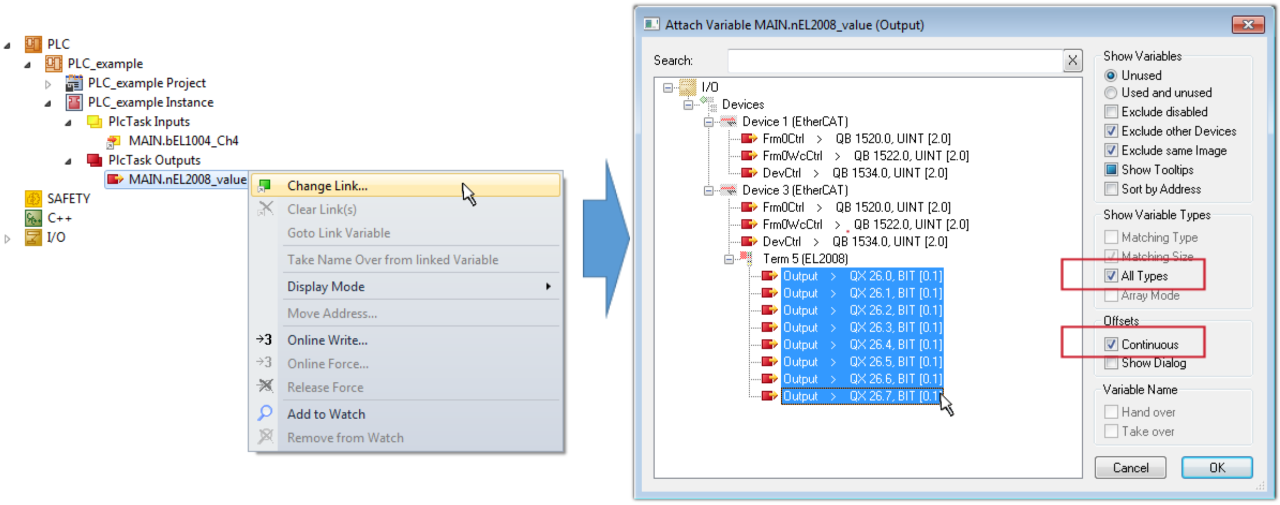


Fig. 58: Selecting several PDOs simultaneously: activate "Continuous" and "All types"

Note that the "Continuous" checkbox was also activated. This is designed to allocate the bits contained in the byte of the variable "nEL2008_value" sequentially to all eight selected output bits of the EL2008 terminal. In this way it is possible to subsequently address all eight outputs of the terminal in the program with a byte corresponding to bit 0 for channel 1 to bit 7 for channel 8 of the PLC. A special symbol ( ) at the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting a "Goto Link Variable” from the context menu of a variable. The object opposite, in this case the PDO, is automatically selected:

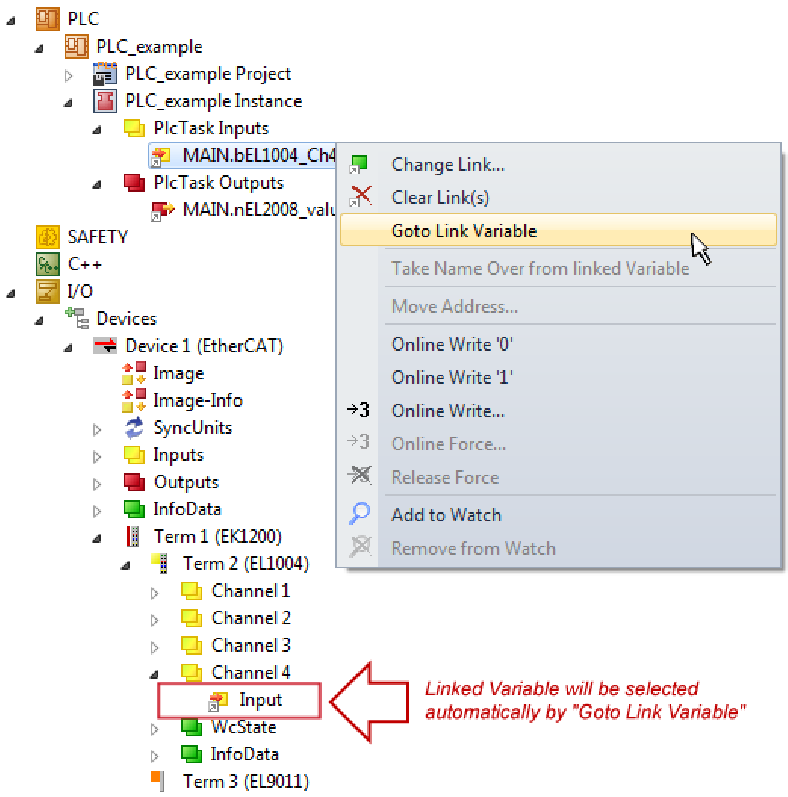
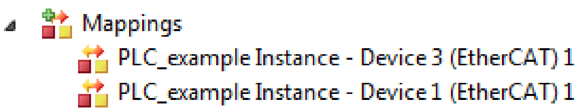


Fig. 59: Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample

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

Activation of the configuration

The allocation of PDO to PLC variables has now established the connection from the controller to the inputs and outputs of the terminals. The configuration can now be activated with or via the menu under "TwinCAT" in order to transfer settings of the development environment to the runtime system. Confirm the messages "Old configurations are overwritten!" and "Restart TwinCAT system in Run mode" with "OK". The corresponding assignments can be seen in the project folder explorer:



A few seconds later the corresponding status of the Run mode is displayed in the form of a rotating symbol at the bottom right of the VS shell development environment. The PLC system can then be started as described below.

Starting the controller

Select the menu option "PLC" → "Login" or click on to link the PLC with the real-time system and load the control program for execution. This results in the message *"No program on the controller! Should the new program be loaded?"*, which should be acknowledged with "Yes". The runtime environment is ready for program start by click on symbol , the "F5" key or via "PLC" in the menu selecting “Start”. The started programming environment shows the runtime values of individual variables:

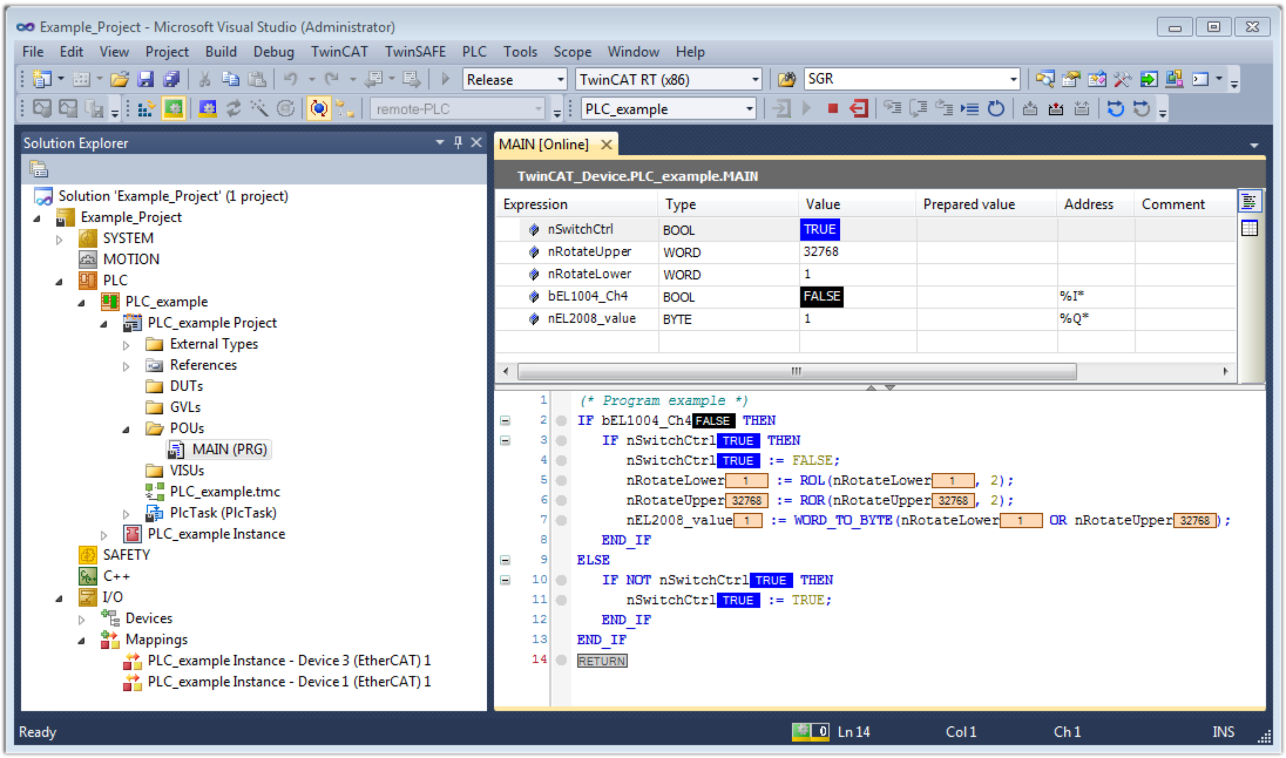


Fig. 60: TwinCAT development environment (VS shell): logged-in, after program startup

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

4.2 TwinCAT Development Environment

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

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

Details:

- **TwinCAT 2:**
    - Connects I/O devices to tasks in a variable-oriented manner
    - Connects tasks to tasks in a variable-oriented manner
    - Supports units at the bit level
    - Supports synchronous or asynchronous relationships
    - Exchange of consistent data areas and process images
    - Datalink on NT - Programs by open Microsoft Standards (OLE, OCX, ActiveX, DCOM+, etc.)
    - Integration of IEC 61131-3-Software-SPS, Software- NC and Software-CNC within Windows NT/2000/XP/Vista, Windows 7, NT/XP Embedded, CE
    - Interconnection to all common fieldbusses
    - More...

Additional features:

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

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

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

4.2.1 Installation of the TwinCAT real-time driver

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

This can be done in several ways. One option is described here.

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

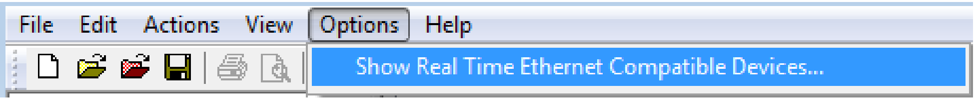


Fig. 61: *System Manager “Options” (TwinCAT 2)*

This have to be called up by the Menü “TwinCAT” within the TwinCAT 3 environment:

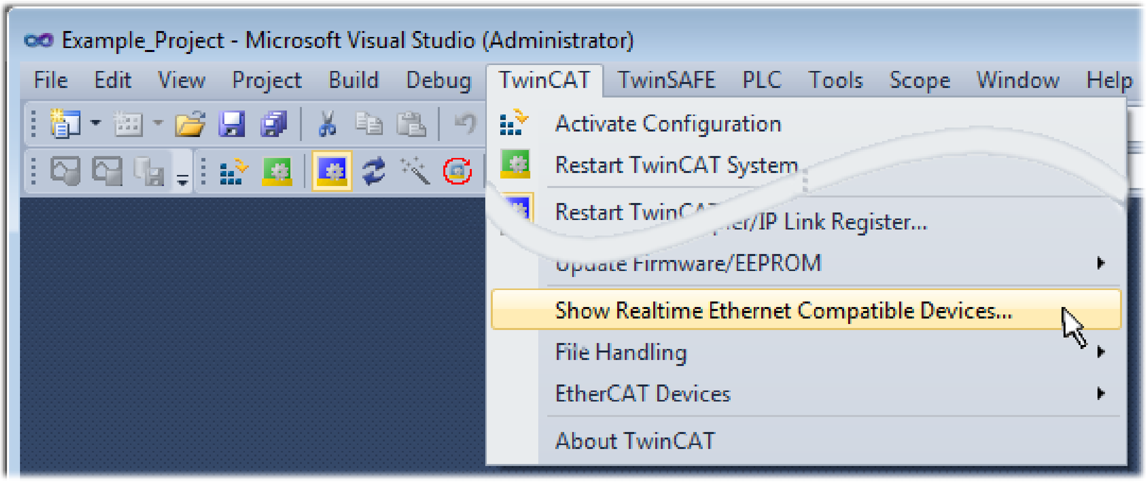


Fig. 62: *Call up under VS Shell (TwinCAT 3)*

The following dialog appears:

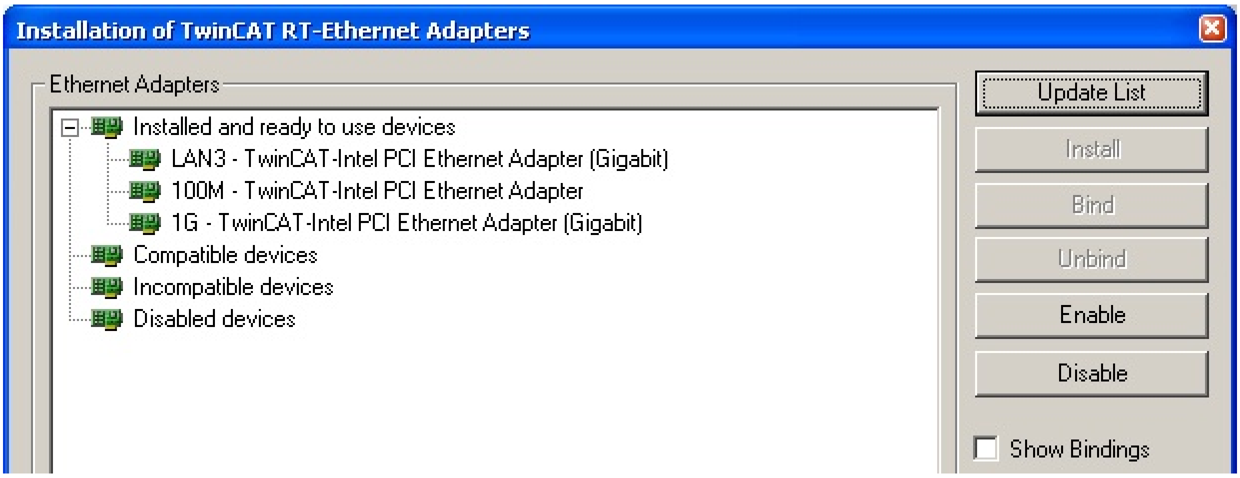


Fig. 63: *Overview of network interfaces*

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

A Windows warning regarding the unsigned driver can be ignored.

Alternatively an EtherCAT-device can be inserted first of all as described in chapter Offline configuration creation, section “Creating the EtherCAT device” [▶ 65] in order to view the compatible ethernet ports via its EtherCAT properties (tab „Adapter“, button „Compatible Devices…“):

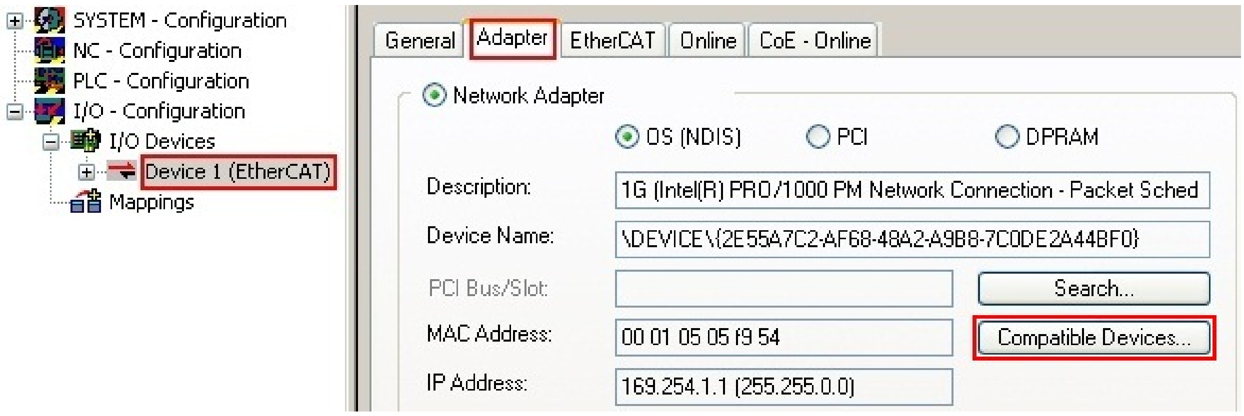
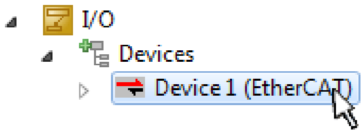


Fig. 64: *EtherCAT device properties(TwinCAT 2): click on „Compatible Devices…“ of tab “Adapter”*

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



After the installation the driver appears activated in the Windows overview for the network interface (Windows Start → System Properties → Network)

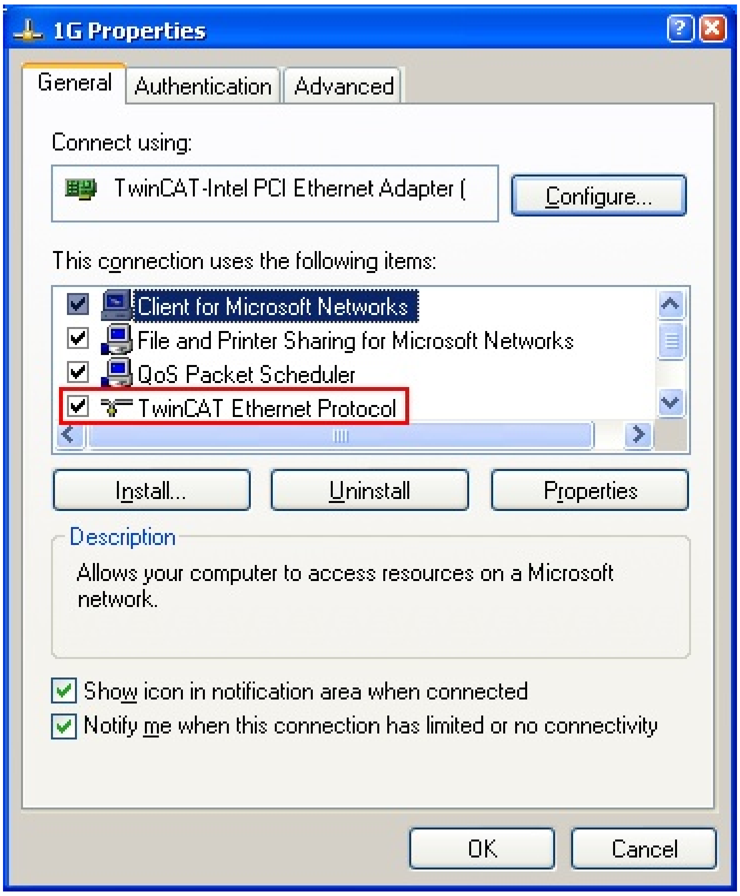


Fig. 65: *Windows properties of the network interface*

A correct setting of the driver could be:

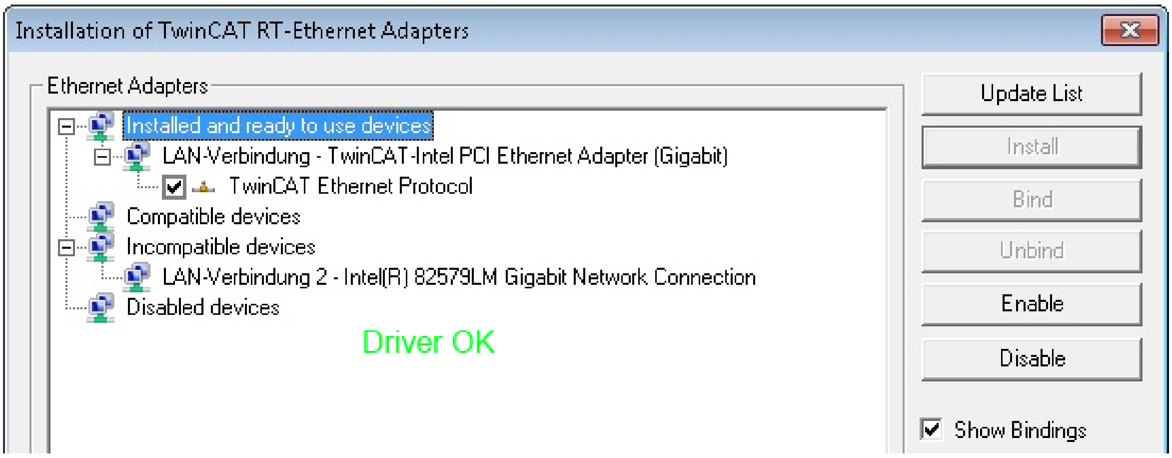


Fig. 66: *Exemplary correct driver setting for the Ethernet port*

Other possible settings have to be avoided:

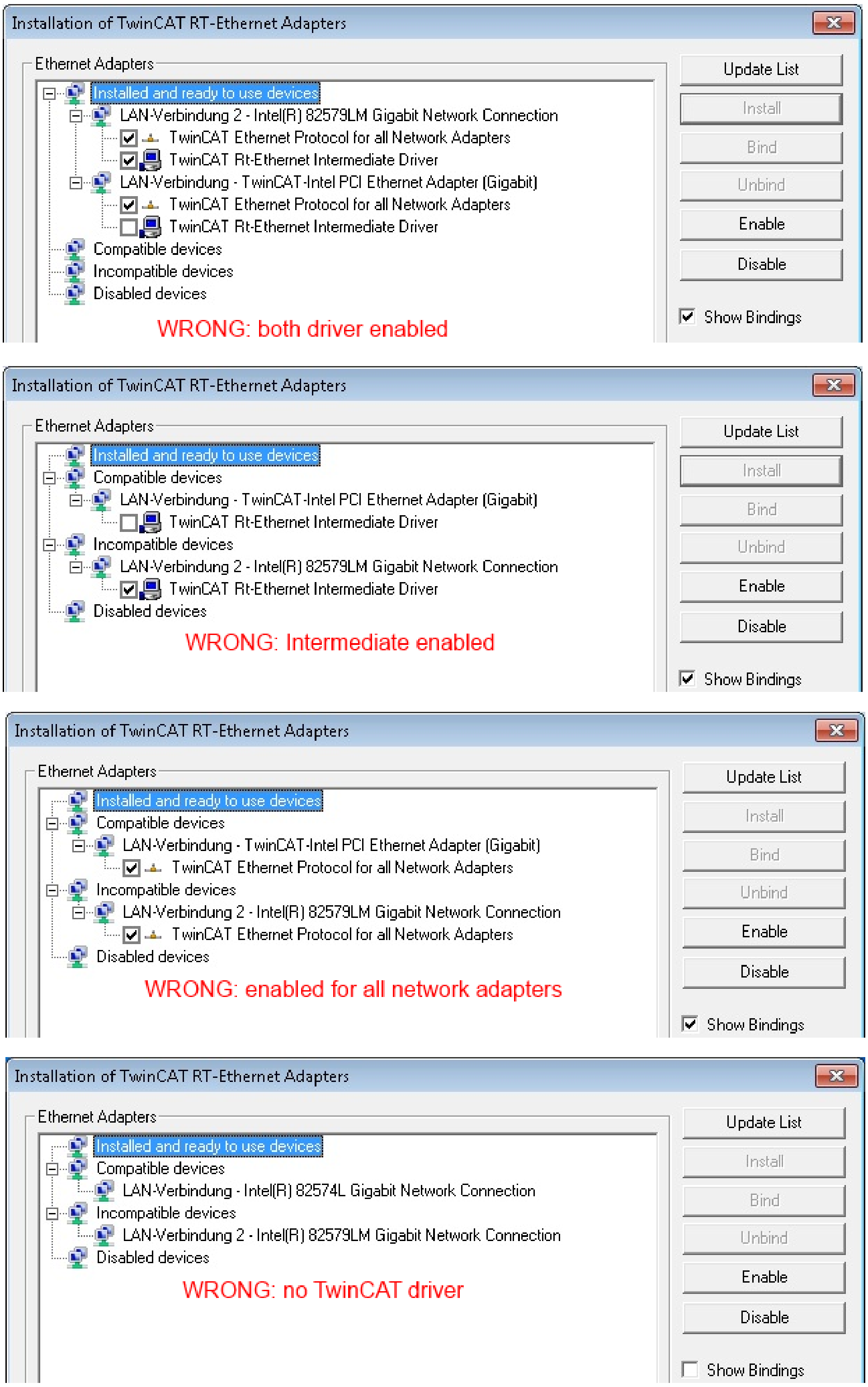


Fig. 67: *Incorrect driver settings for the Ethernet port*

IP address of the port used

Note

IP address/DHCP

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

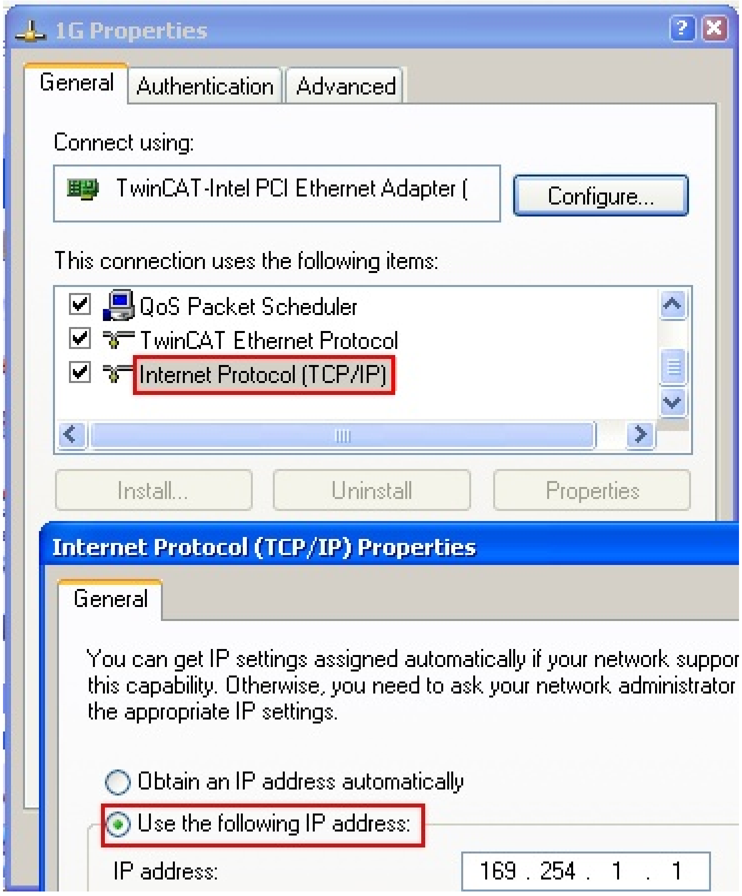


Fig. 68: *TCP/IP setting for the Ethernet port*

4.2.2 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 [▶ 64] is available for this purpose.

Note	**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”

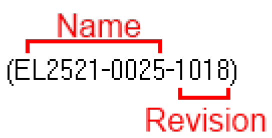


Fig. 69: *Identifier structure*

The order identifier consisting of name + type (here: EL2521-0010) describes the device function. The revision indicates the technical progress and is managed by Beckhoff. In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation. Each revision has its own ESI description. See further notes [▶ 6].

Online description

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

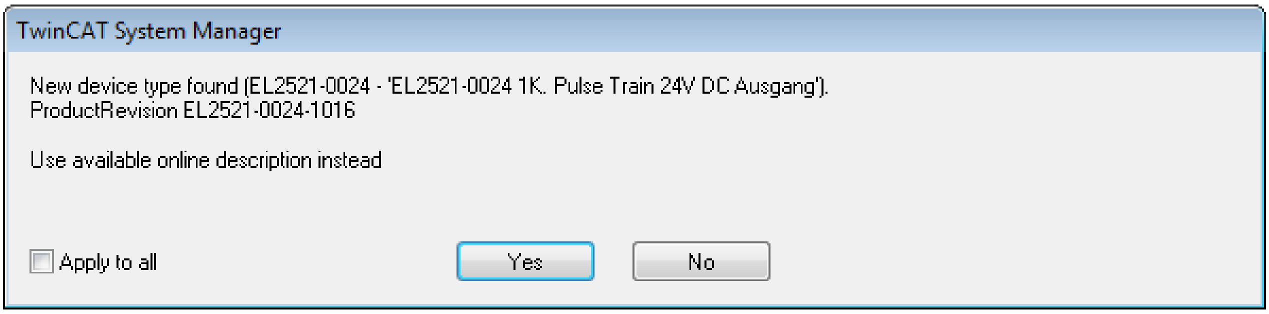


Fig. 70: *OnlineDescription information window (TwinCAT 2)*

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

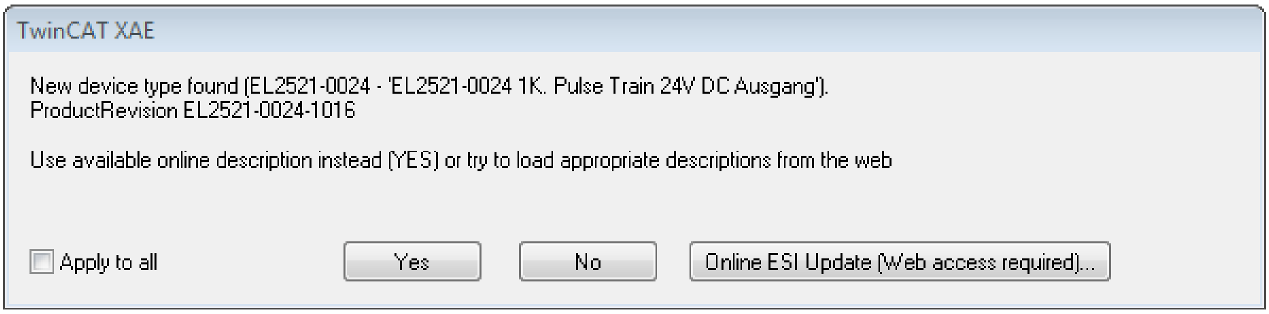


Fig. 71: *Information window OnlineDescription (TwinCAT 3)*

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

Attention	**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’ [▶ 65].

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.

Fig. 72: *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”)*.

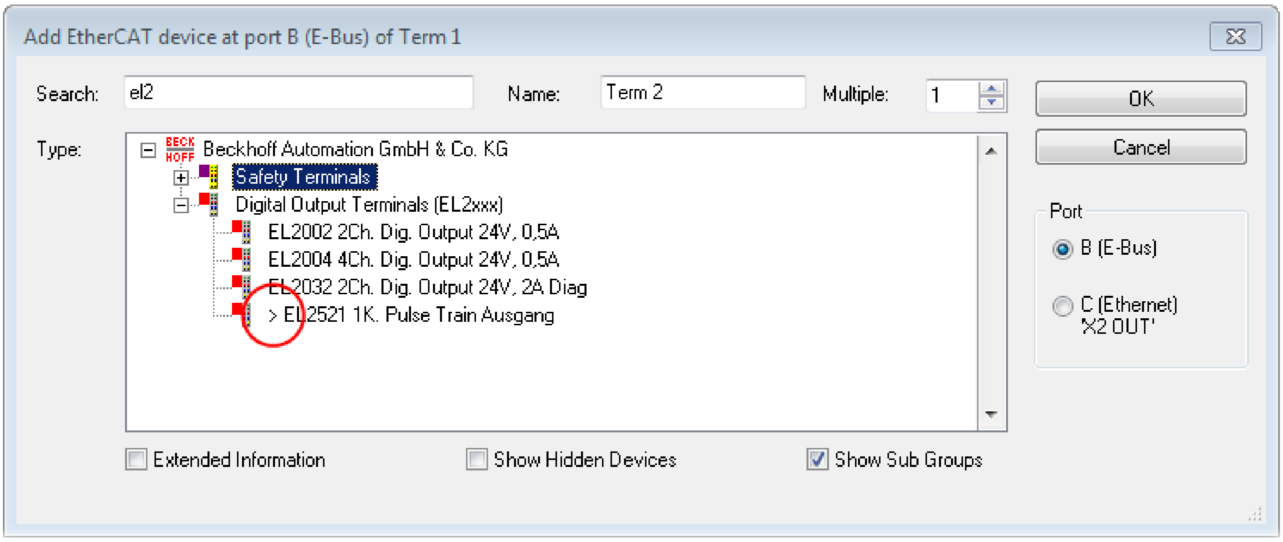


Fig. 73: *Indication of an online recorded ESI of EL2521 as an example*

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

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

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

Note	**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.

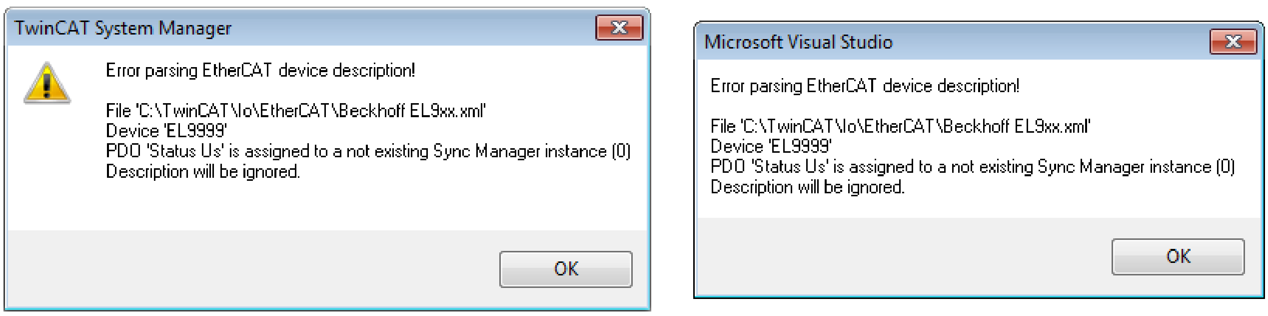


Fig. 74: *Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)*

Reasons may include:

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

4.2.3 TwinCAT ESI Updater

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

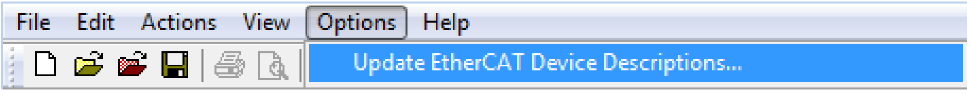


Fig. 75: *Using the ESI Updater (>= TwinCAT 2.11)*

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

Selection under TwinCAT 3:

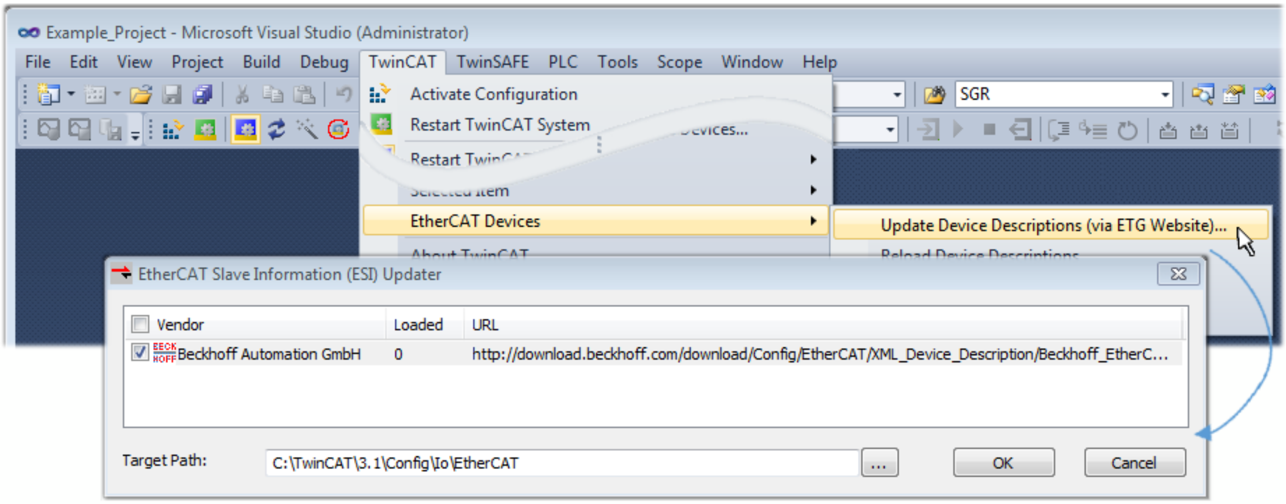


Fig. 76: *Using the ESI Updater (TwinCAT 3)*

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

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

4.2.4 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” [▶ 60].

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 [▸ 70] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▸ 71]. This step can be carried out independent of the preceding step
- troubleshooting [▸ 74]

The scan with existing configuration [▸ 75] can also be carried out for comparison.

4.2.5 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

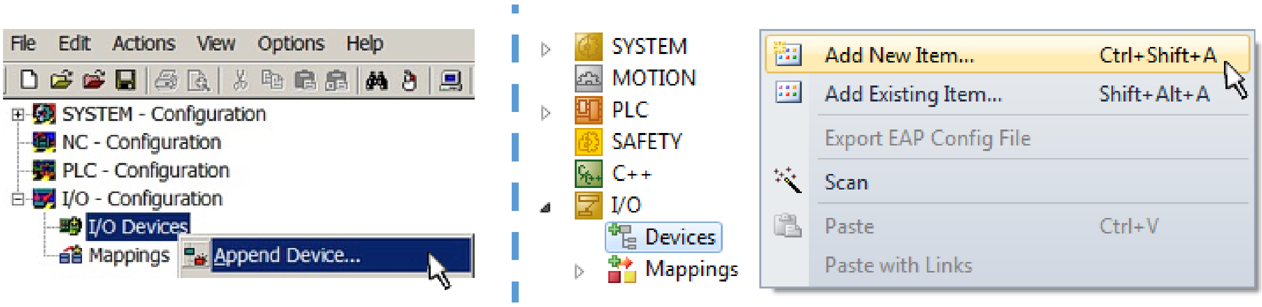


Fig. 77: *Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)*

Select type 'EtherCAT' for an EtherCAT I/O application with EtherCAT slaves. For the present publisher/subscriber service in combination with an EL6601/EL6614 terminal select "EtherCAT Automation Protocol via EL6601".

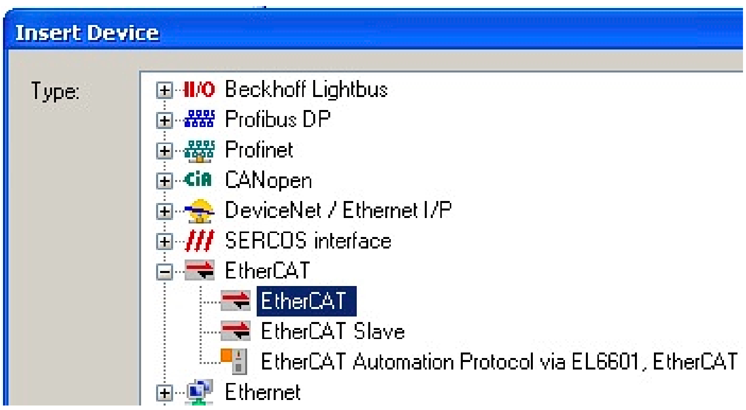


Fig. 78: *Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)*

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

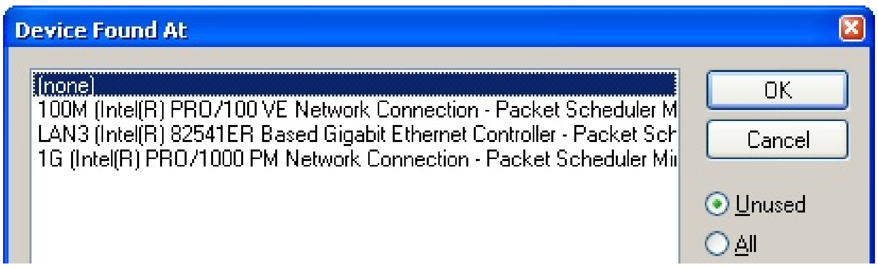


Fig. 79: *Selecting the Ethernet port*

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

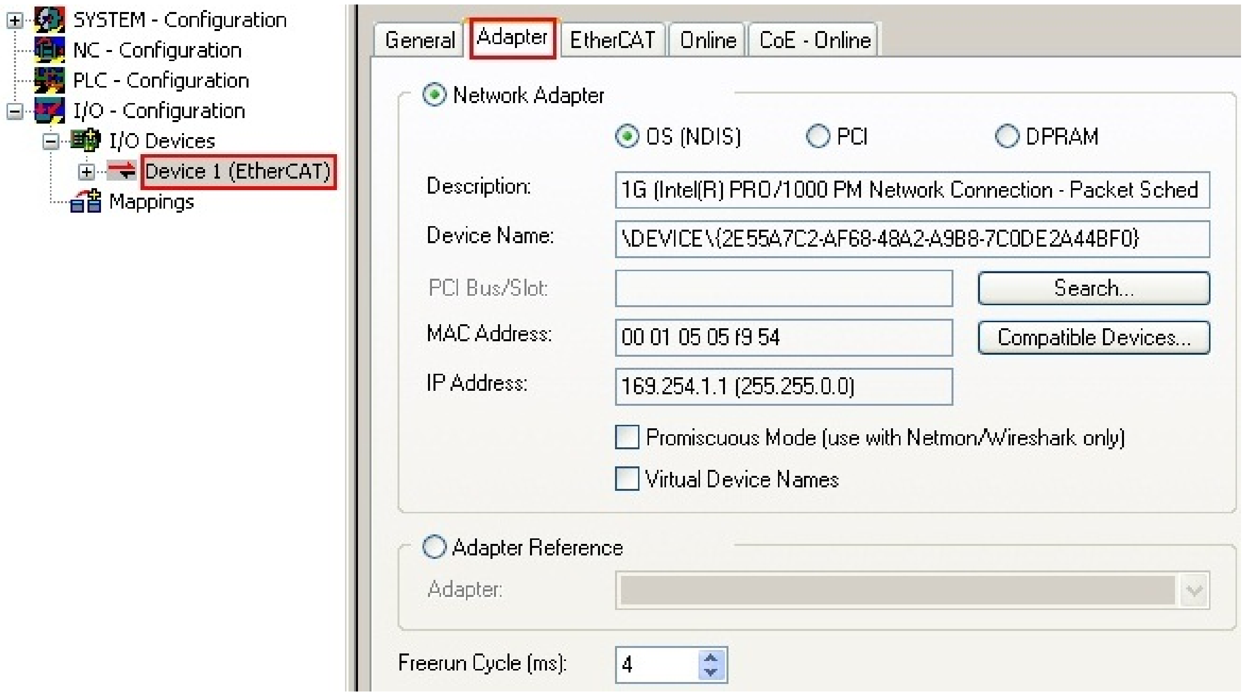


Fig. 80: *EtherCAT device properties (TwinCAT 2)*

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

Selecting the Ethernet port

Note

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 [▶ 54].

Defining EtherCAT slaves

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

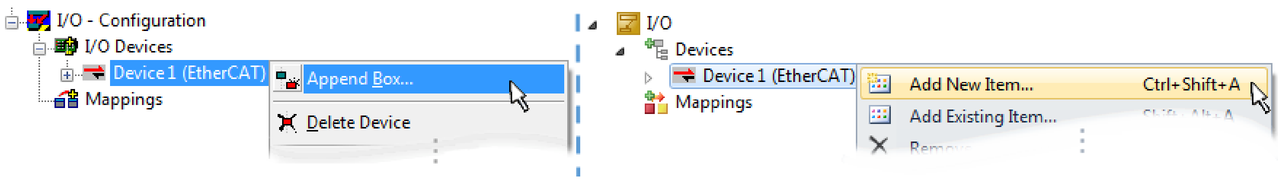


Fig. 81: *Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)*

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

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

Overview of physical layer

- “Ethernet”: cable-based 100BASE-TX: EK couplers, EP boxes, devices with RJ45/M8/M12 connector

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

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

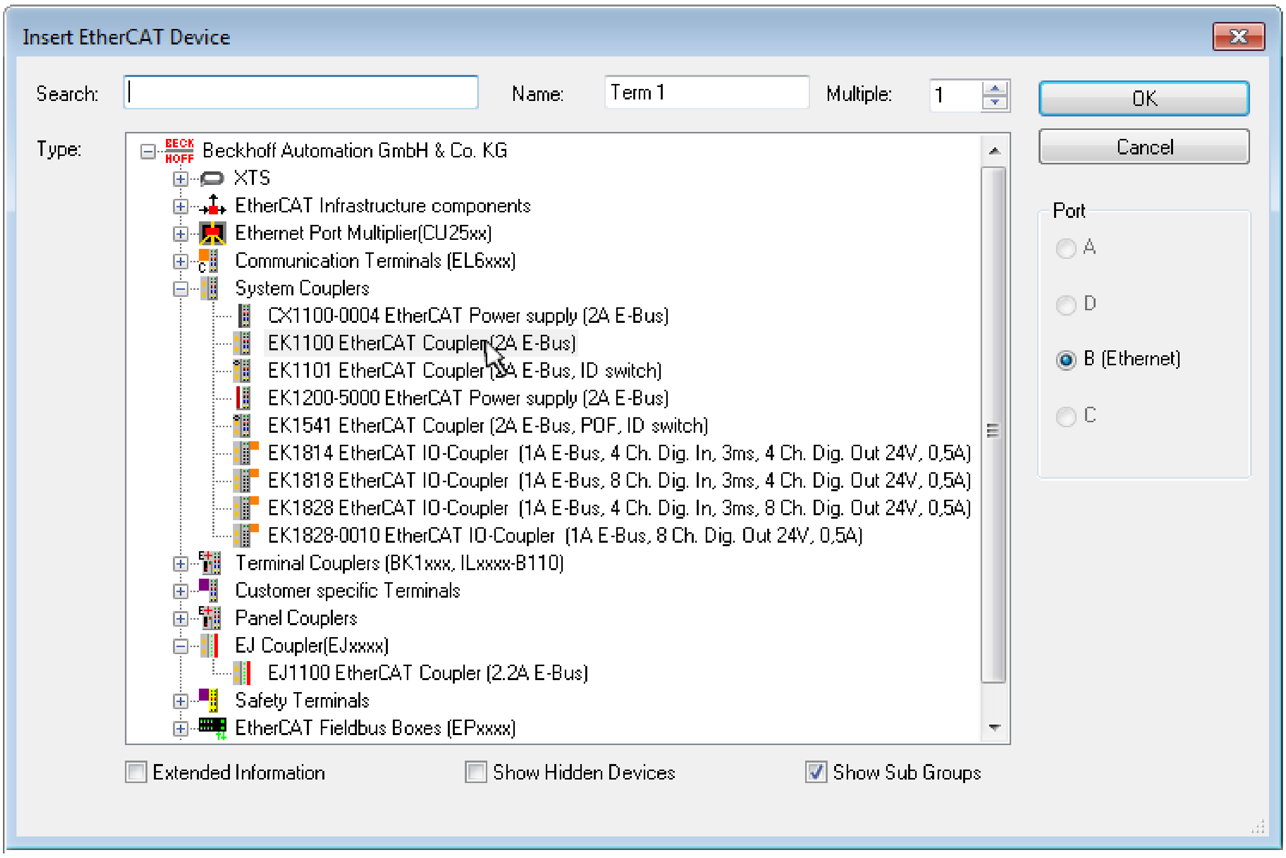


Fig. 82: *Selection dialog for new EtherCAT device*

By default only the name/device type is used as selection criterion. For selecting a specific revision of the device the revision can be displayed as “Extended Information”.

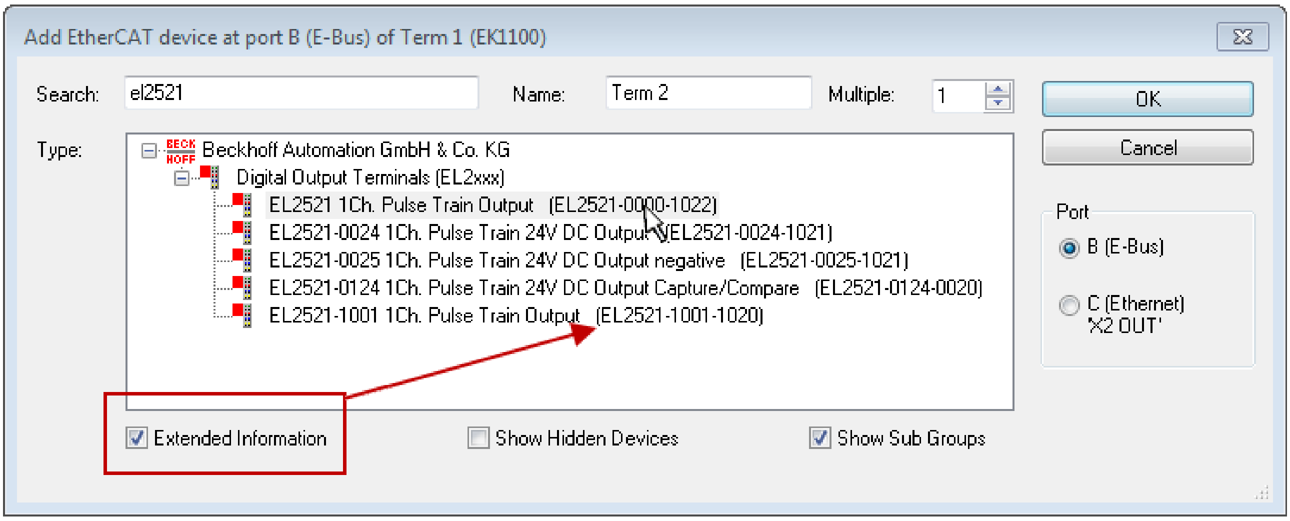


Fig. 83: *Display of device revision*

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

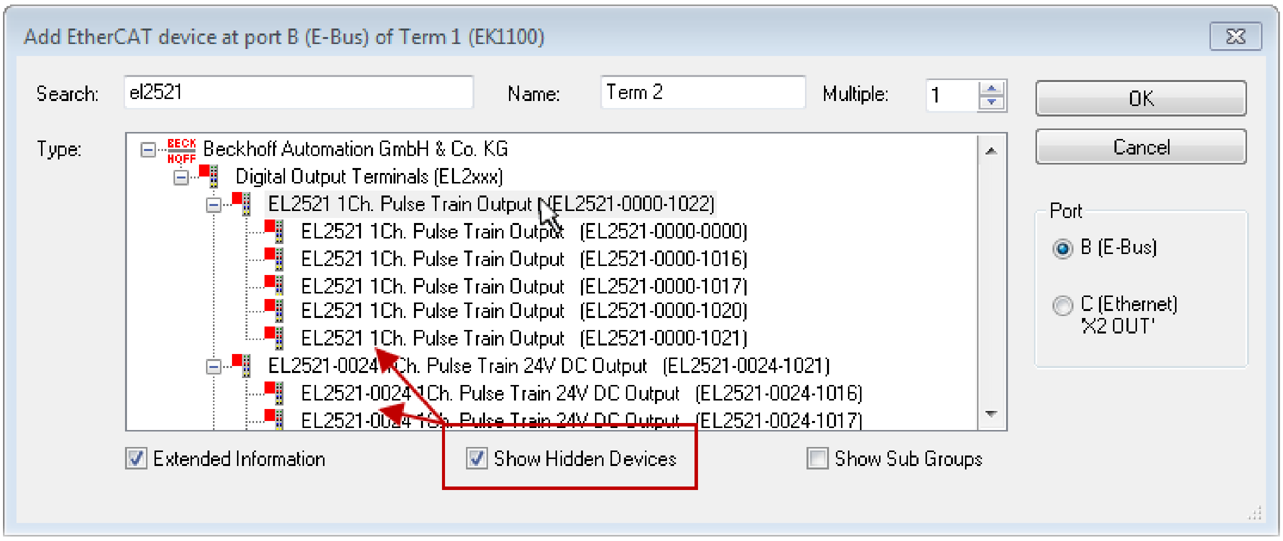


Fig. 84: *Display of previous revisions*

Note

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.

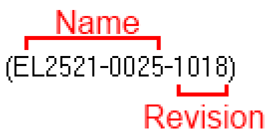


Fig. 85: *Name/revision of the terminal*

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

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

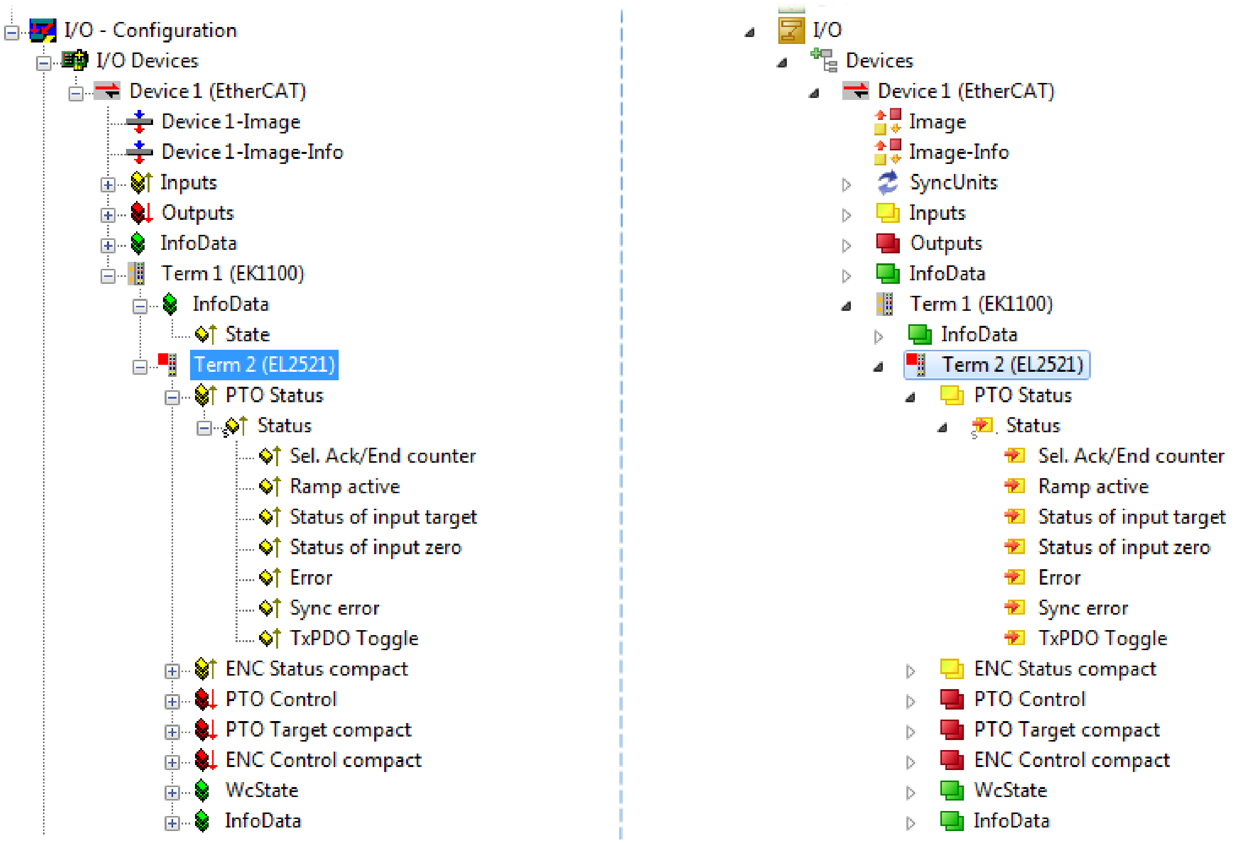


Fig. 86: *EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)*

4.2.6 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)“

Note	**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.

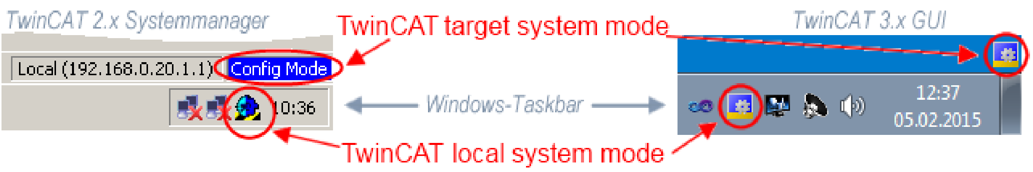


Fig. 87: *Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)*

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

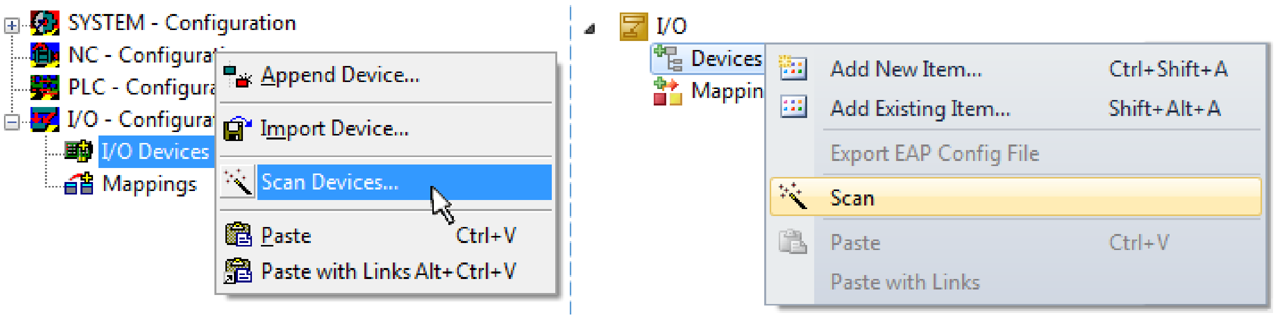


Fig. 88: *Scan Devices (left: TwinCAT 2; right: TwinCAT 3)*

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

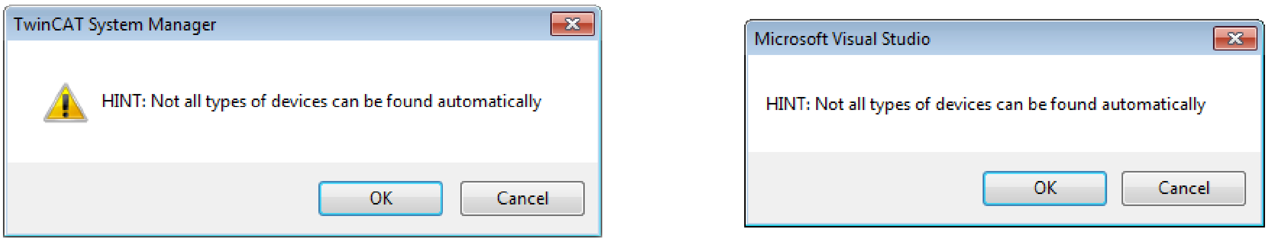


Fig. 89: *Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)*

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

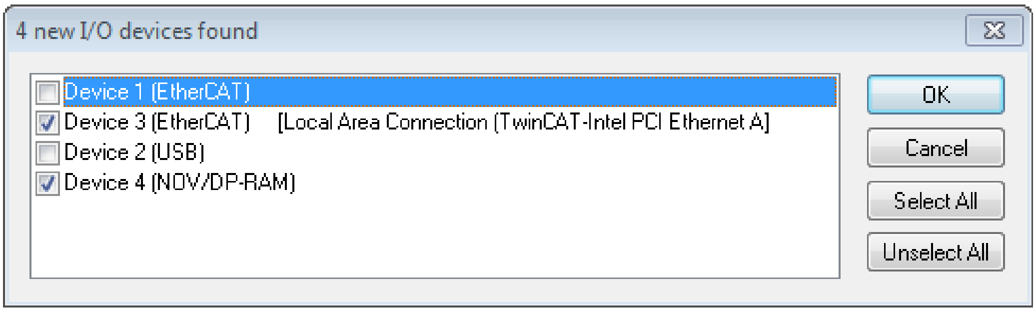


Fig. 90: *Detected Ethernet devices*

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

Note

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 [▶ 54].

Detecting/Scanning the EtherCAT devices

Note

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.

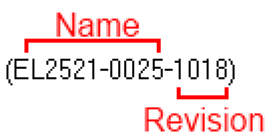


Fig. 91: *Example default state*

Attention

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 [▶ 75] 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:

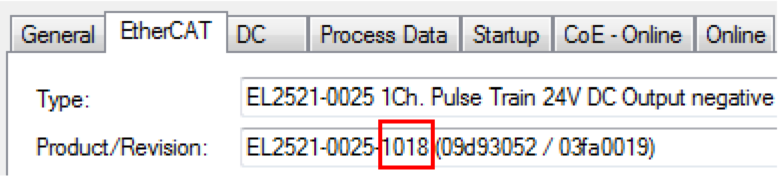


Fig. 92: *Installing EthetCAT terminal with revision -1018*

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 comparative scan [▶ 75] against the initial configuration ‘B.tsm’ in order to check the built machine.

However, if the series machine production department now doesn’t use ‘B.tsm’, but instead carries out a scan to create the productive configuration, the revision **-1019** is automatically detected and built into the configuration:

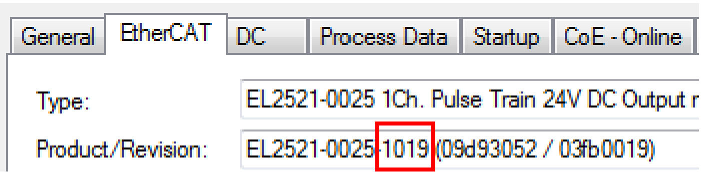


Fig. 93: *Detection of EtherCAT terminal with revision -1019*

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

In addition, it could be the case that, due to the development accompanying production in company A, the new feature C of the EL2521-0025-1019 (for example, an improved analog filter or an additional process data for the diagnosis) is discovered and used without in-house consultation. The previous stock of spare part devices are then no longer to be used for the new configuration ‘B2.tsm’ created in this way.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. 94: *Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)*

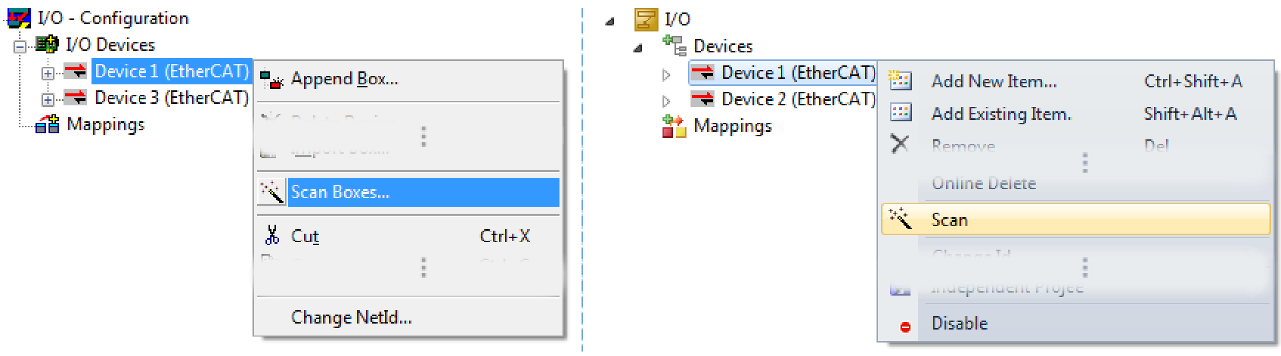


Fig. 95: *Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)*

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

Fig. 96: *Scan progressexemplary by TwinCAT 2*

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

Fig. 97: *Config/FreeRun query (left: TwinCAT 2; right: TwinCAT 3)*

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

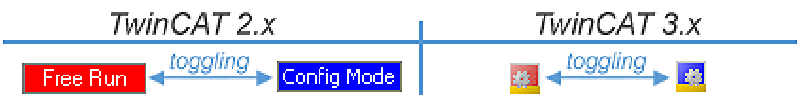


Fig. 98: *Displaying of “Free Run” and “Config Mode” toggling right below in the status bar*

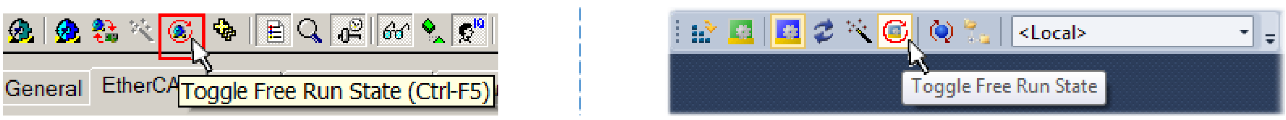


Fig. 99: *TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3)*

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

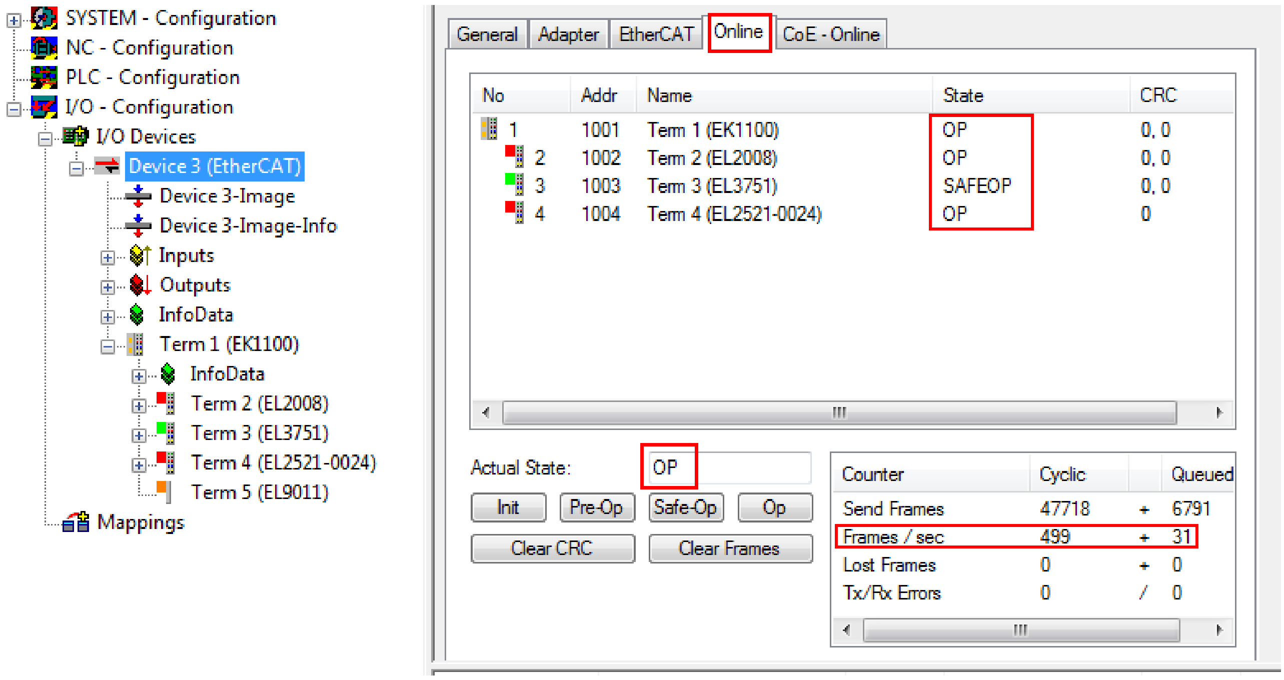


Fig. 100: *Online display example*

Please note:

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

The configuration is now complete. It can be modified as described under manual procedure [▶ 65].

Troubleshooting

Various effects may occur during scanning.

- An **unknown device** is detected, i.e. an EtherCAT slave for which no ESI XML description is available. In this case the System Manager offers to read any ESI that may be stored in the device. This case is described in the chapter "Notes regarding ESI device description".
- **Device are not detected properly**
  Possible reasons include:
  - faulty data links, resulting in data loss during the scan
  - slave has invalid device description
  The connections and devices should be checked in a targeted manner, e.g. via the emergency scan. Then re-run the scan.

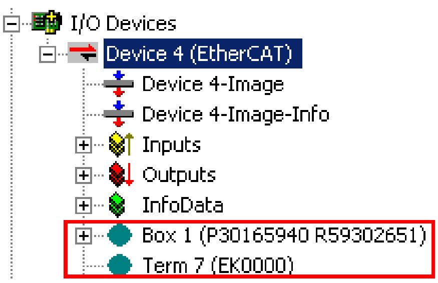


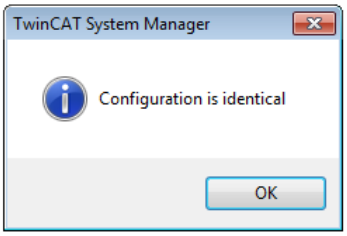
Fig. 101: *Faulty identification*

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

Scan over existing Configuration

Attention	**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.



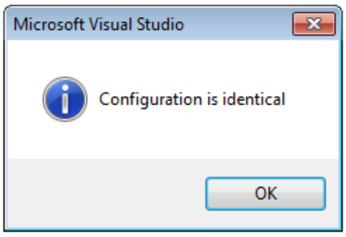


Fig. 102: *Identical configuration (left: TwinCAT 2; right: TwinCAT 3)*

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

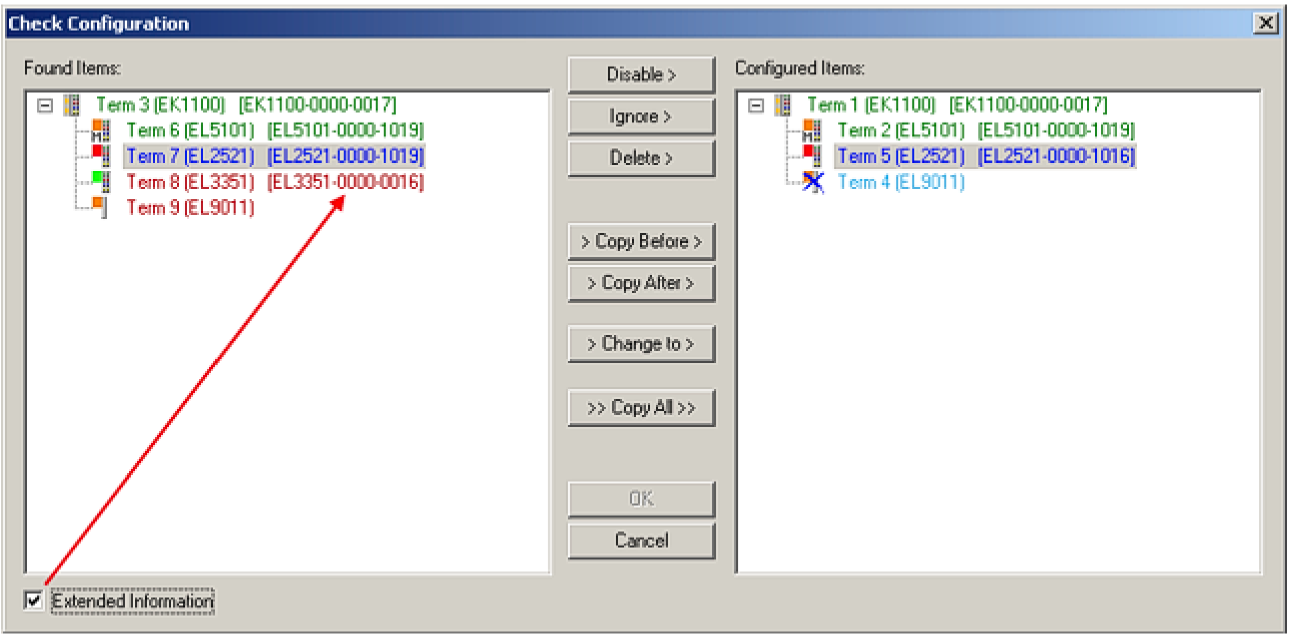


Fig. 103: *Correction dialog*

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

Colour	Explanation
green	This EtherCAT slave matches the entry on the other side. Both type and revision match.
blue	This EtherCAT slave is present on the other side, but in a different revision. This other revision can have other default values for the process data as well as other/additional functions. If the found revision is higher than the configured revision, the slave may be used provided compatibility issues are taken into account. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.
light blue	This EtherCAT slave is ignored (“Ignore” button)
red	• 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.

Note

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.

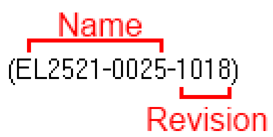


Fig. 104: *Name/revision of the terminal*

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

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

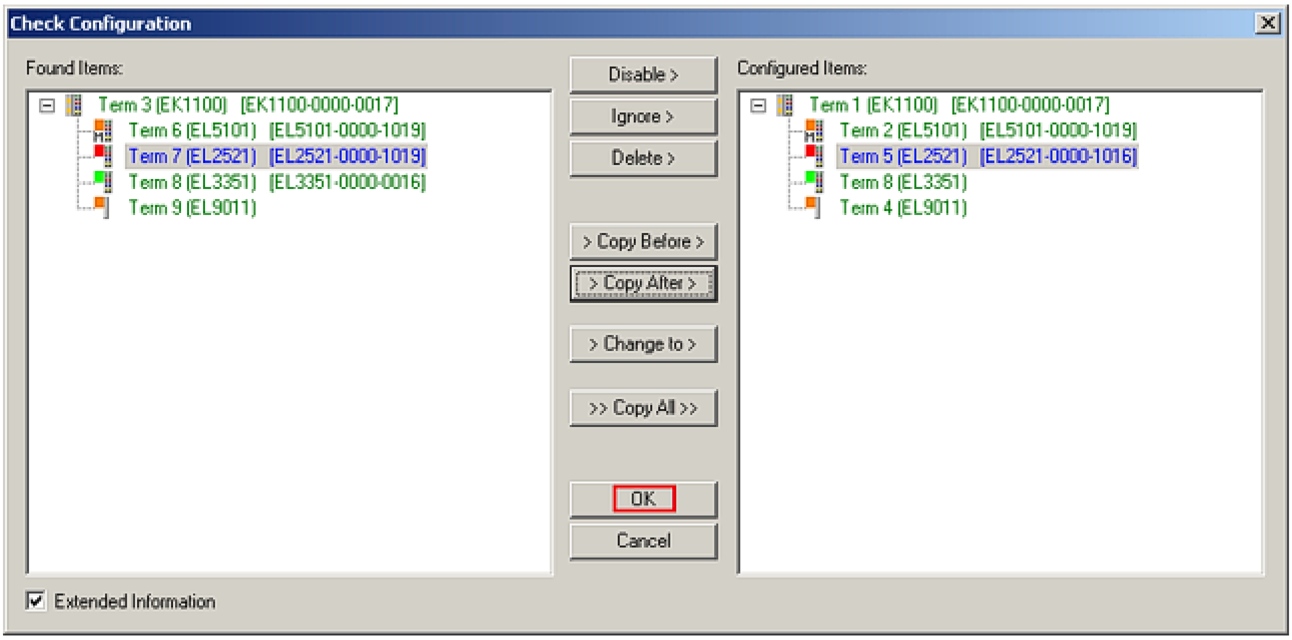


Fig. 105: *Correction dialog with modifications*

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

Change to compatible device

TwinCAT offers a function *"Change to Compatible Type…"* for the exchange of a device whilst retaining the links in the task.

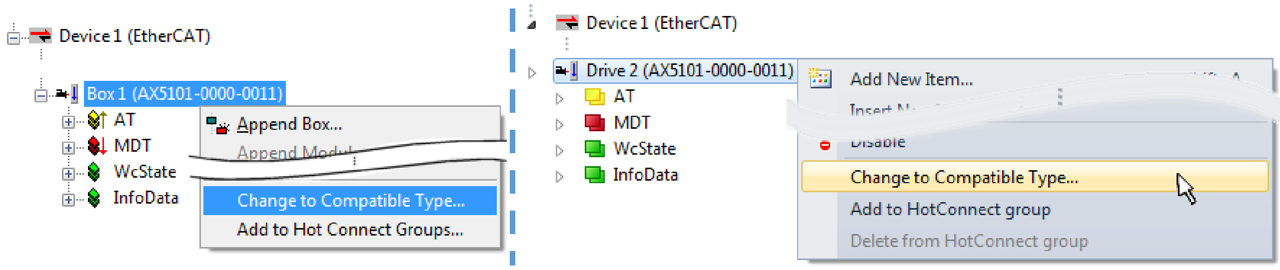


Fig. 106: *Dialog "Change to Compatible Type…" (left: TwinCAT 2; right: TwinCAT 3)*

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

Change to Alternative Type

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

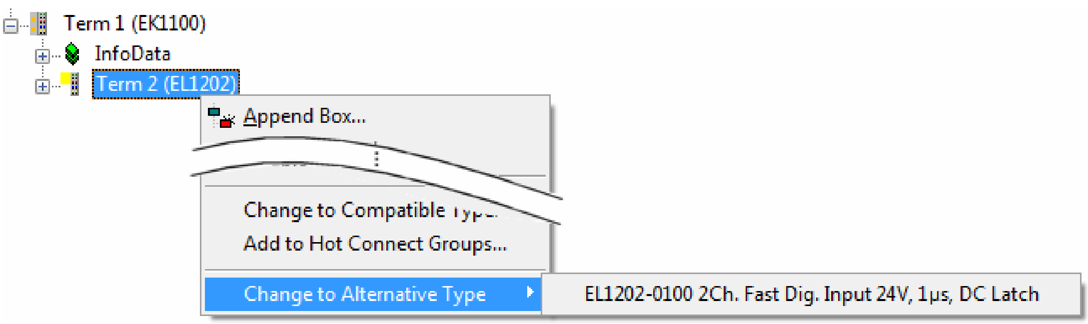


Fig. 107: *TwinCAT 2 Dialog Change to Alternative Type*

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

4.2.7 EtherCAT subscriber configuration

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

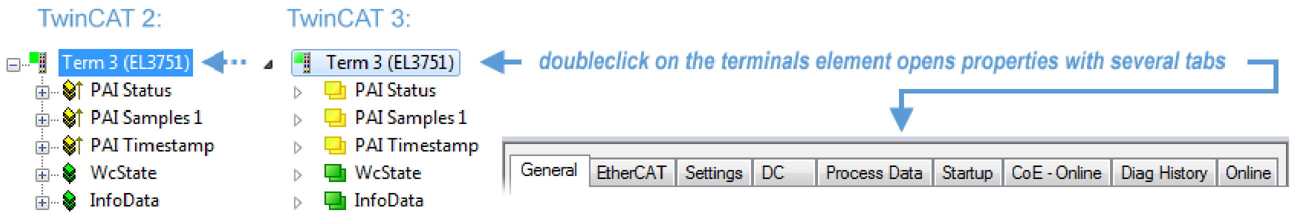


Fig. 108: *Branch element as terminal EL3751*

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

„General“ tab

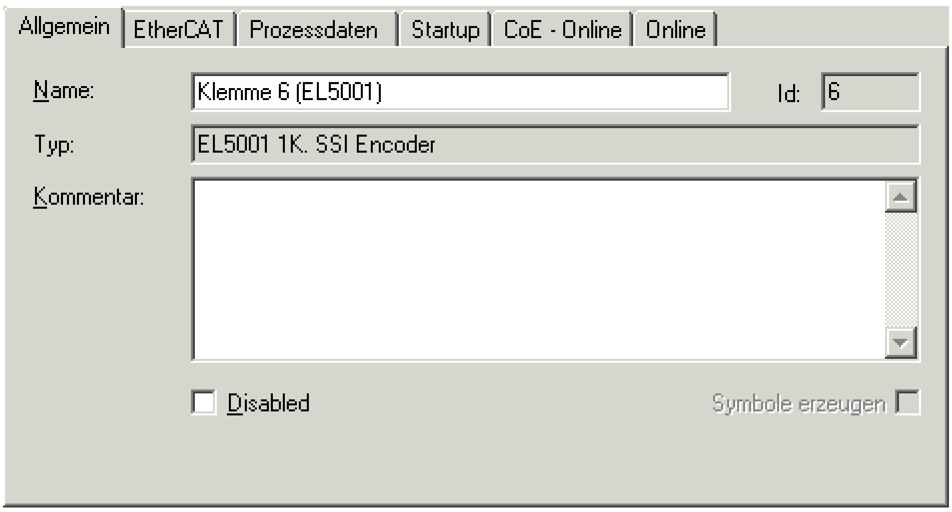


Fig. 109: “General” tab

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

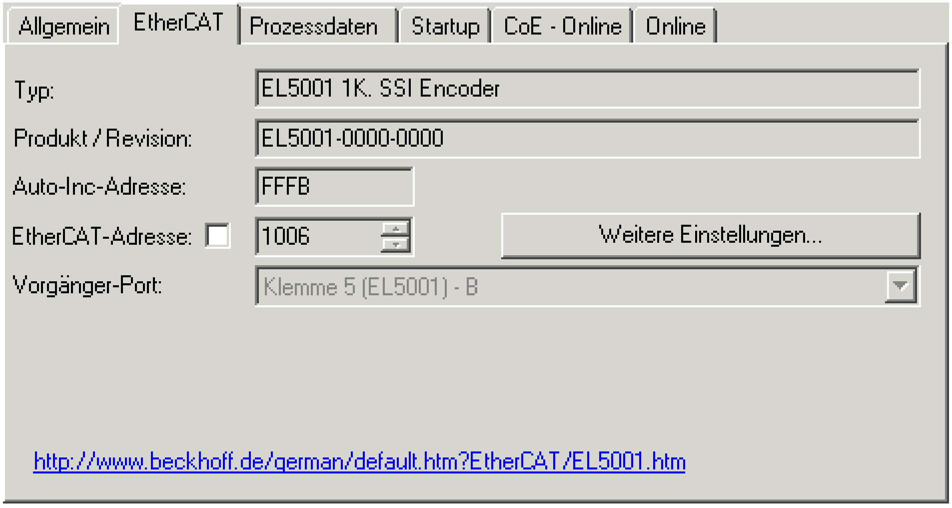


Fig. 110: „EtherCAT“ tab

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.

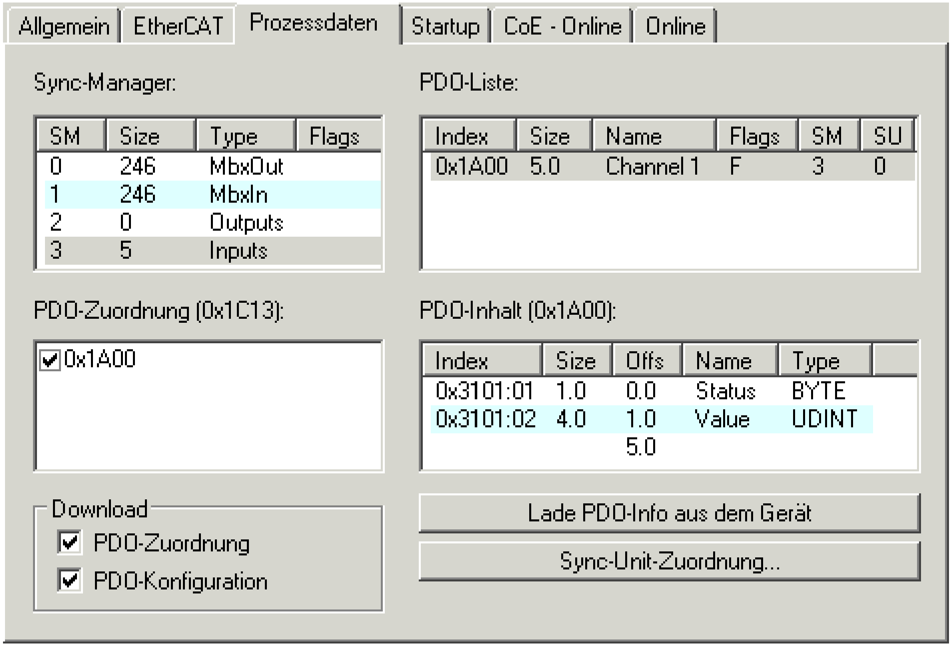


Fig. 111: “Process Data” tab

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

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

- The input/output process data supported by the device are defined by the manufacturer in the ESI/XML description. The TwinCAT EtherCAT Master uses the ESI description to configure the slave correctly.
- The process data can be modified in the system manager. See the device documentation.
  Examples of modifications include: mask out a channel, displaying additional cyclic information, 16-bit display instead of 8-bit data size, etc.
- In so-called “intelligent” EtherCAT devices the process data information is also stored in the CoE directory. Any changes in the CoE directory that lead to different PDO settings prevent successful startup of the slave. It is not advisable to deviate from the designated process data, because the device firmware (if available) is adapted to these PDO combinations.

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

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

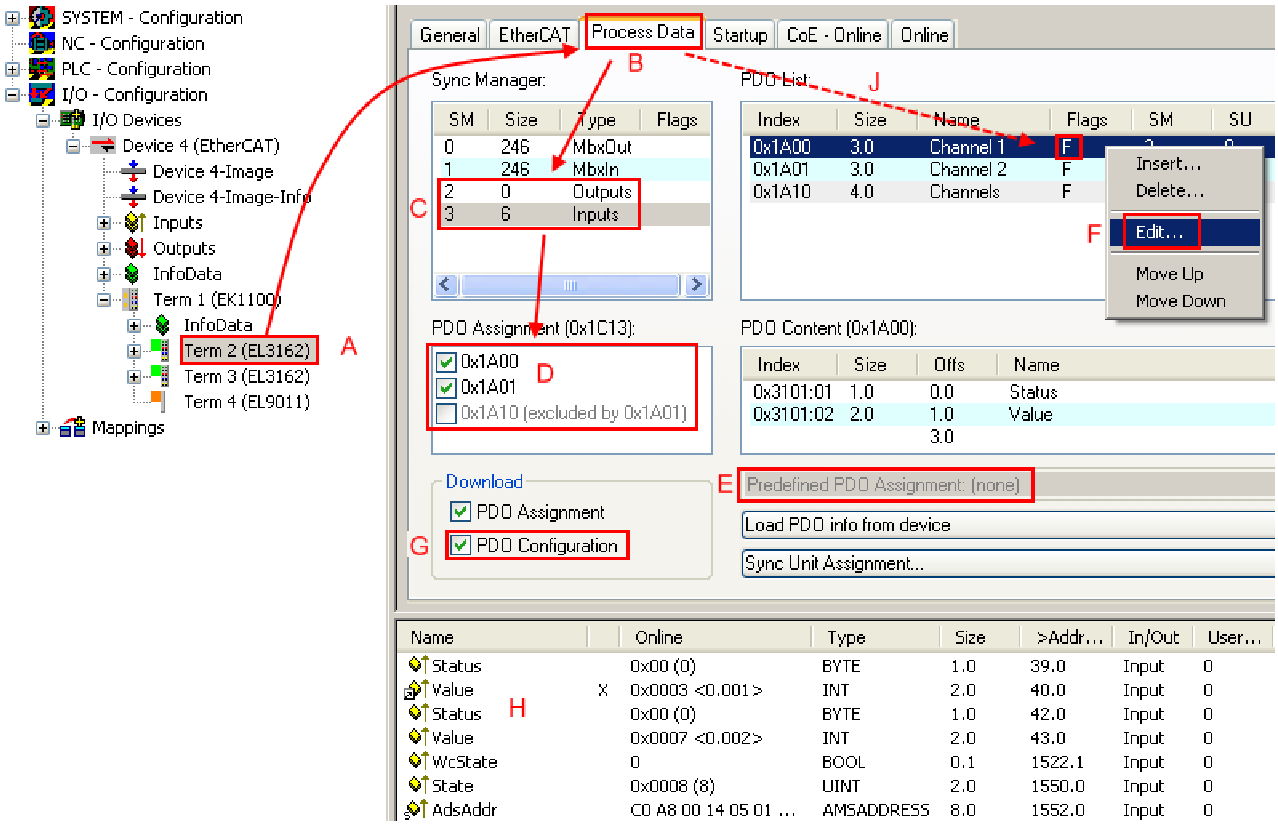


Fig. 112: Configuring the process data

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

A detailed description [▶ 86] can be found at the end of this section.

„Startup“ tab

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

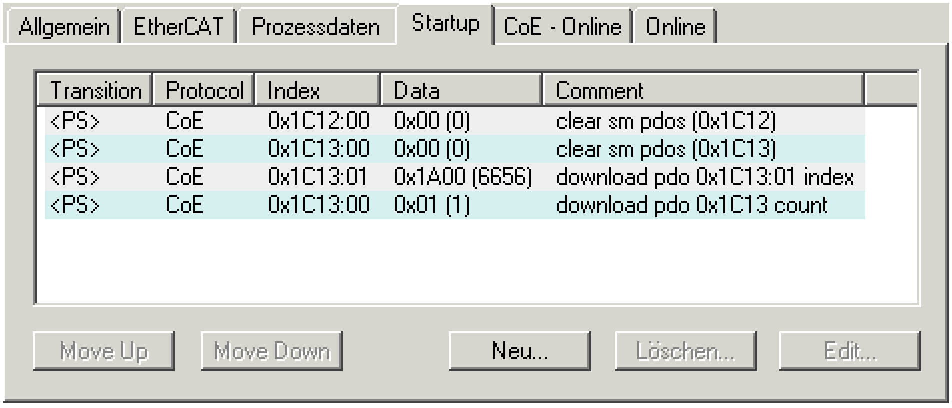


Fig. 113: „Startup“ tab

Column	Description
Transition	Transition to which the request is sent. This can either be • 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.

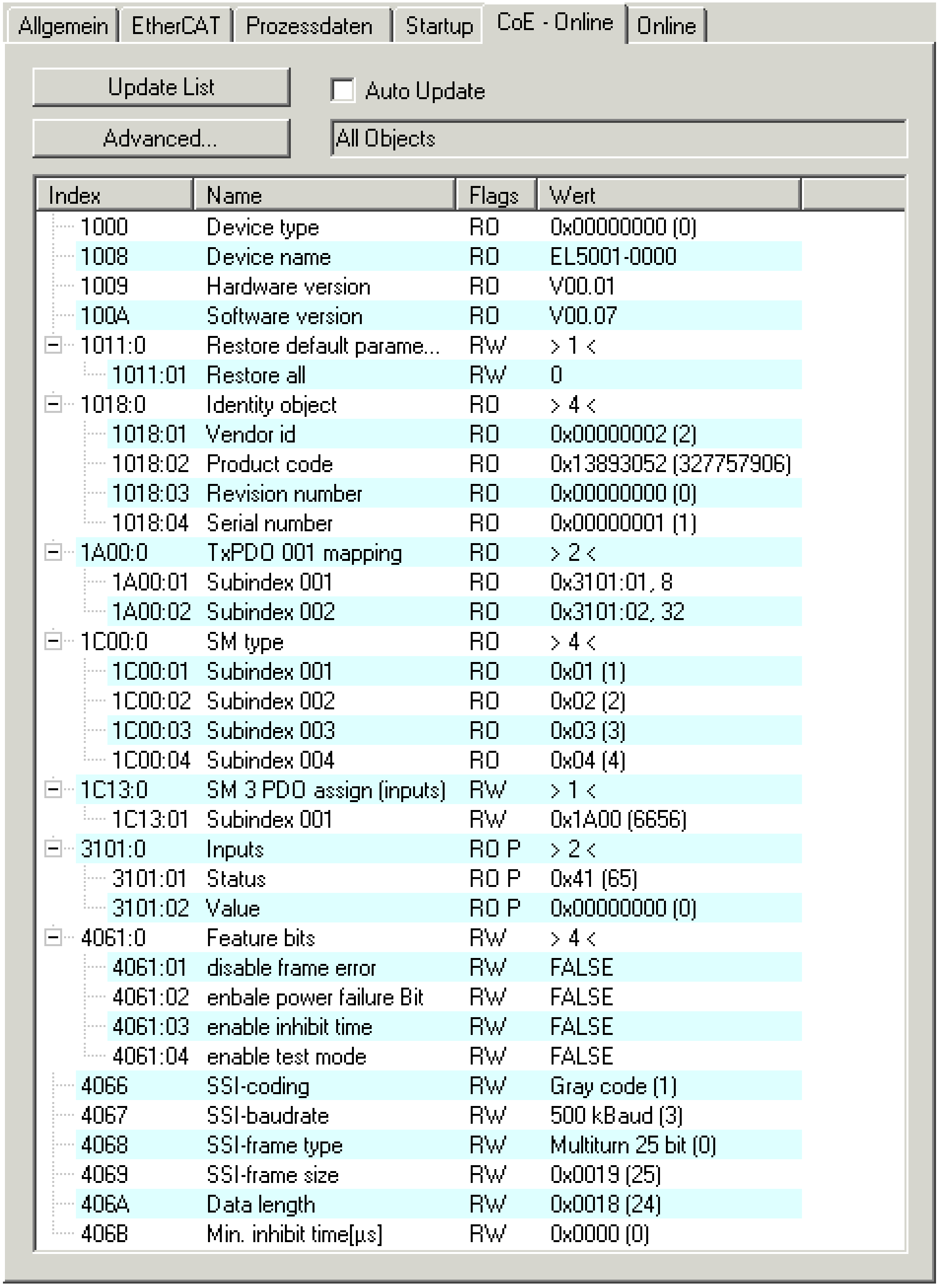


Fig. 114: “CoE – Online” tab

Object list display

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

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

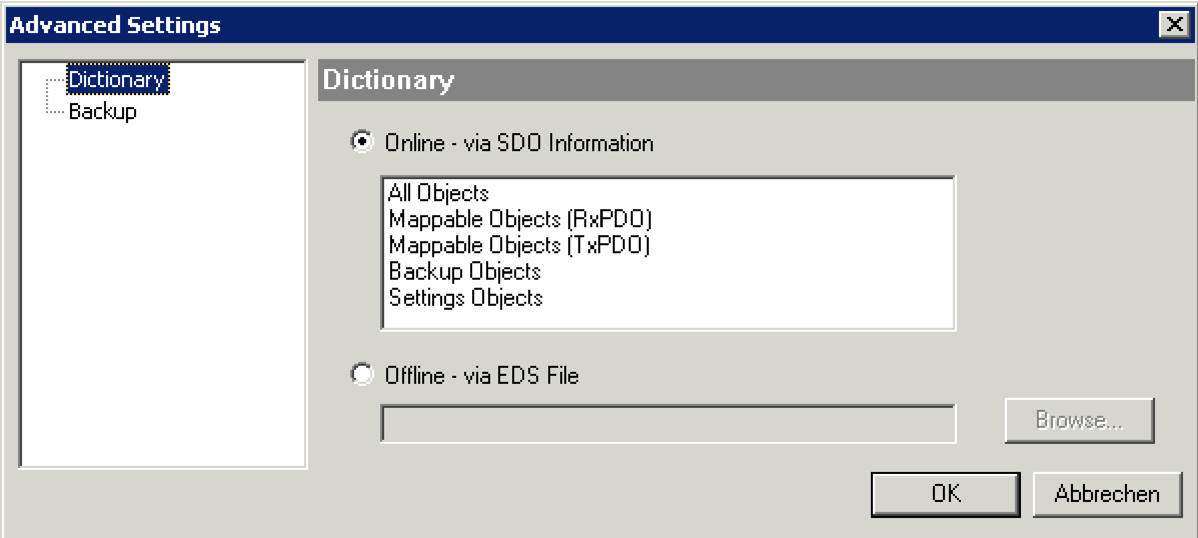


Fig. 115: Dialog “Advanced settings”

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

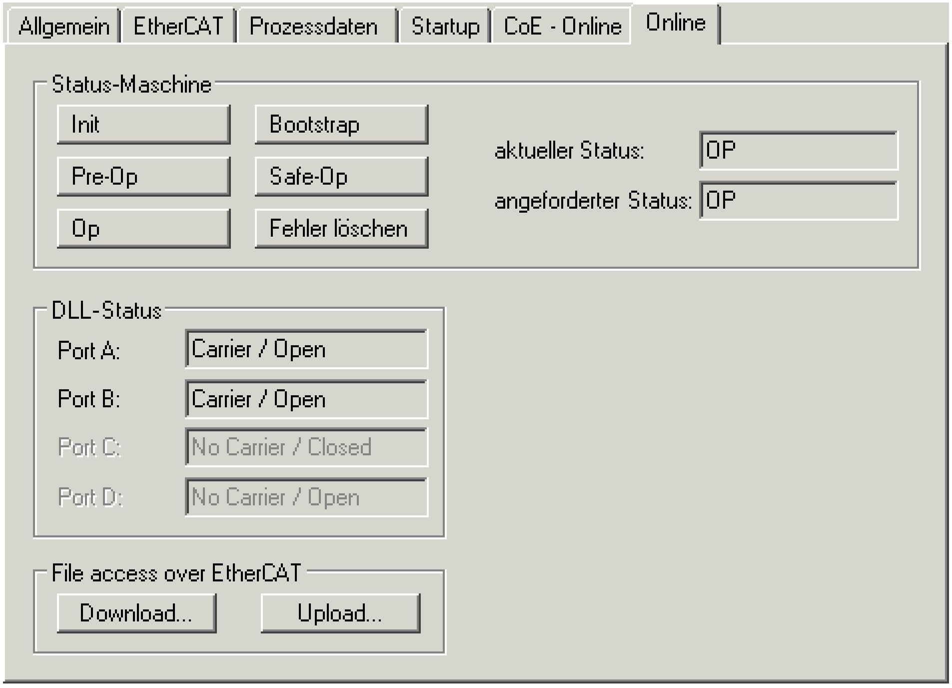


Fig. 116: „Online“ tab

State Machine

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

DLL Status

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

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

File Access over EtherCAT

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

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

"DC" tab (Distributed Clocks)

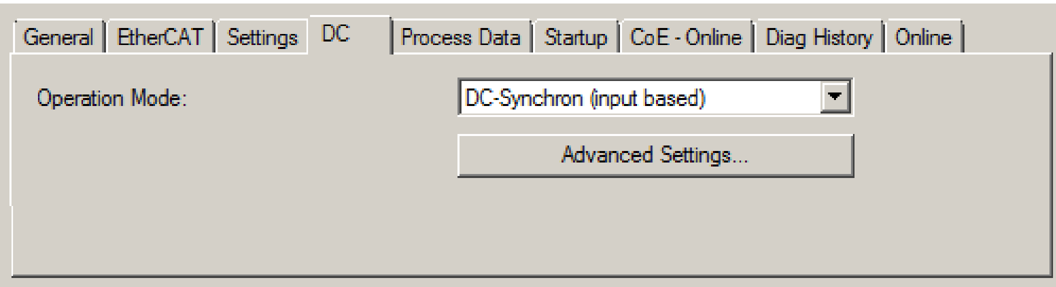


Fig. 117: "DC" tab (Distributed Clocks)

Operation Mode Options (optional):

- FreeRun
- SM-Synchron
- DC-Synchron (Input based)
- DC-Synchron

Advanced Settings… Advanced settings for readjustment of the real time determinant TwinCAT-clock

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

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

4.2.7.1 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.

Note

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 [▶ 85]),

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 Startup [▶ 81] 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.

4.3 General Notes - EtherCAT Slave Application

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

Diagnosis in real time: WorkingCounter, EtherCAT State and Status

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

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

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

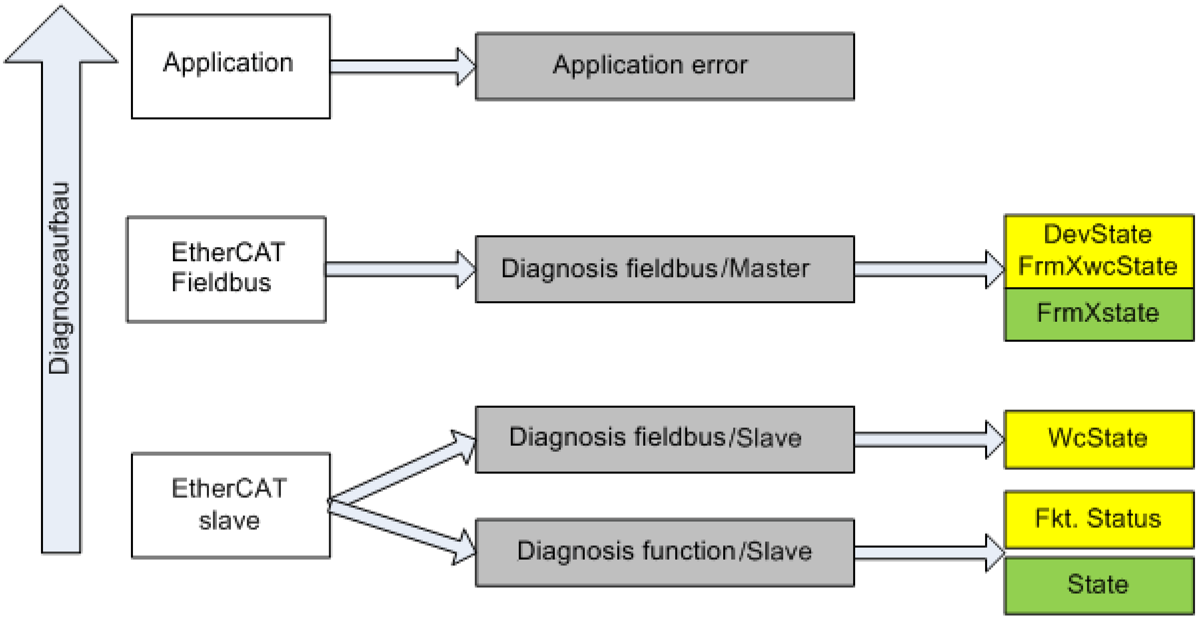


Fig. 118: Selection of the diagnostic information of an EtherCAT Slave

In general, an EtherCAT Slave offers

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

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

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

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

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

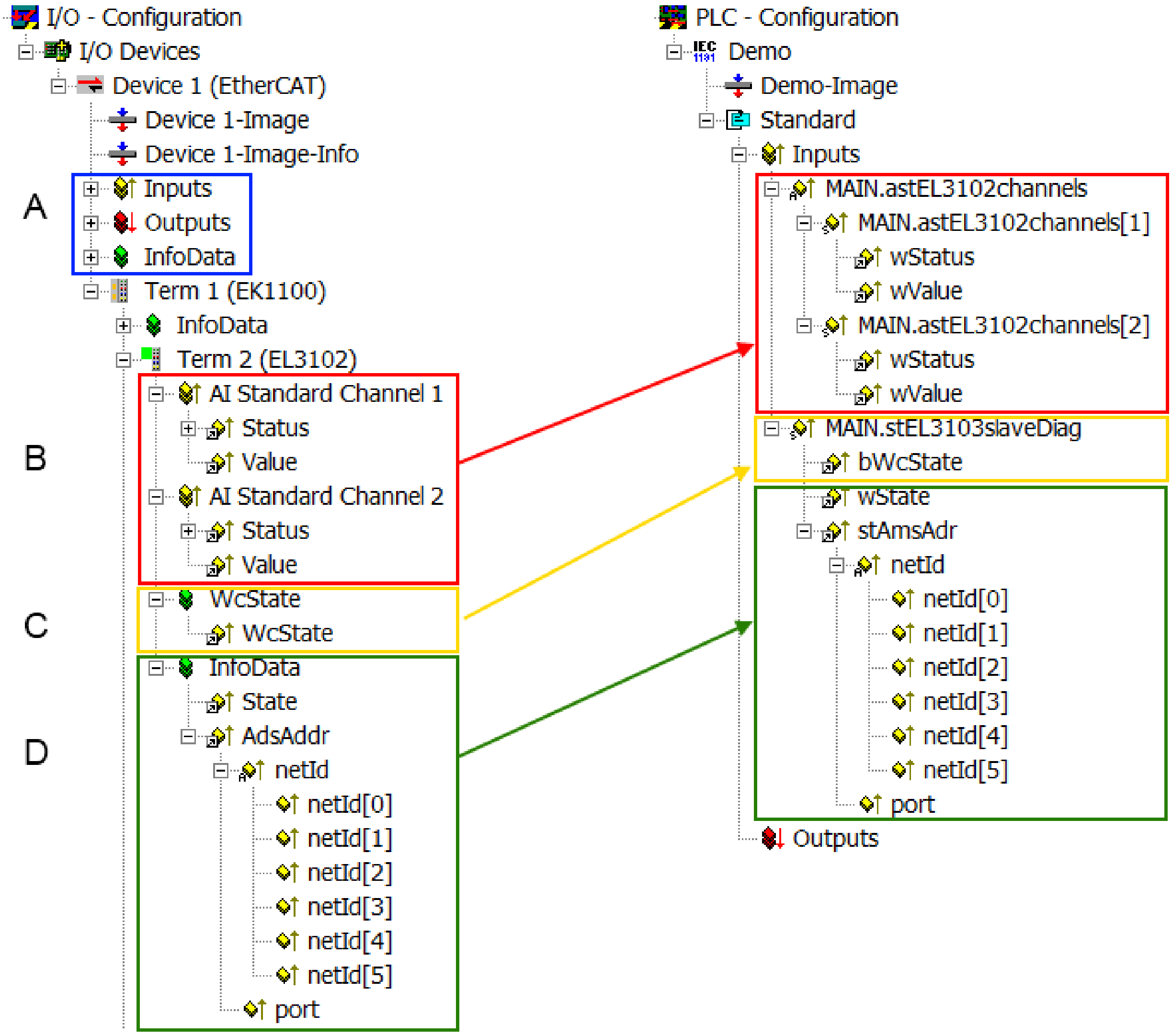


Fig. 119: Basic EtherCAT Slave Diagnosis in the PLC

The following aspects are covered here:

Code	Function	Implementation	Application/evaluation
A	The EtherCAT Master's diagnostic information updated acyclically (yellow) or provided acyclically (green).		At least the DevState is to be evaluated for the most recent cycle in the PLC. The EtherCAT Master's diagnostic information offers many more possibilities than are treated in the EtherCAT System Documentation. A few keywords: • 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.

Attention

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"*:

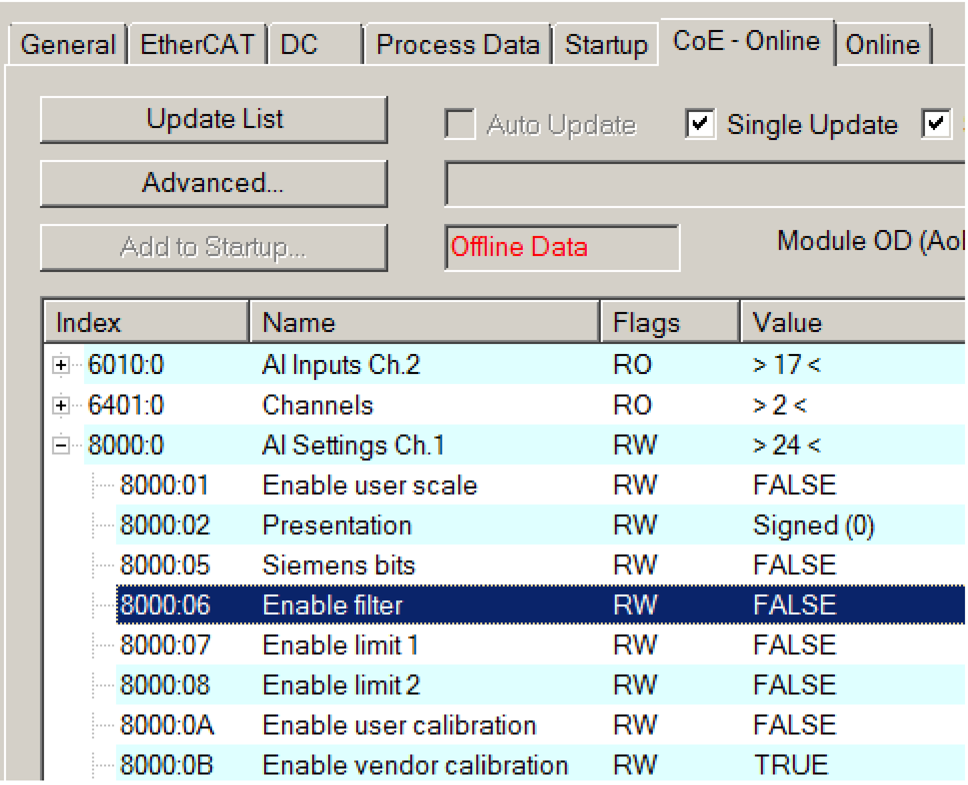


Fig. 120: EL3102, CoE directory

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

A few brief extracts:

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

Commissioning aid in the TwinCAT System Manager

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

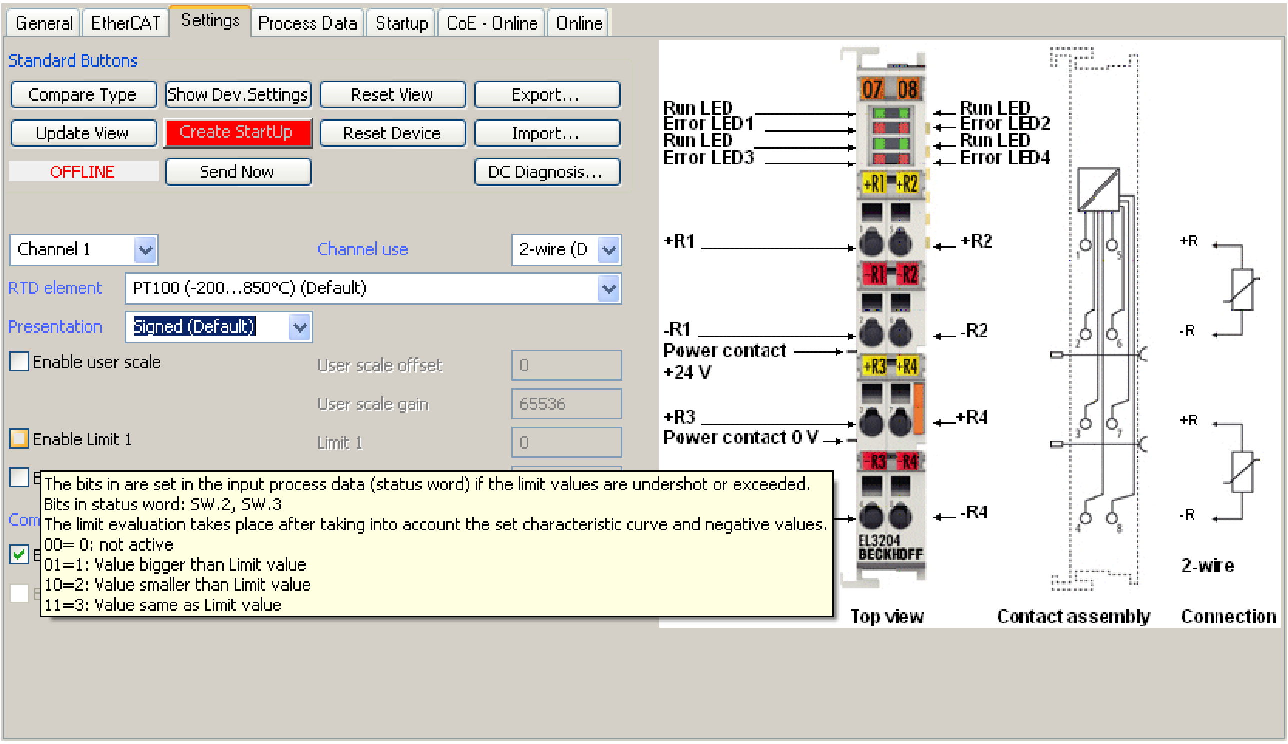


Fig. 121: Example of commissioning aid for a EL3204

This commissioning process simultaneously manages

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

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

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

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

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

- INIT
- PREOP
- SAFEOP
- OP

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

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

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

Standard setting

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

- EtherCAT Master: OP
- Slaves: OP
  This setting applies equally to all Slaves.

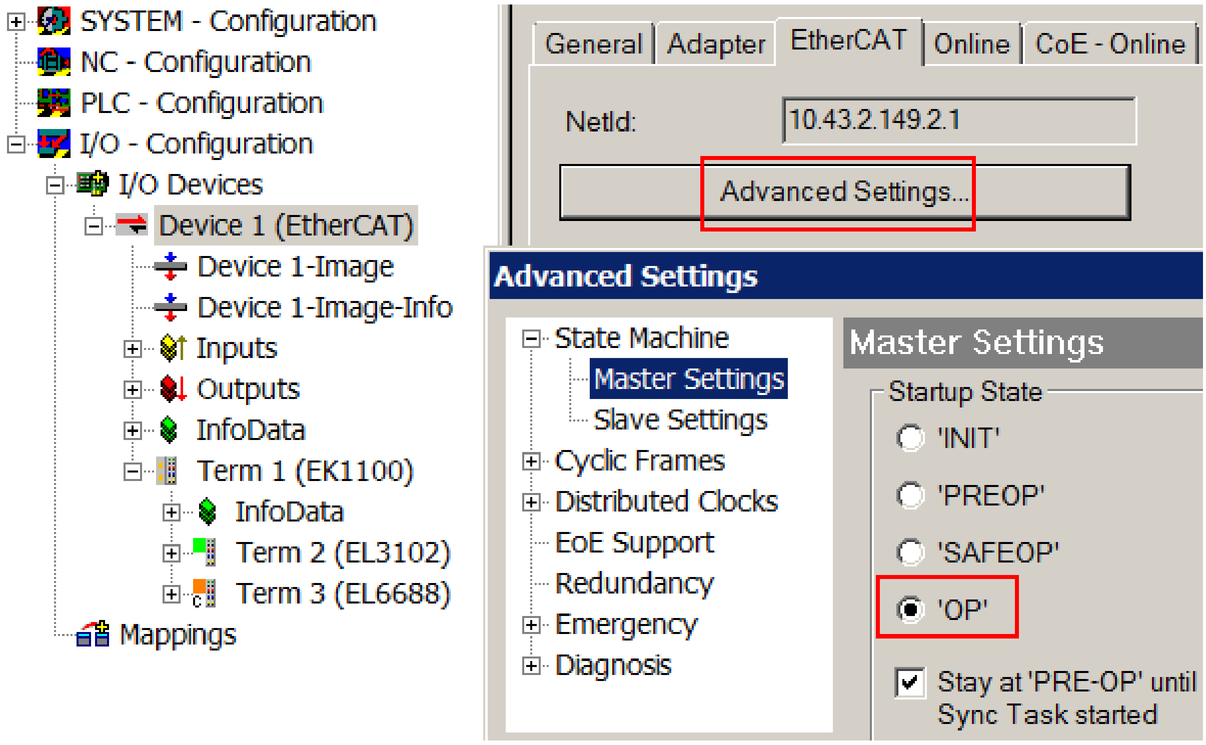


Fig. 122: Default behaviour of the System Manager

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

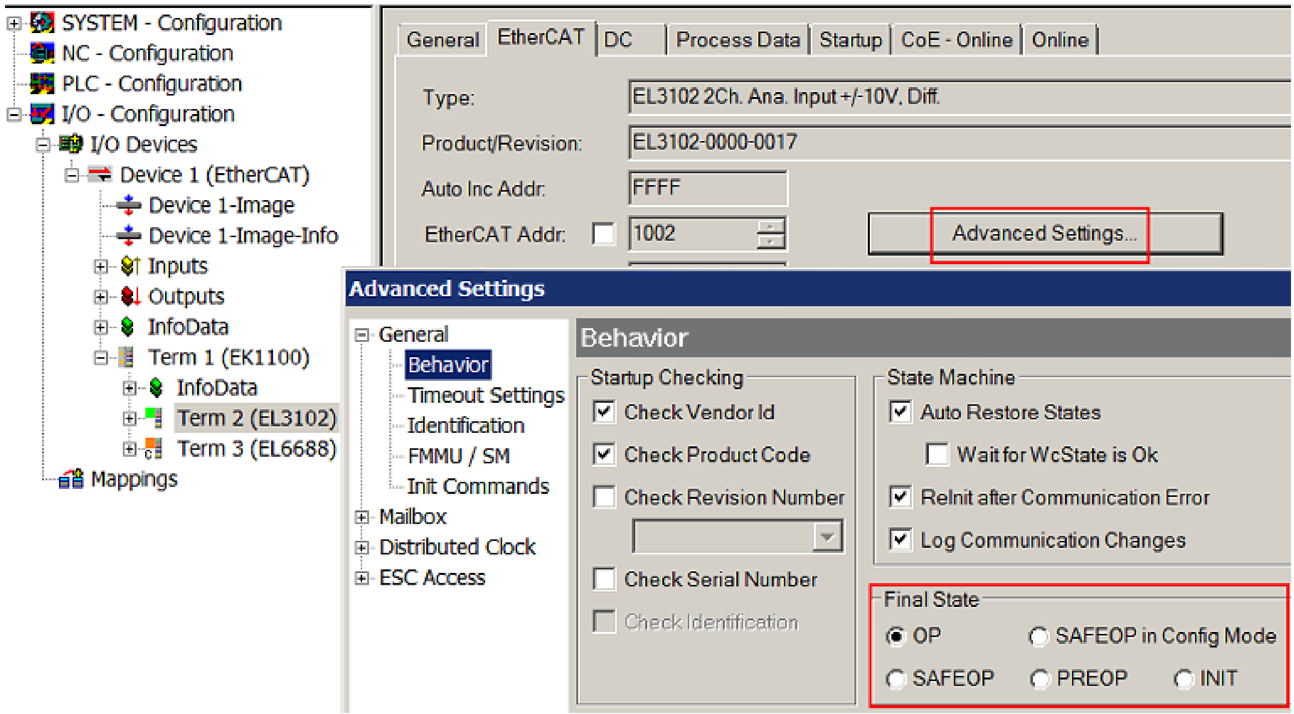


Fig. 123: Default target state in the Slave

Manual Control

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

- for diagnostic reasons
- to induce a controlled restart of axes
- because a change in the times involved in starting is desirable

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

It is then useful to put the settings in the EtherCAT Master to INIT for master and slave.

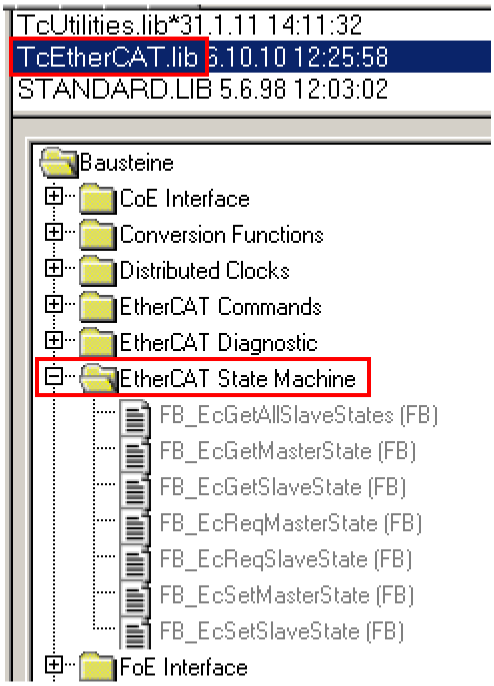
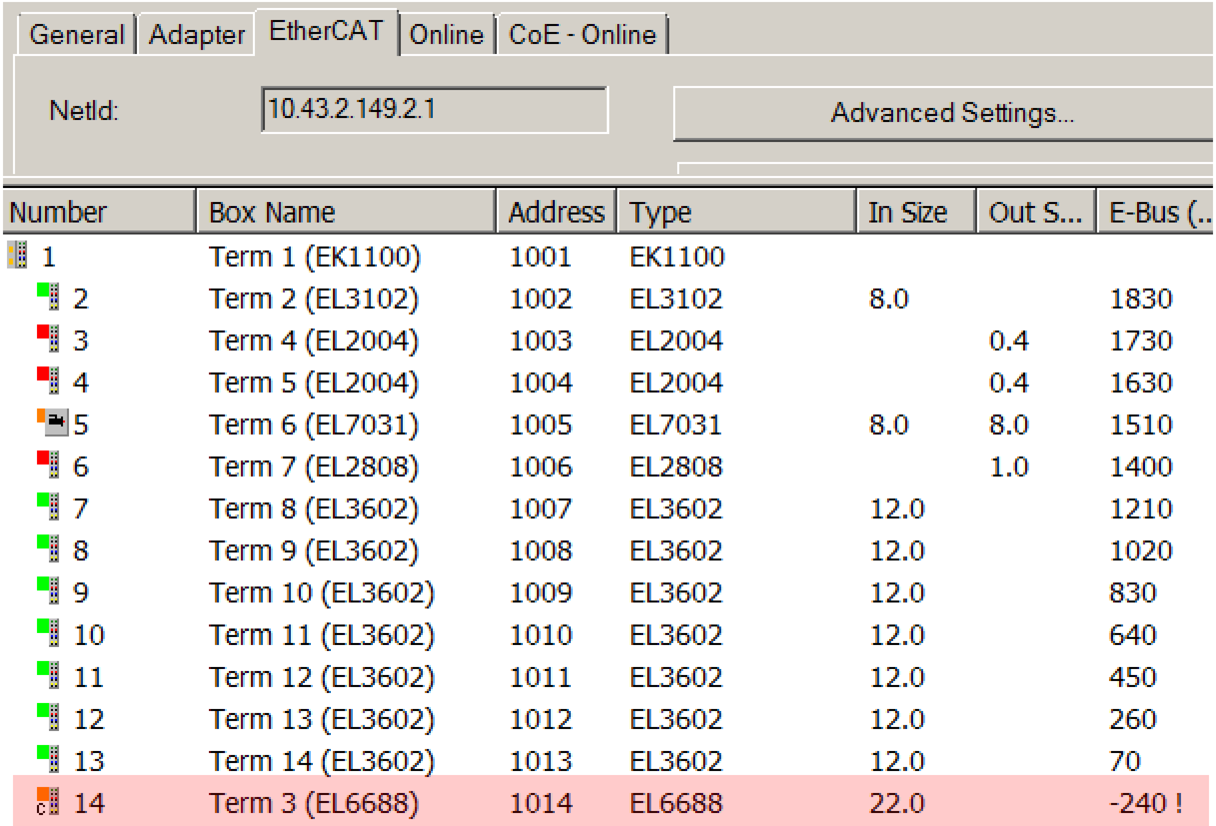


Fig. 124: PLC function blocks

Note regarding E-Bus current

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

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



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

Fig. 125: Illegally exceeding the E-Bus current

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

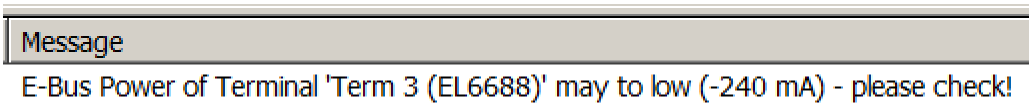


Fig. 126: Warning message for exceeding E-Bus current

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

5 Object description and parameterization

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

Note	**Parameterization via the CoE list (CAN over EtherCAT)** The EtherCAT device is parameterized via the CoE - Online tab [▸ 82] (double-click on the respective object) or via the Process Data tab [▸ 79](allocation of PDOs). Please note the following general CoE notes when using/manipulating the CoE parameters: - Keep a startup list if components have to be replaced - Differentiation between online/offline dictionary, existence of current XML description - use “CoE reload” for resetting changes

Introduction

The CoE overview contains objects for different intended applications:

- Objects required for parameterization [▸ 97] during commissioning
- Objects intended for regular operation [▸ 98], e.g. through ADS access.
- Objects for indicating internal settings [▸ 98] (may be fixed)

The following section first describes the objects required for normal operation, followed by a complete overview of missing objects.

5.1 Objects for commissioning

Index 80n0 AO Settings Ch.1 - 4 (for 2 ≤ n ≤ 5)

Index (hex)	Name	Meaning	Data type	Flags	Default
80n0:0	AO Settings Ch.1 - 4	Maximum subindex	UINT8	RO	0x13 (19_{dec})
80n0:01	Enable user scale	FALSE The user scaling is deactivated TRUE The user scaling (object 0x80n0:11, 0x80n0:12) is active; the output value is then calculated as described in the Calculation example [▶ 14].	BOOLEAN	RW	0x00 (0_{dec})
80n0:02	Presentation	0: Signed Presentation: The output value is output as signed integer in the two's complement 0x7FFF = +10 Volt 0x8001 = -10 Volt 1: Unsigned Presentation: In the negative range, the value is output as an absolute value 0x7FFF = 10 Volt 0x8001 = 10 Volt 2: Enable absolute value with MSB as sign: The output value is displayed in magnitude-sign format 0x7FFF = +10 Volt 0xFFFF = -10 Volt	BIT3	RW	0x00 (0_{dec})
80n0:05	Disable watchdog	FALSE The watchdog is active. In the event of a failure of communication, the analog output value will be set to either the manufacturer's or the user's default value. TRUE The watchdog timer is deactivated. In the event of a failure of communication, the analog output value will not be set to either the manufacturer's or the user's default value.	BOOLEAN	RW	0x00 (0_{dec})
80n0:06	Enable user default output	FALSE The manufacturer's value (0 V) will be output if the watchdog timer triggers TRUE If the watchdog timer triggers, the user-specific output value [▶ 14] (object 0x80n0:13) will be output.	BOOLEAN	RW	0x00 (0_{dec})
80n0:11	Offset	User scaling offset compensation	INT16	RW	0x0000 (0_{dec})
80n0:12	Gain	User scaling gain compensation: The gain is represented in fixed-point format, with the factor 2^{-16}. A value of 1 for the gain factor therefore corresponds to 65536_{dec} (0x00010000).	INT32	RW	0x00010000 (65536_{dec})
80n0:13	Default output	Definition of the user-specific output value [▶ 14] that is presented to the output if the watchdog timer for the cyclic communication triggers.	INT16	RW	0x0000 (0_{dec})

Index 80nE AO Internal data Ch.1 - 4 (for 2 ≤ n ≤ 5)

Index (hex)	Name	Meaning	Data type	Flags	Default
80nE:0	AO Internal data Ch.1 - 4	Maximum subindex	UINT8	RO	0x01 (1_{dec})
80nE:01	DAC raw value	DAC raw value	INT16	RO	0x0000 (0_{dec})

Index 80nF AO Vendor data Ch.1 - 4 (for 2 ≤ n ≤ 5)

Index (hex)	Name	Meaning	Data type	Flags	Default
80nF:0	AO Vendor data Ch.1 - 4	Maximum subindex	UINT8	RO	0x02 (2_{dec})
80nF:01	Calibration offset	Calibration offset	INT16	RW	0x0800 (2048_{dec})
80nF:02	Calibration gain	Calibration gain	INT16	RW	0x7D00 (32000_{dec})

Index 8060 ENC Settings Ch.1 - 4 (for 6 ≤ n ≤ 9)

Index (hex)	Name	Meaning	Data type	Flags	Default
80n0:0	ENC Settings Ch.1 - 4	Maximum subindex	UINT8	RO	0x06 (6_{dec})
80n0:02	Enable reset	The counter is set to zero whenever a latch event occurs	BOOLEAN	RW	0x00 (0_{dec})
80n0:03	Enable up/down counter	Enable up/down counter instead of the encoder	BOOLEAN	RW	0x00 (0_{dec})
80n0:04	Gate polarity	0: Gate disabled 1: Gate input responds to positive edge and locks the counter 2: Gate input responds to negative edge and locks the counter	BIT2	RW	0x01 (1_{dec})
80n0:06	Evaluation mode	0: 1-fold (single evaluation) 1: 2-fold (two-fold evaluation) 3: 4-fold (four-fold evaluation)	BIT2	RW	0x03 (3_{dec})

Index 80A0 PLS Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
80A0:0	PLS Settings	Maximum subindex	UINT8	RO	0x12 (18_{dec})
80A0:01	Encoder as PLS source	0: Channel 1 is the source for PLS functions 1: Channel 2 is the source for PLS functions 2: Channel 3 is the source for PLS functions 3: Channel 4 is the source for PLS functions	BIT2	RW	0x00 (0_{dec})
80A0:11	Output mask	Determines which of the digital outputs are assigned to the PLS. Sample: 0x00FF, outputs 0-7 are assigned to the PLS, outputs 8-15 can be controlled freely by the PLC.	UINT16	RW	0x0000 (0_{dec})
80A0:12	Default output	If no valid table entry is available (counter value is >=0 but less than the first entry in the table), this value is output	UINT16	RW	0x0000 (0_{dec})

5.2 Objects for regular operation

The EM7004 has no such objects.

5.3 Standard objects (0x1000-0x1FFF)

The standard objects have the same meaning for all EtherCAT slaves.

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default
1000:0	Device type	Device type of the EtherCAT slave: The Lo-Word contains the CoE profile used (5001). The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	0x00001389 (5001_{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	EM7004

Index 1009 Hardware version

Index (hex)	Name	Meaning	Data type	Flags	Default
1009:0	Hardware version	Hardware version of the EtherCAT slave	STRING	RO	02

Index 100A Software version

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

Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
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})

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	0x1B5C3452 (459027538_{dec})
1018:03	Revision	Revision number of the EtherCAT slave; the low word (bit 0-15) indicates the special terminal number, the high word (bit 16-31) refers to the device description	UINT32	RO	0x00100000 (1048576_{dec})
1018:04	Serial number	Serial number of the EtherCAT slave; the low byte (bit 0-7) of the low word contains the year of production, the high byte (bit 8-15) of the low word contains the week of production, the high word (bit 16-31) is 0	UINT32	RO	0x00000000 (0_{dec})

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 1601 DO RxPDO-Map

Index (hex)	Name	Meaning	Data type	Flags	Default
1601:0	DO RxPDO-Map	PDO Mapping RxPDO 2	UINT8	RO	0x10 (16_{dec})
1601:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (DO Outputs), entry 0x01 (Output 0))	UINT32	RO	0x7010:01, 1
1601:02	SubIndex 002	2. PDO Mapping entry (object 0x7010 (DO Outputs), entry 0x02 (Output 1))	UINT32	RO	0x7010:02, 1
1601:03	SubIndex 003	3. PDO Mapping entry (object 0x7010 (DO Outputs), entry 0x03 (Output 2))	UINT32	RO	0x7010:03, 1
1601:04	SubIndex 004	4. PDO Mapping entry (object 0x7010 (DO Outputs), entry 0x04 (Output 3))	UINT32	RO	0x7010:04, 1
1601:05	SubIndex 005	5. PDO Mapping entry (object 0x7010 (DO Outputs), entry 0x05 (Output 4))	UINT32	RO	0x7010:05, 1
1601:06	SubIndex 006	6. PDO Mapping entry (object 0x7010 (DO Outputs), entry 0x06 (Output 5))	UINT32	RO	0x7010:06, 1
1601:07	SubIndex 007	7. PDO Mapping entry (object 0x7010 (DO Outputs), entry 0x07 (Output 6))	UINT32	RO	0x7010:07, 1
1601:08	SubIndex 008	8. PDO Mapping entry (object 0x7010 (DO Outputs), entry 0x08 (Output 7))	UINT32	RO	0x7010:08, 1
1601:09	SubIndex 009	9. PDO Mapping entry (object 0x7010 (DO Outputs), entry 0x09 (Output 8))	UINT32	RO	0x7010:09, 1
1601:0A	SubIndex 010	10. PDO Mapping entry (object 0x7010 (DO Outputs), entry 0x0A (Output 9))	UINT32	RO	0x7010:0A, 1
1601:0B	SubIndex 011	11. PDO Mapping entry (object 0x7010 (DO Outputs), entry 0x0B (Output 10))	UINT32	RO	0x7010:0B, 1
1601:0C	SubIndex 012	12. PDO Mapping entry (object 0x7010 (DO Outputs), entry 0x0C (Output 11))	UINT32	RO	0x7010:0C, 1
1601:0D	SubIndex 013	13. PDO Mapping entry (object 0x7010 (DO Outputs), entry 0x0D (Output 12))	UINT32	RO	0x7010:0D, 1
1601:0E	SubIndex 014	14. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1601:0F	SubIndex 015	15. PDO Mapping entry (object 0x7010 (DO Outputs), entry 0x0F (Output 14))	UINT32	RO	0x7010:0F, 1
1601:10	SubIndex 016	16. PDO Mapping entry (object 0x7010 (DO Outputs), entry 0x10 (Output 15))	UINT32	RO	0x7010:10, 1

Index 1602 AO RxPDO-Map Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1602:0	AO RxPDO-Map Ch.1	PDO Mapping RxPDO 3	UINT8	RO	0x01 (1_{dec})
1602:01	SubIndex 001	1. PDO Mapping entry (object 0x7020 (AO Outputs Ch.1), entry 0x11 (Analog output))	UINT32	RO	0x7020:11, 16

Index 1603 AO RxPDO-Map Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1603:0	AO RxPDO-Map Ch.2	PDO Mapping RxPDO 4	UINT8	RO	0x01 (1_{dec})
1603:01	SubIndex 001	1. PDO Mapping entry (object 0x7030 (AO Outputs Ch.2), entry 0x11 (Analog output))	UINT32	RO	0x7030:11, 16

Index 1604 AO RxPDO-Map Ch.3

Index (hex)	Name	Meaning	Data type	Flags	Default
1604:0	AO RxPDO-Map Ch.3	PDO Mapping RxPDO 5	UINT8	RO	0x01 (1_{dec})
1604:01	SubIndex 001	1. PDO Mapping entry (object 0x7040 (AO Outputs Ch.3), entry 0x11 (Analog output))	UINT32	RO	0x7040:11, 16

Index 1605 AO RxPDO-Map Ch.4

Index (hex)	Name	Meaning	Data type	Flags	Default
1605:0	AO RxPDO-Map Ch.4	PDO Mapping RxPDO 6	UINT8	RO	0x01 (1_{dec})
1605:01	SubIndex 001	1. PDO Mapping entry (object 0x7050 (AO Outputs Ch.4), entry 0x11 (Analog output))	UINT32	RO	0x7050:11, 16

Index 1606 ENC RxPDO-Map Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1606:0	ENC RxPDO-Map Ch.1	PDO Mapping RxPDO 7	UINT8	RO	0x06 (6_{dec})
1606:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1606:02	SubIndex 002	2. PDO Mapping entry (object 0x7060 (ENC Outputs Ch.1), entry 0x02 (Enable latch extern on positive edge))	UINT32	RO	0x7060:02, 1
1606:03	SubIndex 003	3. PDO Mapping entry (object 0x7060 (ENC Outputs Ch.1), entry 0x03 (Set counter))	UINT32	RO	0x7060:03, 1
1606:04	SubIndex 004	4. PDO Mapping entry (object 0x7060 (ENC Outputs Ch.1), entry 0x04 (Enable latch extern on negative edge))	UINT32	RO	0x7060:04, 1
1606:05	SubIndex 005	5. PDO Mapping entry (12 bits align)	UINT32	RO	0x0000:00, 12
1606:06	SubIndex 006	6. PDO Mapping entry (object 0x7060 (ENC Outputs Ch.1), entry 0x11 (Set counter value))	UINT32	RO	0x7060:11, 16

Index 1607 ENC RxPDO-Map Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1607:0	ENC RxPDO-Map Ch.2	PDO Mapping RxPDO 8	UINT8	RO	0x06 (6_{dec})
1607:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1607:02	SubIndex 002	2. PDO Mapping entry (object 0x7070 (ENC Outputs Ch.2), entry 0x02 (Enable latch extern on positive edge))	UINT32	RO	0x7070:02, 1
1607:03	SubIndex 003	3. PDO Mapping entry (object 0x7070 (ENC Outputs Ch.2), entry 0x03 (Set counter))	UINT32	RO	0x7070:03, 1
1607:04	SubIndex 004	4. PDO Mapping entry (object 0x7070 (ENC Outputs Ch.2), entry 0x04 (Enable latch extern on negative edge))	UINT32	RO	0x7070:04, 1
1607:05	SubIndex 005	5. PDO Mapping entry (12 bits align)	UINT32	RO	0x0000:00, 12
1607:06	SubIndex 006	6. PDO Mapping entry (object 0x7070 (ENC Outputs Ch.2), entry 0x11 (Set counter value))	UINT32	RO	0x7070:11, 16

Index 1608 ENC RxPDO-Map Ch.3

Index (hex)	Name	Meaning	Data type	Flags	Default
1608:0	ENC RxPDO-Map Ch.3	PDO Mapping RxPDO 9	UINT8	RO	0x06 (6_{dec})
1608:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1608:02	SubIndex 002	2. PDO Mapping entry (object 0x7080 (ENC Outputs Ch.3), entry 0x02 (Enable latch extern on positive edge))	UINT32	RO	0x7080:02, 1
1608:03	SubIndex 003	3. PDO Mapping entry (object 0x7080 (ENC Outputs Ch.3), entry 0x03 (Set counter))	UINT32	RO	0x7080:03, 1
1608:04	SubIndex 004	4. PDO Mapping entry (object 0x7080 (ENC Outputs Ch.3), entry 0x04 (Enable latch extern on negative edge))	UINT32	RO	0x7080:04, 1
1608:05	SubIndex 005	5. PDO Mapping entry (12 bits align)	UINT32	RO	0x0000:00, 12
1608:06	SubIndex 006	6. PDO Mapping entry (object 0x7080 (ENC Outputs Ch.3), entry 0x11 (Set counter value))	UINT32	RO	0x7080:11, 16

Index 1609 ENC RxPDO-Map Ch.4

Index (hex)	Name	Meaning	Data type	Flags	Default
1609:0	ENC RxPDO-Map Ch.4	PDO Mapping RxPDO 10	UINT8	RO	0x06 (6_{dec})
1609:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1609:02	SubIndex 002	2. PDO Mapping entry (object 0x7090 (ENC Outputs Ch.4), entry 0x02 (Enable latch extern on positive edge))	UINT32	RO	0x7090:02, 1
1609:03	SubIndex 003	3. PDO Mapping entry (object 0x7090 (ENC Outputs Ch.4), entry 0x03 (Set counter))	UINT32	RO	0x7090:03, 1
1609:04	SubIndex 004	4. PDO Mapping entry (object 0x7090 (ENC Outputs Ch.4), entry 0x04 (Enable latch extern on negative edge))	UINT32	RO	0x7090:04, 1
1609:05	SubIndex 005	5. PDO Mapping entry (12 bits align)	UINT32	RO	0x0000:00, 12
1609:06	SubIndex 006	6. PDO Mapping entry (object 0x7090 (ENC Outputs Ch.4), entry 0x11 (Set counter value))	UINT32	RO	0x7090:11, 16

Index 160A PLS RxPDO-Map

Index (hex)	Name	Meaning	Data type	Flags	Default
160A:0	PLS RxPDO-Map	PDO Mapping RxPDO 11	UINT8	RO	0x02 (2_{dec})
160A:01	SubIndex 001	1. PDO Mapping entry (object 0x70A0 (PLS Outputs), entry 0x01 (Enable PLS))	UINT32	RO	0x70A0:01, 1
160A:02	SubIndex 002	2. PDO Mapping entry (15 bits align)	UINT32	RO	0x0000:00, 15

Index 1A00 DI TxPDO-Map

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00:0	DI TxPDO-Map	PDO Mapping TxPDO 1	UINT8	RO	0x10 (16_{dec})
1A00:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (DI Inputs), entry 0x01 (Input 0))	UINT32	RO	0x6000:01, 1
1A00:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (DI Inputs), entry 0x02 (Input 1))	UINT32	RO	0x6000:02, 1
1A00:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (DI Inputs), entry 0x03 (Input 2))	UINT32	RO	0x6000:03, 1
1A00:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (DI Inputs), entry 0x04 (Input 3))	UINT32	RO	0x6000:04, 1
1A00:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (DI Inputs), entry 0x05 (Input 4))	UINT32	RO	0x6000:05, 1
1A00:06	SubIndex 006	6. PDO Mapping entry (object 0x6000 (DI Inputs), entry 0x06 (Input 5))	UINT32	RO	0x6000:06, 1
1A00:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (DI Inputs), entry 0x07 (Input 6))	UINT32	RO	0x6000:07, 1
1A00:08	SubIndex 008	8. PDO Mapping entry (object 0x6000 (DI Inputs), entry 0x08 (Input 7))	UINT32	RO	0x6000:08, 1
1A00:09	SubIndex 009	9. PDO Mapping entry (object 0x6000 (DI Inputs), entry 0x09 (Input 8))	UINT32	RO	0x6000:09, 1
1A00:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (DI Inputs), entry 0x0A (Input 9))	UINT32	RO	0x6000:0A, 1
1A00:0B	SubIndex 011	11. PDO Mapping entry (object 0x6000 (DI Inputs), entry 0x0B (Input 10))	UINT32	RO	0x6000:0B, 1
1A00:0C	SubIndex 012	12. PDO Mapping entry (object 0x6000 (DI Inputs), entry 0x0C (Input 11))	UINT32	RO	0x6000:0C, 1
1A00:0D	SubIndex 013	13. PDO Mapping entry (object 0x6000 (DI Inputs), entry 0x0D (Input 12))	UINT32	RO	0x6000:0D, 1
1A00:0E	SubIndex 014	14. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A00:0F	SubIndex 015	15. PDO Mapping entry (object 0x6000 (DI Inputs), entry 0x0F (Input 14))	UINT32	RO	0x6000:0F, 1
1A00:10	SubIndex 016	16. PDO Mapping entry (object 0x6000 (DI Inputs), entry 0x10 (Input 15))	UINT32	RO	0x6000:10, 1

Index 1A06 ENC TxPDO-Map Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A06:0	ENC TxPDO-Map Ch.1	PDO Mapping TxPDO 7	UINT8	RO	0x0D (13$_{dec}$)
1A06:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A06:02	SubIndex 002	2. PDO Mapping entry (object 0x6060 (ENC Inputs Ch.1), entry 0x02 (Latch extern valid))	UINT32	RO	0x6060:02, 1
1A06:03	SubIndex 003	3. PDO Mapping entry (object 0x6060 (ENC Inputs Ch.1), entry 0x03 (Set counter done))	UINT32	RO	0x6060:03, 1
1A06:04	SubIndex 004	4. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1A06:05	SubIndex 005	5. PDO Mapping entry (object 0x6060 (ENC Inputs Ch.1), entry 0x09 (Status of input A))	UINT32	RO	0x6060:09, 1
1A06:06	SubIndex 006	6. PDO Mapping entry (object 0x6060 (ENC Inputs Ch.1), entry 0x0A (Status of input B))	UINT32	RO	0x6060:0A, 1
1A06:07	SubIndex 007	7. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A06:08	SubIndex 008	8. PDO Mapping entry (object 0x6060 (ENC Inputs Ch.1), entry 0x0C (Status of input gate))	UINT32	RO	0x6060:0C, 1
1A06:09	SubIndex 009	9. PDO Mapping entry (object 0x6060 (ENC Inputs Ch.1), entry 0x0D (Status of extern latch))	UINT32	RO	0x6060:0D, 1
1A06:0A	SubIndex 010	10. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1A06:0B	SubIndex 011	11. PDO Mapping entry (object 0x1806, entry 0x09)	UINT32	RO	0x1806:09, 1
1A06:0C	SubIndex 012	12. PDO Mapping entry (object 0x6060 (ENC Inputs Ch.1), entry 0x11 (Counter value))	UINT32	RO	0x6060:11, 16
1A06:0D	SubIndex 013	13. PDO Mapping entry (object 0x6060 (ENC Inputs Ch.1), entry 0x12 (Latch value))	UINT32	RO	0x6060:12, 16

Index 1A07 ENC TxPDO-Map Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A07:0	ENC TxPDO-Map Ch.2	PDO Mapping TxPDO 8	UINT8	RO	0x0D (13$_{dec}$)
1A07:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A07:02	SubIndex 002	2. PDO Mapping entry (object 0x6070 (ENC Inputs Ch.2), entry 0x02 (Latch extern valid))	UINT32	RO	0x6070:02, 1
1A07:03	SubIndex 003	3. PDO Mapping entry (object 0x6070 (ENC Inputs Ch.2), entry 0x03 (Set counter done))	UINT32	RO	0x6070:03, 1
1A07:04	SubIndex 004	4. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1A07:05	SubIndex 005	5. PDO Mapping entry (object 0x6070 (ENC Inputs Ch.2), entry 0x09 (Status of input A))	UINT32	RO	0x6070:09, 1
1A07:06	SubIndex 006	6. PDO Mapping entry (object 0x6070 (ENC Inputs Ch.2), entry 0x0A (Status of input B))	UINT32	RO	0x6070:0A, 1
1A07:07	SubIndex 007	7. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A07:08	SubIndex 008	8. PDO Mapping entry (object 0x6070 (ENC Inputs Ch.2), entry 0x0C (Status of input gate))	UINT32	RO	0x6070:0C, 1
1A07:09	SubIndex 009	9. PDO Mapping entry (object 0x6070 (ENC Inputs Ch.2), entry 0x0D (Status of extern latch))	UINT32	RO	0x6070:0D, 1
1A07:0A	SubIndex 010	10. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1A07:0B	SubIndex 011	11. PDO Mapping entry (object 0x1807, entry 0x09)	UINT32	RO	0x1807:09, 1
1A07:0C	SubIndex 012	12. PDO Mapping entry (object 0x6070 (ENC Inputs Ch.2), entry 0x11 (Counter value))	UINT32	RO	0x6070:11, 16
1A07:0D	SubIndex 013	13. PDO Mapping entry (object 0x6070 (ENC Inputs Ch.2), entry 0x12 (Latch value))	UINT32	RO	0x6070:12, 16

Index 1A08 ENC TxPDO-Map Ch.3

Index (hex)	Name	Meaning	Data type	Flags	Default
1A08:0	ENC TxPDO-Map Ch.3	PDO Mapping TxPDO 9	UINT8	RO	0x0D (13$_{dec}$)
1A08:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A08:02	SubIndex 002	2. PDO Mapping entry (object 0x6080 (ENC Inputs Ch.3), entry 0x02 (Latch extern valid))	UINT32	RO	0x6080:02, 1
1A08:03	SubIndex 003	3. PDO Mapping entry (object 0x6080 (ENC Inputs Ch.3), entry 0x03 (Set counter done))	UINT32	RO	0x6080:03, 1
1A08:04	SubIndex 004	4. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1A08:05	SubIndex 005	5. PDO Mapping entry (object 0x6080 (ENC Inputs Ch.3), entry 0x09 (Status of input A))	UINT32	RO	0x6080:09, 1
1A08:06	SubIndex 006	6. PDO Mapping entry (object 0x6080 (ENC Inputs Ch.3), entry 0x0A (Status of input B))	UINT32	RO	0x6080:0A, 1
1A08:07	SubIndex 007	7. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A08:08	SubIndex 008	8. PDO Mapping entry (object 0x6080 (ENC Inputs Ch.3), entry 0x0C (Status of input gate))	UINT32	RO	0x6080:0C, 1
1A08:09	SubIndex 009	9. PDO Mapping entry (object 0x6080 (ENC Inputs Ch.3), entry 0x0D (Status of extern latch))	UINT32	RO	0x6080:0D, 1
1A08:0A	SubIndex 010	10. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1A08:0B	SubIndex 011	11. PDO Mapping entry (object 0x1808, entry 0x09)	UINT32	RO	0x1808:09, 1
1A08:0C	SubIndex 012	12. PDO Mapping entry (object 0x6080 (ENC Inputs Ch.3), entry 0x11 (Counter value))	UINT32	RO	0x6080:11, 16
1A08:0D	SubIndex 013	13. PDO Mapping entry (object 0x6080 (ENC Inputs Ch.3), entry 0x12 (Latch value))	UINT32	RO	0x6080:12, 16

Index 1A09 ENC TxPDO-Map Ch.4

Index (hex)	Name	Meaning	Data type	Flags	Default
1A09:0	ENC TxPDO-Map Ch.4	PDO Mapping TxPDO 10	UINT8	RO	0x0D (13$_{dec}$)
1A09:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A09:02	SubIndex 002	2. PDO Mapping entry (object 0x6090 (ENC Inputs Ch.4), entry 0x02 (Latch extern valid))	UINT32	RO	0x6090:02, 1
1A09:03	SubIndex 003	3. PDO Mapping entry (object 0x6090 (ENC Inputs Ch.4), entry 0x03 (Set counter done))	UINT32	RO	0x6090:03, 1
1A09:04	SubIndex 004	4. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1A09:05	SubIndex 005	5. PDO Mapping entry (object 0x6090 (ENC Inputs Ch.4), entry 0x09 (Status of input A))	UINT32	RO	0x6090:09, 1
1A09:06	SubIndex 006	6. PDO Mapping entry (object 0x6090 (ENC Inputs Ch.4), entry 0x0A (Status of input B))	UINT32	RO	0x6090:0A, 1
1A09:07	SubIndex 007	7. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A09:08	SubIndex 008	8. PDO Mapping entry (object 0x6090 (ENC Inputs Ch.4), entry 0x0C (Status of input gate))	UINT32	RO	0x6090:0C, 1
1A09:09	SubIndex 009	9. PDO Mapping entry (object 0x6090 (ENC Inputs Ch.4), entry 0x0D (Status of extern latch))	UINT32	RO	0x6090:0D, 1
1A09:0A	SubIndex 010	10. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1A09:0B	SubIndex 011	11. PDO Mapping entry (object 0x1809, entry 0x09)	UINT32	RO	0x1809:09, 1
1A09:0C	SubIndex 012	12. PDO Mapping entry (object 0x6090 (ENC Inputs Ch.4), entry 0x11 (Counter value))	UINT32	RO	0x6090:11, 16
1A09:0D	SubIndex 013	13. PDO Mapping entry (object 0x6090 (ENC Inputs Ch.4), entry 0x12 (Latch value))	UINT32	RO	0x6090:12, 16

Index 1A0A PLS TxPDO-Map

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0A:0	PLS TxPDO-Map	PDO Mapping TxPDO 11	UINT8	RO	0x03 (3$_{dec}$)
1A0A:01	SubIndex 001	1. PDO Mapping entry (object 0x60A0 (PLS Inputs), entry 0x01 (PLS Enabled))	UINT32	RO	0x60A0:01, 1
1A0A:02	SubIndex 002	2. PDO Mapping entry (object 0x60A0 (PLS Inputs), entry 0x02 (Unequal SI:0))	UINT32	RO	0x60A0:02, 1
1A0A:03	SubIndex 003	3. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14

Index 1C00 Sync manager type

Index (hex)	Name	Meaning	Data type	Flags	Default
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	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RO	0x0A (10_{dec})
1C12:01	SubIndex 001	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RO	0x1601 (5633_{dec})
1C12:02	SubIndex 002	2. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RO	0x1602 (5634_{dec})
1C12:03	SubIndex 003	3. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RO	0x1603 (5635_{dec})
1C12:04	SubIndex 004	4. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RO	0x1604 (5636_{dec})
1C12:05	SubIndex 005	5. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RO	0x1605 (5637_{dec})
1C12:06	SubIndex 006	6. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RO	0x1606 (5638_{dec})
1C12:07	SubIndex 007	7. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RO	0x1607 (5639_{dec})
1C12:08	SubIndex 008	8. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RO	0x1608 (5640_{dec})
1C12:09	SubIndex 009	9. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RO	0x1609 (5641_{dec})
1C12:0A	SubIndex 010	10. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RO	0x160A (5642_{dec})

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RO	0x06 (6_{dec})
1C13:01	SubIndex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RO	0x1A00 (6656_{dec})
1C13:02	SubIndex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RO	0x1A06 (6662_{dec})
1C13:03	SubIndex 003	3. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RO	0x1A07 (6663_{dec})
1C13:04	SubIndex 004	4. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RO	0x1A08 (6664_{dec})
1C13:05	SubIndex 005	5. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RO	0x1A09 (6665_{dec})
1C13:06	SubIndex 006	6. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RO	0x1A0A (6666_{dec})

Index 1C32 SM output parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20 (32_{dec})
1C32:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 2 Event • 2: DC-Mode - Synchron with SYNC0 Event • 3: DC-Mode - Synchron with SYNC1 Event	UINT16	RW	0x0001 (1_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchron with SM 2 Event: Master cycle time • DC-Mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x00000000 (0_{dec})
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0 = 1: free run is supported • Bit 1 = 1: Synchronous with SM 2 event is supported • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 10: Output shift with SYNC1 event (only DC mode) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08)	UINT16	RO	0xC007 (49159_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x00000000 (0_{dec})
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:08	Command	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 0x1C32:03 [▸ 106], 0x1C32:05 [▸ 106], 0x1C32:06 [▸ 106], 0x1C32:09 [▸ 106], 0x1C33:03 [▸ 107], 0x1C33:06 [▸ 106], 0x1C33:09 [▸ 107] are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RW	0x0000 (0_{dec})
1C32:09	Delay time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:0C	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})
1C32:0D	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})
1C32:20	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 1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
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 • 1: Synchron with SM 3 Event (no outputs available) • 2: DC - Synchronous with SYNC0 Event • 3: DC - Synchron with SYNC1 Event • 34: Synchron with SM 2 Event (outputs available)	UINT16	RW	0x0022 (34_{dec})
1C33:02	Cycle time	as 0x1C32:02 [▸ 106]	UINT32	RW	0x00000000 (0_{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: free run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron with SM 3 Event is supported (no outputs available) • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 01: input shift through local event (outputs available) • Bit 4-5 = 10: input shift with SYNC1 event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08 [▸ 106] or 0x1C33:08 [▸ 107])	UINT16	RO	0xC007 (49159_{dec})
1C33:05	Minimum cycle time	as 0x1C32:05 [▸ 106]	UINT32	RO	0x00000000 (0_{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})
1C33:08	Command	as 0x1C32:08 [▸ 106]	UINT16	RW	0x0000 (0_{dec})
1C33:09	Delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:0B	SM event missed counter	as 0x1C32:11 [▸ 106]	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	as 0x1C32:12 [▸ 106]	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	as 0x1C32:13 [▸ 106]	UINT16	RO	0x0000 (0_{dec})
1C33:20	Sync error	as 0x1C32:32 [▸ 106]	BOOLEAN	RO	0x00 (0_{dec})

5.4 Profile-specific objects (0x6000-0xFFFF)

The profile-specific objects have the same meaning for all EtherCAT slaves that support the profile 5001.

Index 6000 DI Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
6000:0	DI Inputs	Maximum subindex	UINT8	RO	0x10 (16_{dec})
6000:01	Input 0	Input 0	BOOLEAN	RO	0x00 (0_{dec})
6000:02	Input 1	Input 1	BOOLEAN	RO	0x00 (0_{dec})
6000:03	Input 2	Input 2	BOOLEAN	RO	0x00 (0_{dec})
6000:04	Input 3	Input 3	BOOLEAN	RO	0x00 (0_{dec})
6000:05	Input 4	Input 4	BOOLEAN	RO	0x00 (0_{dec})
6000:06	Input 5	Input 5	BOOLEAN	RO	0x00 (0_{dec})
6000:07	Input 6	Input 6	BOOLEAN	RO	0x00 (0_{dec})
6000:08	Input 7	Input 7	BOOLEAN	RO	0x00 (0_{dec})
6000:09	Input 8	Input 8	BOOLEAN	RO	0x00 (0_{dec})
6000:0A	Input 9	Input 9	BOOLEAN	RO	0x00 (0_{dec})
6000:0B	Input 10	Input 10	BOOLEAN	RO	0x00 (0_{dec})
6000:0C	Input 11	Input 11	BOOLEAN	RO	0x00 (0_{dec})
6000:0D	Input 12	Input 12	BOOLEAN	RO	0x00 (0_{dec})
6000:0F	Input 14	Input 14	BOOLEAN	RO	0x00 (0_{dec})
6000:10	Input 15	Input 15	BOOLEAN	RO	0x00 (0_{dec})

Index 6060 ENC Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
6060:0	ENC Inputs Ch.1	Maximum subindex	UINT8	RO	0x12 (18_{dec})
6060:02	Latch extern valid	The counter value was stored via the external latch (index 0x7060:02 [▸ 110] or 0x7060:04 [▸ 110] is set). To re-activate the latch input, 0x7060:02 [▸ 110] or 0x7060:04 [▸ 110] must be cancelled and then reset.	BOOLEAN	RO	0x00 (0_{dec})
6060:03	Set counter done	Counter was set	BOOLEAN	RO	0x00 (0_{dec})
6060:09	Status of input A	Status of input A	BOOLEAN	RO	0x00 (0_{dec})
6060:0A	Status of input B	Status of input B	BOOLEAN	RO	0x00 (0_{dec})
6060:0C	Status of input gate	Status of the gate input	BOOLEAN	RO	0x00 (0_{dec})
6060:0D	Status of extern latch	Status of the ext. latch input	BOOLEAN	RO	0x00 (0_{dec})
6060:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated.	BOOLEAN	RO	0x00 (0_{dec})
6060:11	Counter value	Counter value	UINT16	RO	0x0000 (0_{dec})
6060:12	Latch value	Latch value	UINT16	RO	0x0000 (0_{dec})

Index 6070 ENC Inputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
6070:0	ENC Inputs Ch.2	Maximum subindex	UINT8	RO	0x12 (18_{dec})
6070:02	Latch extern valid	The counter value was stored via the external latch (index 0x7070:02 [▸ 111] or 0x7070:04 [▸ 111] is set). To re-activate the latch input, 0x7070:02 [▸ 111] or 0x7070:04 [▸ 111] must be cancelled and then reset.	BOOLEAN	RO	0x00 (0_{dec})
6070:03	Set counter done	Counter was set	BOOLEAN	RO	0x00 (0_{dec})
6070:09	Status of input A	Status of input A	BOOLEAN	RO	0x00 (0_{dec})
6070:0A	Status of input B	Status of input B	BOOLEAN	RO	0x00 (0_{dec})
6070:0C	Status of input gate	Status of the gate input	BOOLEAN	RO	0x00 (0_{dec})
6070:0D	Status of extern latch	Status of the ext. latch input	BOOLEAN	RO	0x00 (0_{dec})
6070:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated.	BOOLEAN	RO	0x00 (0_{dec})
6070:11	Counter value	Counter value	UINT16	RO	0x0000 (0_{dec})
6070:12	Latch value	Latch value	UINT16	RO	0x0000 (0_{dec})

Index 6080 ENC Inputs Ch.3

Index (hex)	Name	Meaning	Data type	Flags	Default
6080:0	ENC Inputs Ch.3	Maximum subindex	UINT8	RO	0x12 (18_{dec})
6080:02	Latch extern valid	The counter value was stored via the external latch (index 0x7080:02 [▸ 111] or 0x7080:04 [▸ 111] is set). To re-activate the latch input, 0x7080:02 [▸ 111] or 0x7080:04 [▸ 111] must be cancelled and then reset.	BOOLEAN	RO	0x00 (0_{dec})
6080:03	Set counter done	Counter was set	BOOLEAN	RO	0x00 (0_{dec})
6080:09	Status of input A	Status of input A	BOOLEAN	RO	0x00 (0_{dec})
6080:0A	Status of input B	Status of input B	BOOLEAN	RO	0x00 (0_{dec})
6080:0C	Status of input gate	Status of the gate input	BOOLEAN	RO	0x00 (0_{dec})
6080:0D	Status of extern latch	Status of the ext. latch input	BOOLEAN	RO	0x00 (0_{dec})
6080:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated.	BOOLEAN	RO	0x00 (0_{dec})
6080:11	Counter value	Counter value	UINT16	RO	0x0000 (0_{dec})
6080:12	Latch value	Latch value	UINT16	RO	0x0000 (0_{dec})

Index 6090 ENC Inputs Ch.4

Index (hex)	Name	Meaning	Data type	Flags	Default
6090:0	ENC Inputs Ch.4	Maximum subindex	UINT8	RO	0x12 (18_{dec})
6090:02	Latch extern valid	The counter value was stored via the external latch (index 0x7080:02 [▸ 111] or 0x7080:04 [▸ 111] is set). To re-activate the latch input, 0x7080:02 [▸ 111] or 0x7080:04 [▸ 111] must be cancelled and then reset.	BOOLEAN	RO	0x00 (0_{dec})
6090:03	Set counter done	Counter was set	BOOLEAN	RO	0x00 (0_{dec})
6090:09	Status of input A	Status of input A	BOOLEAN	RO	0x00 (0_{dec})
6090:0A	Status of input B	Status of input B	BOOLEAN	RO	0x00 (0_{dec})
6090:0C	Status of input gate	Status of the gate input	BOOLEAN	RO	0x00 (0_{dec})
6090:0D	Status of extern latch	Status of the ext. latch input	BOOLEAN	RO	0x00 (0_{dec})
6090:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated.	BOOLEAN	RO	0x00 (0_{dec})
6090:11	Counter value	Counter value	UINT16	RO	0x0000 (0_{dec})
6090:12	Latch value	Latch value	UINT16	RO	0x0000 (0_{dec})

Index 60A0 PLS Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
60A0:0	PLS Inputs	Maximum subindex	UINT8	RO	0x02 (2_{dec})
60A0:01	PLS Enabled	PLS function enabled	BOOLEAN	RO	0x00 (0_{dec})
60A0:02	Unequal SI:0	Bit is set if the subindex 0 of indices 0x80A1 [▸ 111] and 0x80A2 [▸ 111] are different	BOOLEAN	RO	0x00 (0_{dec})

Index 7010 DO Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
7010:0	DO Outputs	Maximum subindex	UINT8	RO	0x10 (16_{dec})
7010:01	Output 0	Output 0	BOOLEAN	RO	0x00 (0_{dec})
7010:02	Output 1	Output 1	BOOLEAN	RO	0x00 (0_{dec})
7010:03	Output 2	Output 2	BOOLEAN	RO	0x00 (0_{dec})
7010:04	Output 3	Output 3	BOOLEAN	RO	0x00 (0_{dec})
7010:05	Output 4	Output 4	BOOLEAN	RO	0x00 (0_{dec})
7010:06	Output 5	Output 5	BOOLEAN	RO	0x00 (0_{dec})
7010:07	Output 6	Output 6	BOOLEAN	RO	0x00 (0_{dec})
7010:08	Output 7	Output 7	BOOLEAN	RO	0x00 (0_{dec})
7010:09	Output 8	Output 8	BOOLEAN	RO	0x00 (0_{dec})
7010:0A	Output 9	Output 9	BOOLEAN	RO	0x00 (0_{dec})
7010:0B	Output 10	Output 10	BOOLEAN	RO	0x00 (0_{dec})
7010:0C	Output 11	Output 11	BOOLEAN	RO	0x00 (0_{dec})
7010:0D	Output 12	Output 12	BOOLEAN	RO	0x00 (0_{dec})
7010:0F	Output 14	Output 14	BOOLEAN	RO	0x00 (0_{dec})
7010:10	Output 15	Output 15	BOOLEAN	RO	0x00 (0_{dec})

Index 7020 AO Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
7020:0	AO Outputs Ch.1	Maximum subindex	UINT8	RO	0x11 (17_{dec})
7020:11	Analog output	Analog output value	INT16	RO	0x0000 (0_{dec})

Index 7030 AO Outputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
7030:0	AO Outputs Ch.2	Maximum subindex	UINT8	RO	0x11 (17_{dec})
7030:11	Analog output	Analog output value	INT16	RO	0x0000 (0_{dec})

Index 7040 AO Outputs Ch.3

Index (hex)	Name	Meaning	Data type	Flags	Default
7040:0	AO Outputs Ch.3	Maximum subindex	UINT8	RO	0x11 (17_{dec})
7040:11	Analog output	Analog output value	INT16	RO	0x0000 (0_{dec})

Index 7050 AO Outputs Ch.4

Index (hex)	Name	Meaning	Data type	Flags	Default
7050:0	AO Outputs Ch.4	Maximum subindex	UINT8	RO	0x11 (17_{dec})
7050:11	Analog output	Analog output value	INT16	RO	0x0000 (0_{dec})

Index 7060 ENC Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
7060:0	ENC Outputs Ch.1	Maximum subindex	UINT8	RO	0x11 (17_{dec})
7060:02	Enable latch extern on positive edge	The external latch is enabled with a positive edge	BOOLEAN	RO	0x00 (0_{dec})
7060:03	Set counter	Set counter	BOOLEAN	RO	0x00 (0_{dec})
7060:04	Enable latch extern on negative edge	The external latch is enabled with a negative edge	BOOLEAN	RO	0x00 (0_{dec})
7060:11	Set counter value	Value of the counter value to be set via 0x7060:03 [▶ 110]	UINT16	RO	0x0000 (0_{dec})

Index 7070 ENC Outputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
7070:0	ENC Outputs Ch.2	Maximum subindex	UINT8	RO	0x11 (17_{dec})
7070:02	Enable latch extern on positive edge	The external latch is enabled with a positive edge	BOOLEAN	RO	0x00 (0_{dec})
7070:03	Set counter	Set counter	BOOLEAN	RO	0x00 (0_{dec})
7070:04	Enable latch extern on negative edge	The external latch is enabled with a negative edge	BOOLEAN	RO	0x00 (0_{dec})
7070:11	Set counter value	Value of the counter value to be set via 0x7070:03 [▶ 111]	UINT16	RO	0x0000 (0_{dec})

Index 7080 ENC Outputs Ch.3

Index (hex)	Name	Meaning	Data type	Flags	Default
7080:0	ENC Outputs Ch.3	Maximum subindex	UINT8	RO	0x11 (17_{dec})
7080:02	Enable latch extern on positive edge	The external latch is enabled with a positive edge	BOOLEAN	RO	0x00 (0_{dec})
7080:03	Set counter	Set counter	BOOLEAN	RO	0x00 (0_{dec})
7080:04	Enable latch extern on negative edge	The external latch is enabled with a negative edge	BOOLEAN	RO	0x00 (0_{dec})
7080:11	Set counter value	Value of the counter value to be set via 0x7080:03 [▶ 111]	UINT16	RO	0x0000 (0_{dec})

Index 7090 ENC Outputs Ch.4

Index (hex)	Name	Meaning	Data type	Flags	Default
7090:0	ENC Outputs Ch.4	Maximum subindex	UINT8	RO	0x11 (17_{dec})
7090:02	Enable latch extern on positive edge	The external latch is enabled with a positive edge	BOOLEAN	RO	0x00 (0_{dec})
7090:03	Set counter	Set counter	BOOLEAN	RO	0x00 (0_{dec})
7090:04	Enable latch extern on negative edge	The external latch is enabled with a negative edge	BOOLEAN	RO	0x00 (0_{dec})
7090:11	Set counter value	Value of the counter value to be set via 0x7090:03 [▶ 111]	UINT16	RO	0x0000 (0_{dec})

Index 70A0 PLS Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
70A0:0	PLS Outputs	Maximum subindex	UINT8	RO	0x01 (1_{dec})
70A0:01	Enable PLS	Activates PLS function	BOOLEAN	RO	0x00 (0_{dec})

Index 80A1 PLS Switch values

Index (hex)	Name	Meaning	Data type	Flags	Default
80A1:0	PLS Switch values	Maximum subindex	UINT8	RW	0x00 (0_{dec})
80A1:01	SubIndex 001	Switch value 1	UINT16	RW	0x0000 (0_{dec})
...	...	...	...	...	...
80A1:4A	SubIndex 074	Switch value 74	UINT16	RW	0x0000 (0_{dec})
80A1:4B	SubIndex 075	Switch value 75	UINT16	RW	0x0000 (0_{dec})

Index 80A2 PLS Output data

Index (hex)	Name	Meaning	Data type	Flags	Default
80A2:0	PLS Output data	Maximum subindex	UINT8	RW	0x00 (0_{dec})
80A2:01	SubIndex 001	Switch value 1	UINT16	RW	0x0000 (0_{dec})
...	...	...	...	...	...
80A2:4A	SubIndex 074	Switch value 74	UINT16	RW	0x0000 (0_{dec})
80A2:4B	SubIndex 075	Switch value 75	UINT16	RW	0x0000 (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	0x000B (11_{dec})

Index F008 Code word

Note

Code Wort

The vendor reserves the authority for the basic calibration of the terminals. The code word is therefore at present reserved.

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

Index F010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default
F010:0	Module list	Maximum subindex	UINT8	RW	0x0B (11_{dec})
F010:01	SubIndex 001	Digital Input Profil (100)	UINT32	RW	0x00000064 (100_{dec})
F010:02	SubIndex 002	Digital Output Profil (200)	UINT32	RW	0x000000C8 (200_{dec})
F010:03	SubIndex 003	Analog Output Profil (400)	UINT32	RW	0x00000190 (400_{dec})
F010:04	SubIndex 004	Analog Output Profil (400)	UINT32	RW	0x00000190 (400_{dec})
F010:05	SubIndex 005	Analog Output Profil (400)	UINT32	RW	0x00000190 (400_{dec})
F010:06	SubIndex 006	Analog Output Profil (400)	UINT32	RW	0x00000190 (400_{dec})
F010:07	SubIndex 007	Incremental Encoder Input Profil (510)	UINT32	RW	0x000001FF (511_{dec})
F010:08	SubIndex 008	Incremental Encoder Input Profil (510)	UINT32	RW	0x000001FF (511_{dec})
F010:09	SubIndex 009	Incremental Encoder Input Profil (510)	UINT32	RW	0x000001FF (511_{dec})
F010:0A	SubIndex 010	Incremental Encoder Input Profil (510)	UINT32	RW	0x000001FF (511_{dec})
F010:0B	SubIndex 011	Programmable Limit Switch (PLS) Profil (511)	UINT32	RW	0x00000200 (512_{dec})

6 Appendix

6.1 Ordering information for EM7004 modules and EM/KM connectors

The following tables show the combination options and ordering information for the modules and connectors.

Order identifier	enclosed EM/KM connectors
EM7004-0000	none
EM7004-0002	4x ZS2001-0002; 4x ZS2001-0005
EM7004-0004	4x ZS2001-0004; 4x ZS2001-0005

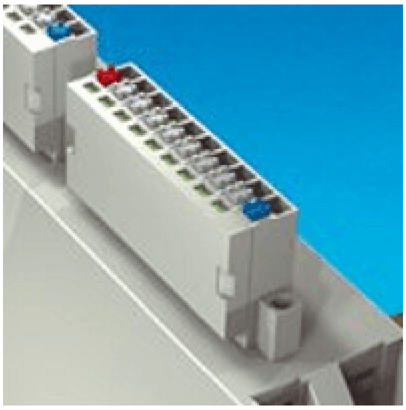


Fig. 127: EM / KM connector ZS2001-0002

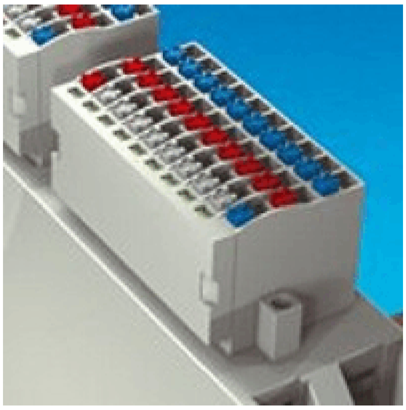


Fig. 128: EM / KM connector ZS2001-0004

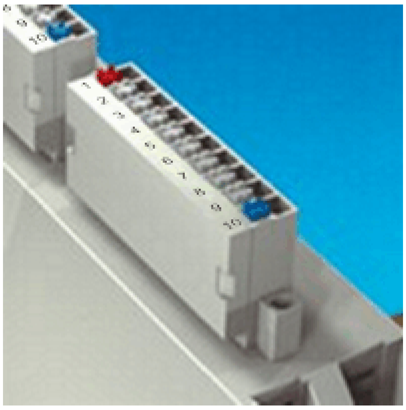


Fig. 129: EM / KM connector ZS2001-0005

Order identifier	Labelling	Signal LEDs	Connection technology		
			Single-conductor	Two-conductor	Three-conductor
ZS2001-0002	24 , 0 - 7, 0 V	yes	yes	no	no
ZS2001-0004	24 , 0 - 7, 0 V	yes	yes	yes	yes
ZS2001-0005	universal (1 - 10)	no	yes	no	no

Note

Suitable connectors

The suitable connectors have to be adapted to the application. The ZS2001-0004 (4x) with signal LEDs is particularly suitable for controlling the digital inputs and outputs. The connector ZS2001-0005 (4x) is particularly suitable for application at the incremental encoder, since the additional 24 V and 0 V terminal segment is not required in this case.

6.2 EtherCAT AL Status Codes

For detailed information please refer to the EtherCAT system description.

6.3 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.

Attention

Risk of damage to the device!

Pay attention to the instructions for firmware updates on the separate page [▶ 114].
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!

EM7004			
Hardware (HW)	**Firmware (FW)**	**Revision No.**	**Date of release**
06 - 10*	04	EM7004-0000-0017	2010/11
	05*		2011/11

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

6.4 Firmware Update EL/ES/EM/EPxxxx

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

Storage locations

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

- Depending on functionality and performance EtherCAT slaves have one or several local controllers for processing I/O data. The corresponding program is the so-called **firmware** in *.efw format.
- In some EtherCAT slaves the EtherCAT communication may also be integrated in these controllers. In this case the controller is usually a so-called **FPGA** chip with *.rbf firmware.
- In addition, each EtherCAT slave has a memory chip, a so-called **ESI-EEPROM**, for storing its own device description (ESI: EtherCAT Slave Information). On power-up this description is loaded and the EtherCAT communication is set up accordingly. The device description is available from the download area of the Beckhoff website at (http://www.beckhoff.de). All ESI files are accessible there as zip files.

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

The TwinCAT System Manager offers mechanisms for programming all 3 parts with new data, if the slave is set up for this purpose. Generally the slave does not check whether the new data are suitable, i.e. it may no longer be able to operate if the data are unsuitable.

Simplified update by bundle firmware

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

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

Attention	**Risk of damage to the device!** Note the following when downloading new device files • Firmware downloads to an EtherCAT device must not be interrupted • Flawless EtherCAT communication must be ensured. CRC errors or LostFrames must be avoided. • 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.

Device description ESI file/XML

Attention	**Notice 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:

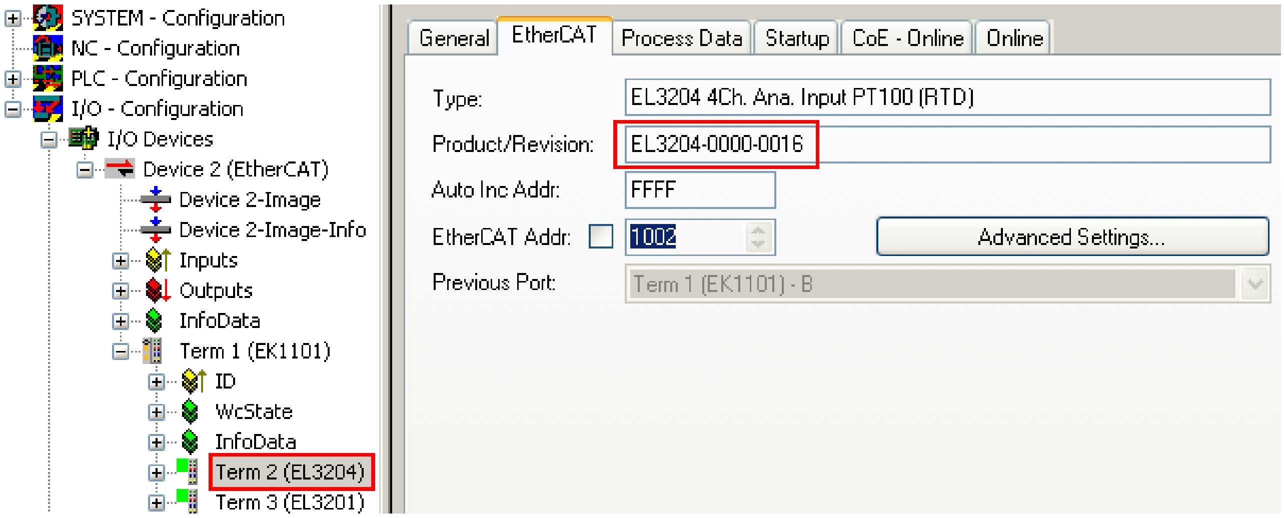


Fig. 130: *Device identifier consisting of name EL3204-0000 and revision -0016*

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.

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

Display of ESI slave identifier

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

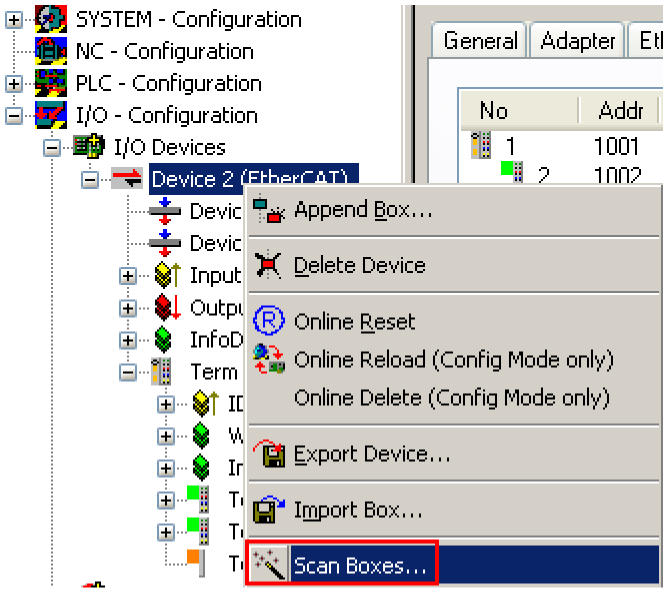


Fig. 131: *Scan the subordinate field by right-clicking on the EtherCAT device in Config/FreeRun mode*

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

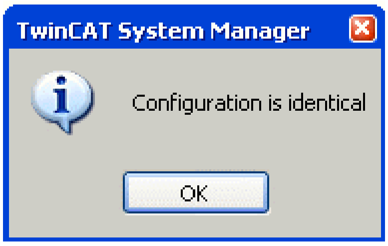


Fig. 132: *Configuration is identical*

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

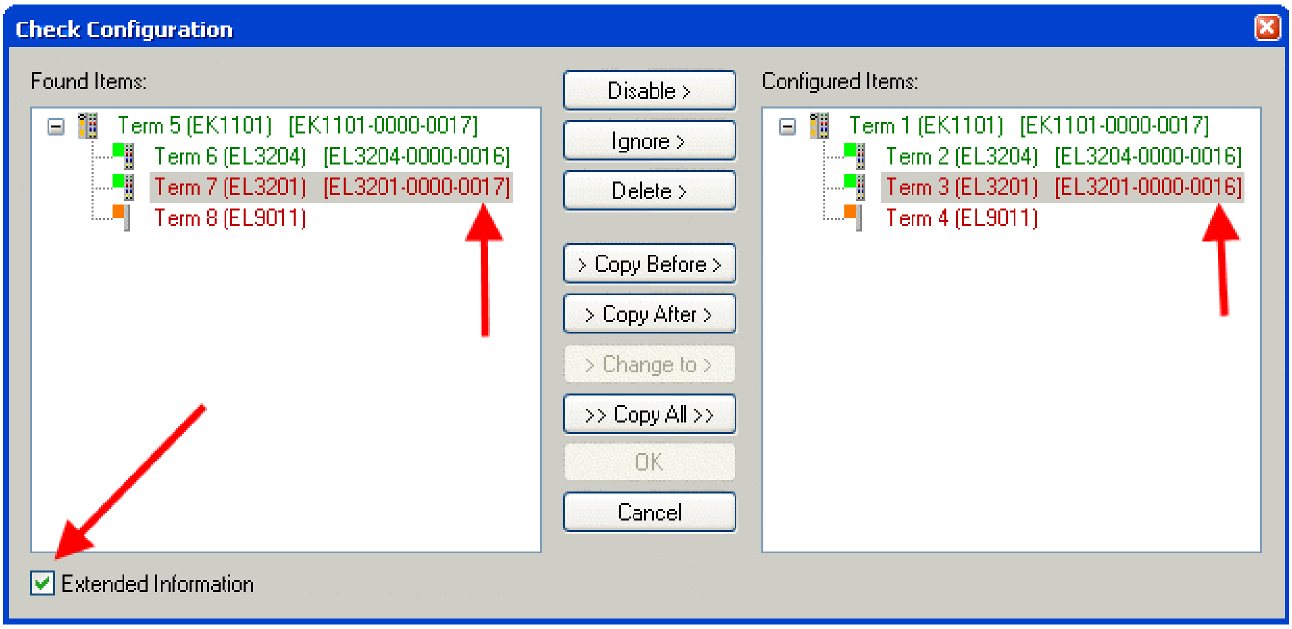


Fig. 133: *Change dialog*

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"*

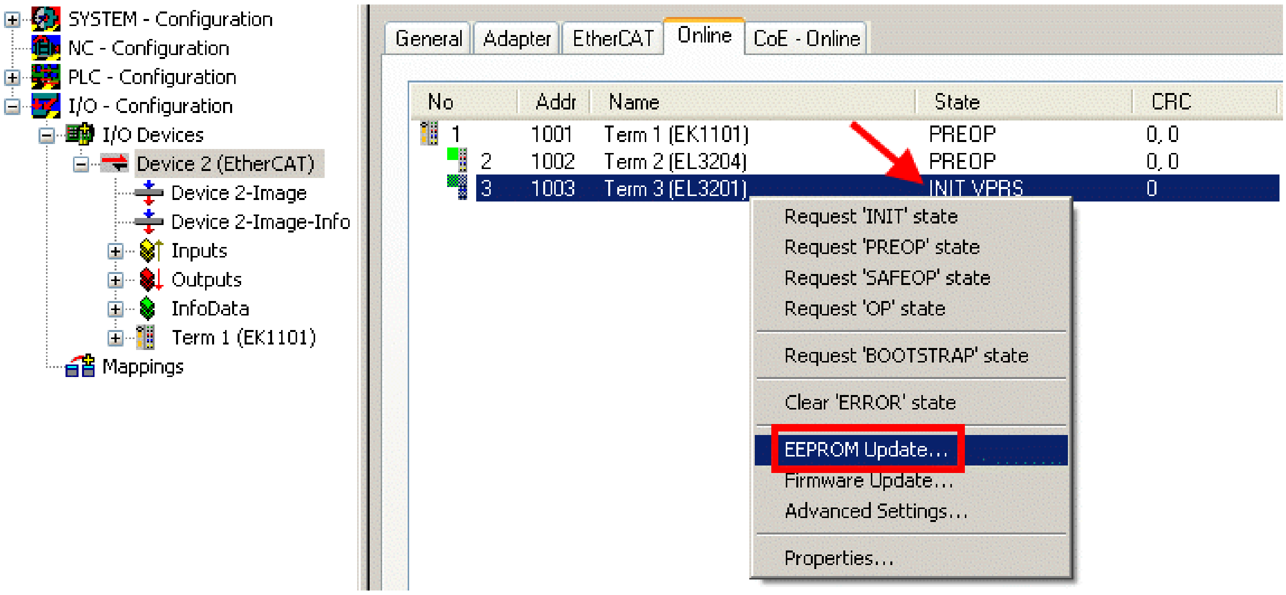


Fig. 134: *EEPROM Update*

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.

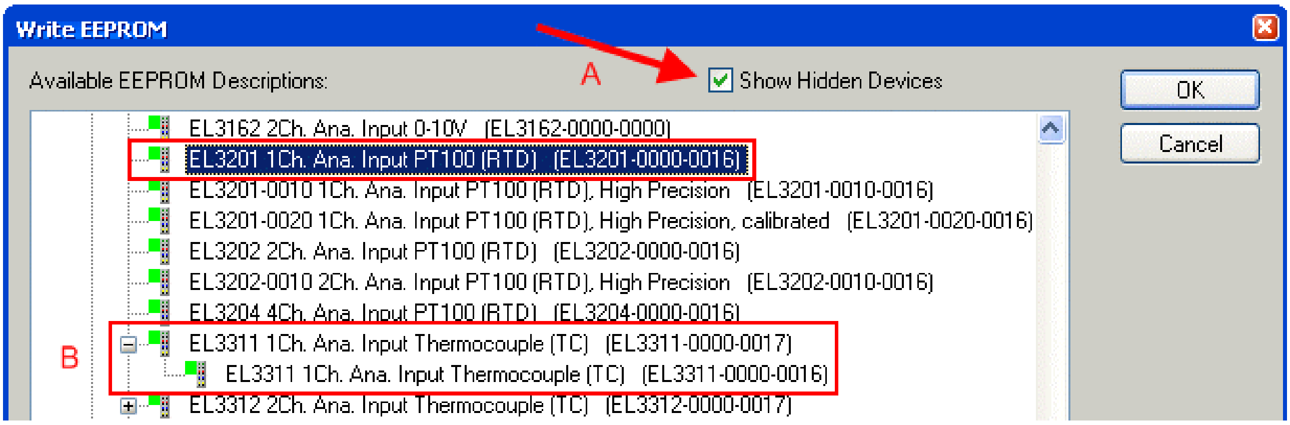


Fig. 135: *Selecting the new ESI*

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

Note	**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.

Determining the firmware version

Determining the version on laser inscription

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

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

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

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

Determining the version via the System Manager

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

Note	**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 does supported it. 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.

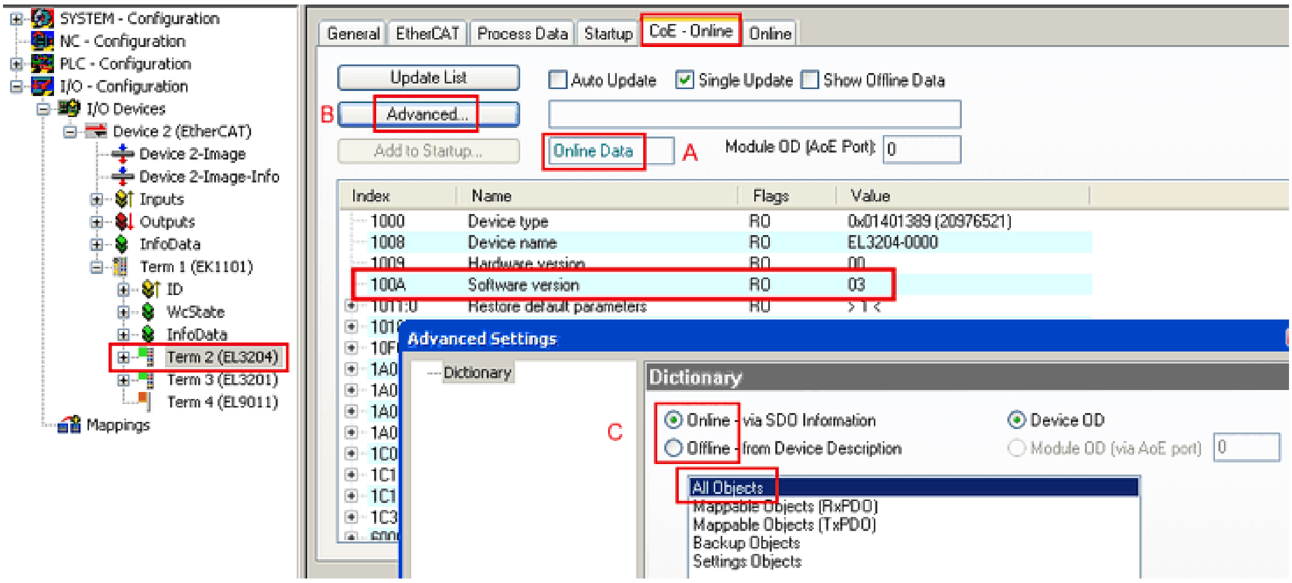


Fig. 136: *Display of EL3204 firmware version*

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

Updating controller firmware *.efw

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

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

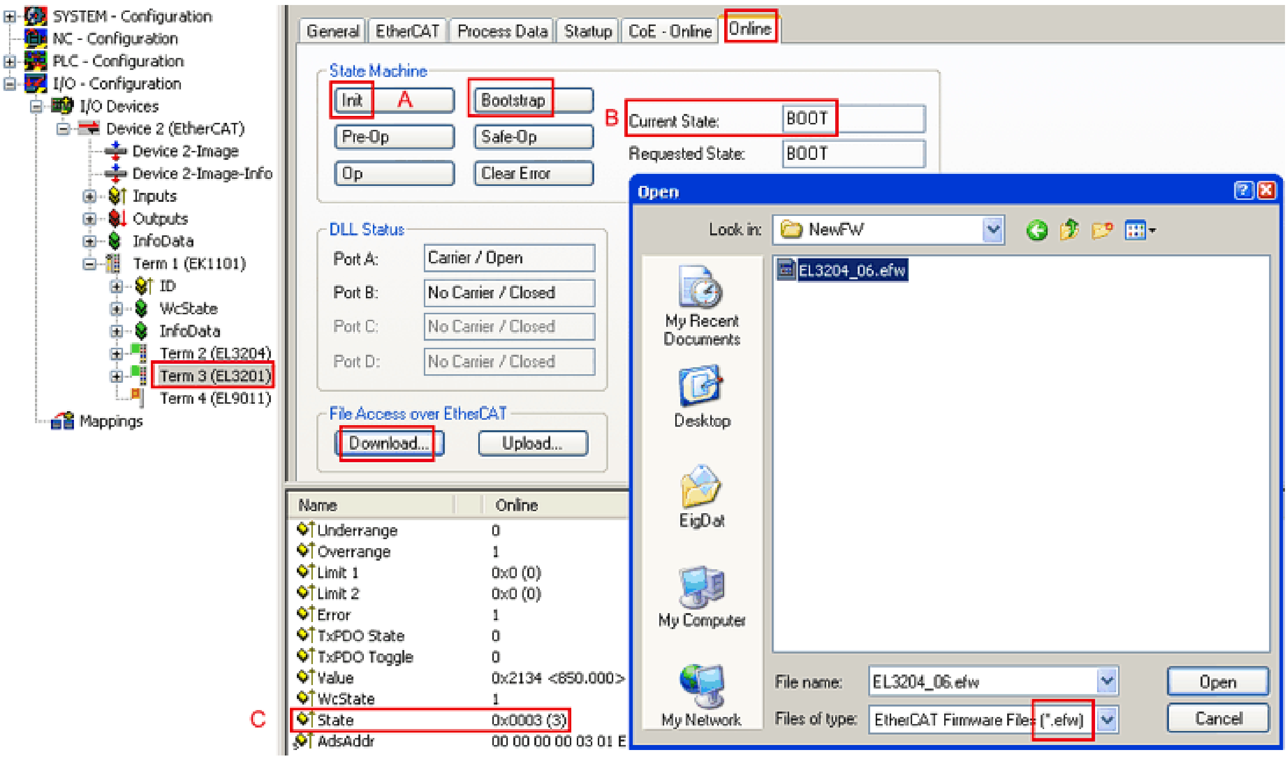


Fig. 137: *Firmware Update*

Proceed as follows, unless instructed otherwise by Beckhoff support.

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

FPGA firmware *.rbf

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

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

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

Determining the version via the System Manager

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

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

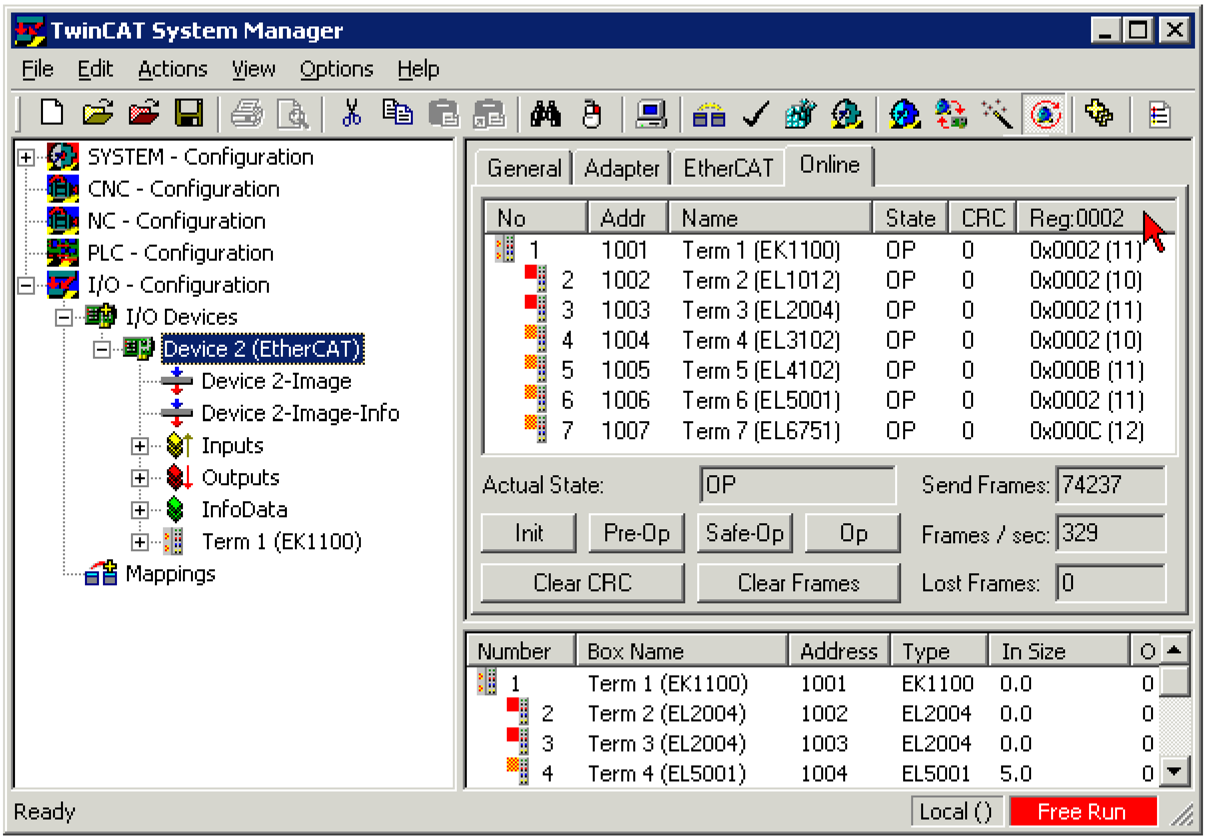


Fig. 138: *FPGA firmware version definition*

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

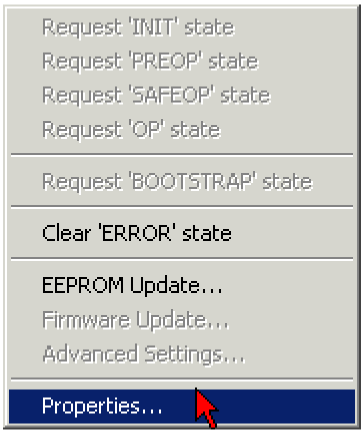


Fig. 139: *Context menu Properties*

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.

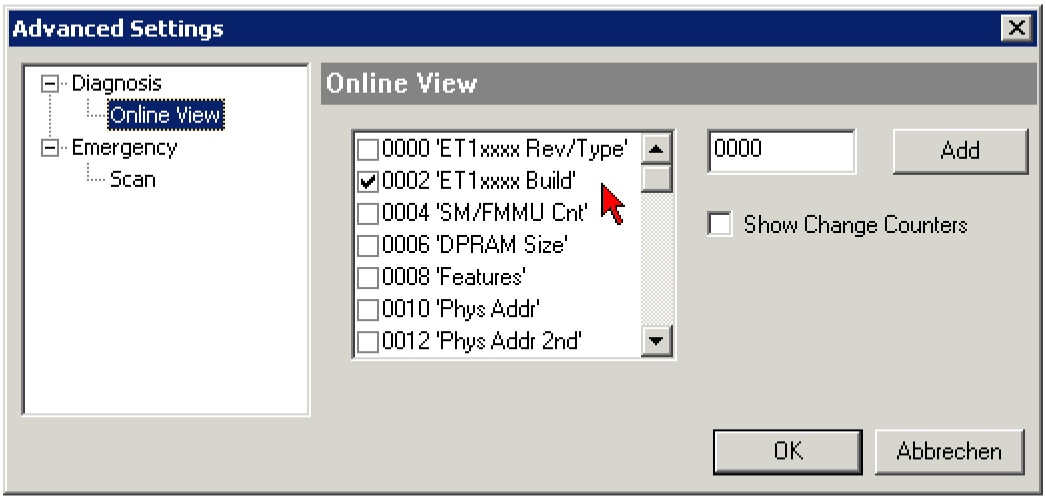


Fig. 140: *Dialog Advanced Settings*

Update

For updating the FPGA firmware

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

Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and
click the *Advanced Settings* button in the *EtherCAT* tab.

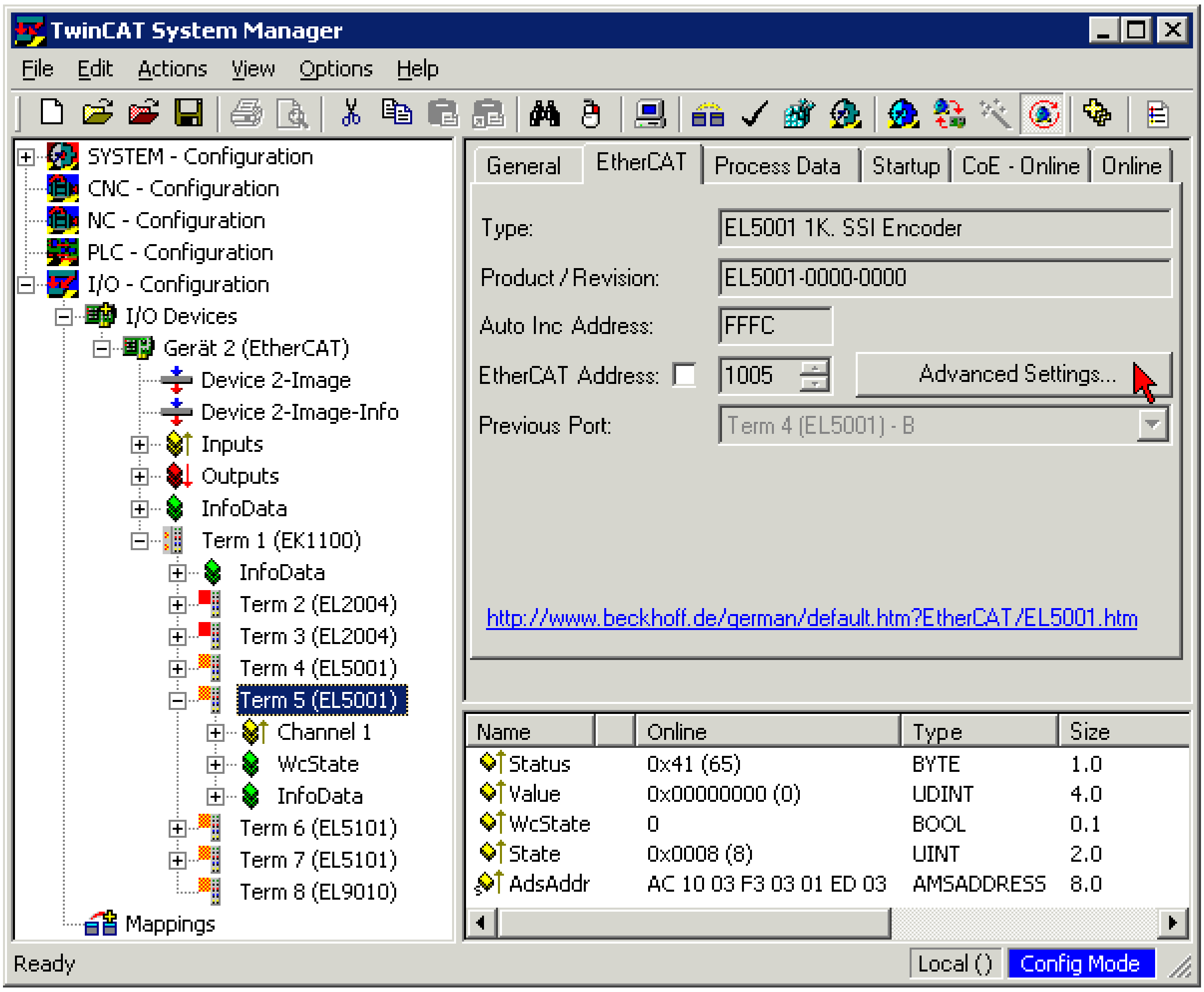


Fig. 141: *Select dialog Advanced Settings*

The *Advanced Settings* dialog appears. Under *ESC Access/E²PROM/***FPGA** click on *Write FPGA* button,

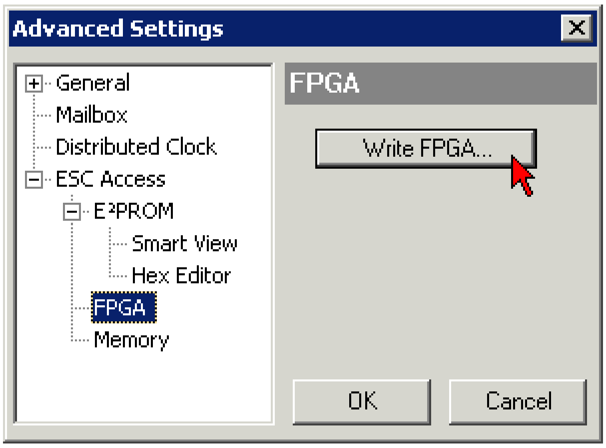


Fig. 142: *Select dialog Write FPGA*

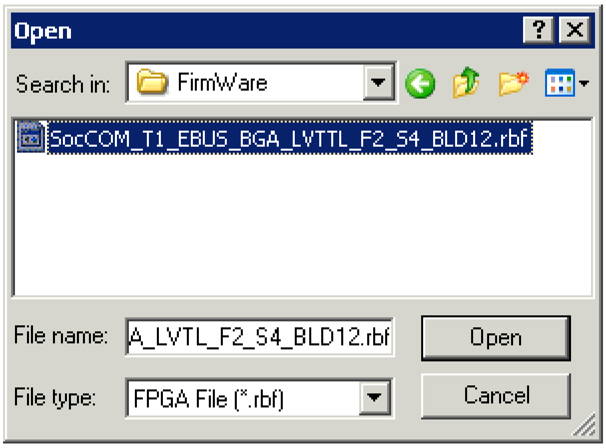


Fig. 143: *Select file*

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

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

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

Simultaneous updating of several EtherCAT devices

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

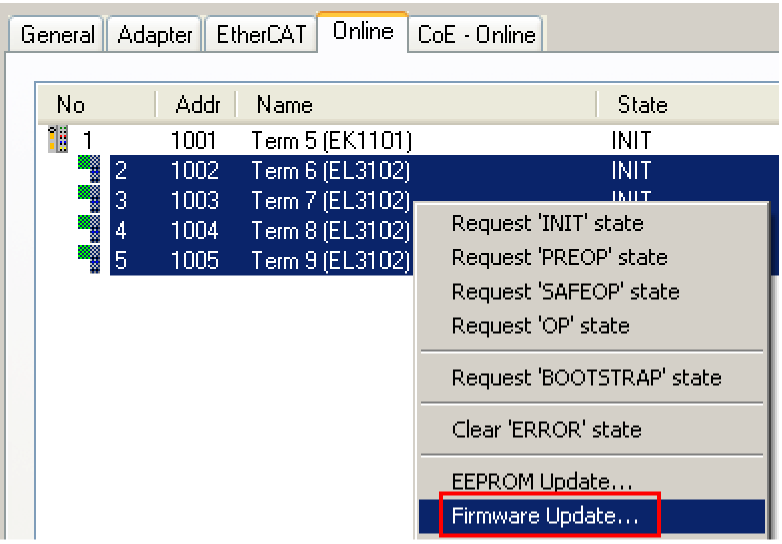


Fig. 144: *Multiple selection and firmware update*

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

6.5 Restoring the delivery state

Restoring the delivery state To restore the delivery state for backup objects in ELxxxx terminals, the CoE object "Restore default parameters", SubIndex 001 can be selected in the TwinCAT System Manager (Config mode) (see Fig. *“Selecting the ‘Restore default parameters’ PDO”*)

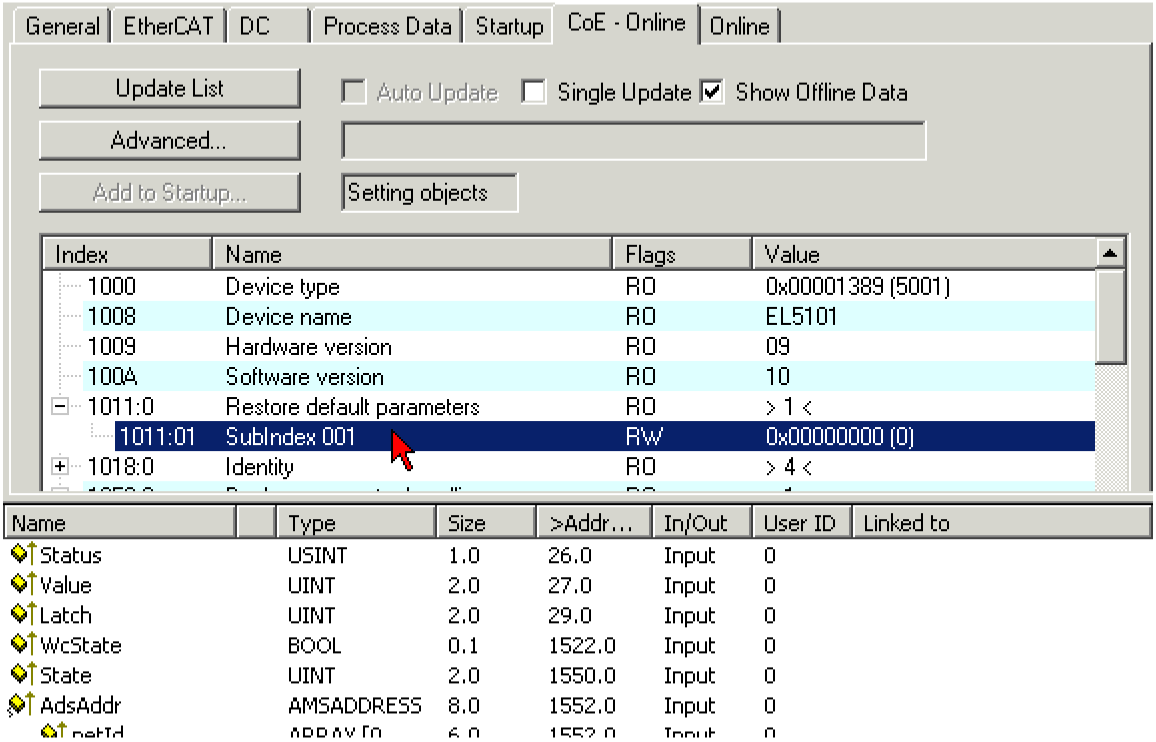


Fig. 145: *Selecting the "Restore default parameters" PDO*

Double-click on SubIndex 001 to enter the Set Value dialog. Enter the value **1684107116** in field "Dec" or the value **0x64616F6C** in field "Hex" and confirm with *OK* (Fig. *“Entering a restore value in the Set Value dialog”*).
All backup objects are reset to the delivery state.

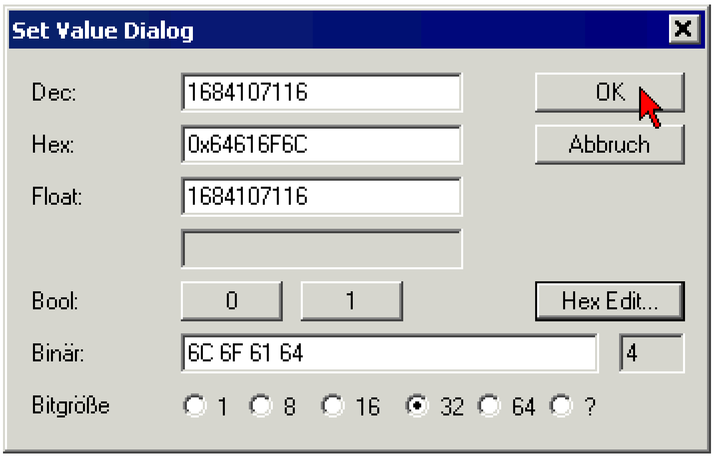


Fig. 146: *Entering a restore value in the Set Value dialog*

Note	**Alternative restore value** In some older terminals the backup objects can be switched with an alternative restore value:Decimal value: "1819238756", Hexadecimal value: "0x6C6F6164"An incorrect entry for the restore value has no effect.

6.6 Support and Service

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

Beckhoff's branch offices and representatives

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

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

You will also find further documentation for Beckhoff components there.

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

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

Beckhoff Support

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

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

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

Beckhoff Service

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

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

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

List of illustrations