

Documentation | EN

EL6740-0010

Interbus Slave Terminal

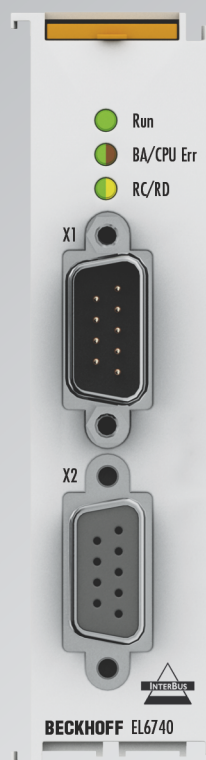


Table of contents

| | |
|--|-----------|
| 1 Foreword | 5 |
| 1.1 Notes on the documentation | 5 |
| 1.2 Safety instructions | 6 |
| 1.3 Documentation issue status | 7 |
| 1.4 Version identification of EtherCAT devices | 8 |
| 1.4.1 General notes on marking..... | 8 |
| 1.4.2 Version identification of EL terminals..... | 9 |
| 1.4.3 Beckhoff Identification Code (BIC)..... | 10 |
| 1.4.4 Electronic access to the BIC (eBIC)..... | 12 |
| 2 Product overview | 14 |
| 2.1 Introduction | 14 |
| 2.2 Technical data..... | 15 |
| 2.3 Interbus - introduction | 16 |
| 2.4 Basic function principles..... | 18 |
| 2.4.1 Configuration..... | 18 |
| 2.4.2 Changing the Interbus baud rate..... | 20 |
| 2.4.3 Resetting a device..... | 21 |
| 2.4.4 Reloading an EtherCAT device..... | 22 |
| 2.4.5 Operation of EtherCAT and Interbus master on a PC..... | 22 |
| 2.4.6 Saving the Interbus configuration..... | 24 |
| 3 Mounting and wiring | 25 |
| 3.1 Instructions for ESD protection | 25 |
| 3.2 Mounting and demounting - traction lever unlocking..... | 26 |
| 3.3 Mounting and demounting - top front unlocking | 28 |
| 3.4 Disposal | 29 |
| 3.5 Positioning of passive Terminals..... | 30 |
| 3.6 Installation positions | 31 |
| 3.7 Note - Power supply | 32 |
| 3.8 Wiring the Bus System..... | 33 |
| 3.9 Topology | 34 |
| 3.10 LEDs and connection | 35 |
| 3.11 Explosion protection..... | 37 |
| 3.11.1 ATEX - Special conditions (standard temperature range)..... | 37 |
| 3.11.2 IECEx - Special conditions..... | 38 |
| 3.11.3 Continuative documentation for ATEX and IECEx..... | 39 |
| 3.12 UL notice | 39 |
| 4 Basics communication | 40 |
| 4.1 EtherCAT basics | 40 |
| 4.2 EtherCAT State Machine | 40 |
| 4.3 General notes for setting the watchdog | 41 |
| 4.4 CoE Interface | 43 |
| 5 Commissioning | 48 |
| 5.1 TwinCAT Quick Start..... | 48 |

| | | |
|----------|--|------------|
| 5.1.1 | TwinCAT 2 | 51 |
| 5.1.2 | TwinCAT 3 | 61 |
| 5.2 | TwinCAT Development Environment | 74 |
| 5.2.1 | Installation of the TwinCAT real-time driver | 75 |
| 5.2.2 | Notes regarding ESI device description | 81 |
| 5.2.3 | TwinCAT ESI Updater | 85 |
| 5.2.4 | OFFLINE configuration creation..... | 85 |
| 5.2.5 | ONLINE configuration creation | 90 |
| 5.2.6 | EtherCAT subscriber configuration | 98 |
| 5.3 | General Commissioning Instructions for an EtherCAT Slave | 107 |
| 5.4 | Object description and parameterization..... | 115 |
| 5.4.1 | Standard objects | 115 |
| 5.4.2 | Profile-specific objects (0x6000-0xFFFF) | 118 |
| 6 | Appendix..... | 120 |
| 6.1 | EtherCAT AL Status Codes | 120 |
| 6.2 | Firmware compatibility | 120 |
| 6.3 | Firmware Update EL/ES/EM/ELM/EPxxxx..... | 121 |
| 6.3.1 | Device description ESI file/XML | 122 |
| 6.3.2 | Firmware explanation..... | 125 |
| 6.3.3 | Updating controller firmware *.efw | 126 |
| 6.3.4 | FPGA firmware *.rbf..... | 128 |
| 6.3.5 | Simultaneous updating of several EtherCAT devices..... | 132 |
| 6.4 | Support and Service..... | 133 |

1 Foreword

1.1 Notes on the documentation

Intended audience

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

It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.

The qualified personnel is obliged to always use the currently valid documentation.

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

Disclaimer

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

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

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

Trademarks

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

Patent Pending

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



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

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

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.

Signal words

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

Personal injury warnings

⚠ DANGER

Hazard with high risk of death or serious injury.

⚠ WARNING

Hazard with medium risk of death or serious injury.

⚠ CAUTION

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

Warning of damage to property or environment

NOTICE

The environment, equipment, or data may be damaged.

Information on handling the product



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

1.3 Documentation issue status

| Version | Comment |
|---------|---|
| 2.4 | <ul style="list-style-type: none"> • Chapter "Recommended mounting rails" removed • Update chapter "Technical data" • Chapter "Instructions for ESD protection" added • Chapter "Note – Power supply" added • Chapter "IECEX - Special conditions" added |
| 2.3 | <ul style="list-style-type: none"> • Update chapter "Technical data" • Update chapter "Version identification of EtherCAT devices" • Update structure • Update notes • Update revision status • Chapter Disposal added |
| 2.2 | <ul style="list-style-type: none"> • Update chapter "UL notice" • Update chapter "Firmware compatibility" • Update structure |
| 2.1 | <ul style="list-style-type: none"> • Update chapter "Notes on the documentation" • Correction of Technical data • Update chapter "TwinCAT 2.1x" -> "TwinCAT Development Environment" and "TwinCAT Quick Start" |
| 2.0 | <ul style="list-style-type: none"> • Migration • Update structure • Revision level updated |
| 1.5 | <ul style="list-style-type: none"> • "Technical data" section updated • Revision level updated • Update structure |
| 1.4 | <ul style="list-style-type: none"> • Notes on installation position added |
| 1.3 | <ul style="list-style-type: none"> • Technical description added |
| 1.2 | <ul style="list-style-type: none"> • Description of status word amended |
| 1.1 | <ul style="list-style-type: none"> • IBS parameters extended |
| 1.0 | <ul style="list-style-type: none"> • Technical data modified, first publication |
| 0.3 | <ul style="list-style-type: none"> • Technical data modified |
| 0.2 | <ul style="list-style-type: none"> • Technical data amended |
| 0.1 | <ul style="list-style-type: none"> • provisional documentation for EL6740-0010 |

1.4 Version identification of EtherCAT devices

1.4.1 General notes on marking

Designation

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

- family key
- type
- version
- revision

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

Notes

- The elements mentioned above result in the **technical designation**. EL3314-0000-0016 is used in the example below.
- EL3314-0000 is the order identifier, in the case of "-0000" usually abbreviated to EL3314. "-0016" is the EtherCAT revision.
- The **order identifier** is made up of
 - family key (EL, EP, CU, ES, KL, CX, etc.)
 - type (3314)
 - version (-0000)
- The **revision** -0016 shows the technical progress, such as the extension of features with regard to the EtherCAT communication, and is managed by Beckhoff.
In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation.
Associated and synonymous with each revision there is usually a description (ESI, EtherCAT Slave Information) in the form of an XML file, which is available for download from the Beckhoff web site.
From 2014/01 the revision is shown on the outside of the IP20 terminals, see Fig. "EL5021 EL terminal, standard IP20 IO device with batch number and revision ID (since 2014/01)".
- The type, version and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

1.4.2 Version identification of EL terminals

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

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

KK - week of production (CW, calendar week)

YY - year of production

FF - firmware version

HH - hardware version

Example with serial number 12 06 3A 02:

12 - production week 12

06 - production year 2006

3A - firmware version 3A

02 - hardware version 02



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

1.4.3 Beckhoff Identification Code (BIC)

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

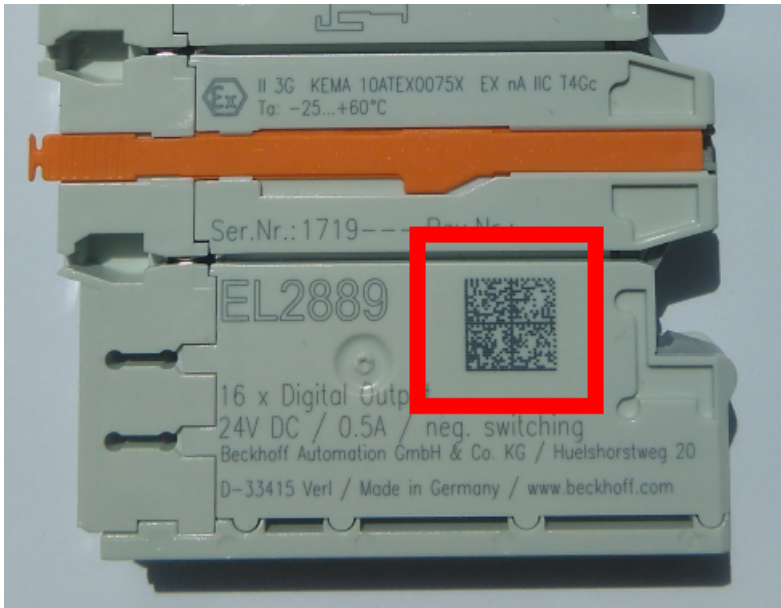


Fig. 2: BIC as data matrix code (DMC, code scheme ECC200)

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

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

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

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

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

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

| Position | Type of information | Explanation | Data identifier | Number of digits incl. data identifier | Example |
|----------|------------------------------------|---|-----------------|--|-------------------------|
| 1 | Beckhoff order number | Beckhoff order number | 1P | 8 | 1P 072222 |
| 2 | Beckhoff Traceability Number (BTN) | Unique serial number, see note below | SBTN | 12 | SBTN k4p562d7 |
| 3 | Article description | Beckhoff article description, e.g. EL1008 | 1K | 32 | 1K EL1809 |
| 4 | Quantity | Quantity in packaging unit, e.g. 1, 10, etc. | Q | 6 | Q1 |
| 5 | Batch number | Optional: Year and week of production | 2P | 14 | 2P 401503180016 |
| 6 | ID/serial number | Optional: Present-day serial number system, e.g. with safety products | 51S | 12 | 51S 678294 |
| 7 | Variant number | Optional: Product variant number on the basis of standard products | 30P | 32 | 30P F971, 2*K183 |
| ... | | | | | |

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

Structure of the BIC

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

1P072222**SBTN**k4p562d7**1K**EL1809 **Q1** **51S**678294

Accordingly as DMC:



Fig. 3: Example DMC **1P**072222**SBTN**k4p562d7**1K**EL1809 **Q1** **51S**678294

BTN

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

NOTICE

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

1.4.4 Electronic access to the BIC (eBIC)

Electronic BIC (eBIC)

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

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

K-bus devices (IP20, IP67)

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

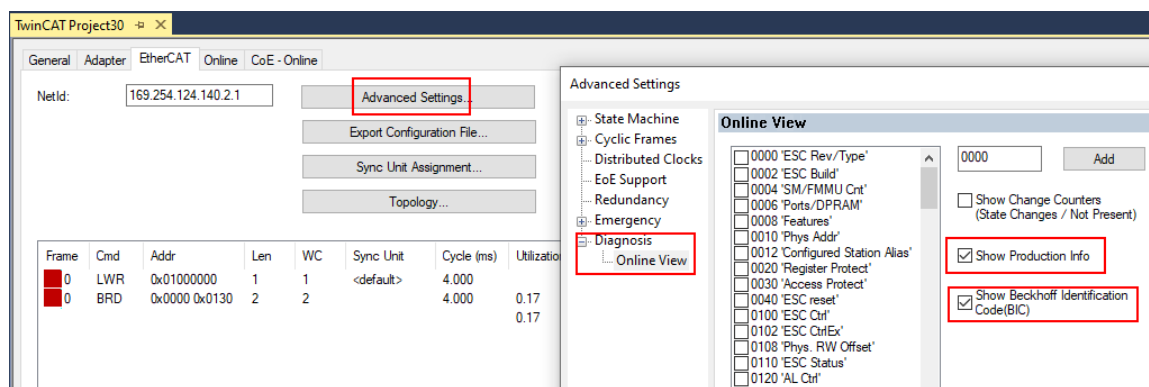
EtherCAT devices (IP20, IP67)

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

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

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

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



- The BTN and its contents are then displayed:

| No | Addr | Name | State | CRC | Fw | Hw | Production Data | ItemNo | BTN | Description | Quantity | BatchNo | SerialNo |
|----|------|-----------------|-------|-----|----|----|-----------------|--------|----------|-------------|----------|---------|----------|
| 1 | 1001 | Term 1 (EK1100) | OP | 0.0 | 0 | 0 | --- | | | | | | |
| 2 | 1002 | Term 2 (EL1018) | OP | 0.0 | 0 | 0 | 2020 KW36 Fr | 072222 | k4p562d7 | EL1809 | 1 | | 678294 |
| 3 | 1003 | Term 3 (EL3204) | OP | 0.0 | 7 | 6 | 2012 KW24 Sa | | | | | | |
| 4 | 1004 | Term 4 (EL2004) | OP | 0.0 | 0 | 0 | --- | 072223 | k4p562d7 | EL2004 | 1 | | 678295 |
| 5 | 1005 | Term 5 (EL1008) | OP | 0.0 | 0 | 0 | --- | | | | | | |
| 6 | 1006 | Term 6 (EL2008) | OP | 0.0 | 0 | 12 | 2014 KW14 Mo | | | | | | |
| 7 | 1007 | Term 7 (EK1110) | OP | 0 | 1 | 8 | 2012 KW25 Mo | | | | | | |

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

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

| Index | Name | Flags | Value |
|---------|---|-------|---|
| 1000 | Device type | RO | 0x015E1389 (22942601) |
| 1008 | Device name | RO | ELM3704-0000 |
| 1009 | Hardware version | RO | 00 |
| 100A | Software version | RO | 01 |
| 100B | Bootloader version | RO | J0.1.27.0 |
| 1011:0 | Restore default parameters | RO | > 1 < |
| 1018:0 | Identity | RO | > 4 < |
| 10E2:0 | Manufacturer-specific Identification C... | RO | > 1 < |
| 10E2:01 | SubIndex 001 | RO | 1P158442SBTN0008jckp1KELM3704 Q1 2P482001000016 |
| 10F0:0 | Backup parameter handling | RO | > 1 < |
| 10F3:0 | Diagnosis History | RO | > 21 < |
| 10F8 | Actual Time Stamp | RO | 0x170fb277e |

- The object 0x10E2 will be introduced into stock products in the course of a necessary firmware revision.
- From TwinCAT 3.1. build 4024.24 the functions *FB_EcCoEReadBIC* and *FB_EcCoEReadBTN* are available in the *Tc2_EtherCAT Library* from v3.3.19.0 for reading into the PLC.
- For processing the BIC/BTN data in the PLC, the following auxiliary functions are available in *Tc2_Uilities* from TwinCAT 3.1 build 4024.24 onwards
 - *F_SplitBIC*: The function splits the Beckhoff Identification Code (BIC) *sBICValue* into its components based on known identifiers and returns the recognized partial strings in a structure *ST_SplitBIC* as return value.
 - *BIC_TO_BTN*: The function extracts the BTN from the BIC and returns it as a value.
- Note: in the case of electronic further processing, the BTN is to be handled as a string(8); the identifier "SBTN" is not part of the BTN.
- Technical background
 The new BIC information is additionally written as a category in the ESI-EEPROM during the device production. The structure of the ESI content is largely dictated by the ETG specifications, therefore the additional vendor-specific content is stored with the help of a category according to ETG.2010. ID 03 indicates to all EtherCAT masters that they must not overwrite these data in case of an update or restore the data after an ESI update.
 The structure follows the content of the BIC, see there. This results in a memory requirement of approx. 50..200 bytes in the EEPROM.
- Special cases
 - If multiple, hierarchically arranged ESCs are installed in a device, only the top-level ESC carries the eBIC Information.
 - If multiple, non-hierarchically arranged ESCs are installed in a device, all ESCs carry the eBIC Information.
 - If the device consists of several sub-devices with their own identity, but only the top-level device is accessible via EtherCAT, the eBIC of the top-level device is located in the CoE object directory 0x10E2:01 and the eBICs of the sub-devices follow in 0x10E2:nn.

PROFIBUS, PROFINET, DeviceNet devices etc.

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

2 Product overview

2.1 Introduction

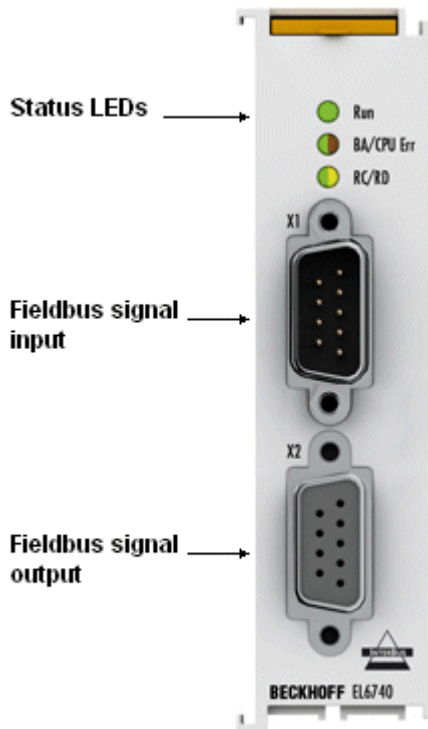


Fig. 4: EL6740

Interbus slave terminal

The Interbus slave terminal enables data exchange between EtherCAT and Interbus. For both bus systems the terminal "mirrors" up to 2 x 32 words (2 x 64 bytes) to the other system. The outputs are written to the inputs of the other bus with minimum delay. The terminal can use the Interbus protocol up to a baud rate of 2 Mbit. Due to the connection via EtherCAT, no PCI slots are required in the PC.

The Interbus baud rate can be changed manually from 2 Mbaud (default) to 500 kbaud in the CoE directory.

The EL6740-0010 is only suitable for connection to a remote bus, not a local bus.

2.2 Technical data

| Technical data | EL6740-0010 |
|--|---|
| Fieldbus | Interbus |
| Function | Slave |
| Number of fieldbus channels | 1 |
| Data transfer rates | 2 Mbit (default) / 500 kbaud |
| Bus interface | 2 x D sub plug, 9-pin, plug and socket with screening and vibration lock |
| Max. number of bytes, fieldbus | 32 Word input and 32 Word output |
| Hardware diagnostics | Status LEDs |
| Supply voltage for electronics | via the E-bus |
| Current consumption via E-bus | typ. 450 mA |
| Electrical isolation | 500 V (E-bus/field voltage) |
| Configuration | via TwinCAT System Manager |
| Weight | approx. 80 g |
| Permissible ambient temperature range during operation | 0 °C ... + 55 °C (aligned in standard installation position) 0 °C ... + 45 °C (all other installation positions, see notice [▶ 31]) |
| Permissible ambient temperature range during storage | -25 °C ... + 85 °C |
| Permissible relative humidity | 95 %, no condensation |
| Dimensions (W x H x D) | approx. 26 mm x 100 mm x 52 mm (width aligned: 23 mm) |
| Mounting [▶ 26] | 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 | see notice [▶ 31] |
| Approvals / markings* | CE, cULus [▶ 39] , UKCA, EAC, ATEX [▶ 37] , IECEX [▶ 38] |

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

Ex markings

| Standard | Marking |
|----------|------------------------|
| ATEX | II 3 G Ex nA IIC T4 Gc |
| IECEX | Ex nA IIC T4 Gc |

2.3 Interbus - introduction

Introduction to the system

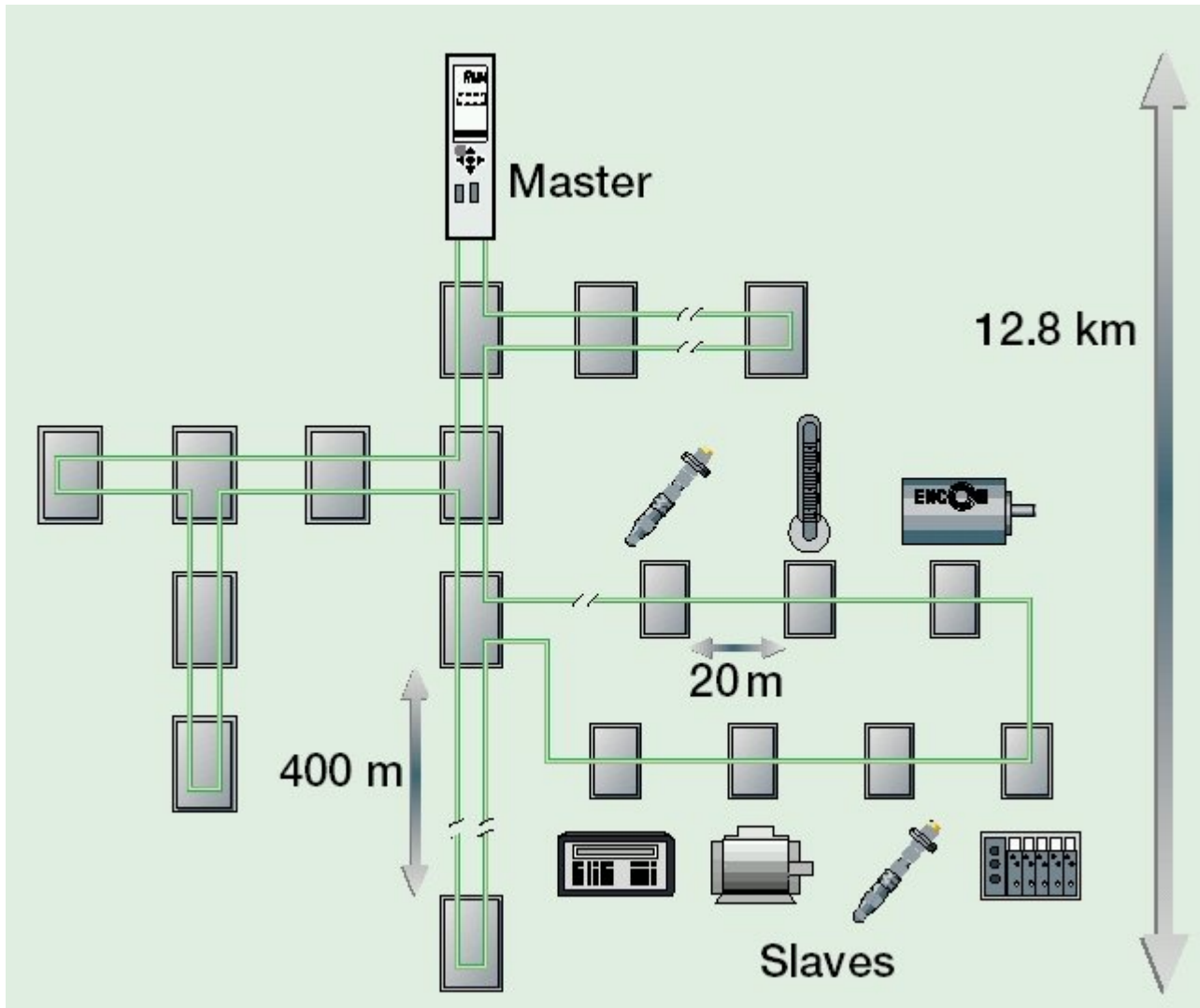


Fig. 5: Interbus data ring

The Interbus system is configured as a data ring with a central master/slave access procedure. It has the structure of a spatial distributed shift register. Each device, with its registers of different length, is part of the shift register ring. The master pushes data through the ring serially. The use of the ring structure offers the option of sending and receiving data simultaneously. The two data directions of the ring are located in a single cable. Each Interbus system device has an ID register (identification register). This register contains information about the module type, the number of input and output registers, and status and error conditions. The Interbus system basically knows two operating modes:

- **The ID cycle,**
which is carried out for the initialization of the Interbus system and on demand. In the ID cycle, the coupling module reads the ID registers of all devices attached to the bus system and uses this information to build the process image.
- **The data cycle;**
this is the actual duty cycle dealing with the data transmission. In the data cycle, the input data are transferred from the registers to the coupling module, and the output data from the coupling module the devices.

System configuration and device types

The Interbus club has a large number of different ID codes. Apart from 6 ID codes, Phoenix Contact has assigned these ID codes to couplers with digital and analog periphery. The manufacturer can therefore not be identified via the ID code. (Detailed commentary in Chapter "ID code and ID length"). The handling of the BK4xx0/BC4xx0 Interbus couplers does not differ from the equipment of other manufacturers.

2.4 Basic function principles

2.4.1 Configuration

The EL6740-0010 acts as an Interbus slave for linking process data with an Interbus strand or reading process data from the Interbus telegrams. The EL6740-0010 first has to be configured via EtherCAT so that the Interbus master can detect it.

Proceed as follows to configure the EL6740-0010:

- Insert the terminal manually into the configuration tree or scan the EtherCAT strand with the EL6740-0010. EL6740-0010 appears without process data, see Fig. *EL6740-0010*. In this state the EL6740-0010 remains in SAFE-OP.

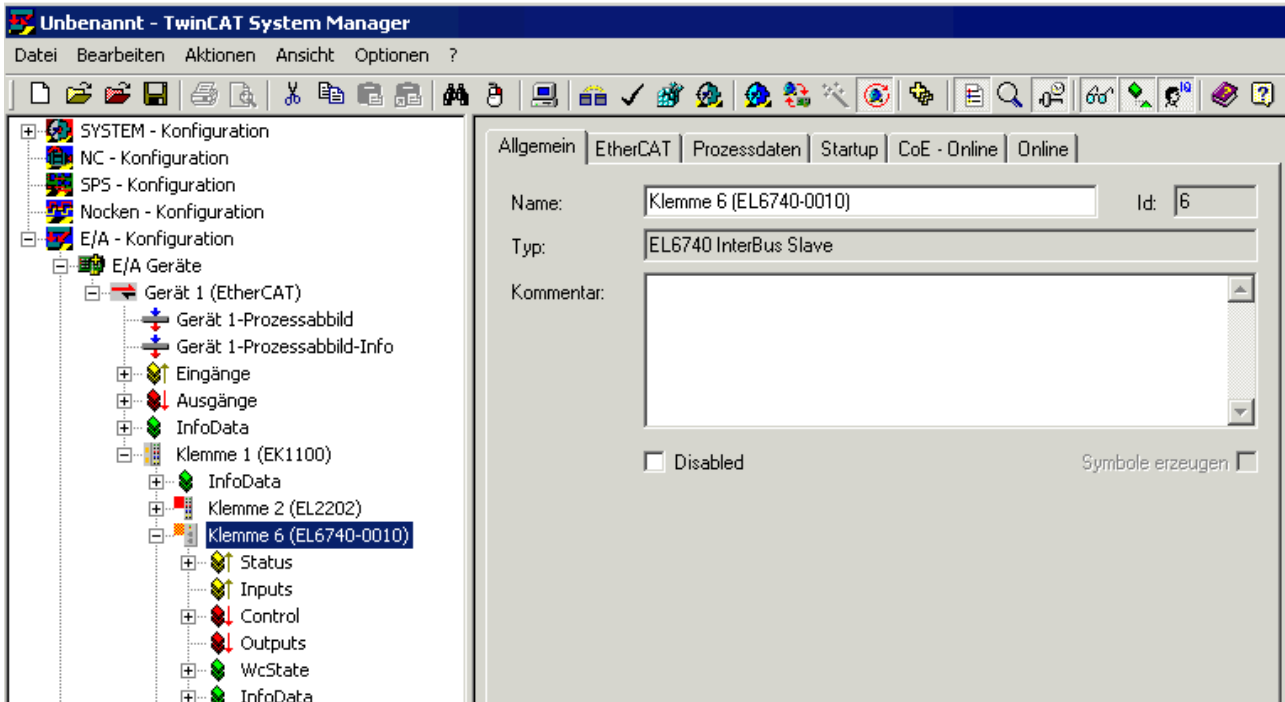


Fig. 6: EL6740-0010

- Manually add the required number of process data for this Interbus slave by right-clicking on inputs and outputs (Fig. *Appending process data + Appending process data, number*).

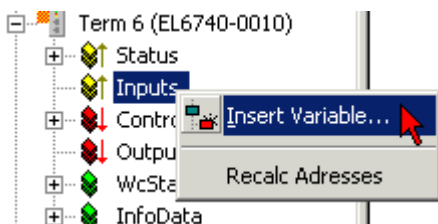


Fig. 7: Adding process data

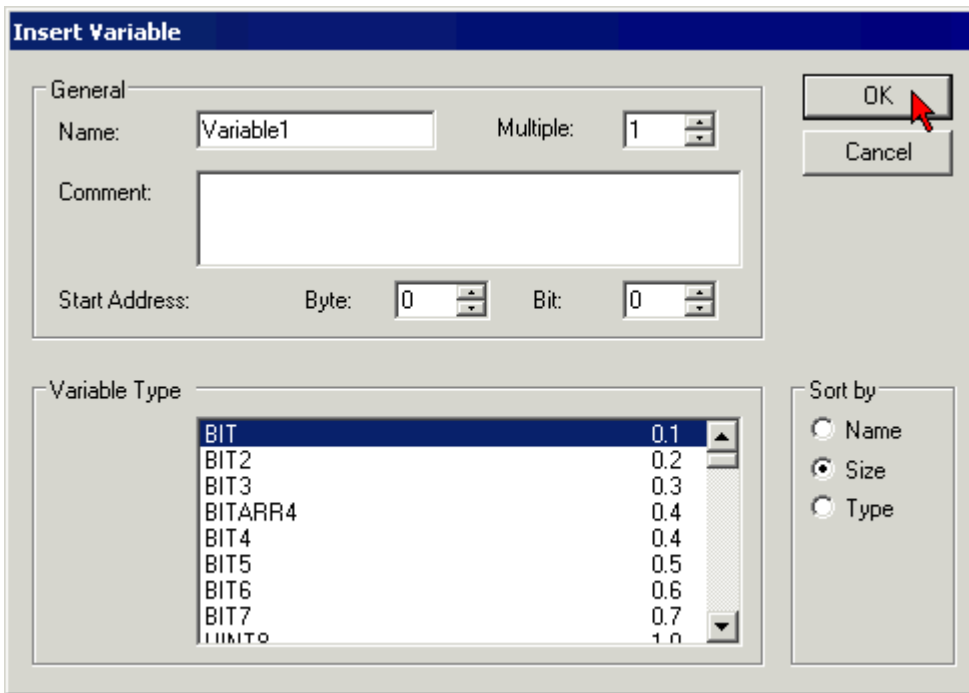


Fig. 8: Adding process data, number

Note regarding process data



The process data of the EL6740-0010 must have the following characteristics:

- Only WORD variables (16 bit) are permitted
 - Up to 32 words input variables and 32 words output variables can be created
-
- After reloading the EtherCAT device the EL6740-0010 is ready for operation as an Interbus slave and reaches the OP state.

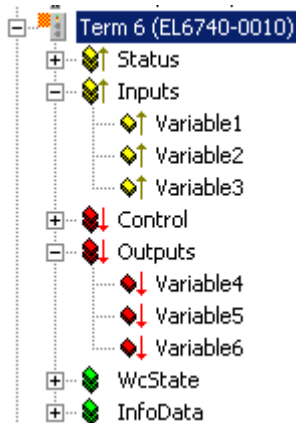


Fig. 9: Configured EL6740-0010

- The Interbus master can now detect it as slave with the configured process data.

EL6740-0010 "INIT" status



SUPI reset with each EL6740-0010 init; an error occurs at the IB, which must be acknowledged there. While the EL6740-0010 is in init state the SUPI remains in reset.

IBS status



As long as the EL6740-0010 is not in OP status, the terminal signals /STATERR on the Interbus. In this case, no valid process data inputs are available for the IBS. /STATERR is also reported if the SyncManager watchdog of the EL6740-0010 drops, e.g. due to a fault in the EtherCAT communication.

i Status word of the EL6740-0010

Error messages with the following values are reported to the status input of the EL6740-0010:

0x2000: SUP1 station Error,
No data have been sent to the Interbus

0x4000: SUP1 invalid data received from Interbus
Invalid data receives from the Interbus

2.4.2 Changing the Interbus baud rate

The CoE index [0xF800:01 \[▶ 118\]](#) enables the baud rate to be switched from 2 Mbit (default) to 500 kbaud. Then press the Reload button (Fig. *Changing the Interbus baud rate + reloading the I/O devices*).

- After double-clicking on index [0xF800:01 \[▶ 118\]](#) (Config mode, CoE tab) enter 0 (FALSE) in the *SetValue* dialog to switch to 500 kbaud.
- Then press the Reload button (Fig. *Changing the Interbus baud rate + reloading the I/O devices*).

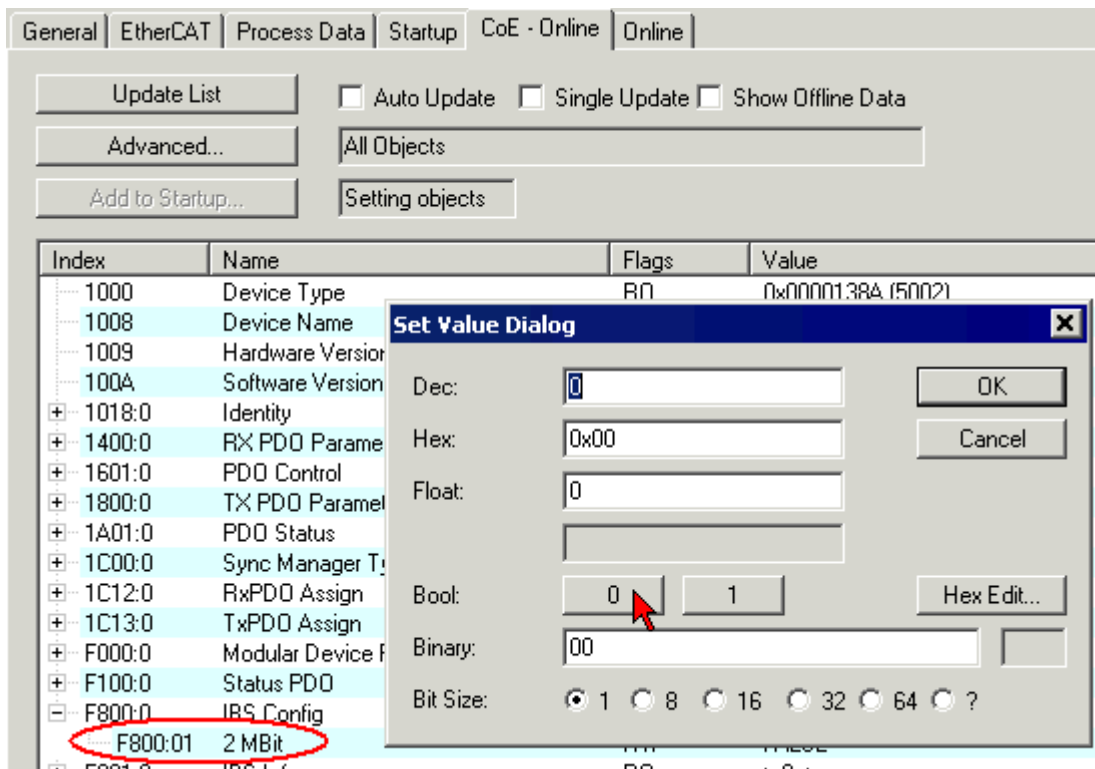


Fig. 10: Changing the Interbus baud rate



Fig. 11: Reloading the I/O devices

Alternatively, you can enter the baud rate in the startup list, so that the correct transfer rate setting is ensured when the I/O devices are loaded (Fig. *Editing the startup entry for the transfer rate*).

- On the "Startup" tab click "New..."
- After double-clicking on index [0xF800:01 \[▶ 118\]](#) in the "Edit CANopen entry" window enter "00" (for FALSE = 500 kbaud) in line "Data (bin)".
- Confirm with "OK" and press the "Reload" button.

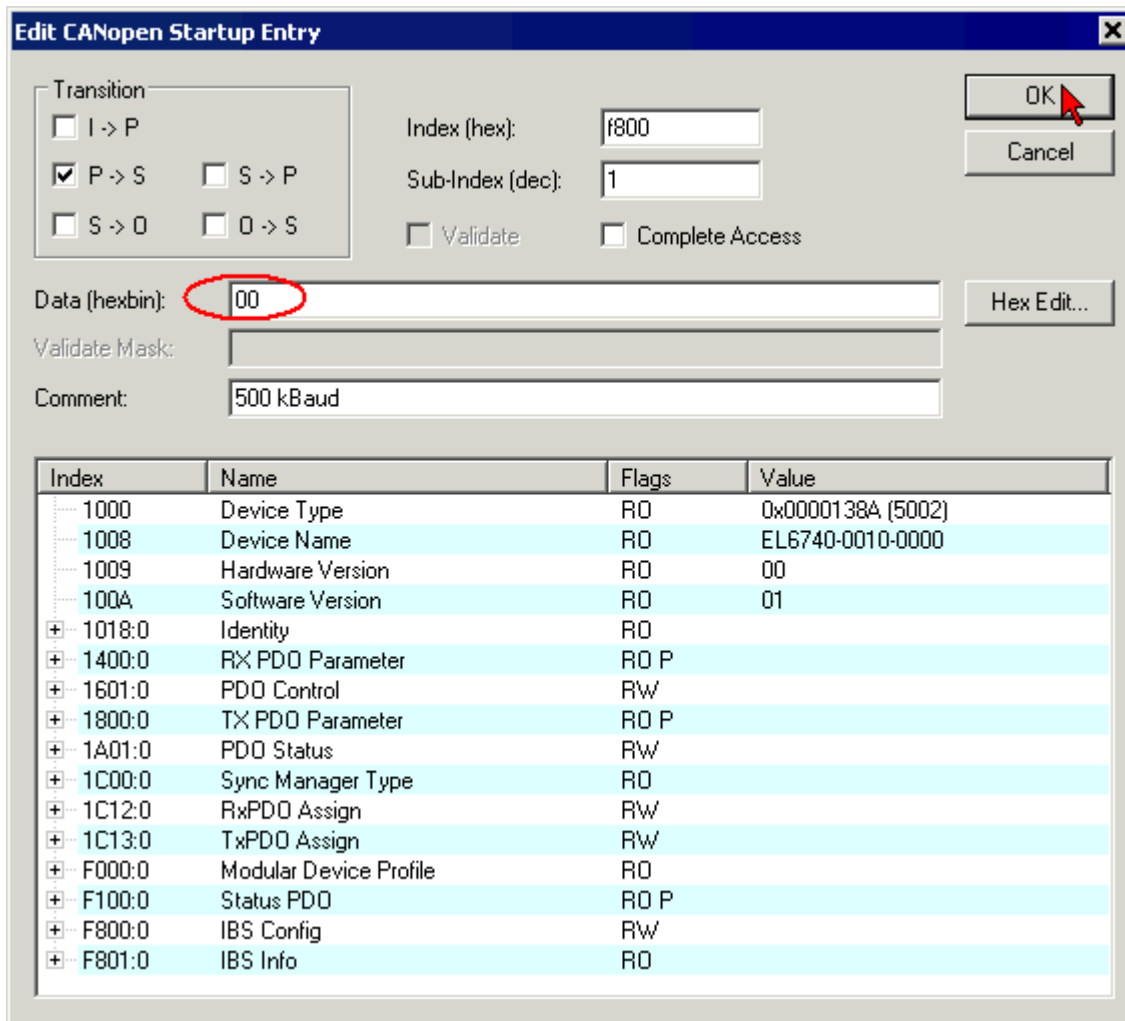


Fig. 12: Editing the startup entry for the data transfer rate

2.4.3 Resetting a device

In the TwinCAT configuration tree each configured fieldbus is allocated to a "device". A "device" may be an actual hardware object such as a PCI_Fieldbus master card (e.g. Lightbus card), or it may be a virtual device, such as an Ethernet or EtherCAT interface.

If errors occur in the subsidiary fieldbus during the configuration phase (errors will be displayed in the logger window of the System Manager, for example), the device may have to be reset after acknowledging the errors. Generally this is achieved through a complete restart of TwinCAT in Run or Config mode, which affects *all* connected devices.

In some cases a *selective* reset of an individual device may be desirable. This is achieved by right-clicking on the device in the configuration tree, see Fig. *Selective reset of a configured device*.



Fig. 13: Selective reset of a configured device

The "logger window" of the System Manager can be displayed for monitoring purposes.

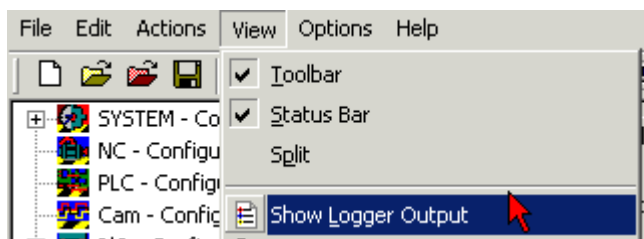


Fig. 14: Logger display

Depending on the cause of the fault the device may have to be reset several times.

2.4.4 Reloading an EtherCAT device

During a reload a reset is performed, the process data structure is reconfigured, the start procedures for all slaves are rearranged based on the current configuration, and the hardware tree associated with the device is started, either in Run or Config (FreeRun) mode as required.

An EtherCAT device (i.e. the EtherCAT strand) can be reloaded in several ways:

- TwinCAT restart in run mode
Effect: *all* devices in the configuration are restarted (EtherCAT master, Profibus master, Ethernet interfaces, ...)
- TwinCAT restart in config mode
Effect: *all* devices in the configuration are restarted (EtherCAT master, Profibus master, Ethernet interfaces, ...)
- Reload of all configured devices via icon or F4 shortcut
Effect: *all* devices in the configuration are restarted (EtherCAT master, Profibus master, Ethernet interfaces, ...)



Fig. 15: Reload of all devices

- Selective reload of individual devices through right-click on the device in the System Manager;
Effect: *only* the selected device is restarted

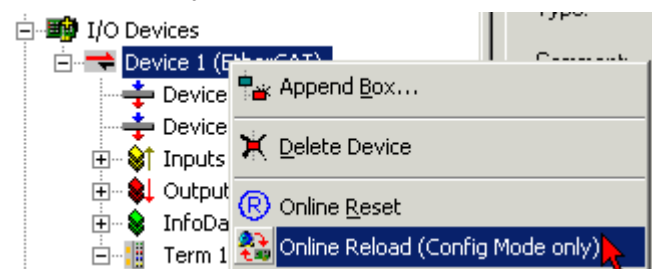


Fig. 16: Selective reload of the EtherCAT device

2.4.5 Operation of EtherCAT and Interbus master on a PC

If the EtherCAT master and the Interbus master are operated on a PC via TwinCAT for test purposes, a collision of the startup timers of the two devices will occur when TwinCAT is restarted. For this atypical case the following start sequence should be followed:

- Configure the EL6740-0010 in the EtherCAT device based on the above specifications, TwinCAT in Config mode/Free-Run.
- Manual selective reload of the EtherCAT device.
- Activate FreeRun. The EL6740-0010 is now in OP state.

- Reset the Interbus master through a selective reset until "Reset started, Reset completed successfully" appears in the logger window.
Several attempts may be required. In order to facilitate the reset it may be helpful link any Interbus box with the Interbus master, followed by resetting and deleting of the Interbus box.
- Initiate a "box scan" on the Interbus master, after which the Interbus slaves will be detected.
- Restart TwinCAT in Config mode/FreeRun. Data transmission can now be tested.
- Use the configuration in Run mode.

To operate the EtherCAT master and the Interbus master on the same PC permanently, it is advisable to use software-based process data transfer via TwinCAT or ADS, rather than a hardware link via the Interbus.

An example configuration with an Interbus master card "IBS PCI SC/I-T" appears in the System Manager as shown in Fig. *Selective reload of the EtherCAT device.*

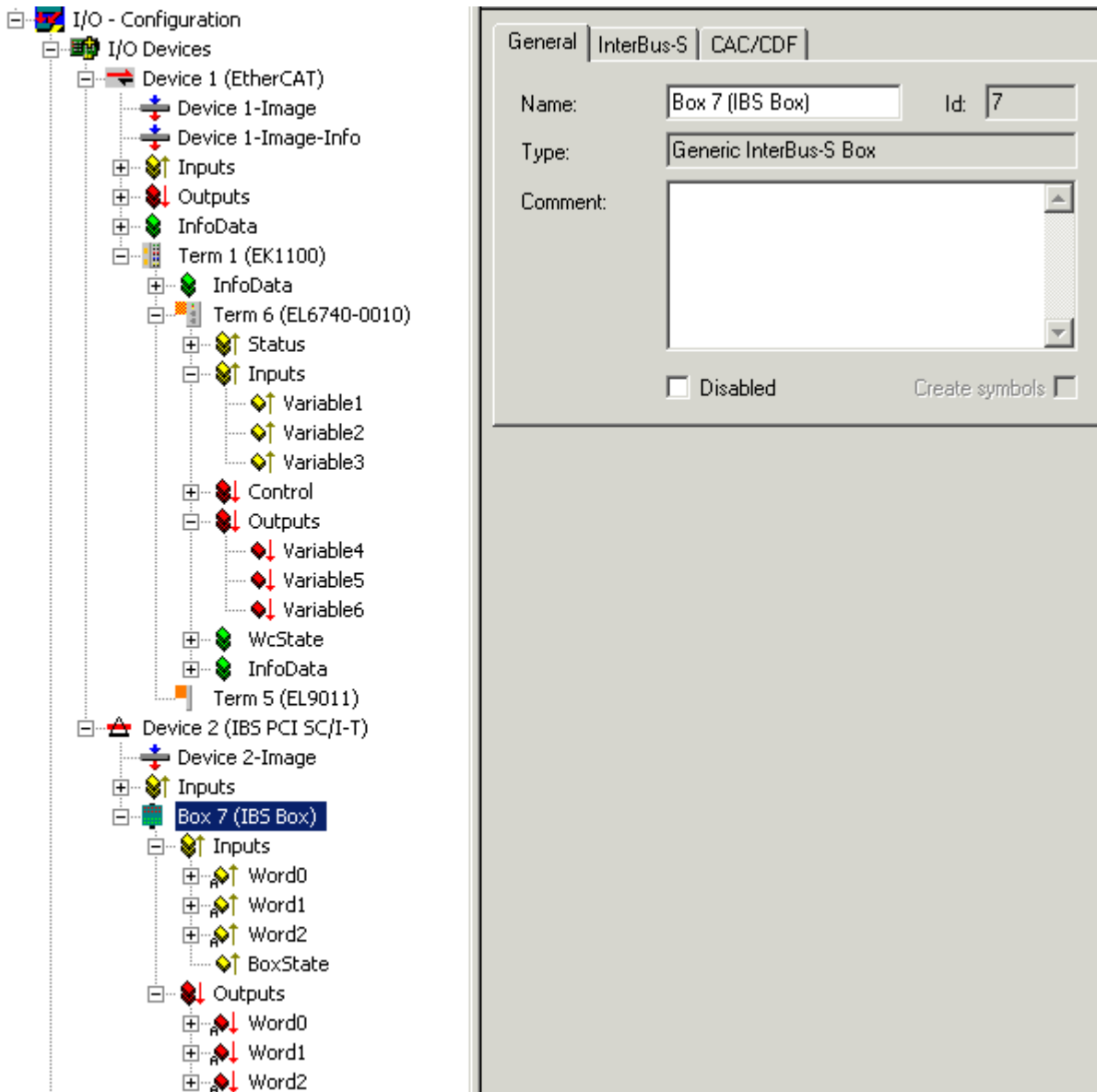


Fig. 17: Example configuration of EtherCAT and Interbus master on the same PC

Data that are output by the PLC via the output process data of the EL6740-0010 "Variable 4..6" are transported via the Interbus and appear as "Word0..2" in the Interbus device.

2.4.6 Saving the Interbus configuration

Saving the Interbus configuration in the EL6740-0010 (from FW 02)

The EL6740-0010 enables the current Interbus configuration (baud rate and ID in CoE object [0xF800 \[► 118\]](#)) to be stored in order to prevent inadvertent changes in the relationship between the slave and the Interbus master. To this end the value 0x65766173 has to be written after object [0x1010:01 \[► 115\]](#) in OP or SAFE-OP state via acyclic communication. Enter 0x64616F6C into object [0x1011:01 \[► 115\]](#) to enable a reset.

An attempt to start an EtherCAT configuration with different Interbus parameters aborts the corresponding EL6740-0010 with abort code 0x0026.

For acyclic communication with the CoE-directory of EL6740-0010 CoE e.g. from the PLC blocks such as FB_EcCoESdoWrite are available in TcEtherCAT.lib.

3 Mounting and wiring

3.1 Instructions for ESD protection

NOTICE

Destruction of the devices by electrostatic discharge possible!

The devices contain components at risk from electrostatic discharge caused by improper handling.

- Please ensure you are electrostatically discharged and avoid touching the contacts of the device directly.
- Avoid contact with highly insulating materials (synthetic fibers, plastic film etc.).
- Surroundings (working place, packaging and personnel) should be grounded probably, when handling with the devices.
- Each assembly must be terminated at the right hand end with an [EL9011](#) or [EL9012](#) bus end cap, to ensure the protection class and ESD protection.

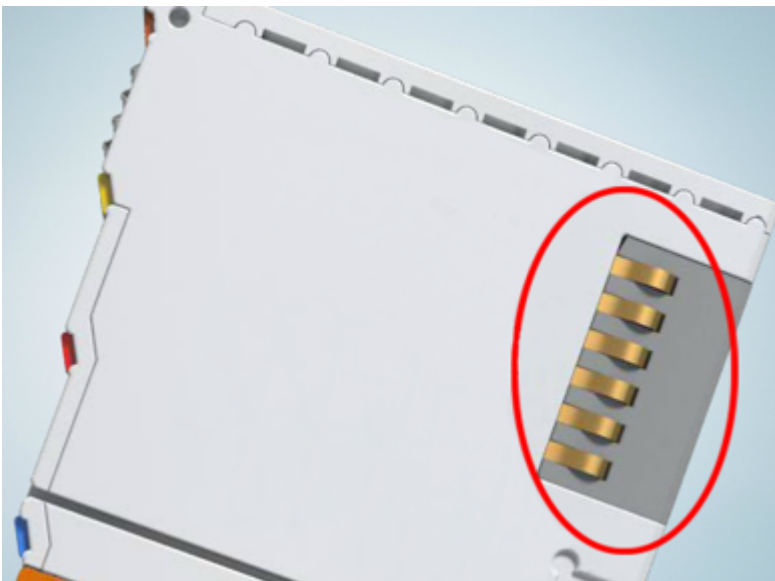


Fig. 18: Spring contacts of the Beckhoff I/O components

3.2 Mounting and demounting - 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).

● Fixing of mounting rails

i 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).

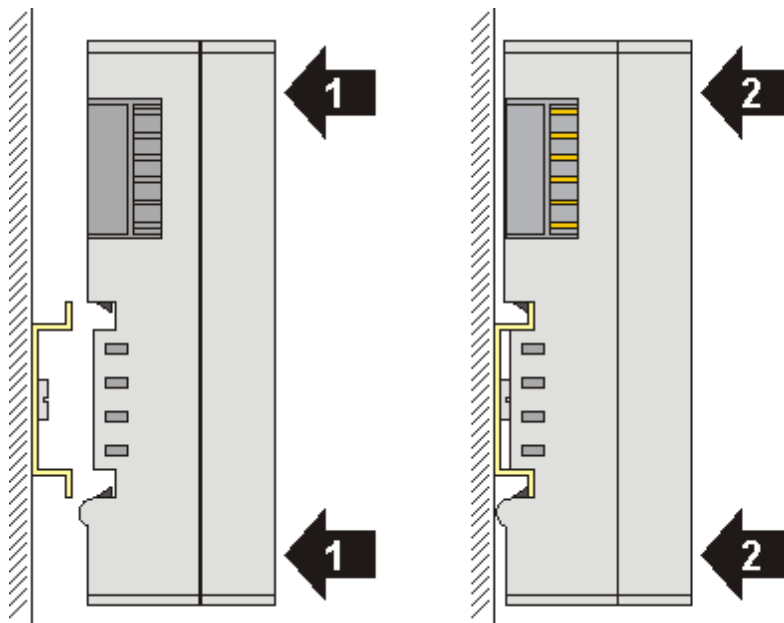
⚠ WARNING

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.

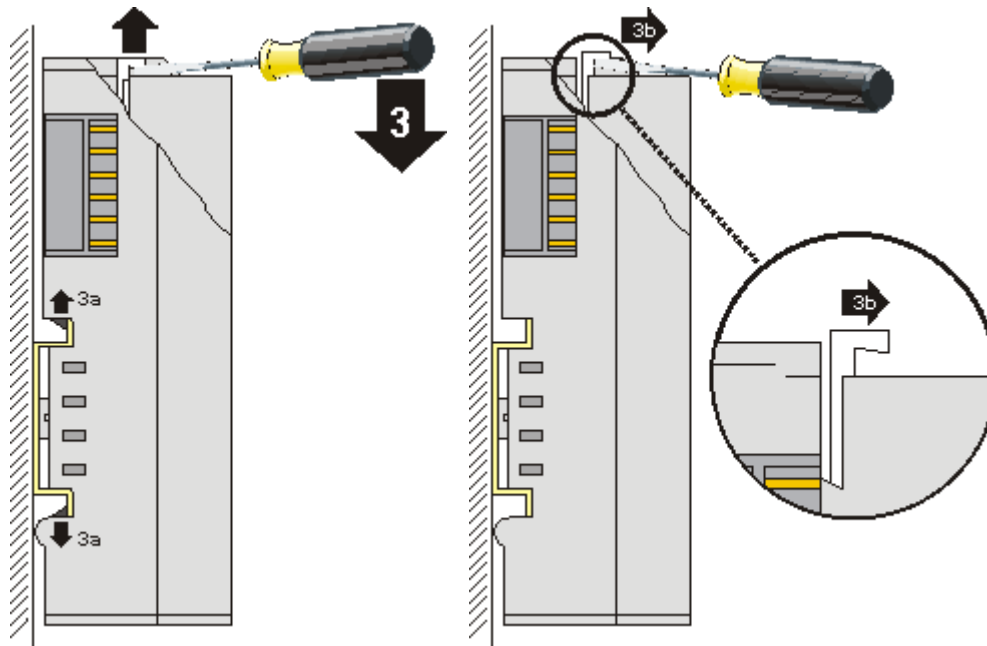


and press (1) the terminal module against the mounting rail until it latches in place on the mounting rail (2).

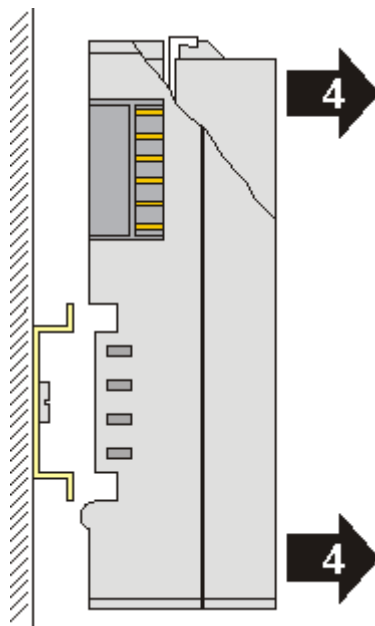
- Attach the cables.

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.



3.3 Mounting and demounting - top front unlocking

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

● Fixing of mounting rails

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

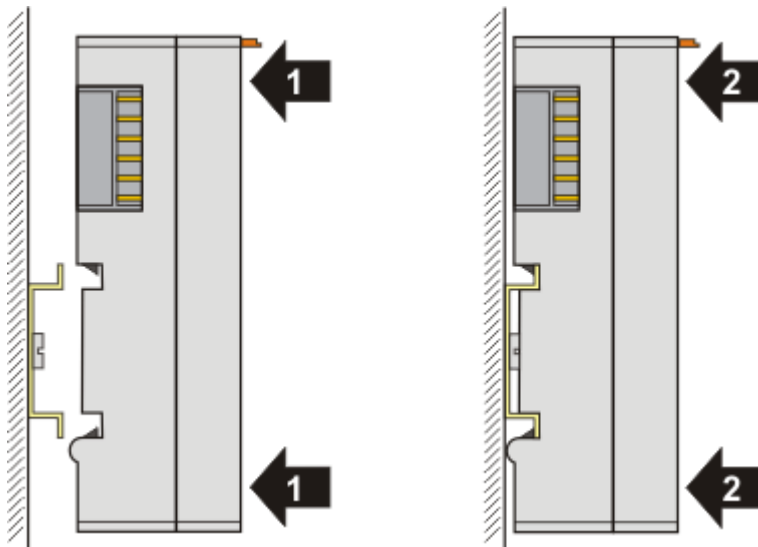
⚠ WARNING

Risk of electric shock and damage of device!

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

Mounting

- Fit the mounting rail to the planned assembly location.

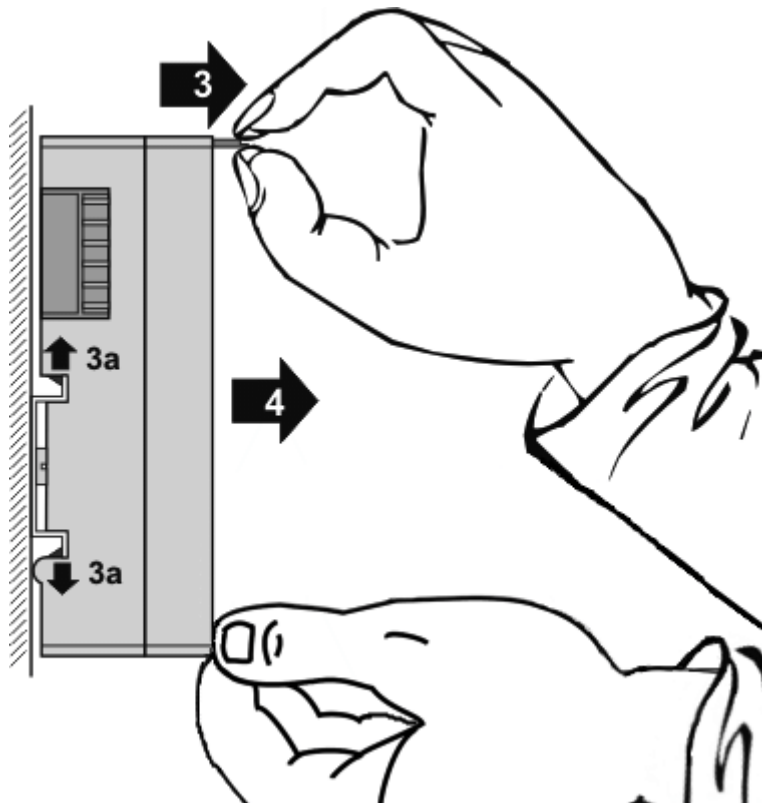


and press (1) the terminal module against the mounting rail until it latches in place on the mounting rail (2).

- Attach the cables.

Demounting

- Remove all the cables.
- Lever the unlatching hook back with thumb and forefinger (3). An internal mechanism pulls the two latching lugs (3a) from the top hat rail back into the terminal module.



- Pull (4) the terminal module away from the mounting surface.
Avoid canting of the module; you should stabilize the module with the other hand, if required.

3.4 Disposal



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

3.5 Positioning of passive Terminals

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

EtherCAT Terminals (ELxxxx / ESxxxx), which do not take an active part in data transfer within the bus terminal block are so called passive terminals. The passive terminals have no current consumption out of the E-Bus.

To ensure an optimal data transfer, you must not directly string together more than two passive terminals!

Examples for positioning of passive terminals (highlighted)

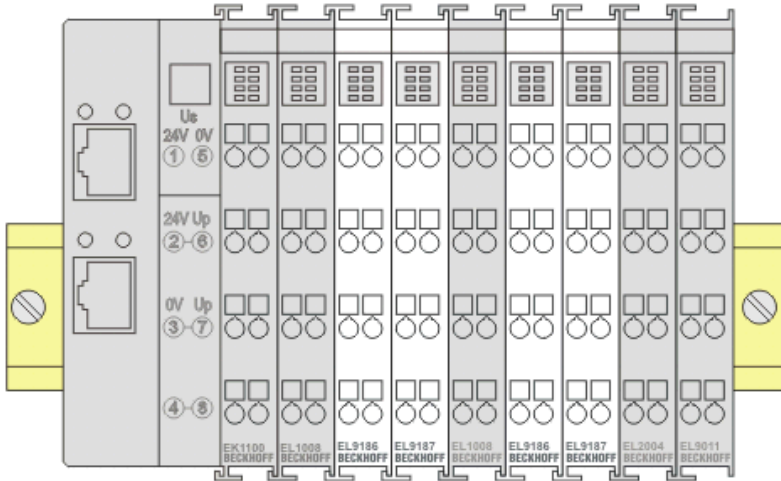


Fig. 19: Correct positioning

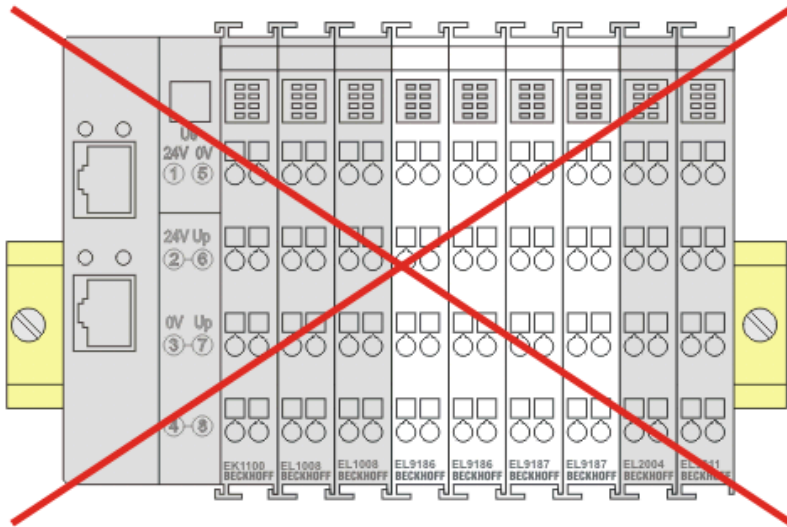


Fig. 20: Incorrect positioning

3.6 Installation positions

NOTICE

Constraints regarding installation position and operating temperature range

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

Optimum installation position (standard)

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

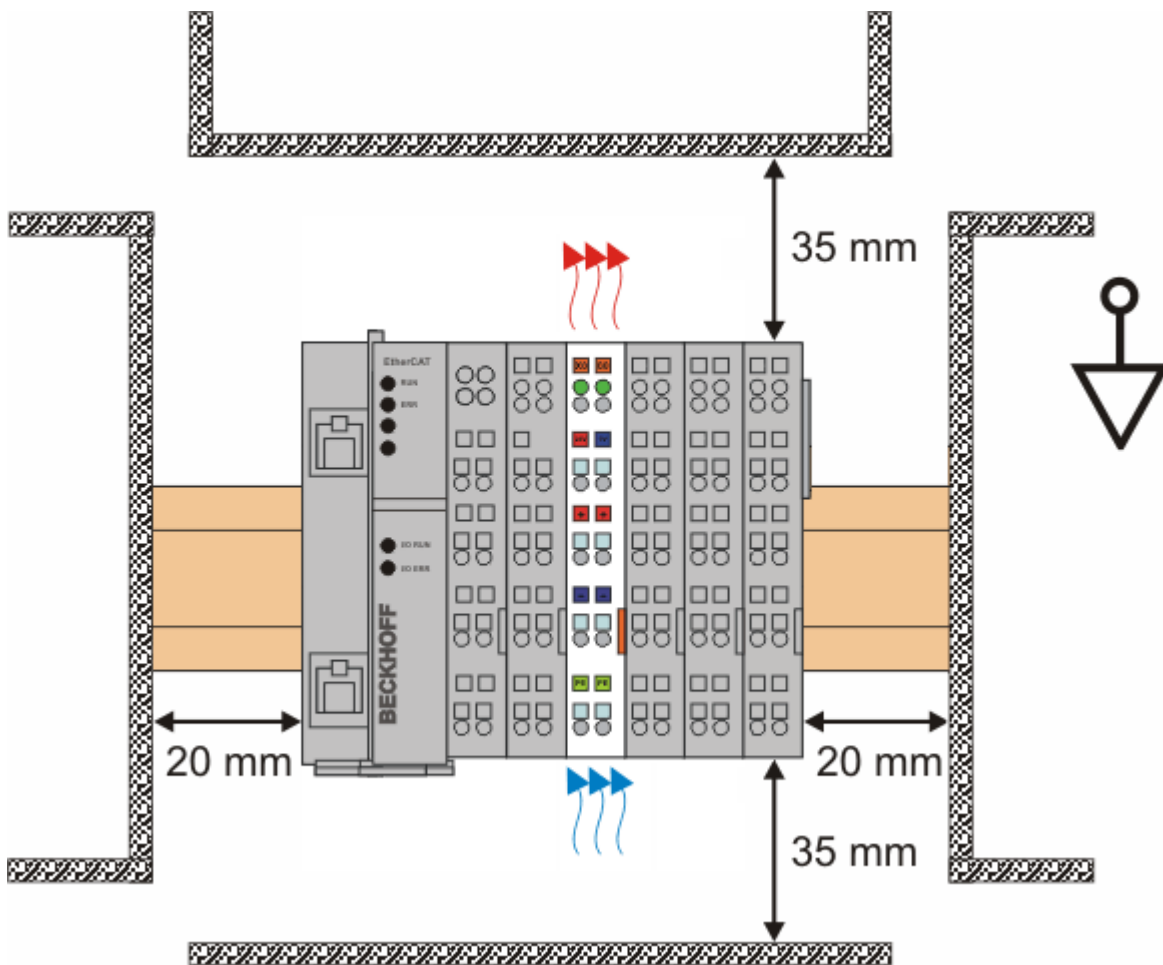


Fig. 21: Recommended distances for standard installation position

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.

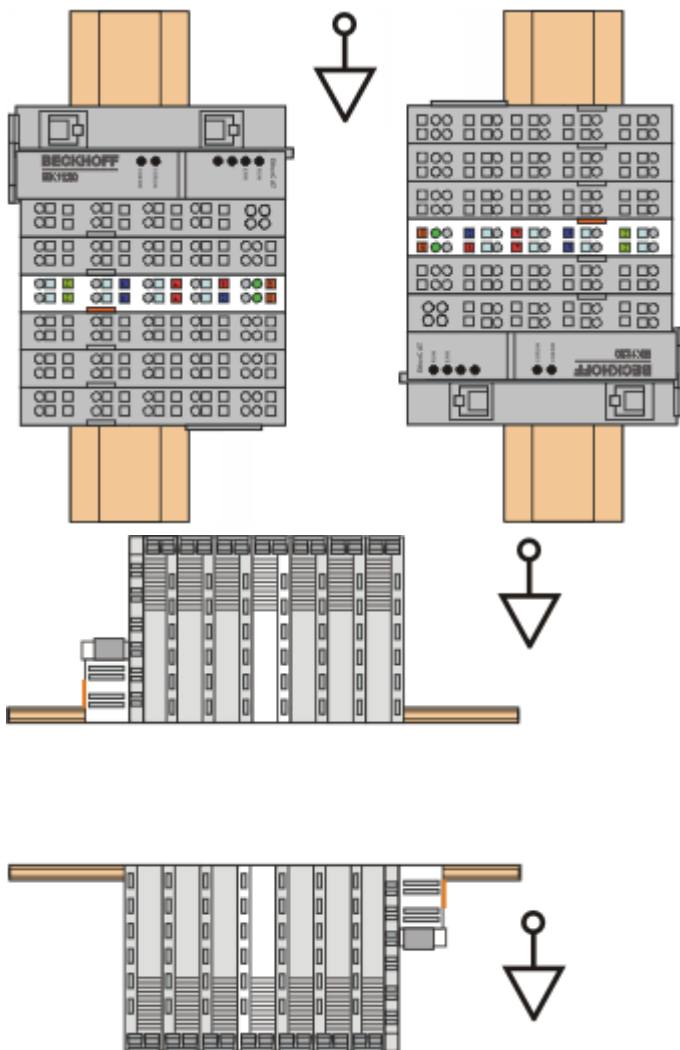


Fig. 22: Other installation positions

3.7 Note - Power supply

⚠ WARNING

Power supply from SELV/PELV power supply unit!

SELV/PELV circuits (Safety Extra Low Voltage, Protective Extra Low Voltage) according to IEC 61010-2-201 must be used to supply this device.

Notes:

- SELV/PELV circuits may give rise to further requirements from standards such as IEC 60204-1 et al, for example with regard to cable spacing and insulation.
- A SELV (Safety Extra Low Voltage) supply provides safe electrical isolation and limitation of the voltage without a connection to the protective conductor, a PELV (Protective Extra Low Voltage) supply also requires a safe connection to the protective conductor.

3.8 Wiring the Bus System

The INTERBUS differentiates between remote bus, peripheral bus and remote installation bus. The INTERBUS Bus Coupler is equipped with the remote bus interface. The INTERBUS Bus Coupler has an incoming and outgoing interface based on D-SUB plug and coupling.

INTERBUS topology

INTERBUS is a ring bus system. The INTERBUS line includes the return line. The cabling therefore appears to be a single line.

Bus length

| Baud rate | Bus length |
|------------|------------------------------------|
| 500 kbit/s | 12.8 km (400 m between 2 stations) |

INTERBUS cable

The use of pair-wise twisted, screened cables is recommended for the INTERBUS wiring (3 x twisted pair with screening).

ZB4200 INTERBUS cable

Beckhoff offers an INTERBUS cable with the order number ZB4200.

Cable preparation

Cable colours for using the Beckhoff INTERBUS cable at Bus Terminal and Fieldbus Box:

| Pin BK40x0/BC4000 | | Function | ZB4200 cable color |
|-----------------------------------|-------------------------------------|----------|--------------------|
| D sub plug outgoing remote bus | D sub socket incoming remote bus | | |
| 6 | 6 | /DO | green |
| 1 | 1 | DO | yellow |
| 7 | 7 | /DI | pink |
| 2 | 2 | DI | grey |
| 3 | 3 | COM | brown |
| 5 | - | jumper | - |
| 9 | - | jumper | - |

INTERBUS D sub coupling ZB4100 and plug ZB4101

For the connection with the INTERBUS cable and the Bus Coupler, we recommend the use of D sub plug and coupling.

ZB4100 incoming remote bus, coupling
 ZB4101 outgoing remote bus, plug

3.9 Topology

- Ring bus system with active coupling
- 400 m between two fieldbus devices
- 12.8 km total length
- max. number of devices 256

All Beckhoff Bus Couplers BK4xxx and BC4000 are remote bus devices and can only be operated as such on the INTERBUS.

Coupling modules

Examples for INTERBUS master controls:

PC cards:

Hilscher CIF 30-IBM

Hilscher CIF 40-IBM

Hilscher CIF 50-IBM

Hilscher CIF 60-IBM

Hilscher CIF 104-IBM

Phoenix IBS PC ISA SC/I-T

Phoenix IBS PC ISA SC/RI/RT-LK

Phoenix IBS PC PCI SC/I-T

Phoenix IBS PC PCI SC/RI/RT-LK

Modules:

Phoenix IBS S5 DCB/I-T

Phoenix IBS S5 DSC/I-T

Phoenix 100 CB-T

3.10 LEDs and connection

LEDs

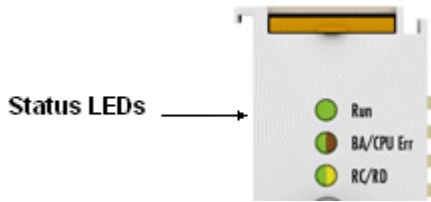


Fig. 23: LEDs

| LED | Colours | Meaning | |
|--------------|---------|---|---|
| RUN 1 | green | These LEDs indicate the terminal's operating state: | |
| | | off | State of the EtherCAT State Machine [▶ 40] : INIT = initialization of the terminal |
| | | flashing uniformly | State of the EtherCAT State Machine: PREOP = function for mailbox communication and different standard-settings set |
| | | flashing slowly | State of the EtherCAT State Machine: SAFEOP = verification of the Sync Manager [▶ 99] channels |
| | | on | State of the EtherCAT State Machine: OP = normal operating state; mailbox and process data communication is possible |
| | | flashing rapidly | State of the EtherCAT State Machine: BOOTSTRAP = function for terminal firmware updates |
| RC/RD | red | RD signal: outgoing Interbus switched off - if not: outgoing Interbus switched on | |
| RC/RD | green | RC signal: incoming Interbus connected - if not: incoming Interbus not connected | |
| RC/RD | orange | RD signal (red) and RC signal (green) at the same time | |
| BA/CPU error | red | CPU signal error | |
| BA/CPU error | green | BA signal: Interbus ASIC bus active | |

Pin assignment D sub9 socket/plug connector (plan view)

● Pin assignment before hardware version 01

- i** Pins 4 and 8 were not assigned before hardware version 01!
- The lower diagram shows the pin assignment from hardware version 01.
- VCC / GND: E-bus supply voltage
- VCC_ISO / GND_ISO: Electrically isolated auxiliary voltage (max. load 60 mA)

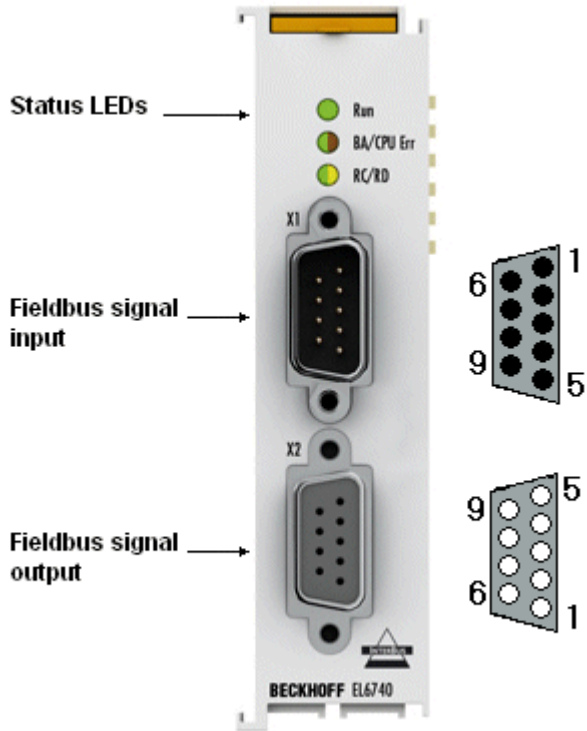


Fig. 24: Connection

Fieldbus signal input

| Pin | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------|----|----|---------|-----|---------|-----|-----|-----|----|
| Signal | DO | DI | GND_ISO | GND | VCC_ISO | /DO | /DI | VCC | NC |

Fieldbus signal output

| Pin | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|--------|----|----|-----|-----|-----|-----|-----|-----|------|
| Signal | DO | DI | GND | GND | VCC | /DO | /DI | VCC | RBST |

3.11 Explosion protection

3.11.1 ATEX - Special conditions (standard temperature range)

⚠ WARNING

Observe the special conditions for the intended use of Beckhoff fieldbus components with standard temperature range in potentially explosive areas (directive 2014/34/EU)!

- The certified components are to be installed in a suitable housing that guarantees a protection class of at least IP54 in accordance with EN 60079-15! The environmental conditions during use are thereby to be taken into account!
- For dust (only the fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9): The equipment shall be installed in a suitable enclosure providing a degree of protection of IP54 according to EN 60079-31 for group IIIA or IIIB and IP6X for group IIIC, taking into account the environmental conditions under which the equipment is used!
- If the temperatures during rated operation are higher than 70°C at the feed-in points of cables, lines or pipes, or higher than 80°C at the wire branching points, then cables must be selected whose temperature data correspond to the actual measured temperature values!
- Observe the permissible ambient temperature range of 0 to 55°C for the use of Beckhoff fieldbus components standard temperature range in potentially explosive areas!
- Measures must be taken to protect against the rated operating voltage being exceeded by more than 40% due to short-term interference voltages!
- The individual terminals may only be unplugged or removed from the Bus Terminal system if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The connections of the certified components may only be connected or disconnected if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The fuses of the KL92xx/EL92xx power feed terminals may only be exchanged if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- Address selectors and ID switches may only be adjusted if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!

Standards

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

- EN 60079-0:2012+A11:2013
- EN 60079-15:2010
- EN 60079-31:2013 (only for certificate no. KEMA 10ATEX0075 X Issue 9)

Marking

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



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

II 3D KEMA 10ATEX0075 X Ex tc IIIC T135°C Dc Ta: 0 ... +55°C
(only for fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9)

or



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

II 3D KEMA 10ATEX0075 X Ex tc IIIC T135°C Dc Ta: 0 ... +55°C
(only for fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9)

3.11.2 IECEx - Special conditions

WARNING

Observe the special conditions for the intended use of Beckhoff fieldbus components in potentially explosive areas!

- For gas: The equipment shall be installed in a suitable enclosure providing a degree of protection of IP54 according to IEC 60079-15, taking into account the environmental conditions under which the equipment is used!
- For dust (only the fieldbus components of certificate no. IECEx DEK 16.0078X Issue 3): The equipment shall be installed in a suitable enclosure providing a degree of protection of IP54 according to EN 60079-31 for group IIIA or IIIB and IP6X for group IIIC, taking into account the environmental conditions under which the equipment is used!
- The equipment shall only be used in an area of at least pollution degree 2, as defined in IEC 60664-1!
- Provisions shall be made to prevent the rated voltage from being exceeded by transient disturbances of more than 119 V!
- If the temperatures during rated operation are higher than 70°C at the feed-in points of cables, lines or pipes, or higher than 80°C at the wire branching points, then cables must be selected whose temperature data correspond to the actual measured temperature values!
- Observe the permissible ambient temperature range for the use of Beckhoff fieldbus components in potentially explosive areas!
- The individual terminals may only be unplugged or removed from the Bus Terminal system if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The connections of the certified components may only be connected or disconnected if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- Address selectors and ID switches may only be adjusted if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The front hatch of certified units may only be opened if the supply voltage has been switched off or a non-explosive atmosphere is ensured!

Standards

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

- EN 60079-0:2011
- EN 60079-15:2010
- EN 60079-31:2013 (only for certificate no. IECEx DEK 16.0078X Issue 3)


Marking

Beckhoff fieldbus components that are certified in accordance with IECEx for use in areas subject to an explosion hazard bear the following markings:




| | |
|--|-----------------------------|
| Marking for fieldbus components of certificate no. IECEx DEK 16.0078X Issue 3: | IECEx DEK 16.0078 X |
| | Ex nA IIC T4 Gc |
| | Ex tc IIIC T135°C Dc |

| | |
|--|----------------------------|
| Marking for fieldbus components of certificates with later issues: | IECEx DEK 16.0078 X |
| | Ex nA IIC T4 Gc |

3.11.3 Continulative documentation for ATEX and IECEx

| NOTICE | |
|---|---|
|  | <p>Continulative documentation about explosion protection according to ATEX and IECEx</p> <p>Pay also attention to the continuative documentation</p> <p>Ex. Protection for Terminal Systems</p> <p>Notes on the use of the Beckhoff terminal systems in hazardous areas according to ATEX and IECEx,</p> <p>that is available for download within the download area of your product on the Beckhoff homepage www.beckhoff.com!</p> |

3.12 UL notice

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

Basic principles

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



4 Basics communication

4.1 EtherCAT basics

Please refer to the [EtherCAT System Documentation](#) for the EtherCAT fieldbus basics.

4.2 EtherCAT State Machine

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

A distinction is made between the following states:

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

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

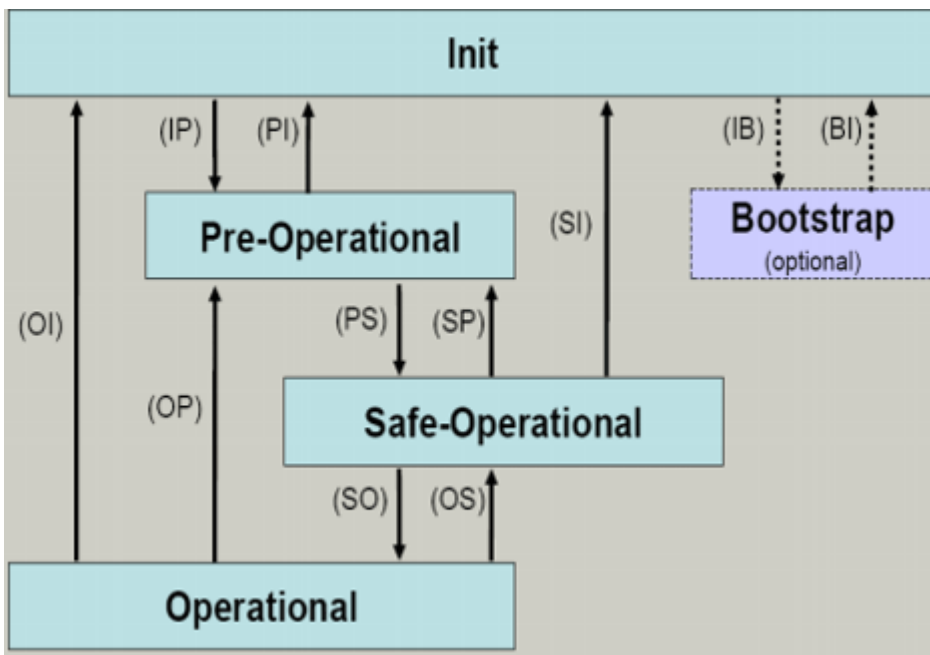


Fig. 25: States of the EtherCAT State Machine

Init

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

Pre-Operational (Pre-Op)

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

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

Safe-Operational (Safe-Op)

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

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

● Outputs in SAFEOP state

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

Operational (Op)

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

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

Boot

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

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

4.3 General notes for setting the watchdog

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

The EtherCAT slave controller (ESC) features two watchdogs:

- SM watchdog (default: 100 ms)
- PDI watchdog (default: 100 ms)

Their times are individually parameterized in TwinCAT as follows:

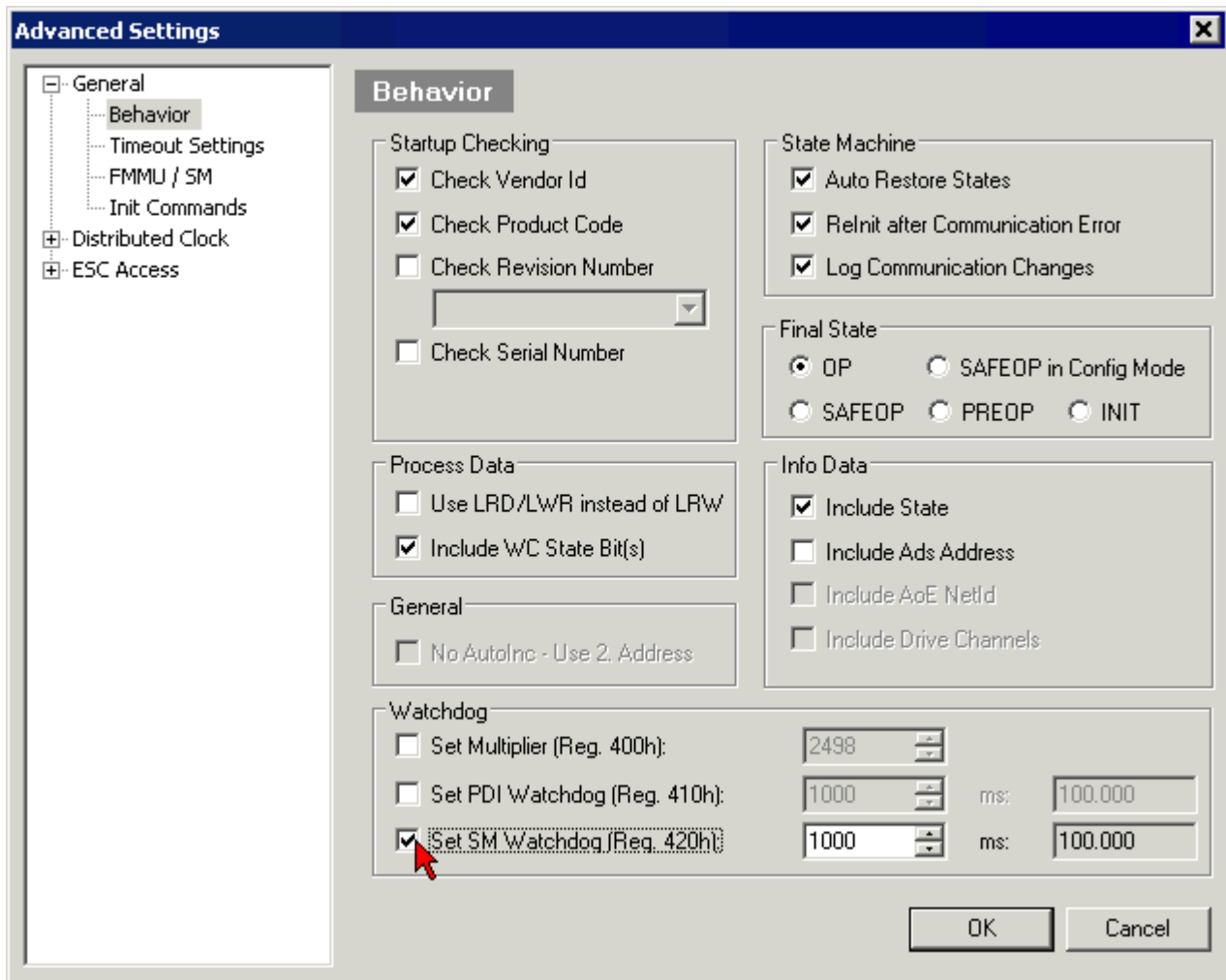


Fig. 26: eEtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

- the Multiplier Register 400h (hexadecimal, i. e. x0400) is valid for both watchdogs.
- each watchdog has its own timer setting 410h or 420h, which together with the Multiplier results in a resulting time.
- important: the Multiplier/Timer setting is only loaded into the slave at EtherCAT startup if the checkbox in front of it is activated.
- if it is not checked, nothing is downloaded and the setting located in the ESC remains unchanged.
- the downloaded values can be seen in the ESC registers x0400/0410/0420: ESC Access -> Memory

SM watchdog (SyncManager Watchdog)

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

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

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

PDI watchdog (Process Data Watchdog)

If there is no PDI communication with the EtherCAT slave controller (ESC) for longer than the set and activated PDI watchdog time, this watchdog is triggered.

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

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

Calculation

Watchdog time = $[1/25 \text{ MHz} * (\text{Watchdog multiplier} + 2)] * \text{PDI/SM watchdog}$

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

The value in Multiplier + 2 corresponds to the number of 40ns base ticks representing one watchdog tick.

⚠ CAUTION

Undefined state possible!

The function for switching off the SM watchdog via SM watchdog = 0 is only implemented in terminals from version -0016. In previous versions this operating mode should not be used.

⚠ CAUTION

Damage of devices and undefined state possible!

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

4.4 CoE Interface

General description

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

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

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

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

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

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

The relevant ranges for EtherCAT fieldbus users are:

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

Other important ranges are:

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

● Availability



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

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

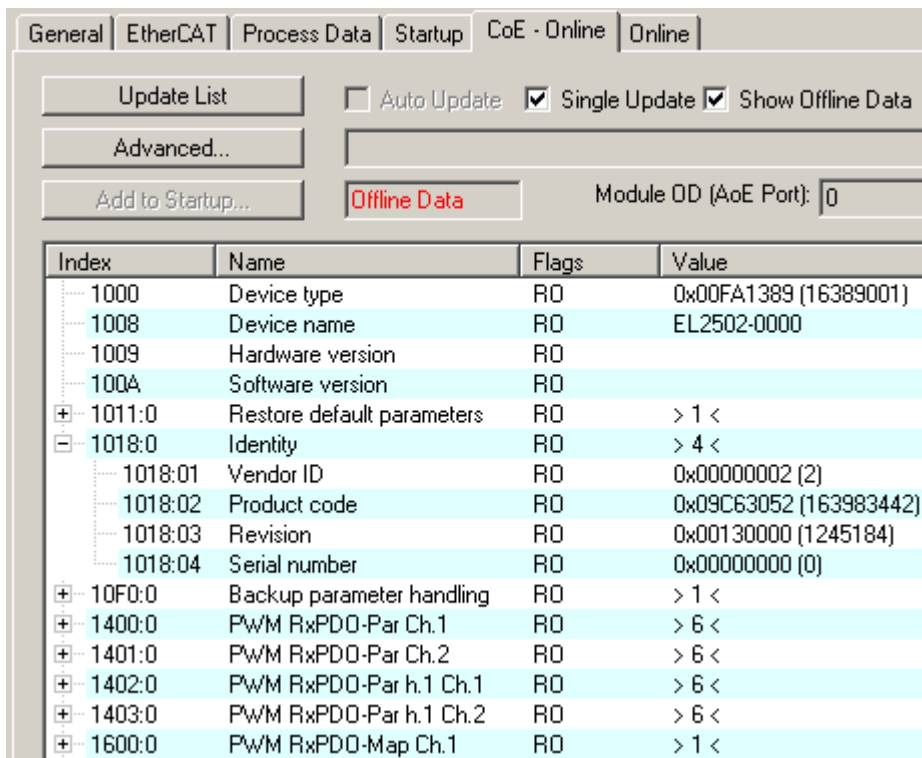


Fig. 27: "CoE Online" tab

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

NOTICE

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

When using/manipulating the CoE parameters observe the general CoE notes in chapter "[CoE interface](#)" of the EtherCAT system documentation:

- Keep a startup list if components have to be replaced,
- Distinction between online/offline dictionary,
- Existence of current XML description (download from the [Beckhoff website](#)),
- "CoE-Reload" for resetting the changes
- Program access during operation via PLC (see [TwinCAT3 | PLC Library: Tc2 EtherCAT](#) and [Example program R/W CoE](#))

Data management and function “NoCoeStorage”

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

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

i Data management

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

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

Please refer to the technical data in this documentation as to whether this applies to the respective device.

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

i Startup list

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

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

Recommended approach for manual modification of CoE parameters

- Make the required change in the System Manager
The values are stored locally in the EtherCAT slave
- If the value is to be stored permanently, enter it in the Startup list.
The order of the Startup entries is usually irrelevant.

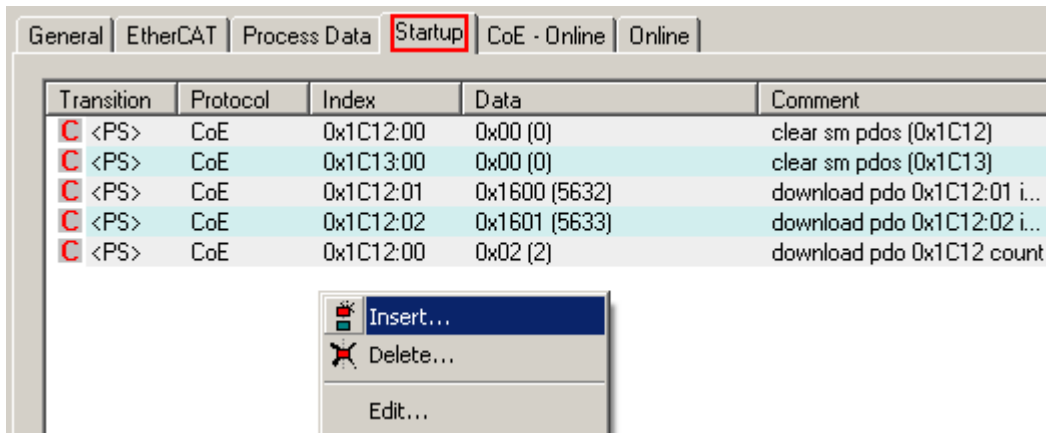


Fig. 28: Startup list in the TwinCAT System Manager

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

Online/offline list

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

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

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

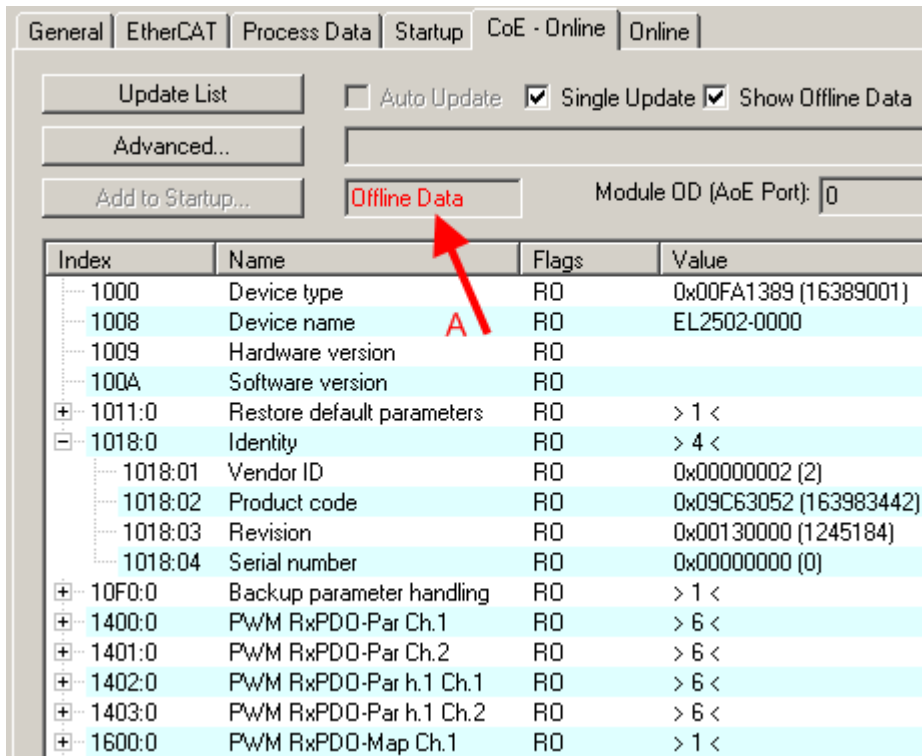


Fig. 29: Offline list

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

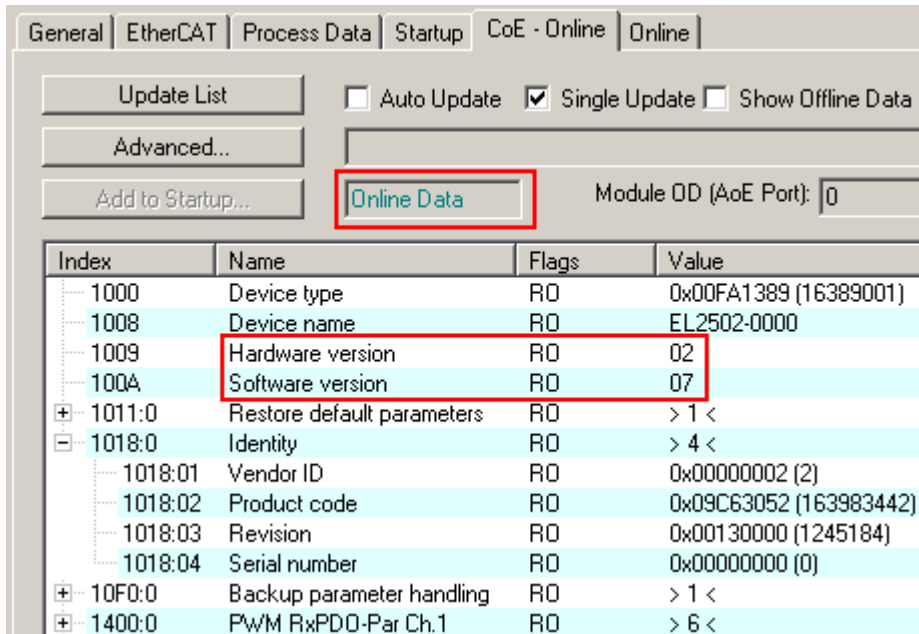


Fig. 30: Online list

Channel-based order

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

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

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

This is generally written as 0x80n0.

Detailed information on the CoE interface can be found in the [EtherCAT system documentation](#) on the Beckhoff website.

5 Commissioning

5.1 TwinCAT Quick Start

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

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

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

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

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

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

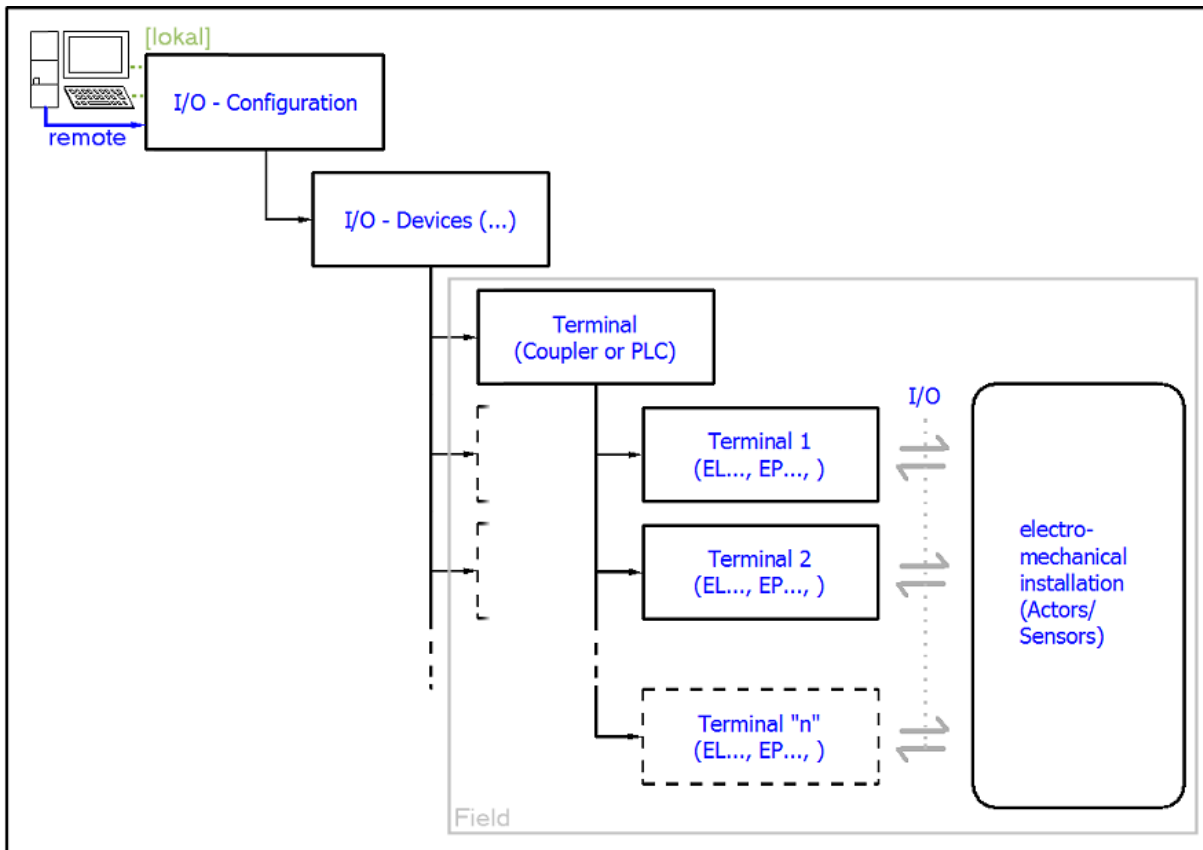


Fig. 31: Relationship between user side (commissioning) and installation

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

Example configuration (actual configuration)

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

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

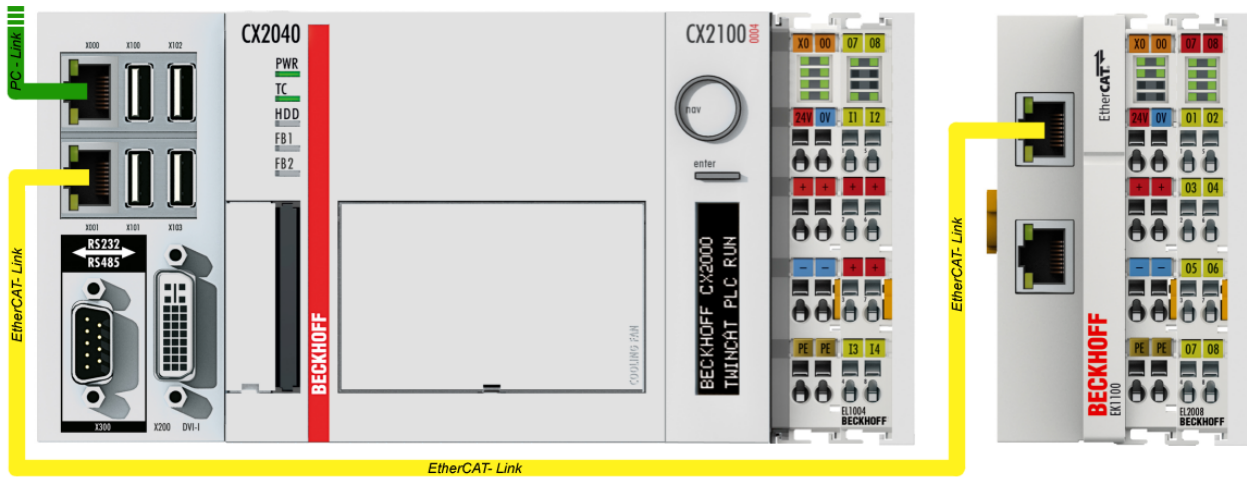


Fig. 32: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

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

5.1.1 TwinCAT 2

Startup

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

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

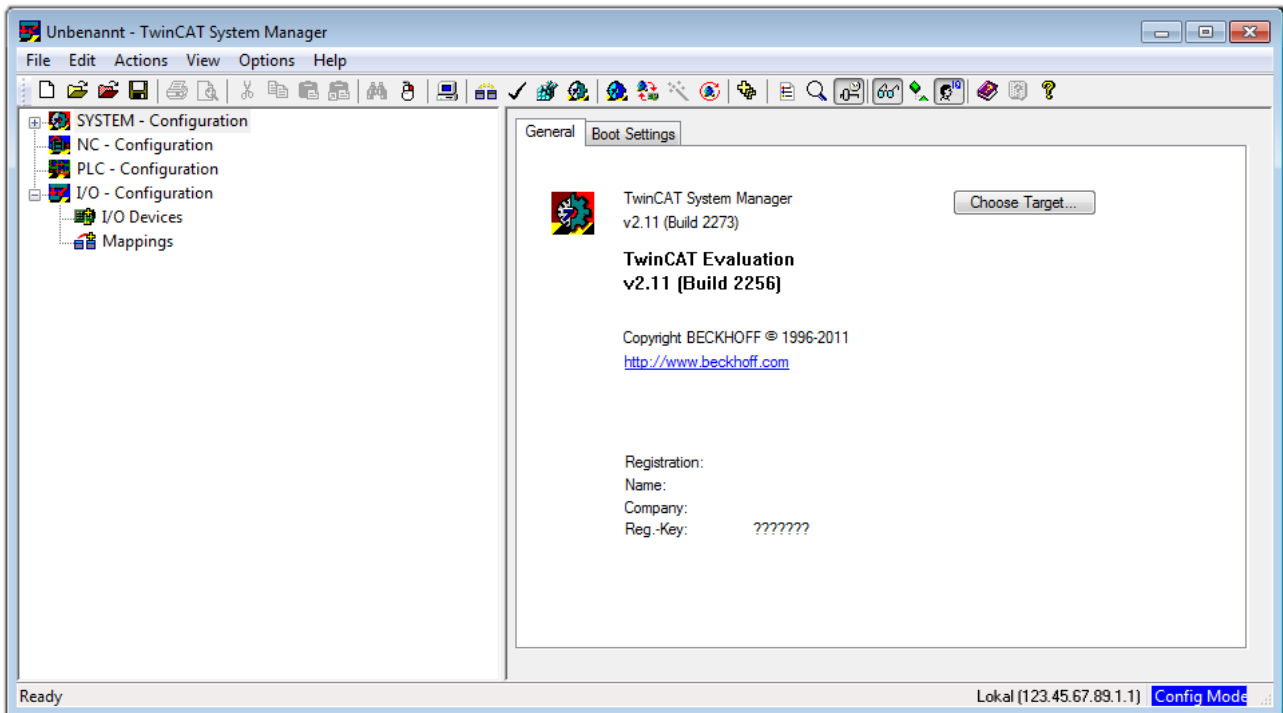



Fig. 33: Initial TwinCAT 2 user interface

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

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

“Actions” → “Choose Target System...”, the following window is opened for this via the symbol “” or the “F8” key:

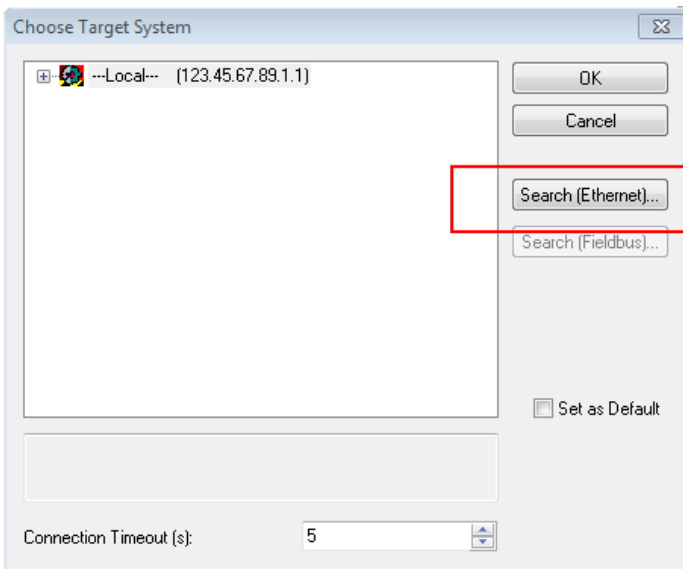


Fig. 34: Selection of the target system

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

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

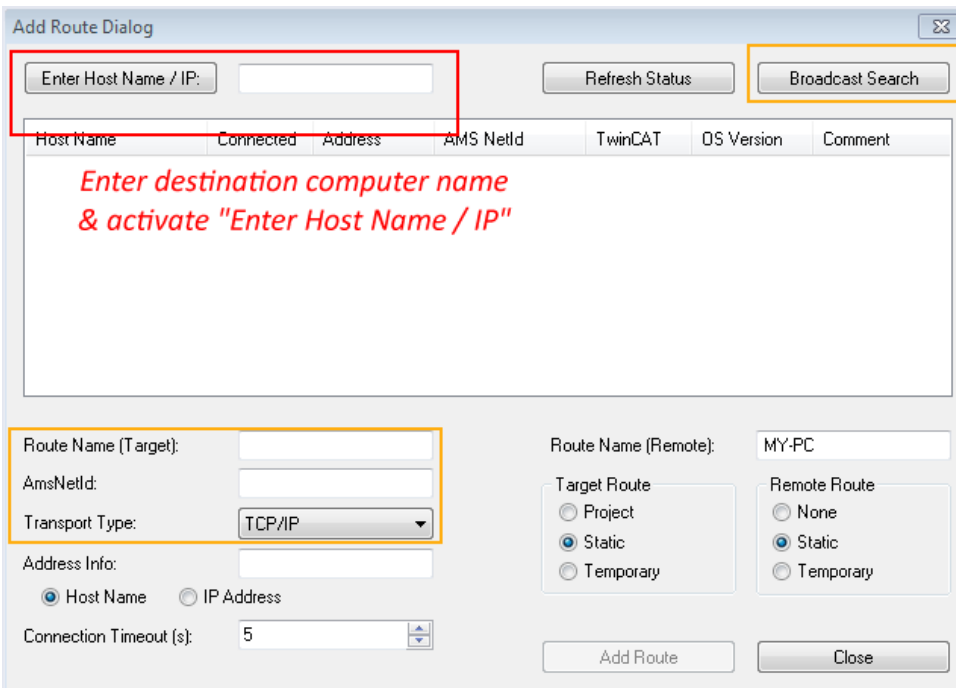
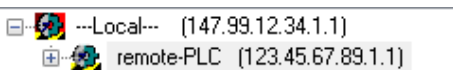


Fig. 35: specify the PLC for access by the TwinCAT System Manager: selection of the target system



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



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

Adding devices

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

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

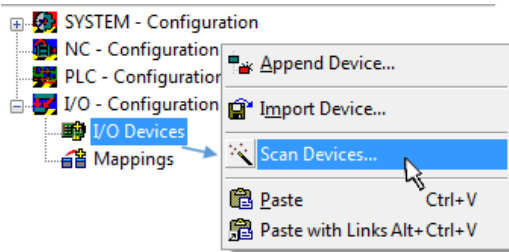


Fig. 36: Select “Scan Devices...”

Confirm the warning message, which follows, and select the “EtherCAT” devices in the dialog:

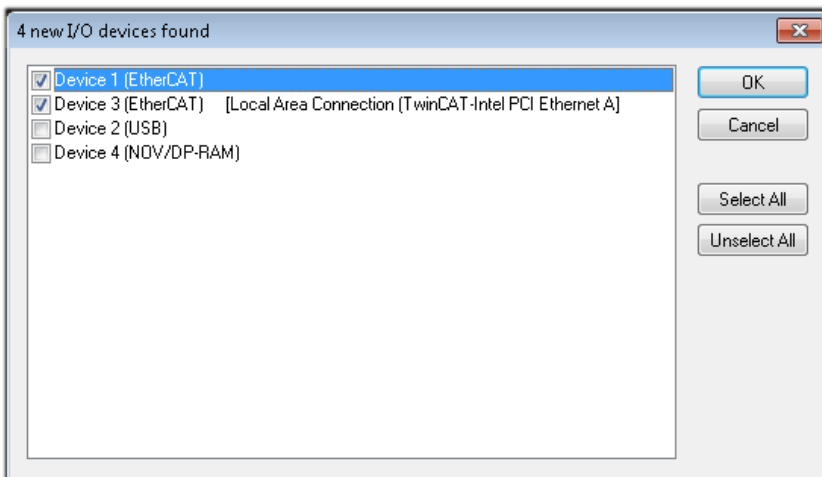


Fig. 37: Automatic detection of I/O devices: selection of the devices to be integrated

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

Based on the [example configuration \[▶ 49\]](#) described at the beginning of this section, the result is as follows:

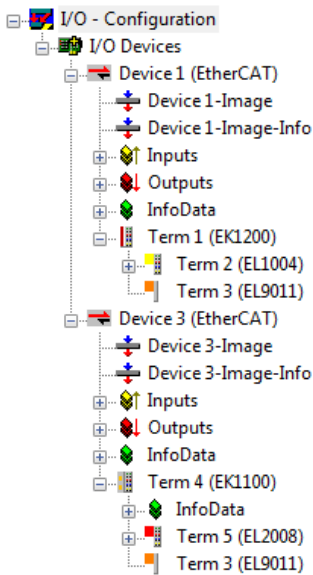


Fig. 38: Mapping of the configuration in the TwinCAT 2 System Manager

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

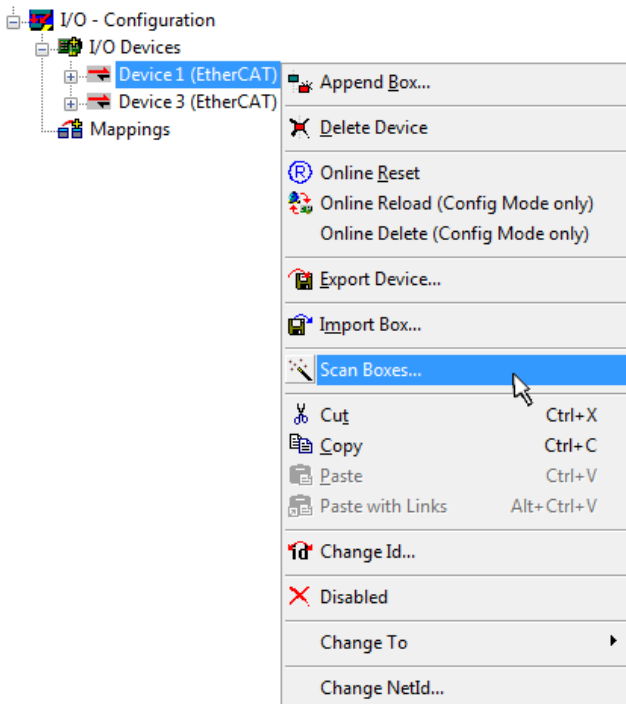


Fig. 39: Reading of individual terminals connected to a device

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

Programming and integrating the PLC

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

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

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

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

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

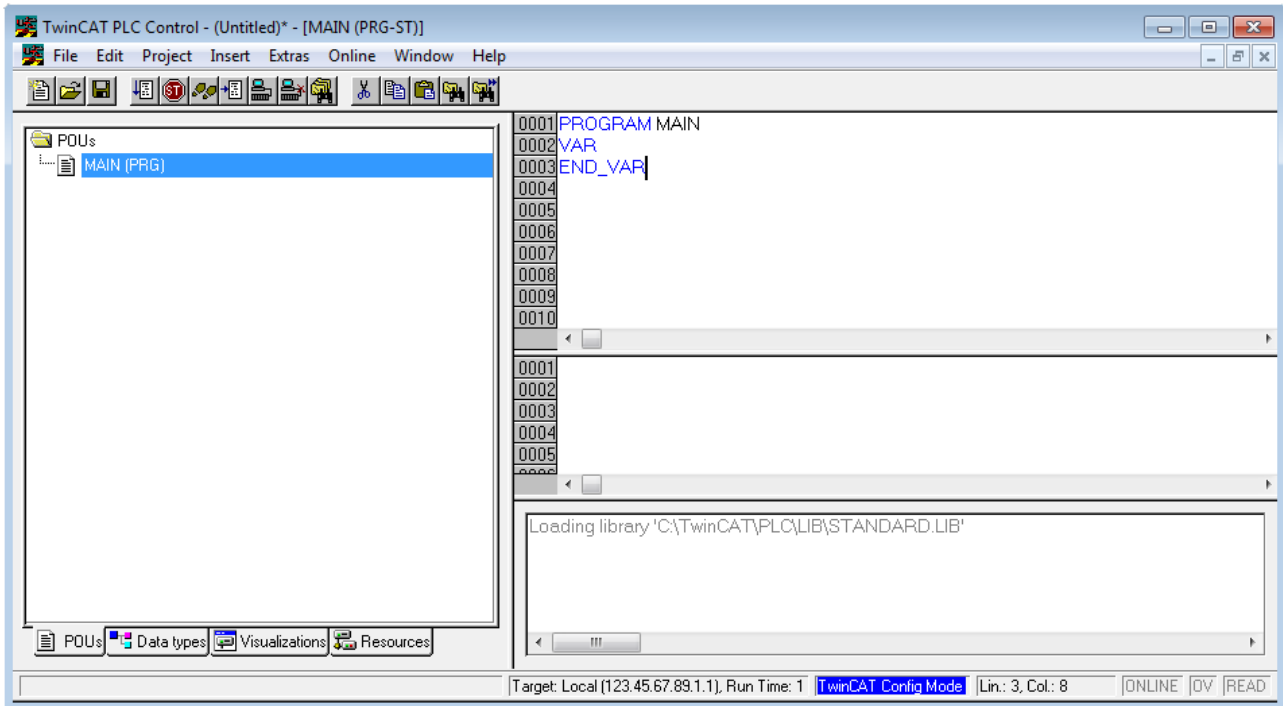


Fig. 40: TwinCAT PLC Control after startup

Example variables and an example program have been created and stored under the name “PLC_example.pro”:

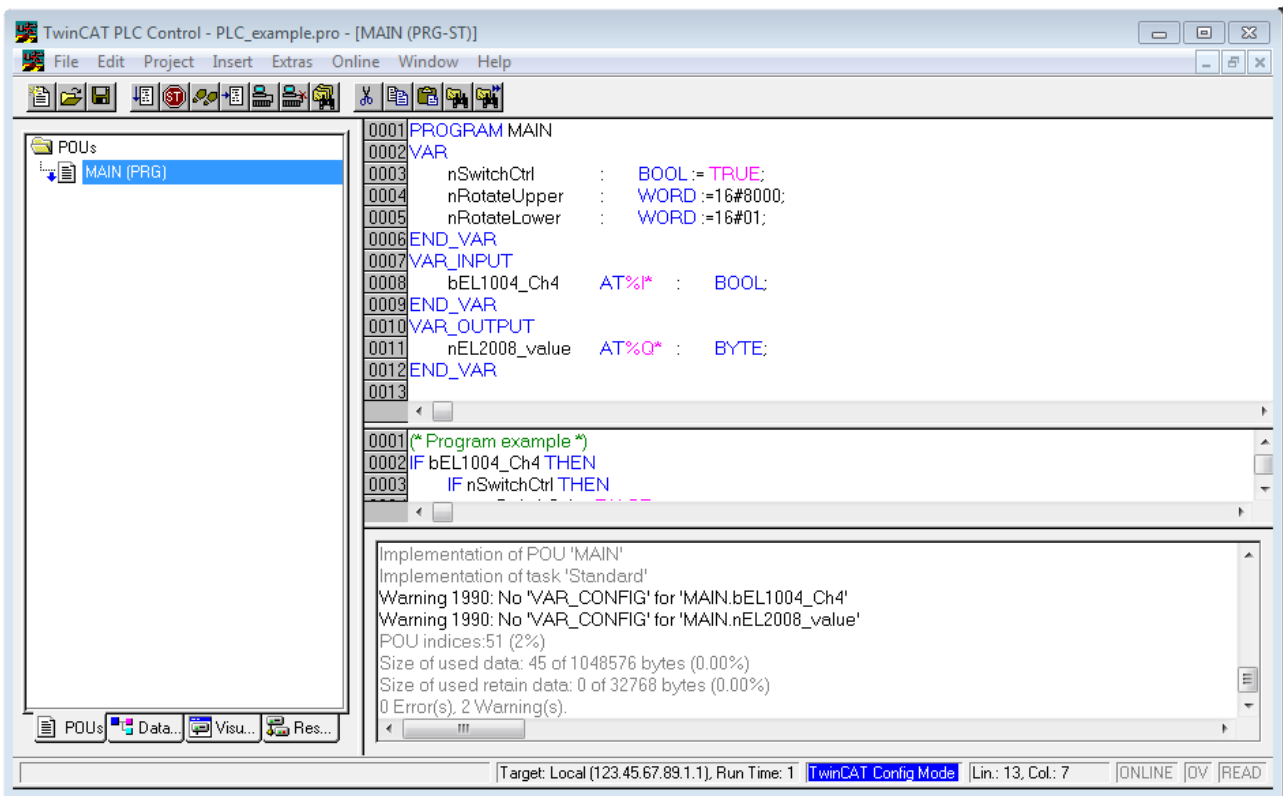


Fig. 41: Example program with variables after a compile process (without variable integration)

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

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

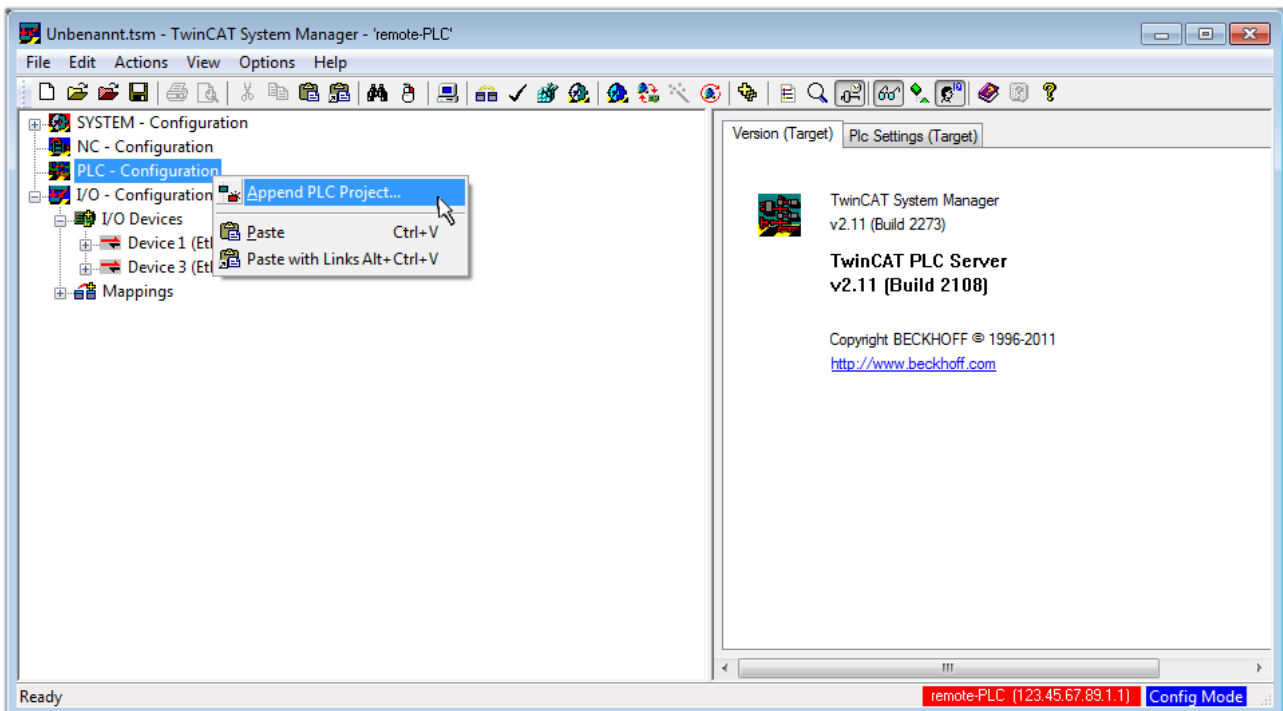


Fig. 42: Appending the TwinCAT PLC Control project

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:

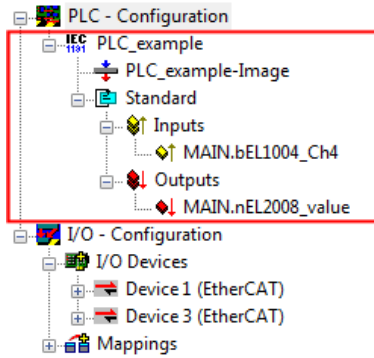


Fig. 43: PLC project integrated in the PLC configuration of the System Manager

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”:

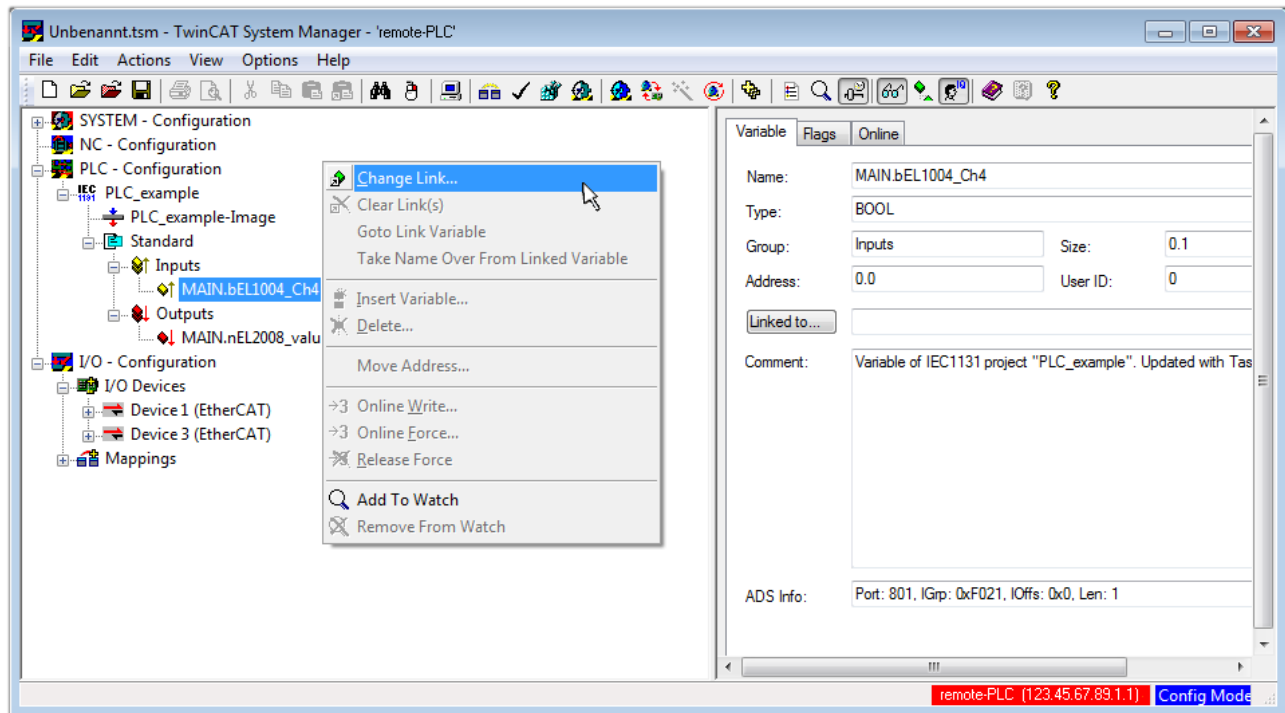


Fig. 44: Creating the links between PLC variables and process objects

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

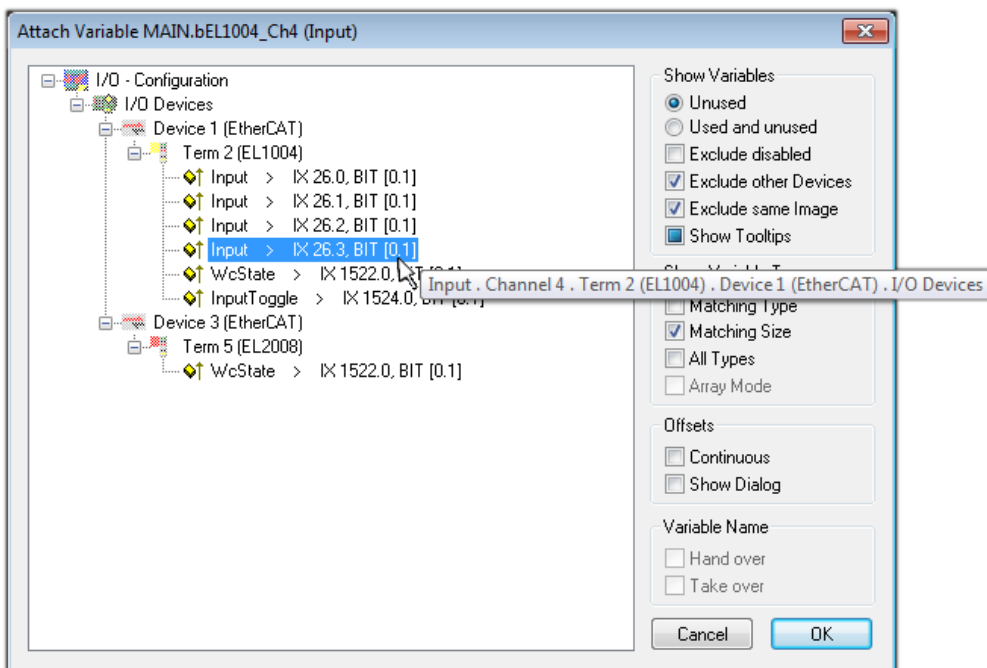


Fig. 45: Selecting BOOL-type PDO

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

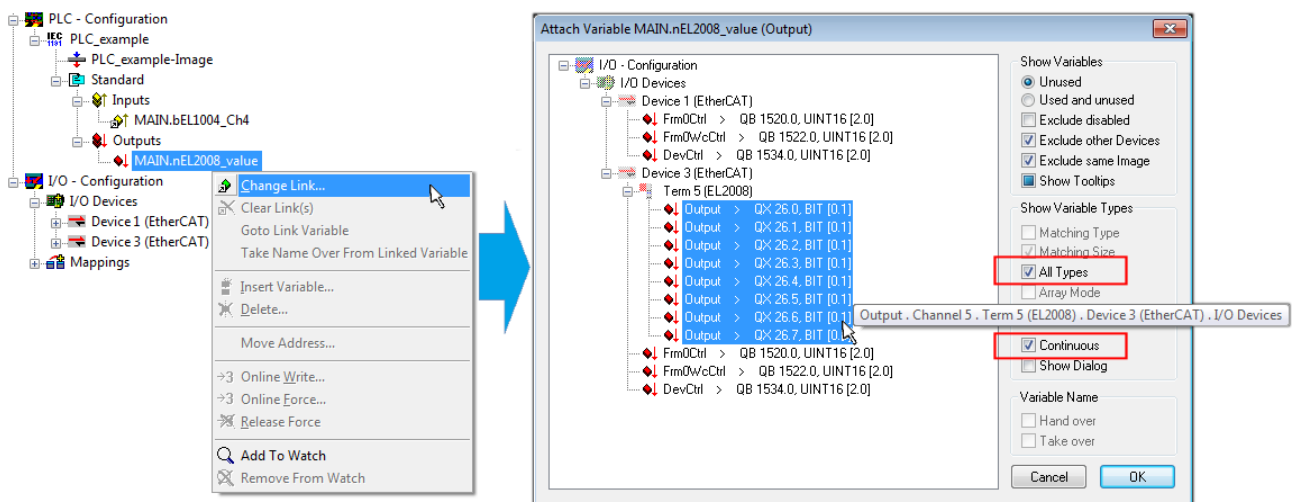



Fig. 46: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

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

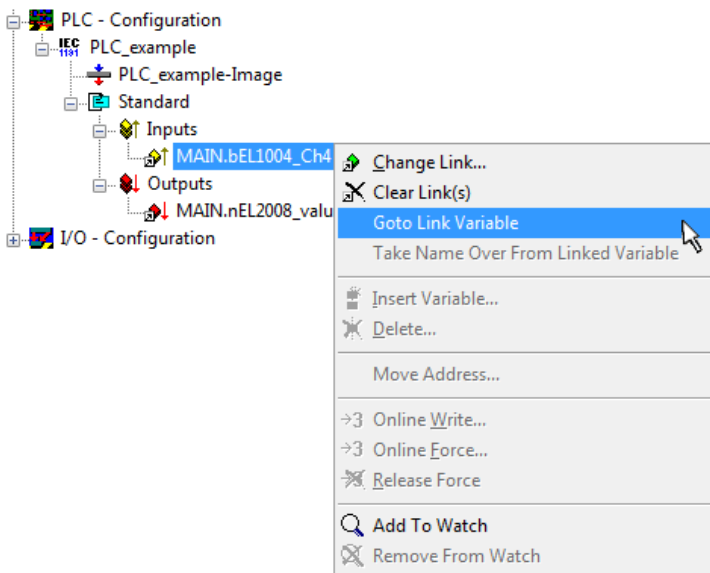

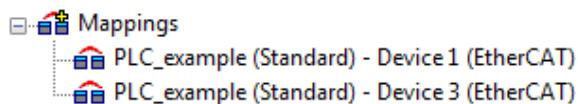


Fig. 47: Application of a “Goto Link Variable”, using “MAIN.bEL1004_Ch4” as an example

The process of assigning variables to the PDO is completed via the menu option “Actions” → “Create assignment”, or via  .


This can be visualized in the configuration:




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

Activation of the configuration

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

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

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

A few seconds later, the real-time status **RTime 0%** is displayed at the bottom right in the System Manager. The PLC system can then be started as described below.

Starting the controller

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

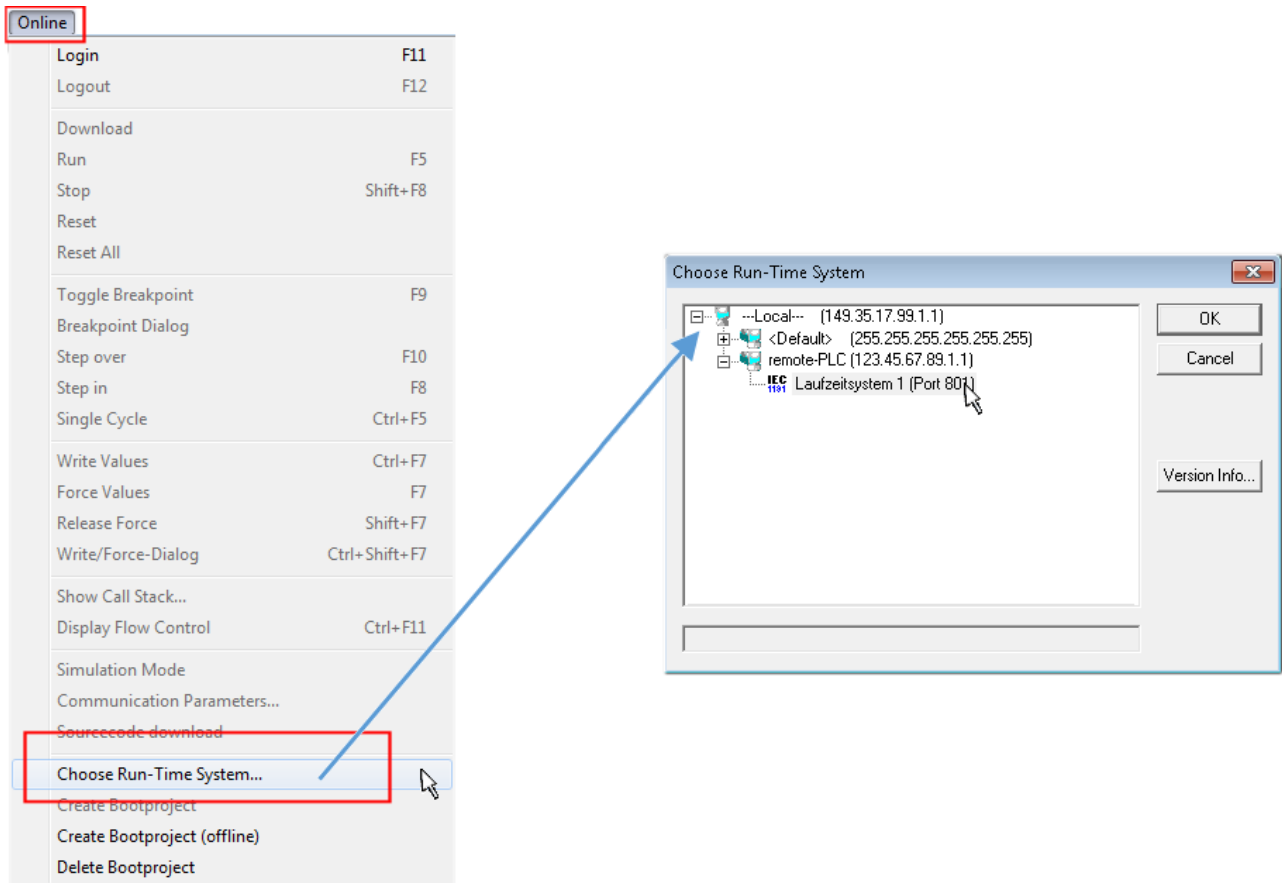

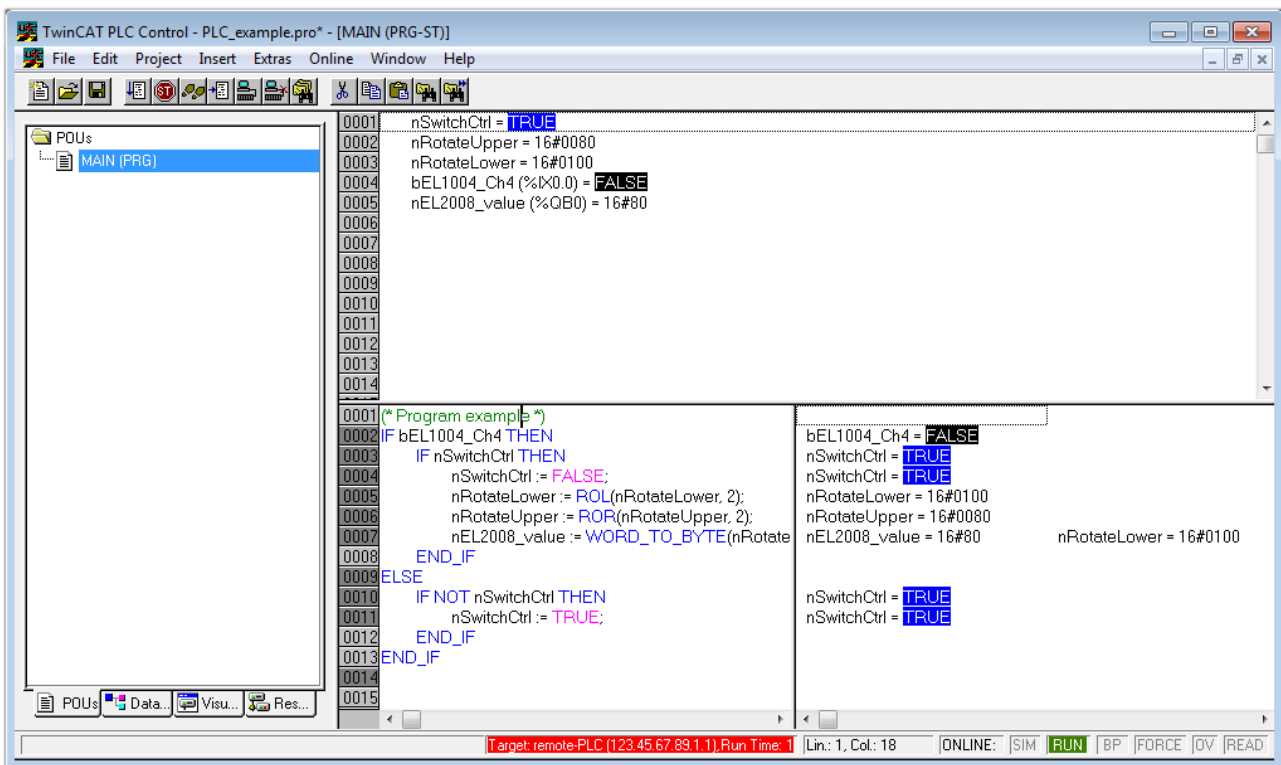


Fig. 48: Choose target system (remote)

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

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



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

5.1.2 TwinCAT 3


Startup

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

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



Fig. 50: Initial TwinCAT 3 user interface

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

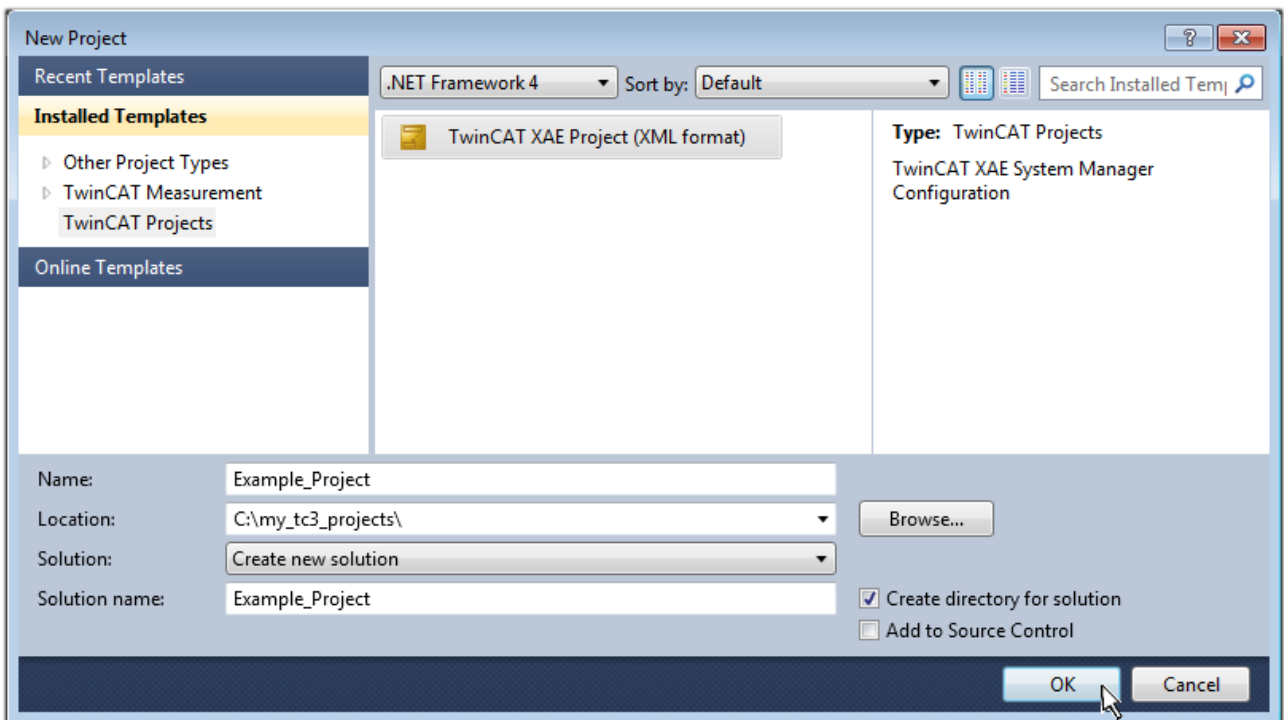


Fig. 51: Create new TwinCAT 3 project

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

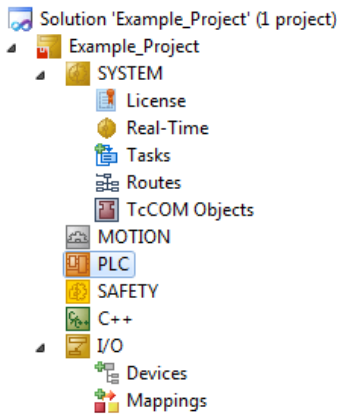
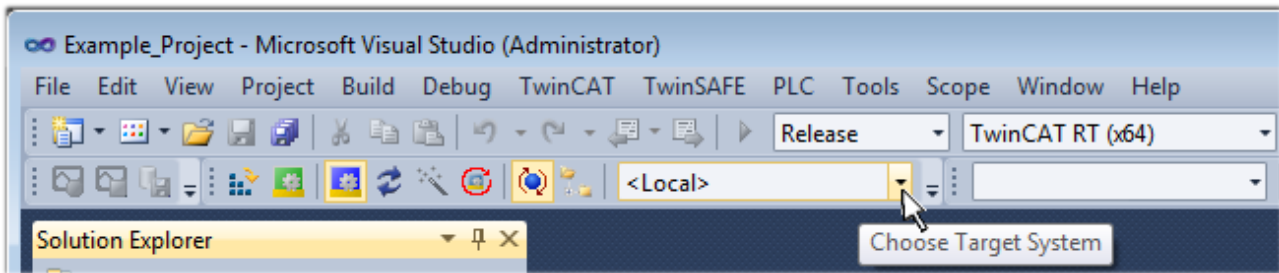


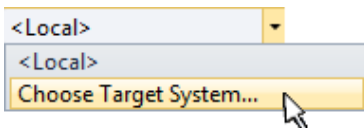
Fig. 52: New TwinCAT 3 project in the project folder explorer

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC (locally), TwinCAT can be used in local mode and the process can be continued with the next step, “Insert Device [▶ 64]”.

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



expand the pull-down menu:



and open the following window:

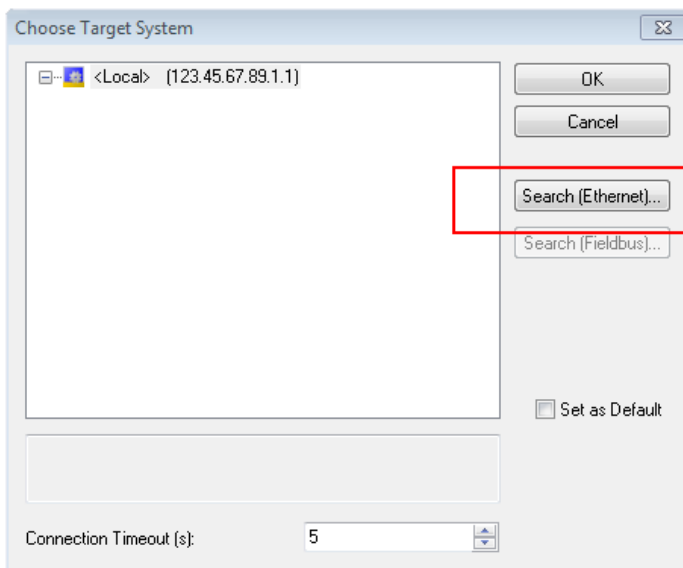


Fig. 53: Selection dialog: Choose the target system

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

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

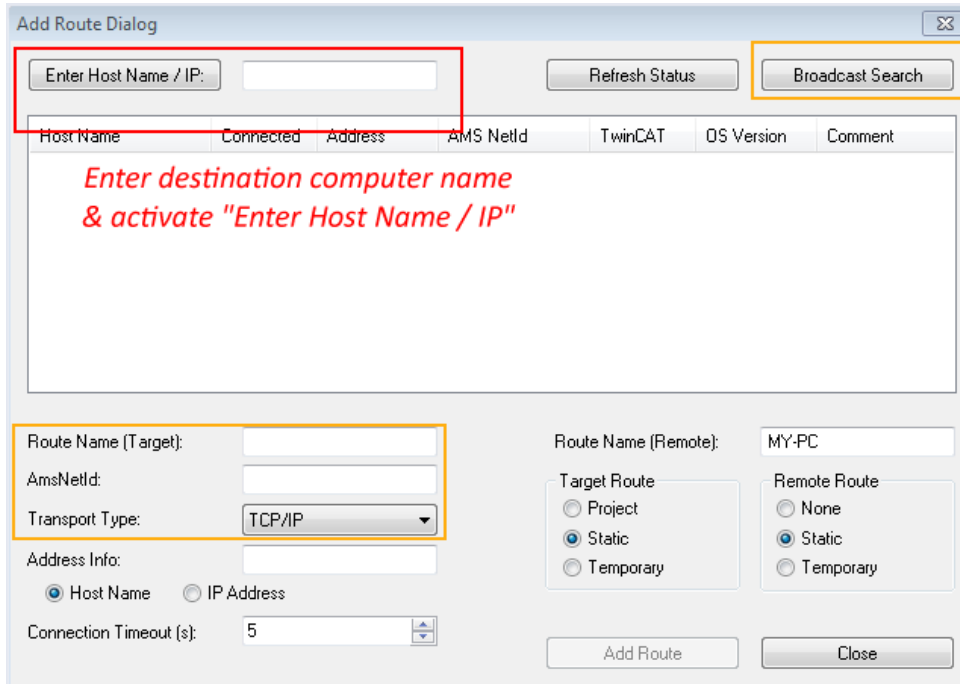
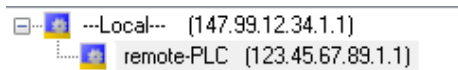


Fig. 54: specify the PLC for access by the TwinCAT System Manager: selection of the target system


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




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

Adding devices

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

element “I/O”, then right-click to open a context menu and select “Scan” or start the action via  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)”.

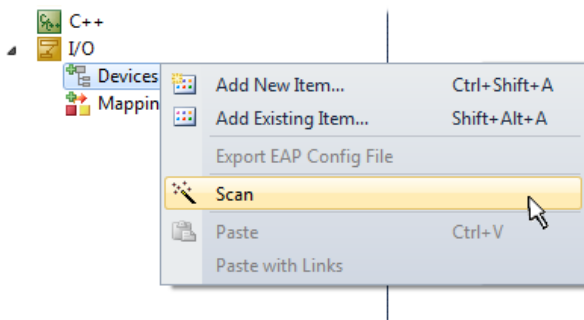


Fig. 55: Select “Scan”

Confirm the warning message, which follows, and select the “EtherCAT” devices in the dialog:

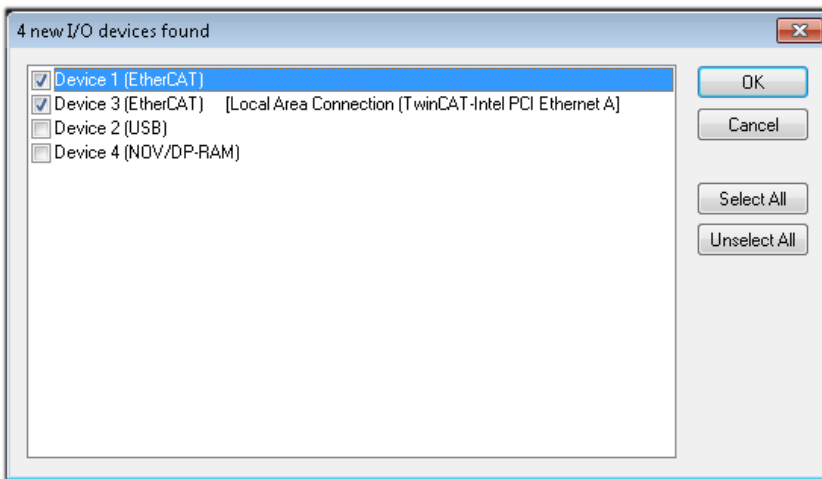


Fig. 56: Automatic detection of I/O devices: selection of the devices to be integrated

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

Based on the [example configuration \[▶ 49\]](#) described at the beginning of this section, the result is as follows:

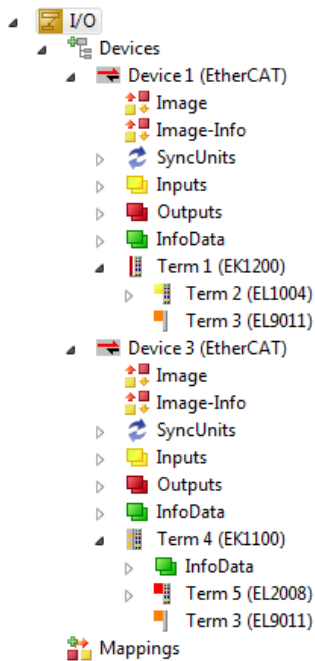


Fig. 57: Mapping of the configuration in VS shell of the TwinCAT 3 environment

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

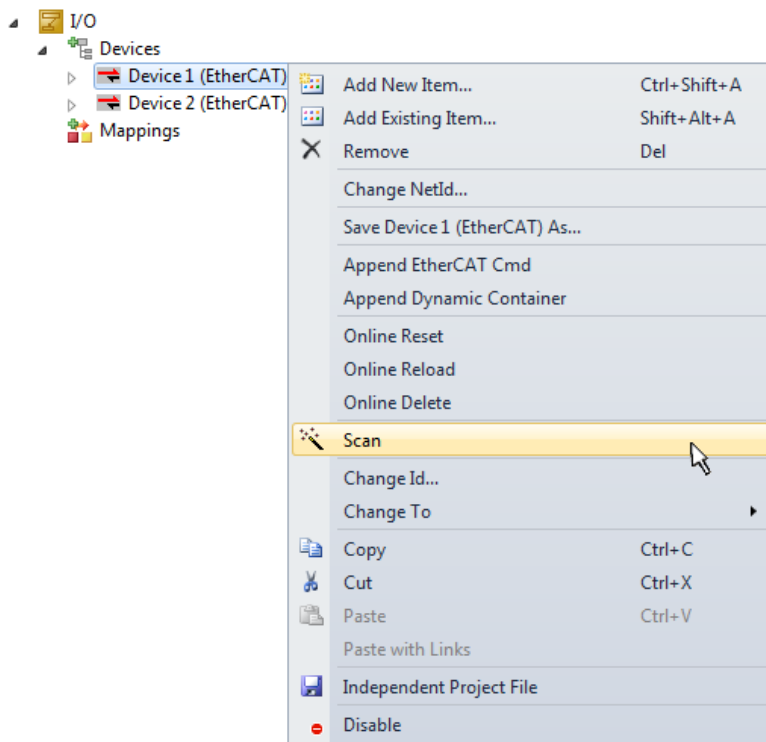


Fig. 58: Reading of individual terminals connected to a device

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

Programming the PLC

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

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

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

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

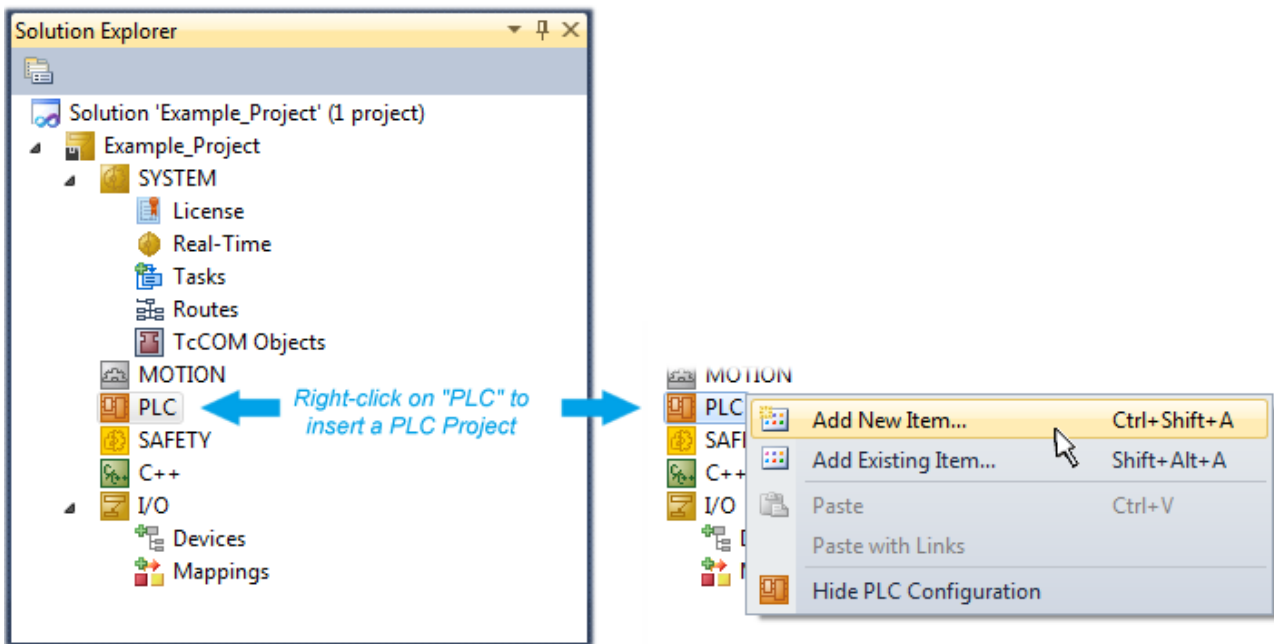


Fig. 59: Adding the programming environment in “PLC”

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

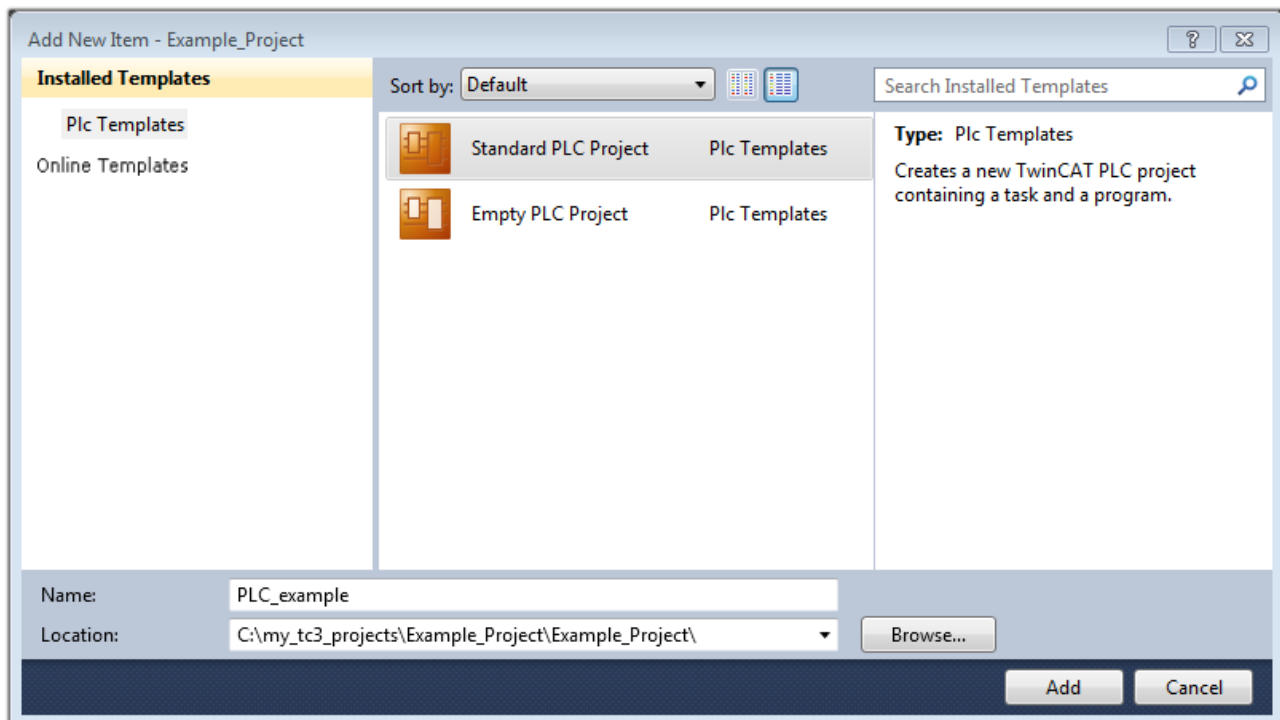


Fig. 60: Specifying the name and directory for the PLC programming environment

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

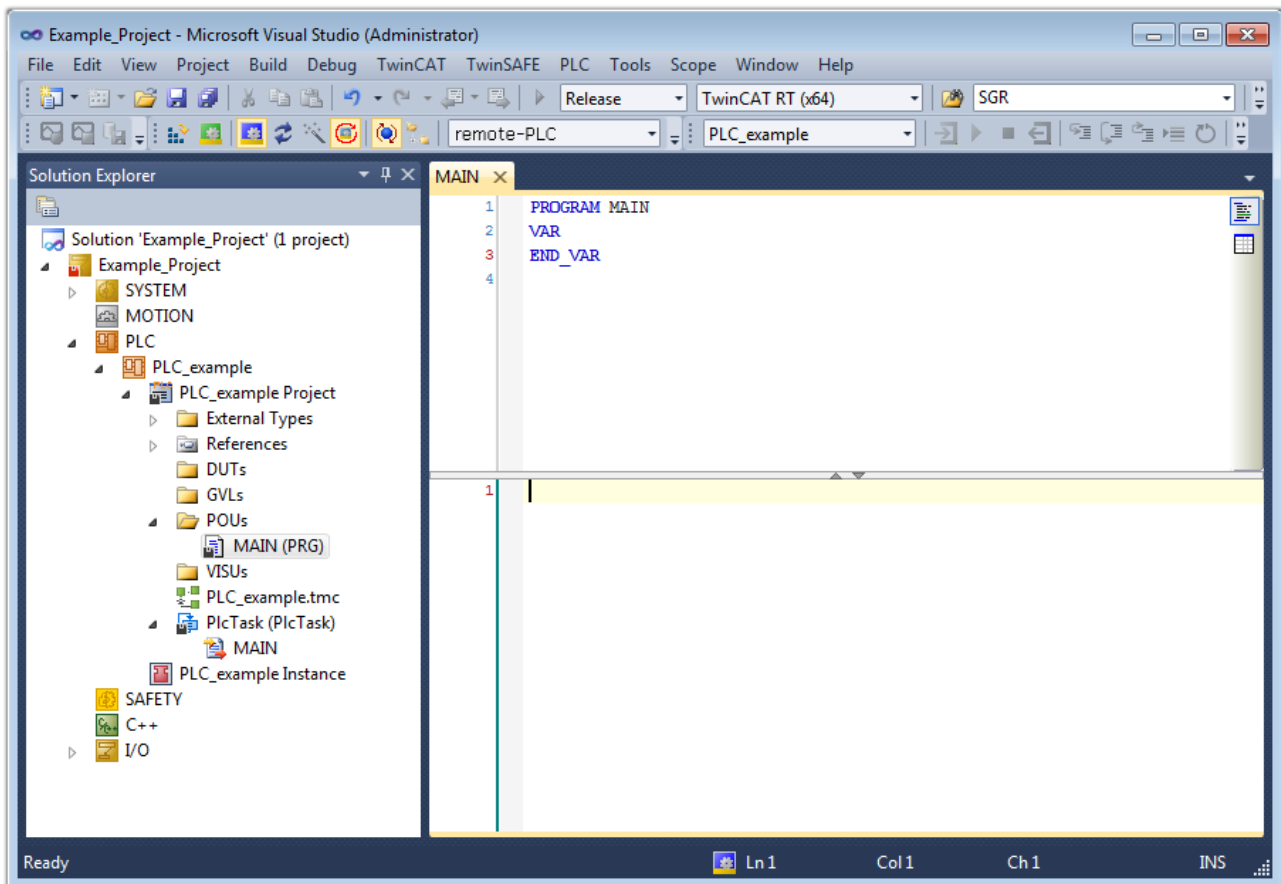


Fig. 61: Initial “Main” program for the standard PLC project

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

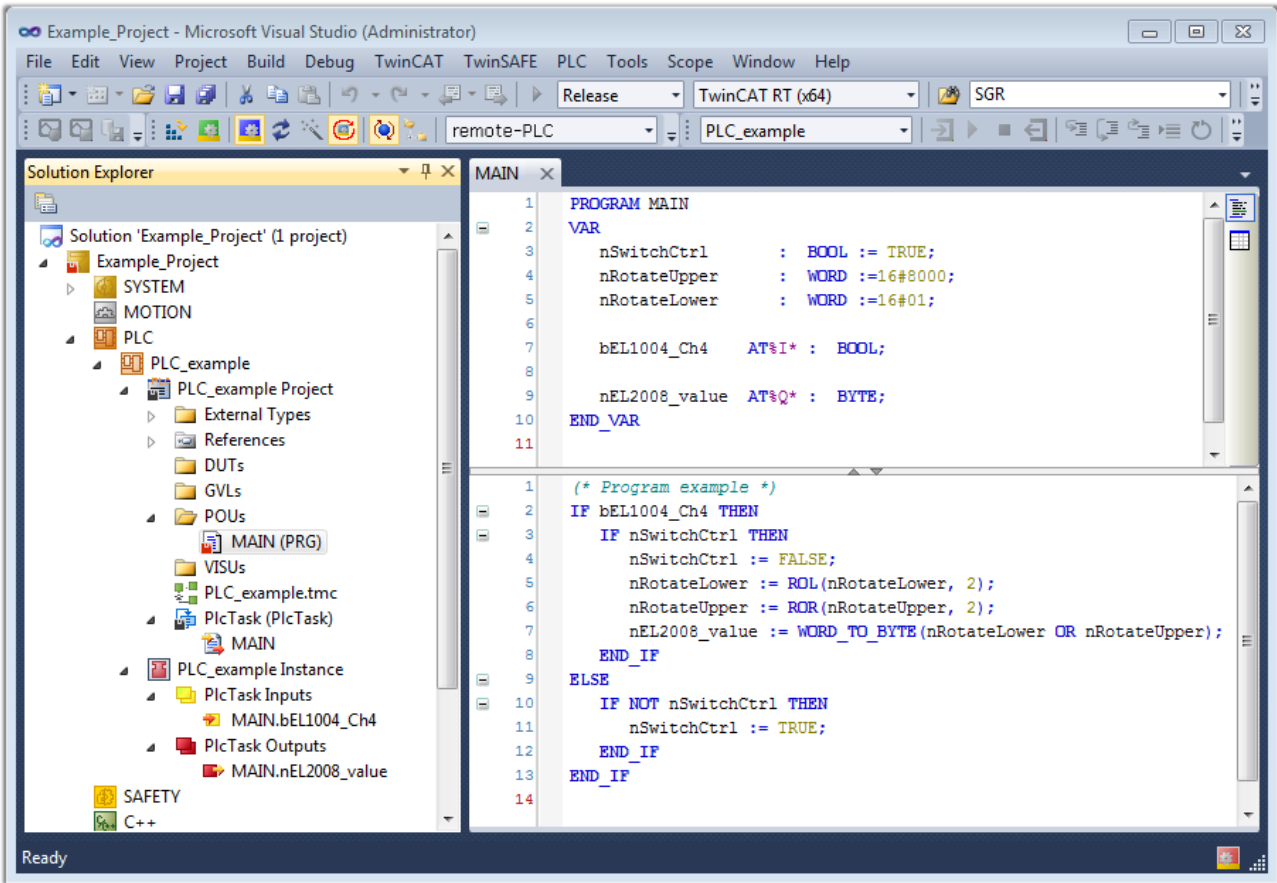


Fig. 62: Example program with variables after a compile process (without variable integration)

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

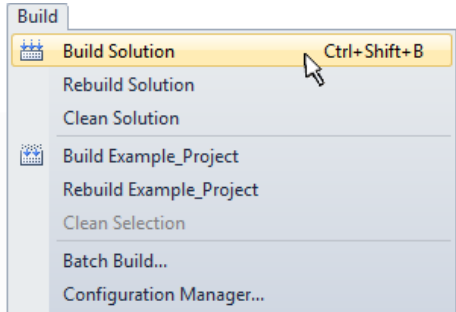
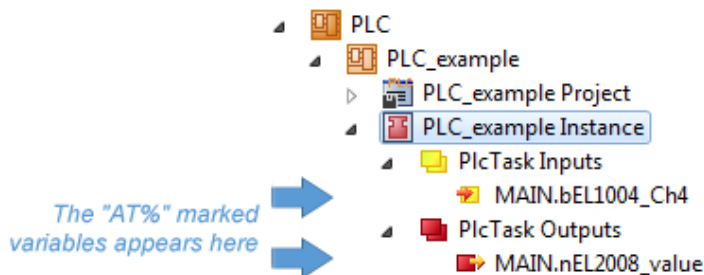


Fig. 63: Start program compilation

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



Assigning variables

Via the menu of an instance – variables in the “PLC” context, use the “Modify Link...” option to open a window to select a suitable process object (PDO) for linking:

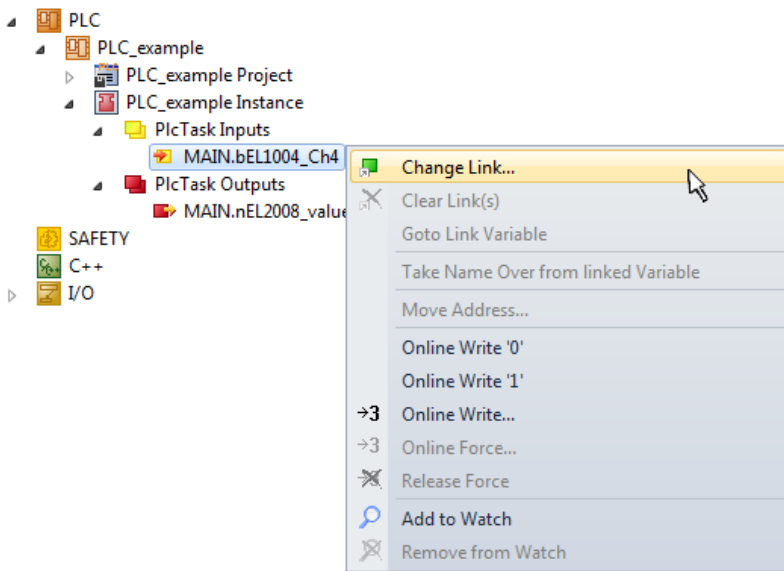


Fig. 64: Creating the links between PLC variables and process objects

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

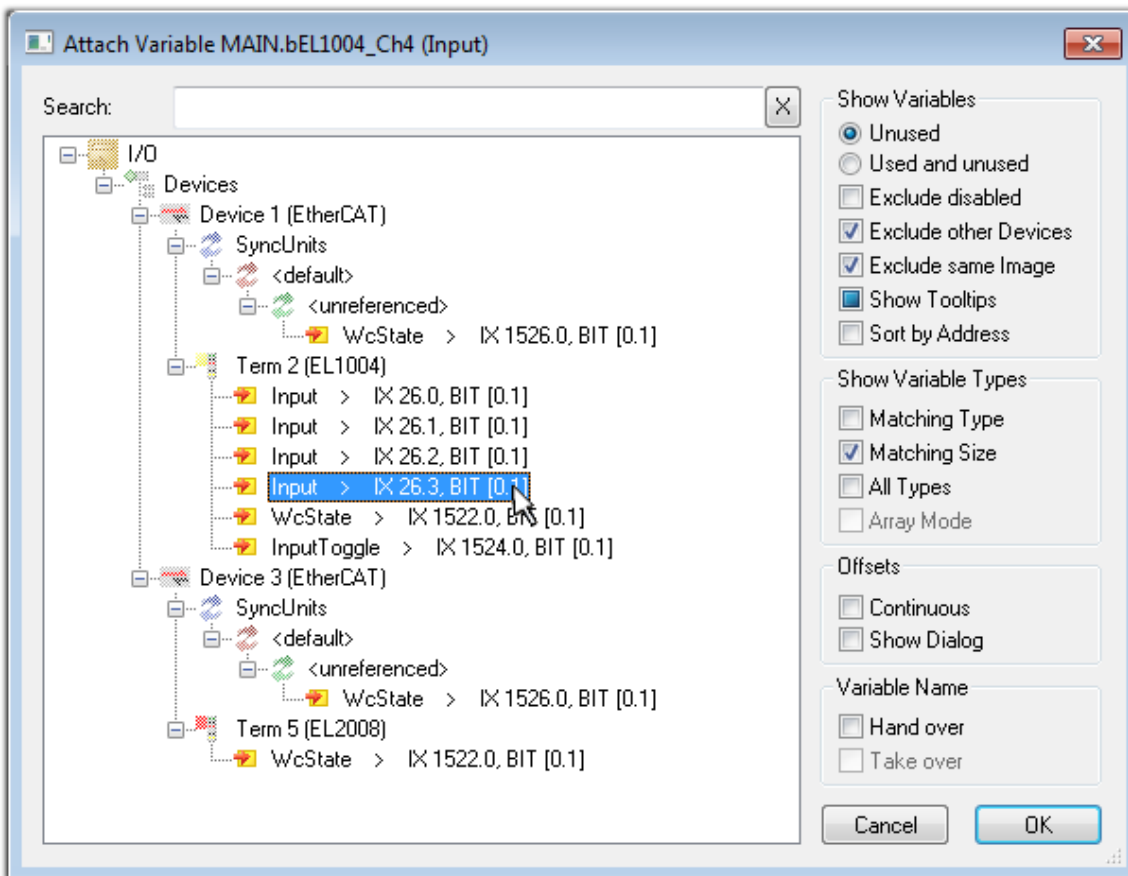


Fig. 65: Selecting BOOL-type PDO

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

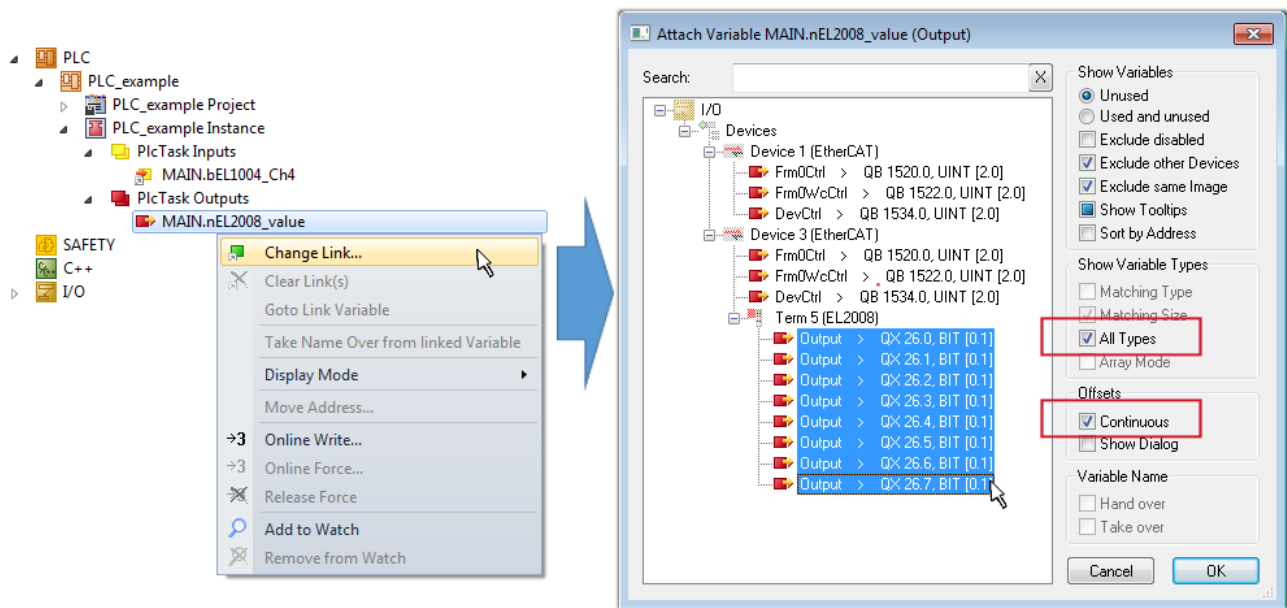



Fig. 66: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

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

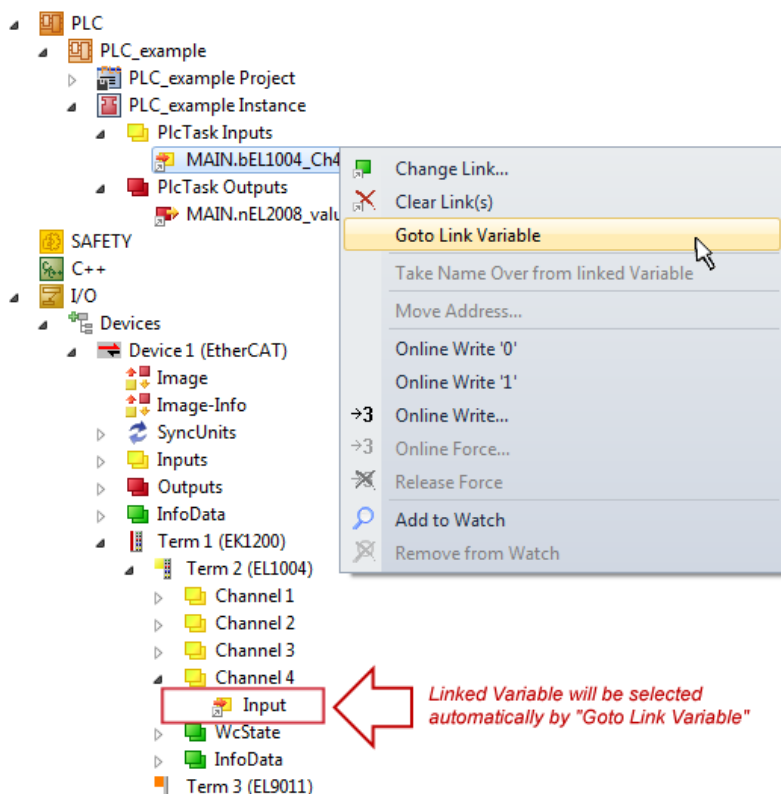


Fig. 67: Application of a “Goto Link Variable”, using “MAIN.bEL1004_Ch4” as an example

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

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

● Note on type of variable assignment



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

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

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

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

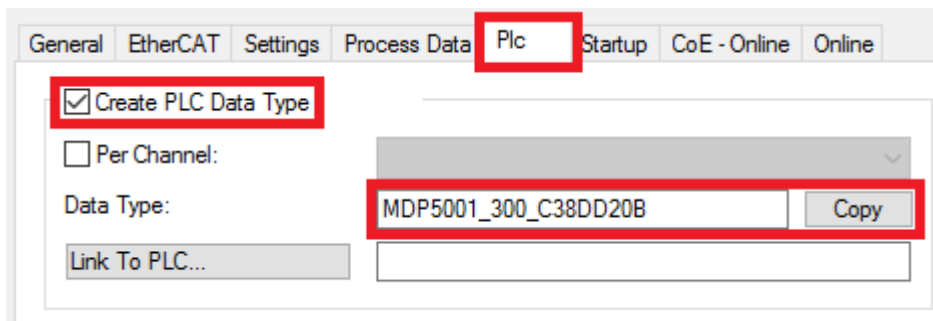


Fig. 68: Creating a PLC data type

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

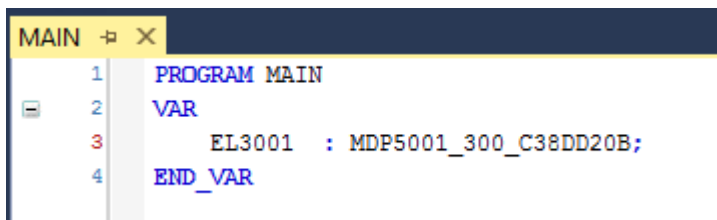


Fig. 69: Instance_of_struct

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

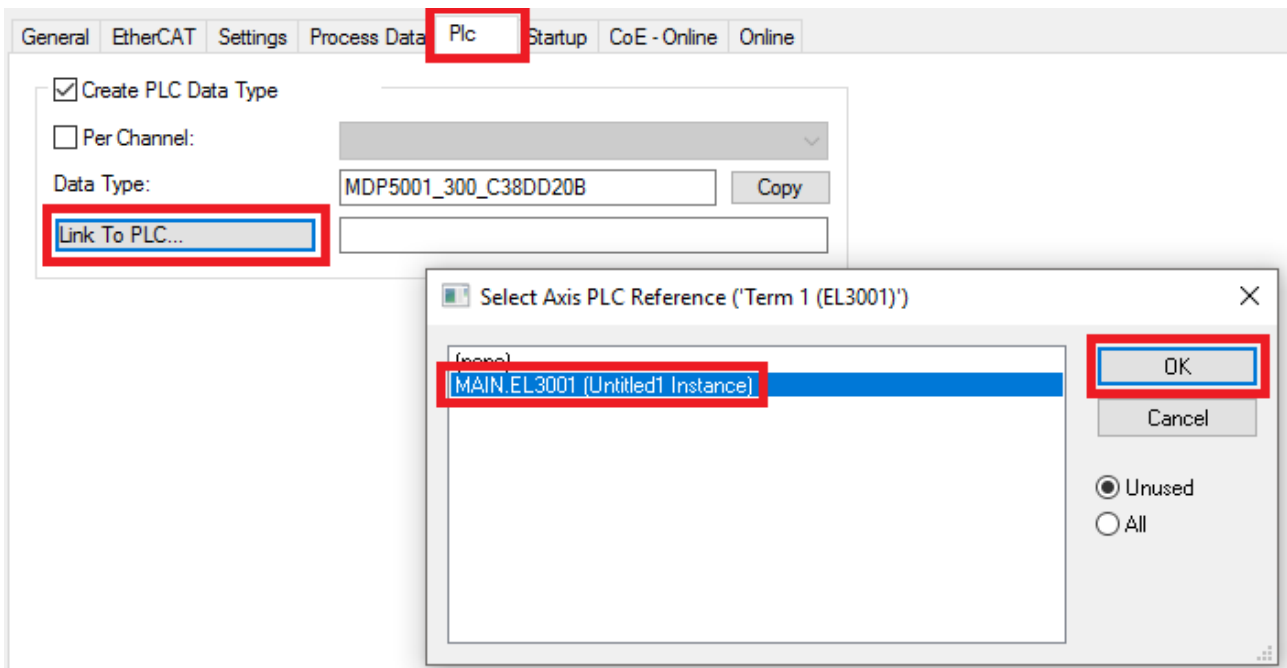


Fig. 70: Linking the structure

7. In the PLC, the process data can then be read or written via the structure in the program code.

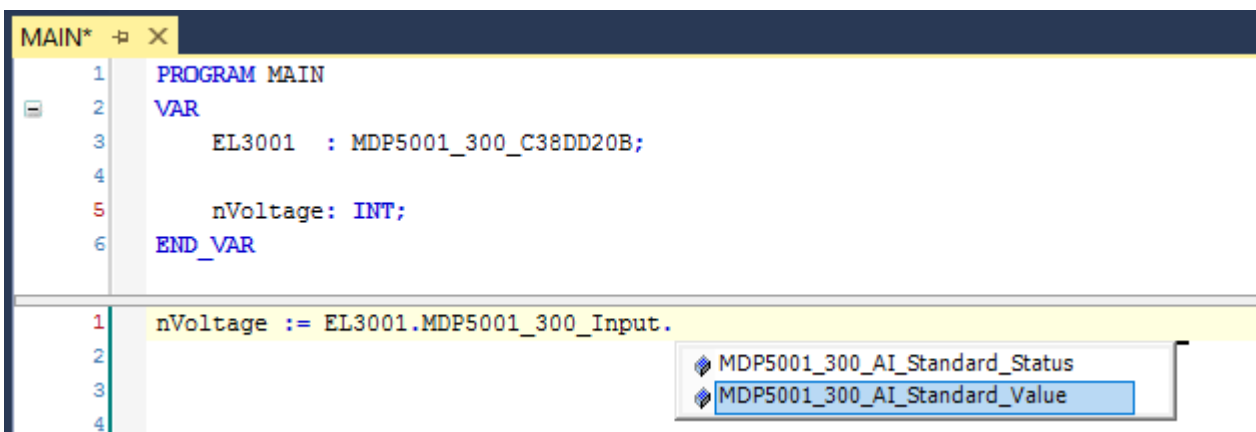

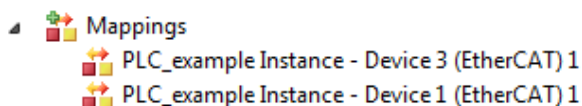


Fig. 71: Reading a variable from the structure of the process data

Activation of the configuration

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

and outputs of the terminals. The configuration can now be activated with  or via the menu under “TwinCAT” in order to transfer the settings of the development environment to the runtime system. Confirm the messages “Old configurations will be overwritten!” and “Restart TwinCAT system in Run mode” with “OK”. The corresponding assignments can be seen in the project folder explorer:





A few seconds later, the corresponding status of the Run mode is displayed in the form of a rotating symbol



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

Starting the controller

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

the program to be started by clicking on symbol , the “F5” key or via “PLC” in the menu, by selecting “Start”. The started programming environment shows the runtime values of individual variables:

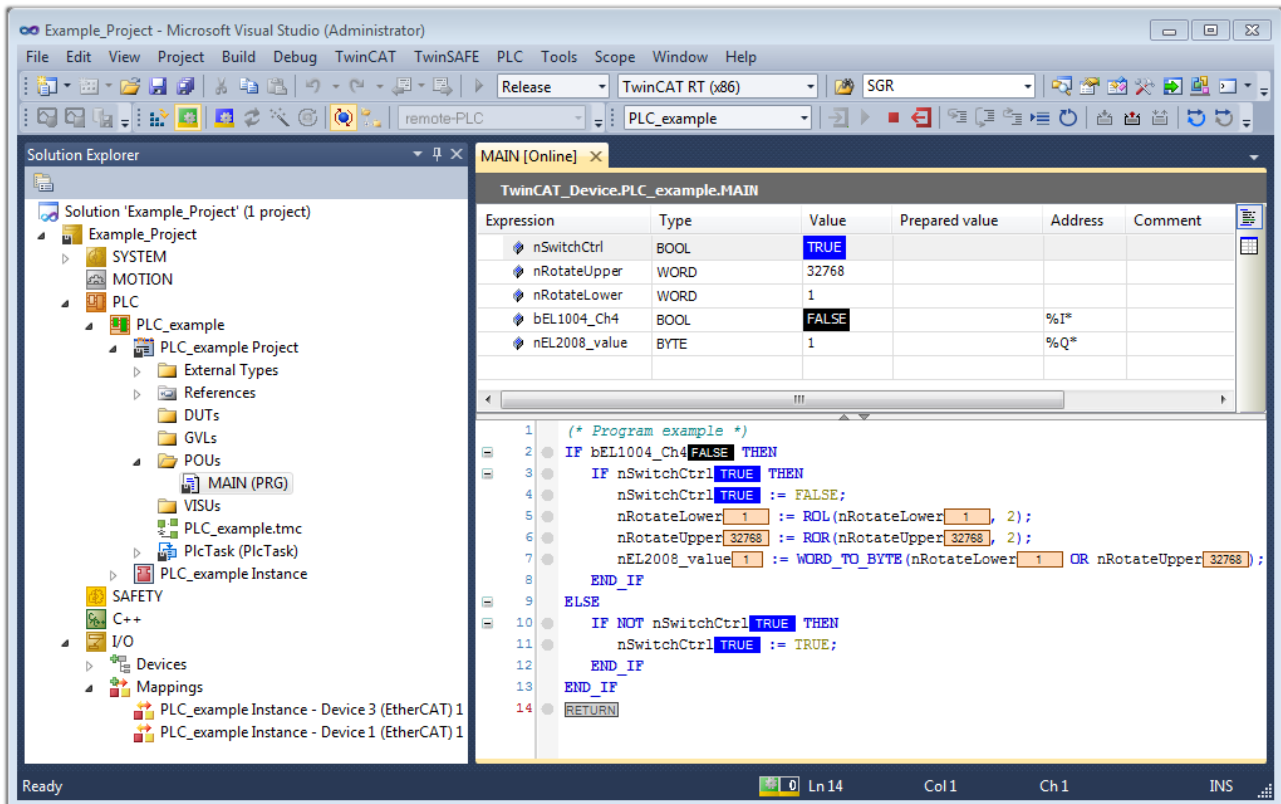




Fig. 72: TwinCAT 3 development environment (VS shell): logged-in, after program startup

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

5.2 TwinCAT Development Environment

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

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

Details:

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

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

Additional features:

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

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

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

5.2.1 Installation of the TwinCAT real-time driver

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

This can be done in several ways.

A: Via the TwinCAT Adapter dialog

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

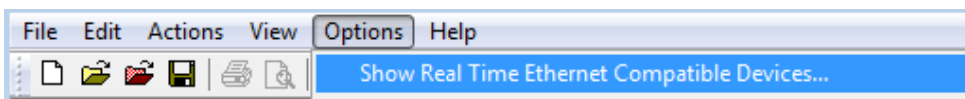


Fig. 73: System Manager "Options" (TwinCAT 2)

This have to be called up by the menu "TwinCAT" within the TwinCAT 3 environment:

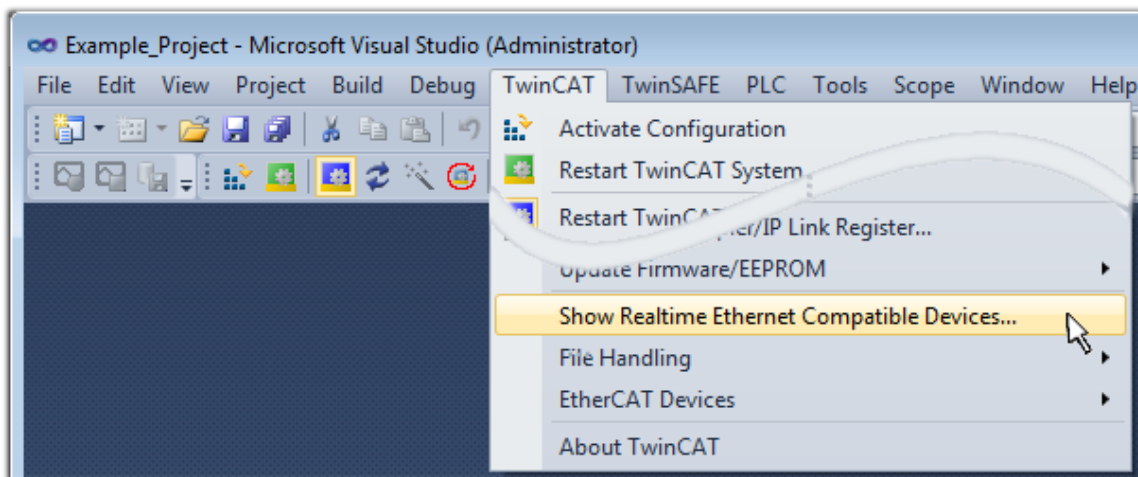


Fig. 74: Call up under VS Shell (TwinCAT 3)

B: Via TcRtelInstall.exe in the TwinCAT directory

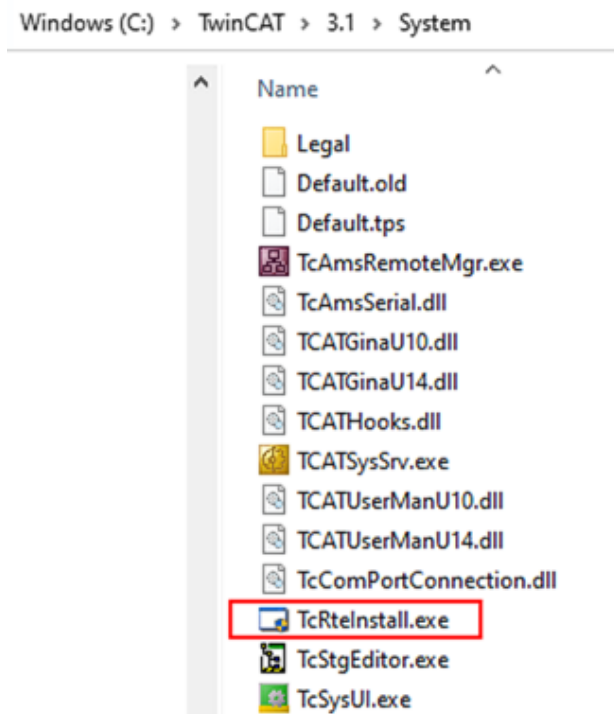


Fig. 75: TcRtelInstall in the TwinCAT directory

In both cases, the following dialog appears:

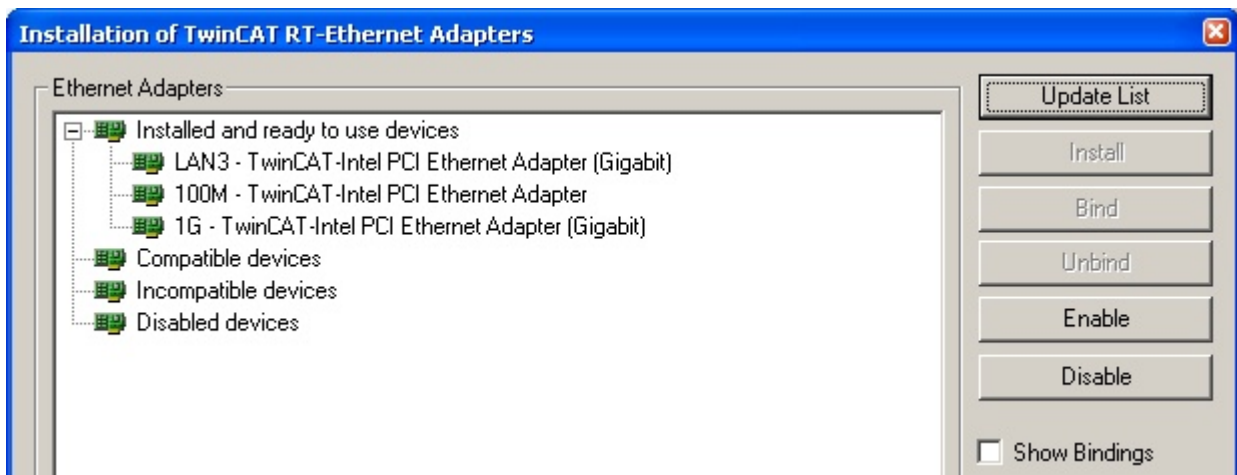


Fig. 76: Overview of network interfaces

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

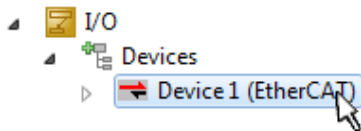
A Windows warning regarding the unsigned driver can be ignored.

Alternatively an EtherCAT-device can be inserted first of all as described in chapter [Offline configuration creation](#), section “Creating the EtherCAT device” [▶ 85] in order to view the compatible ethernet ports via its EtherCAT properties (tab “Adapter”, button “Compatible Devices...”):



Fig. 77: 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)

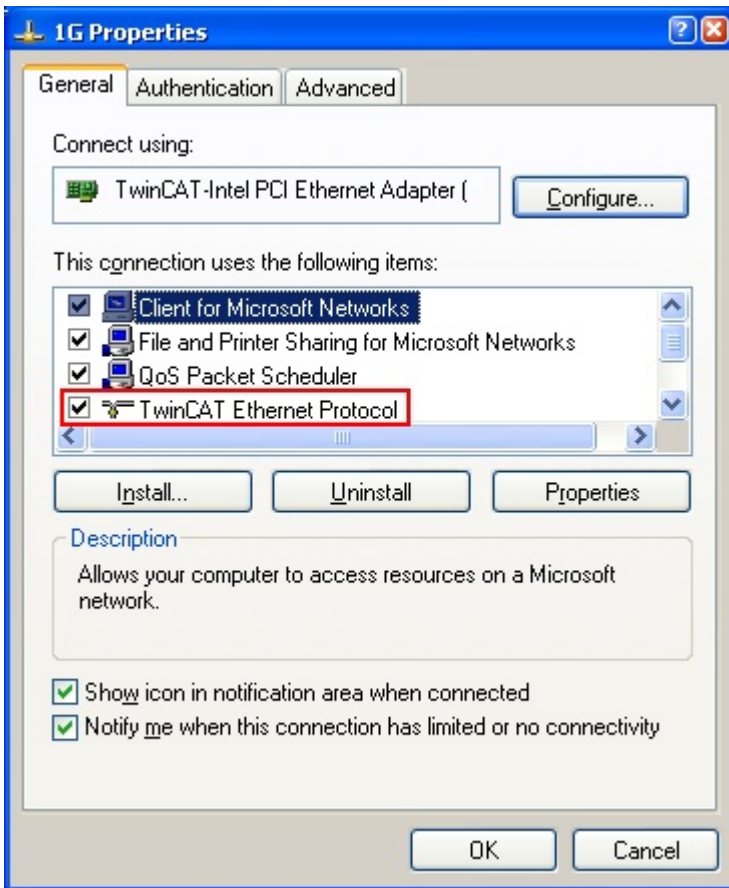


Fig. 78: Windows properties of the network interface

A correct setting of the driver could be:

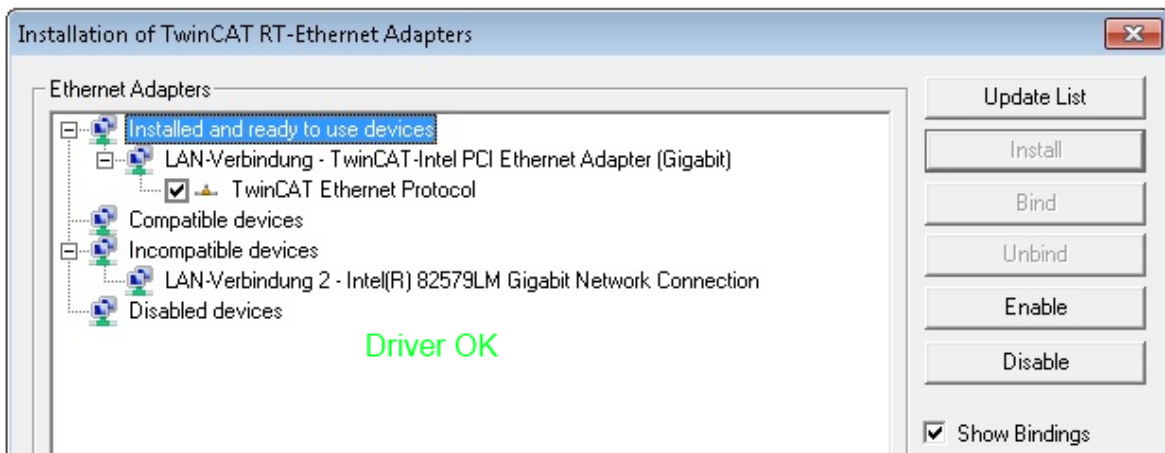


Fig. 79: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

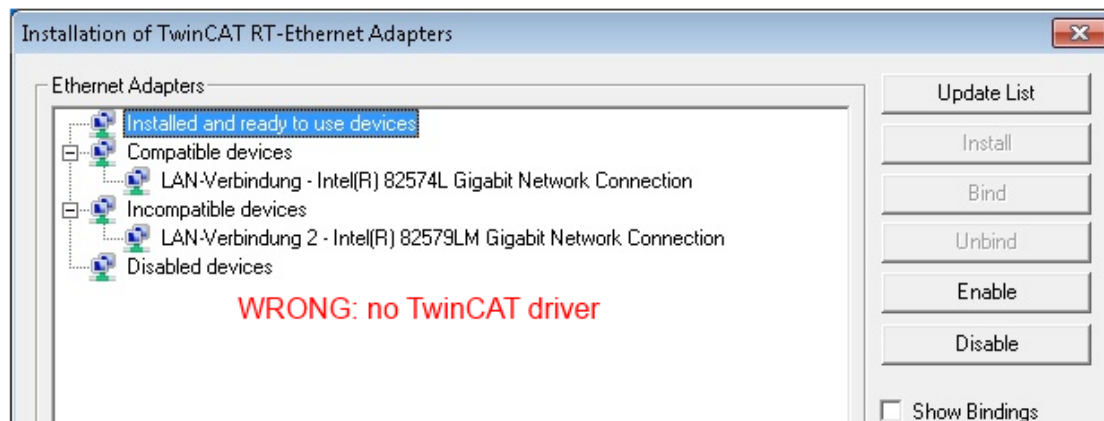
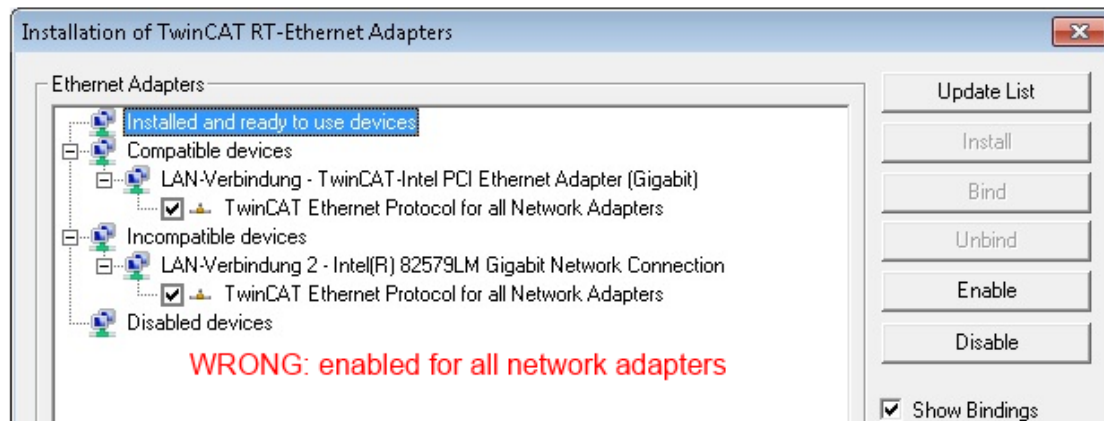
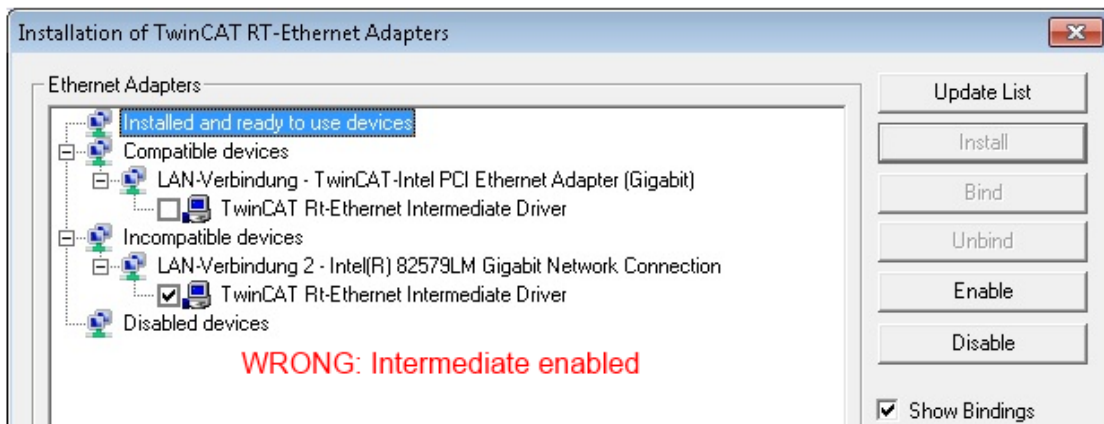
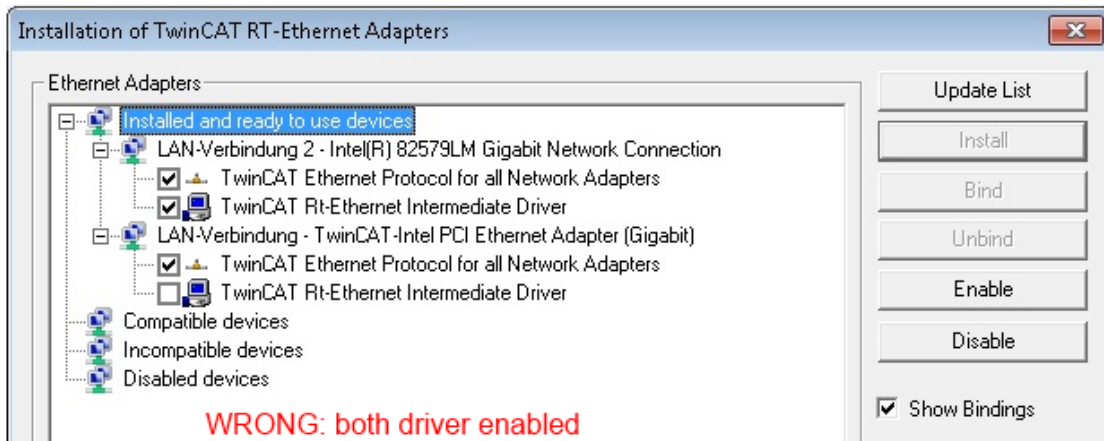


Fig. 80: Incorrect driver settings for the Ethernet port

IP address of the port used

● IP address/DHCP

i 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.

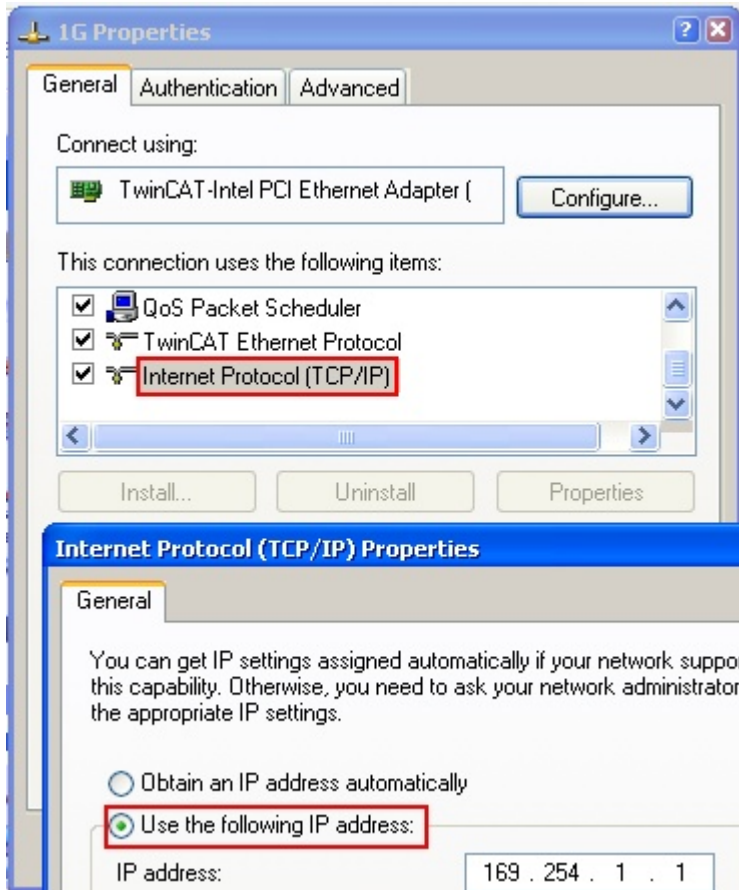


Fig. 81: TCP/IP setting for the Ethernet port

5.2.2 Notes regarding ESI device description

Installation of the latest ESI device description

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

The ESI files for Beckhoff EtherCAT devices are available on the [Beckhoff website](#).

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

Default settings:

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

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

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

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

- **TwinCAT 2:** Option → “Update EtherCAT Device Descriptions”
- **TwinCAT 3:** TwinCAT → EtherCAT Devices → “Update Device Descriptions (via ETG Website)...”

The [TwinCAT ESI Updater](#) [▶ 85] is available for this purpose.



ESI

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

Device differentiation

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

- family key “EL”
- name “2521”
- type “0025”
- and revision “1018”

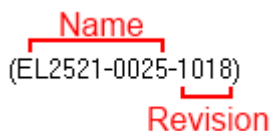


Fig. 82: Identifier structure

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

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.

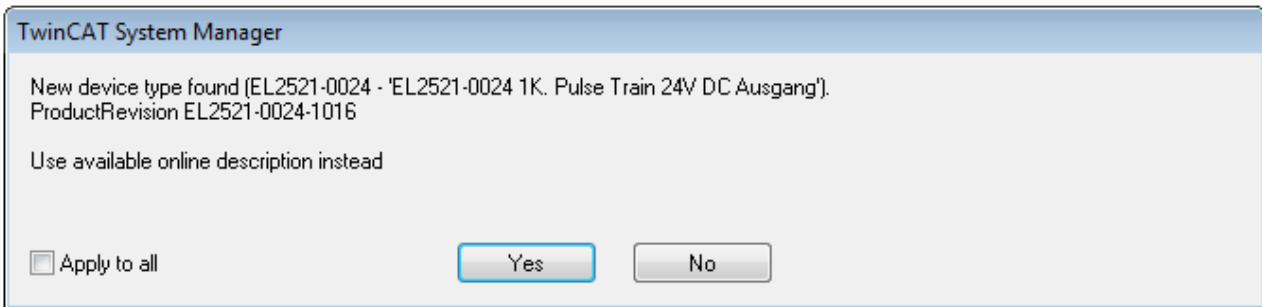


Fig. 83: OnlineDescription information window (TwinCAT 2)

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

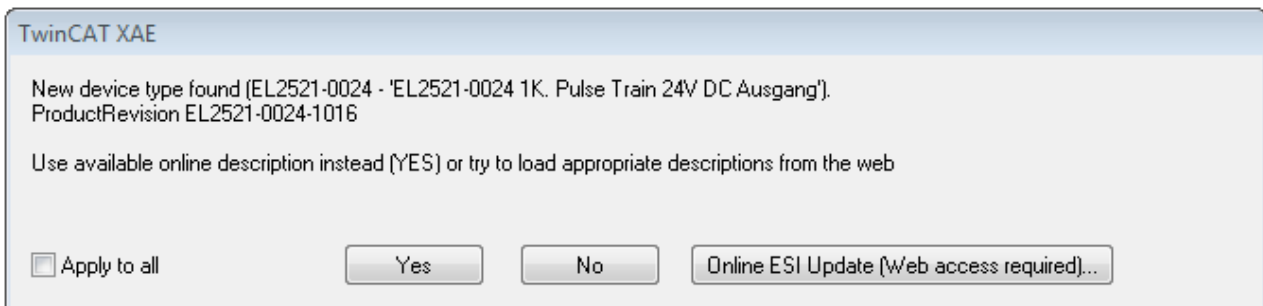


Fig. 84: Information window OnlineDescription (TwinCAT 3)

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

NOTICE

Changing the “usual” configuration through a scan

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

Refer in particular to the chapter “[General notes on the use of Beckhoff EtherCAT IO components](#)” and for manual configuration to the chapter “[Offline configuration creation \[▶ 85\]](#)”.

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

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

OnlineDescriptionCache00000002.xml

Fig. 85: File OnlineDescription.xml created by the System Manager

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

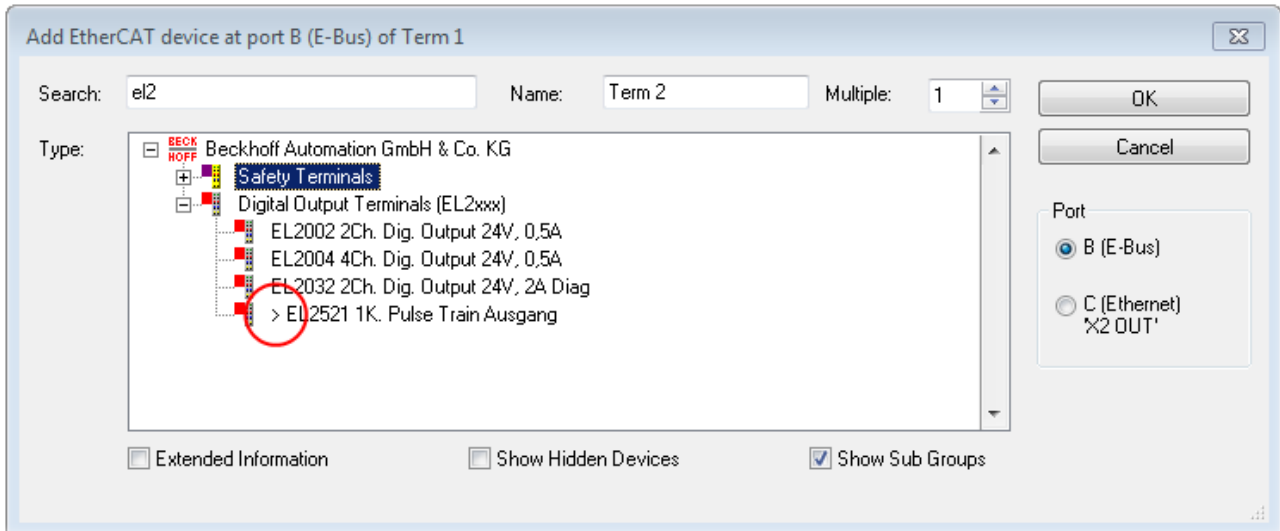


Fig. 86: Indication of an online recorded ESI of EL2521 as an example

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

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

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

i OnlineDescription for TwinCAT 3.x

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

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

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

Faulty ESI file

If an ESI file is faulty and the System Manager is unable to read it, the System Manager brings up an information window.

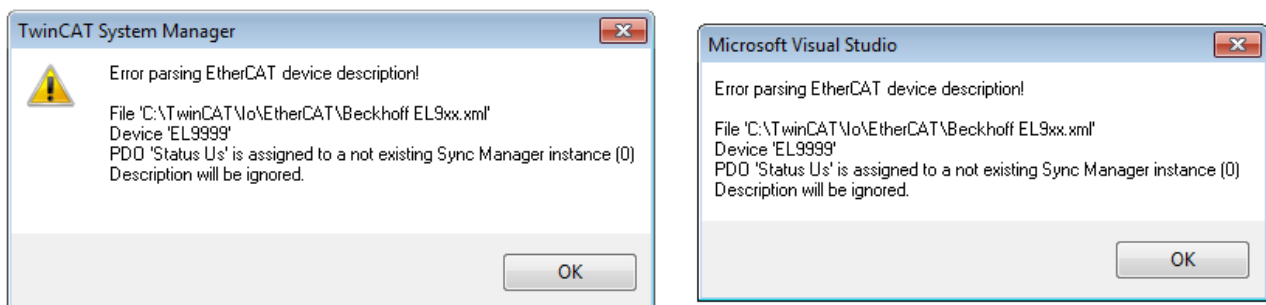


Fig. 87: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

Reasons may include:

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

5.2.3 TwinCAT ESI Updater

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

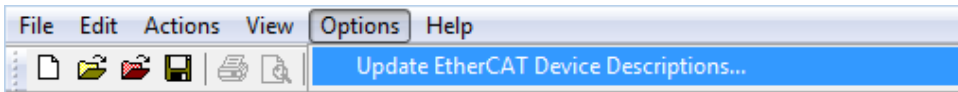


Fig. 88: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:

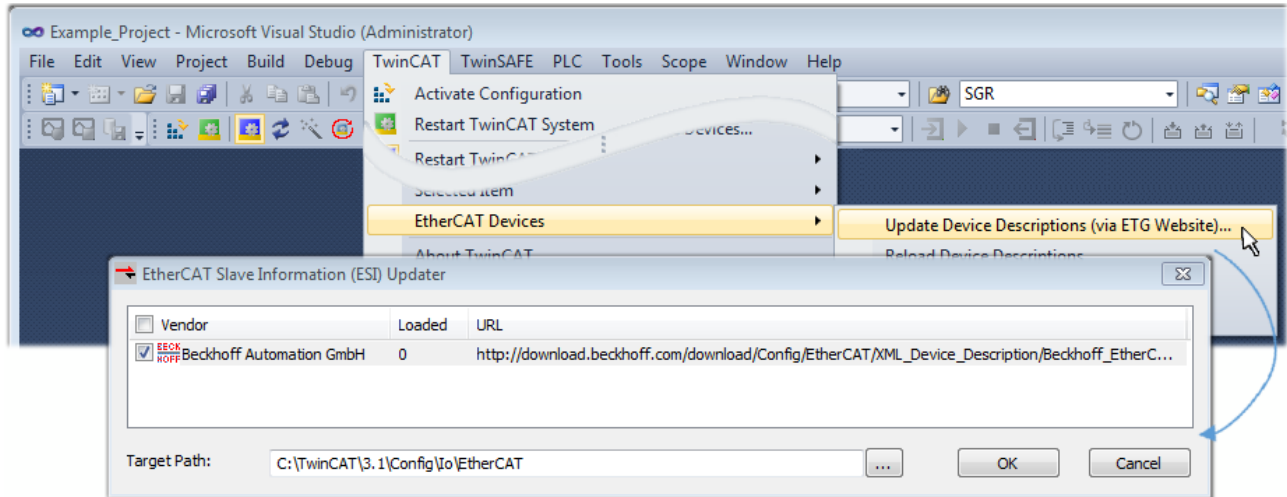


Fig. 89: Using the ESI Updater (TwinCAT 3)

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

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

5.2.4 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

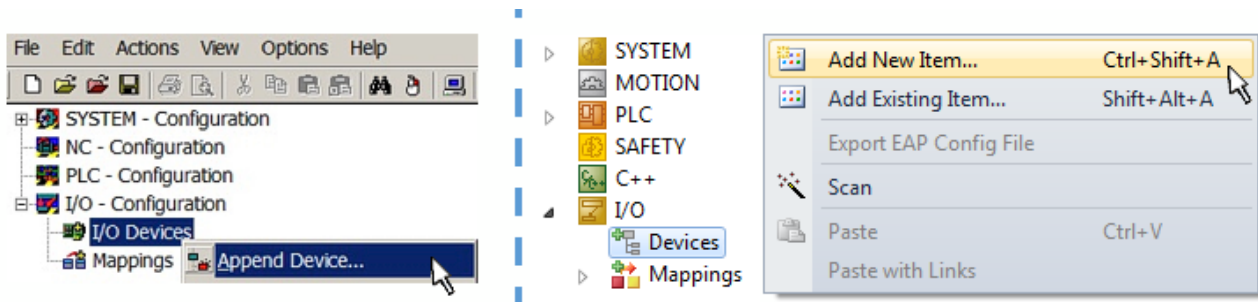


Fig. 90: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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”.

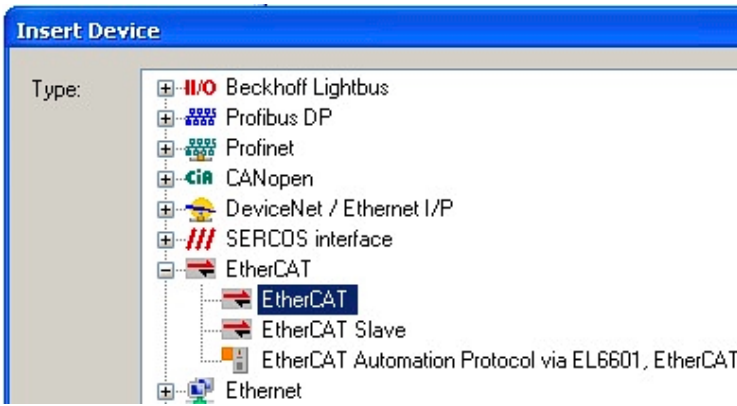


Fig. 91: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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



Fig. 92: 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)”.

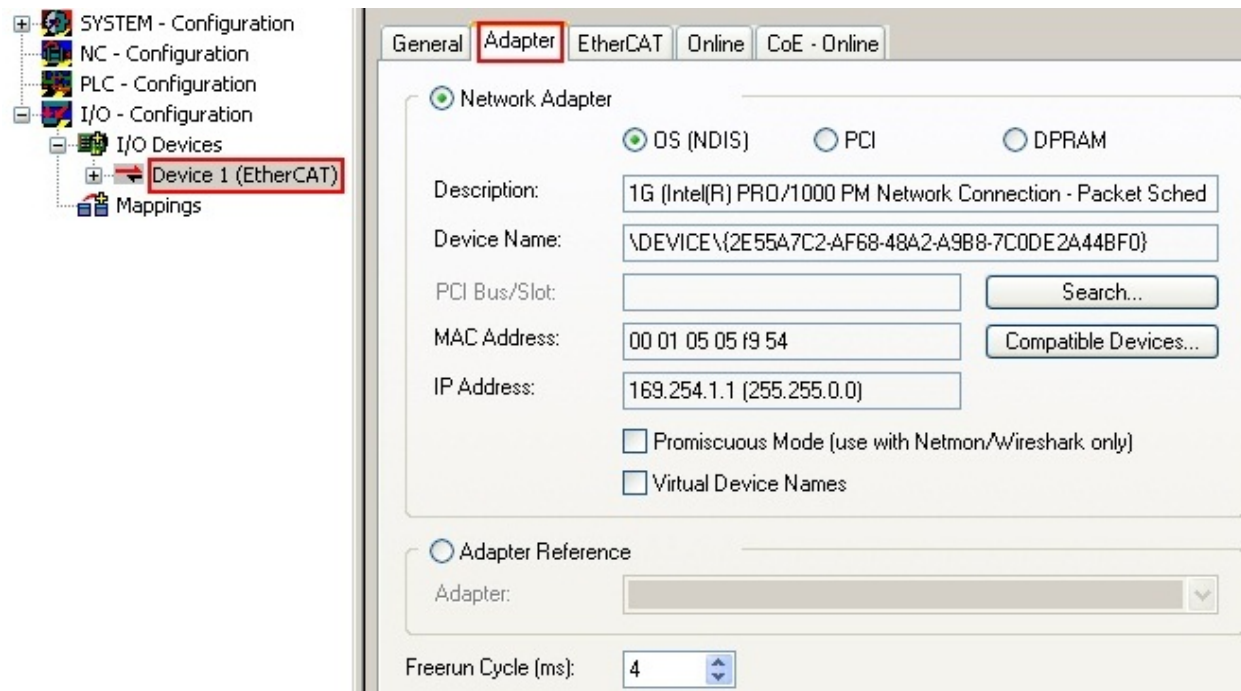
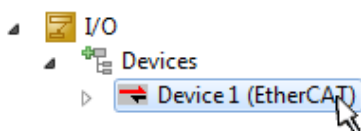


Fig. 93: EtherCAT device properties (TwinCAT 2)

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



i **Selecting the Ethernet port**

Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective [installation page](#) [p. 75].

Defining EtherCAT slaves

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

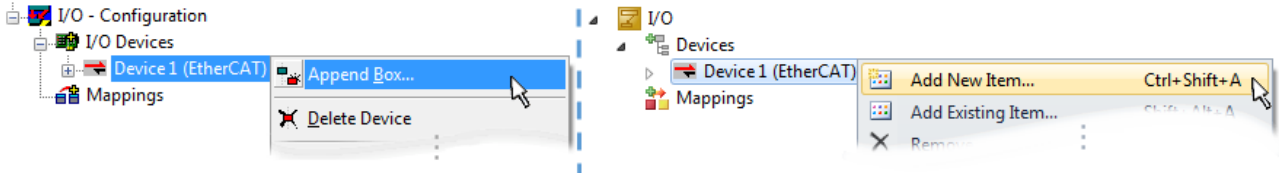


Fig. 94: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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

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

Overview of physical layer

- "Ethernet": cable-based 100BASE-TX: couplers, box modules, devices with RJ45/M8/M12 connector
- "E-Bus": LVDS "terminal bus", EtherCAT plug-in modules (EJ), EtherCAT terminals (EL/ES), various modular modules

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

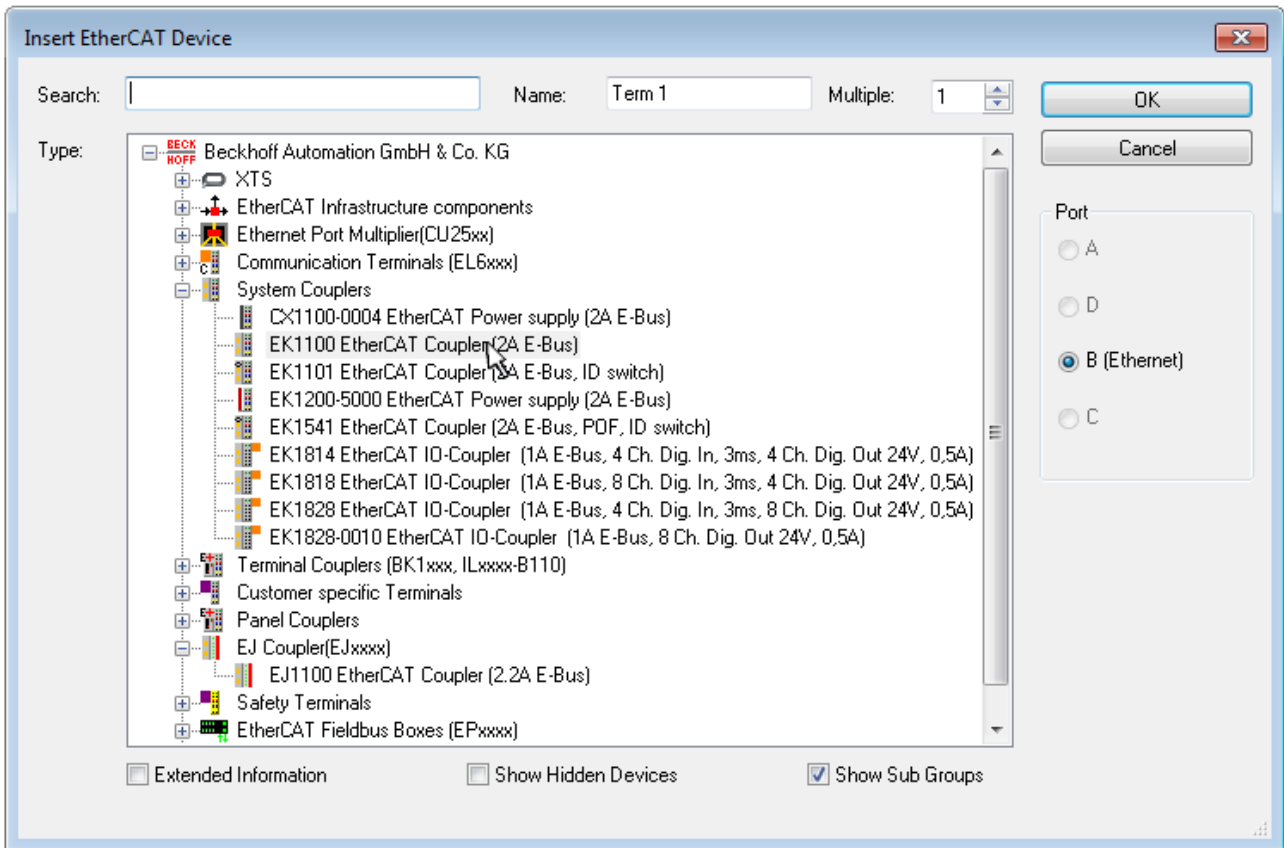


Fig. 95: Selection dialog for new EtherCAT device

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”.

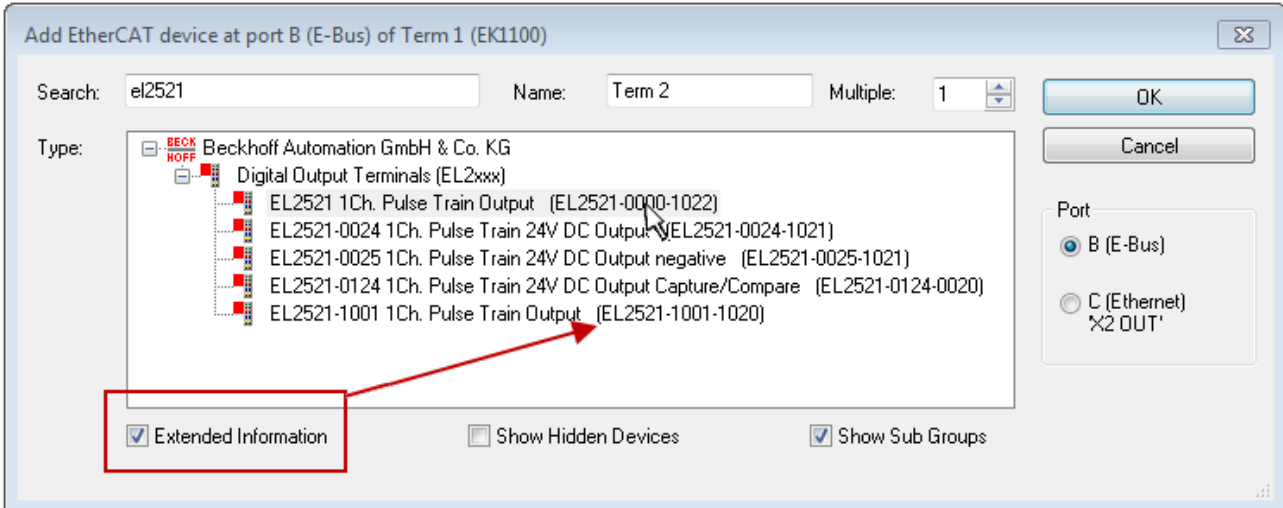


Fig. 96: Display of device revision

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”.

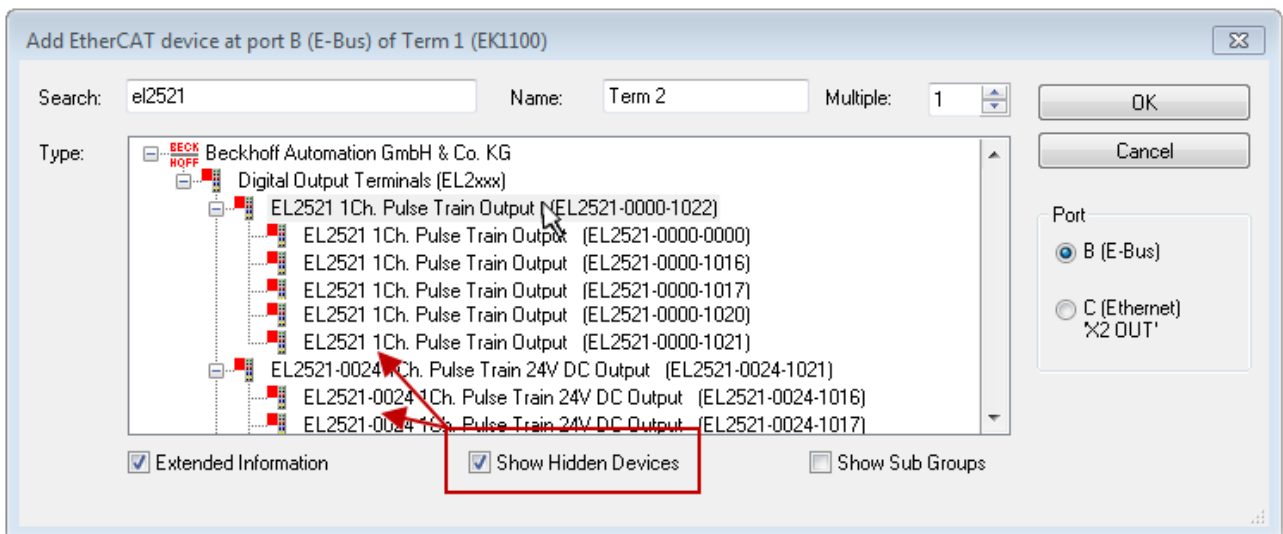


Fig. 97: Display of previous revisions

i Device selection based on revision, compatibility

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

device revision in the system \geq 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.

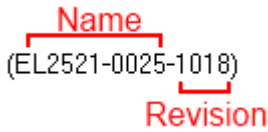


Fig. 98: Name/revision of the terminal

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

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

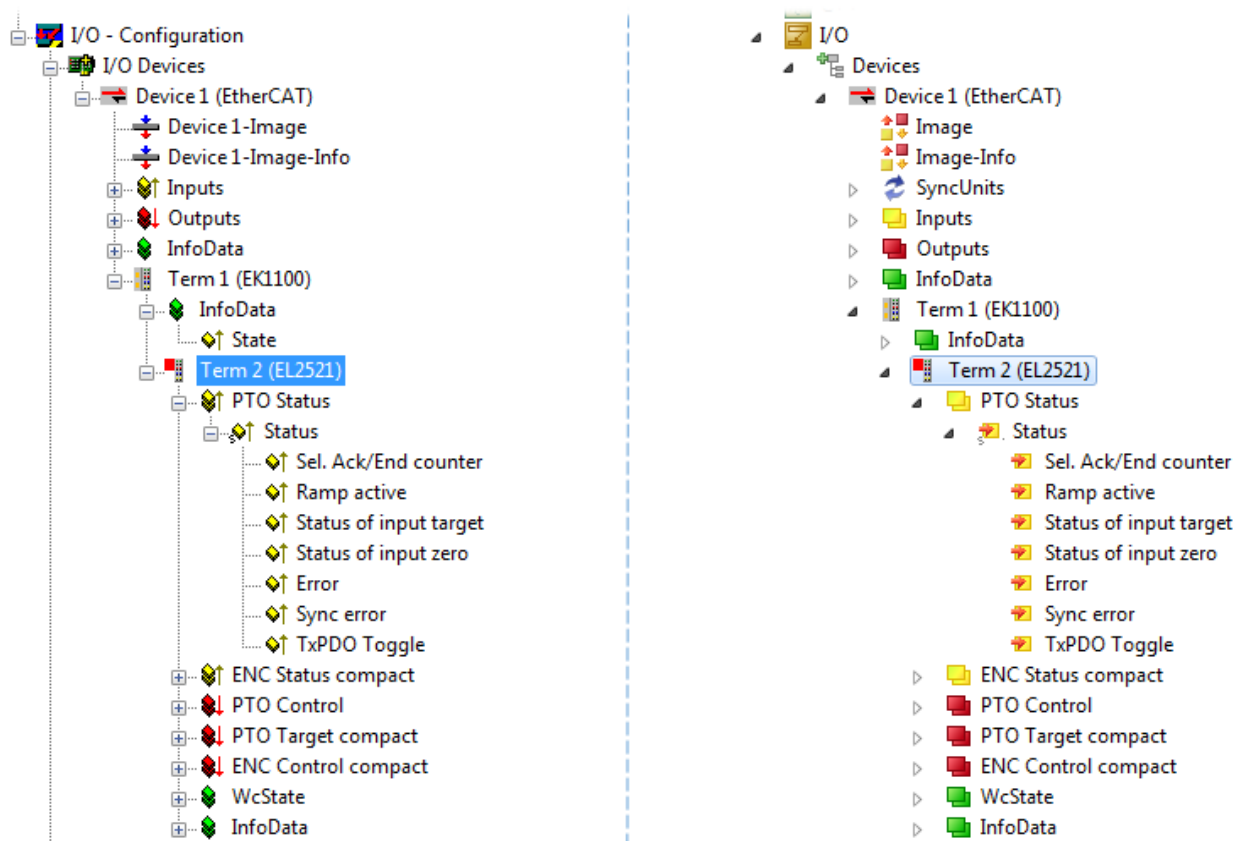




Fig. 99: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)



5.2.5 ONLINE configuration creation

Detecting/scanning of the EtherCAT device

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



- on TwinCAT 2 by a blue display “Config Mode” within the System Manager window:  .
- on TwinCAT 3 within the user interface of the development environment by a symbol  .

TwinCAT can be set into this mode:

- TwinCAT 2: by selection of  in the Menubar or by “Actions” → “Set/Reset TwinCAT to Config Mode...”
- TwinCAT 3: by selection of  in the Menubar or by “TwinCAT” → “Restart TwinCAT (Config Mode)”

● Online scanning in Config mode

i 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.

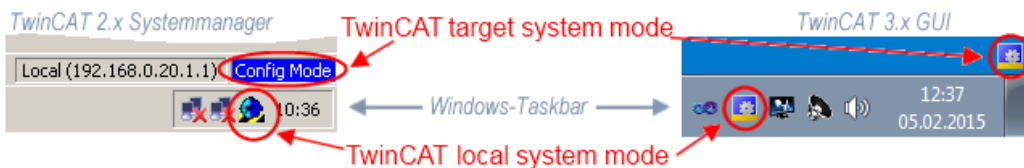


Fig. 100: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

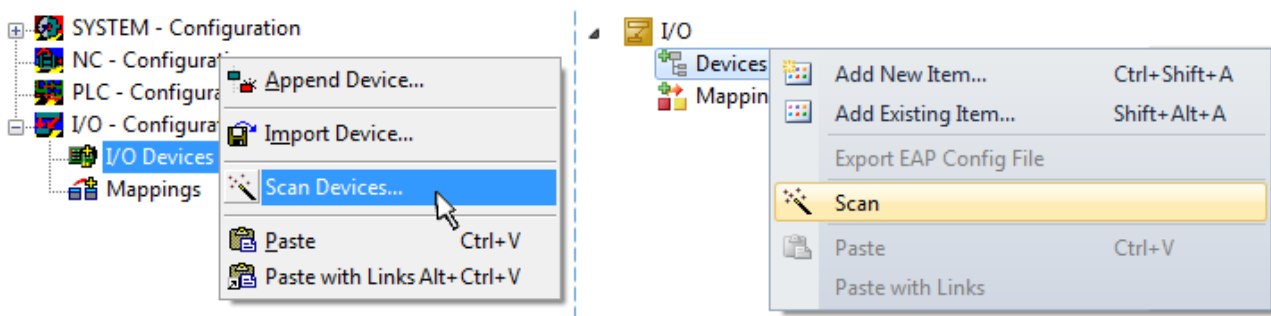


Fig. 101: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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.

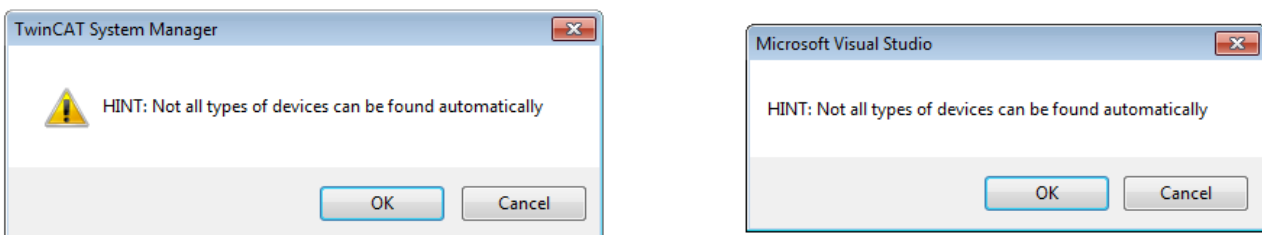


Fig. 102: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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” .

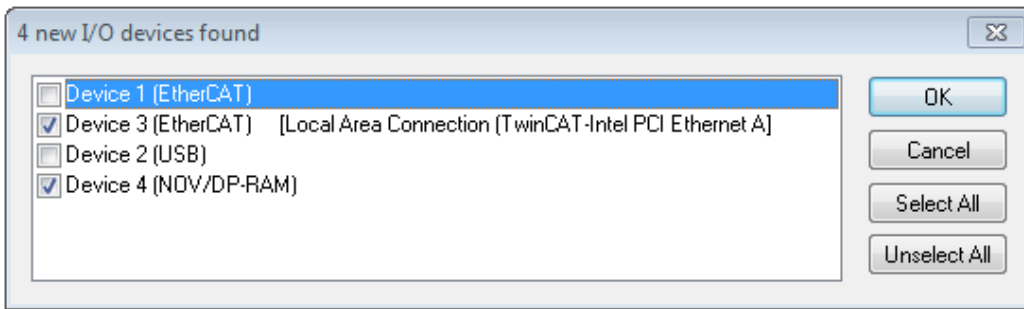


Fig. 103: Detected Ethernet devices

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

● Selecting the Ethernet port



Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective [installation page](#) [▶ 75].

Detecting/Scanning the EtherCAT devices

● Online scan functionality



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

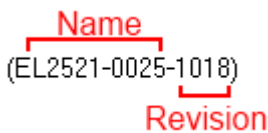


Fig. 104: Example default state

NOTICE

Slave scanning in practice in series machine production

The scanning function should be used with care. It is a practical and fast tool for creating an initial configuration as a basis for commissioning. In series machine production or reproduction of the plant, however, the function should no longer be used for the creation of the configuration, but if necessary for [comparison](#) [▶ 95] 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:

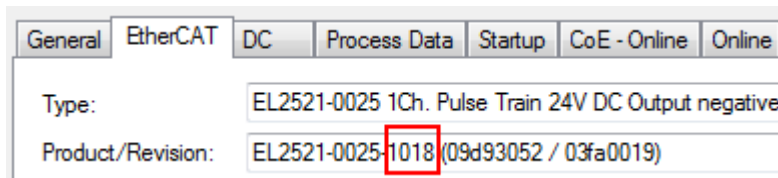


Fig. 105: Installing EtherCAT terminal with revision -1018

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

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

After some time Beckhoff extends the EL2521-0025 by a new feature C. Therefore the FW is changed, outwardly recognizable by a higher FW version and a **new revision -1019**. Nevertheless the new device naturally supports functions and interfaces of the predecessor version(s); an adaptation of “B.tsm” or even “B.pro” is therefore unnecessary. The series-produced machines can continue to be built with “B.tsm” and “B.pro”; it makes sense to perform a comparative scan [► 95] 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:

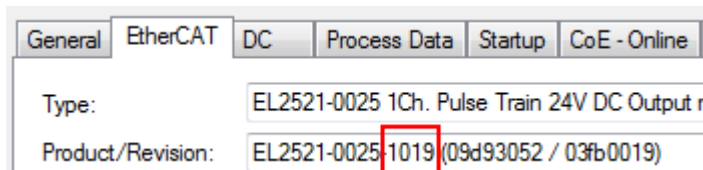


Fig. 106: Detection of EtherCAT terminal with revision -1019

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

In addition, it could be the case that, due to the development accompanying production in company A, the new feature C of the EL2521-0025-1019 (for example, an improved analog filter or an additional process data for the diagnosis) is discovered and used without in-house consultation. The previous stock of spare part devices are then no longer to be used for the new configuration “B2.tsm” created in this way. 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. 107: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

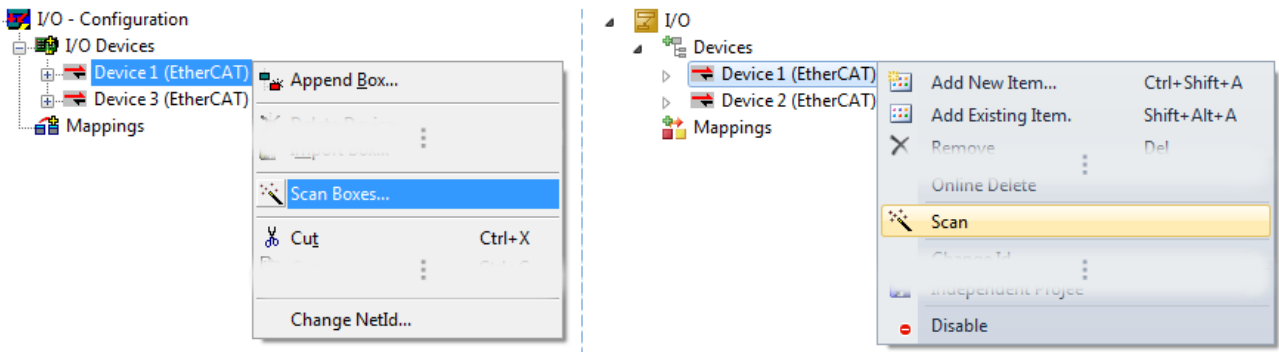


Fig. 108: Manual scanning for devices on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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



Fig. 109: Scan progress exemplary by TwinCAT 2

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



Fig. 110: 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. 111: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar



Fig. 112: 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*.

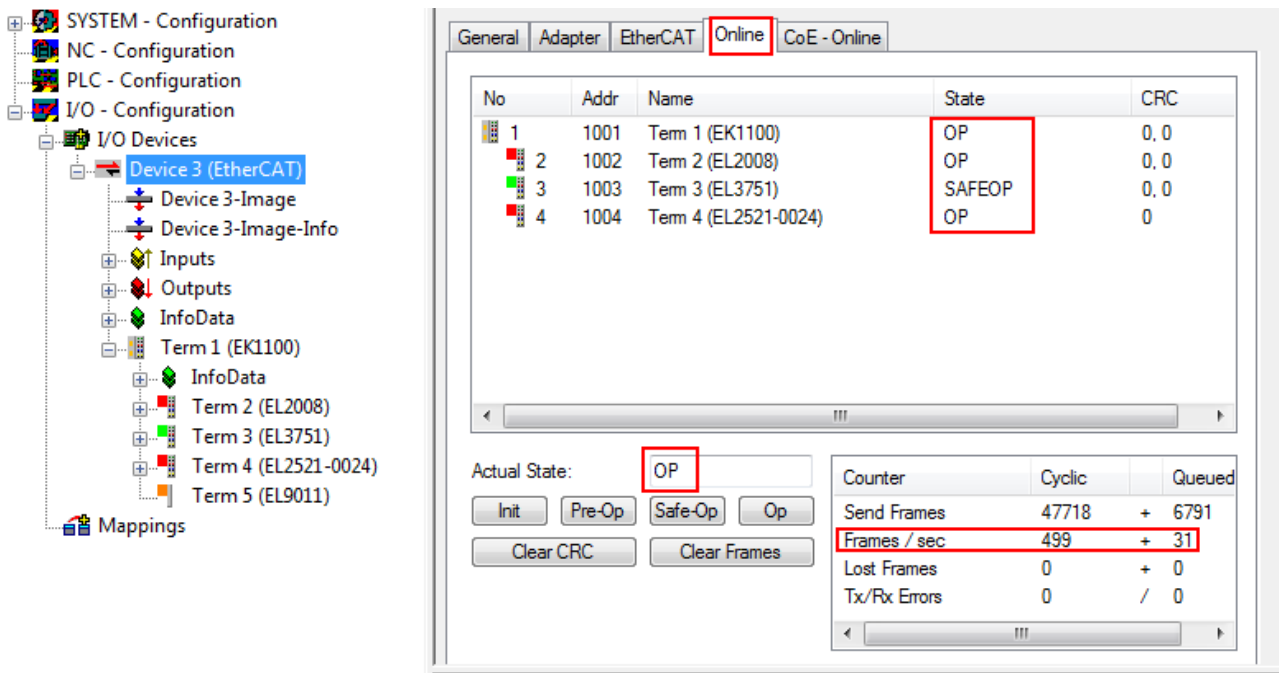


Fig. 113: Online display example

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 \[► 85\]](#).

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.

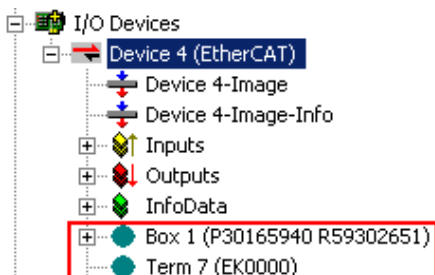


Fig. 114: Faulty identification

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

Scan over existing Configuration

NOTICE

Change of the configuration after comparison

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

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

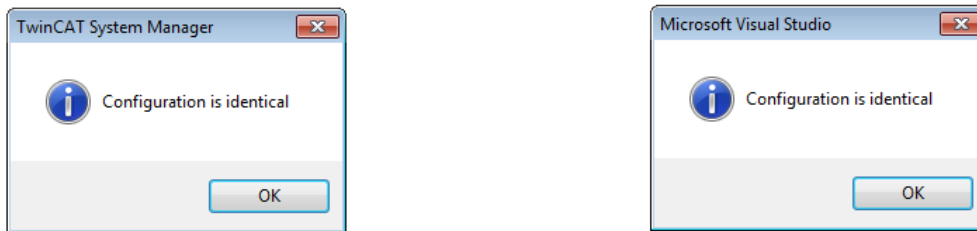


Fig. 115: Identical configuration (left: TwinCAT 2; right: TwinCAT 3)

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

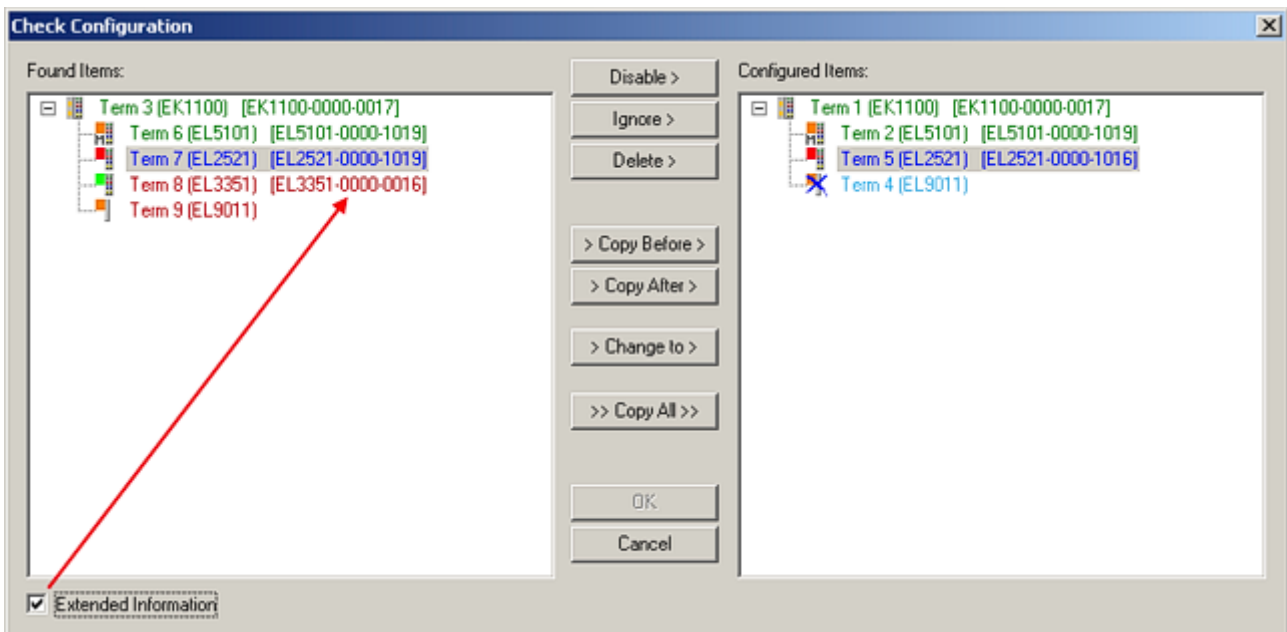


Fig. 116: Correction dialog

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

| Color | Explanation |
|------------|---|
| green | This EtherCAT slave matches the entry on the other side. Both type and revision match. |
| blue | This EtherCAT slave is present on the other side, but in a different revision. This other revision can have other default values for the process data as well as other/additional functions. If the found revision is higher than the configured revision, the slave may be used provided compatibility issues are taken into account. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number. |
| light blue | This EtherCAT slave is ignored ("Ignore" button) |
| red | <ul style="list-style-type: none"> 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. |

i Device selection based on revision, compatibility

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

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

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

Example

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

Name
(EL2521-0025-1018)
Revision

Fig. 117: Name/revision of the terminal

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

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

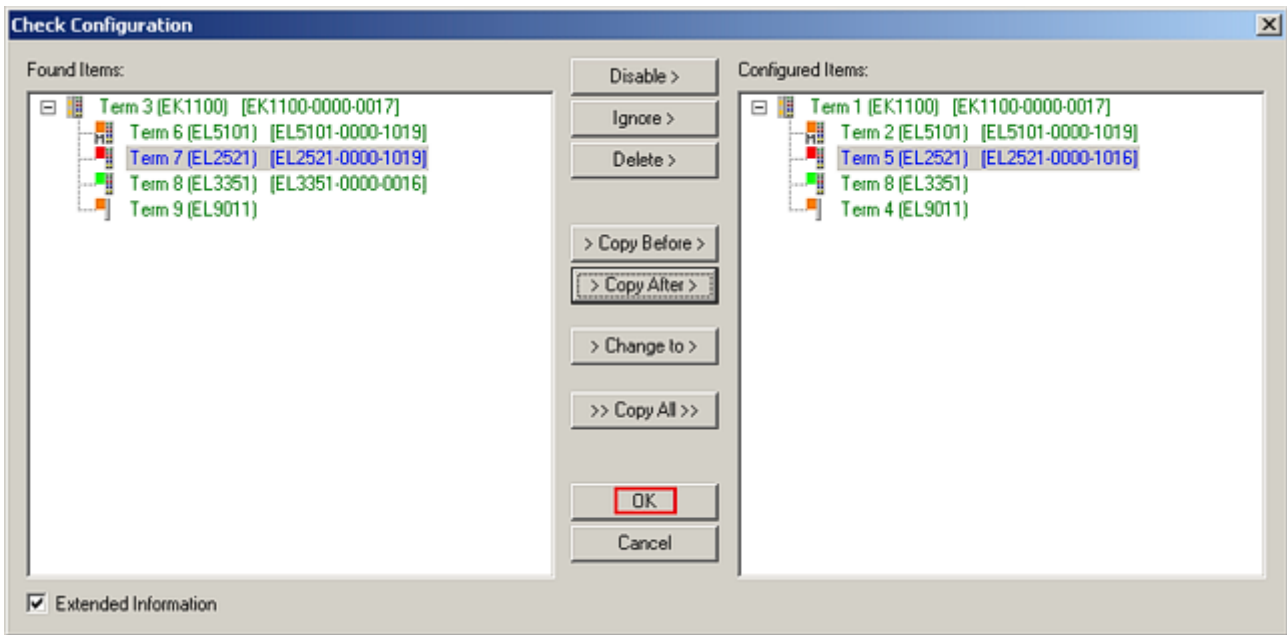


Fig. 118: Correction dialog with modifications

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

Change to Compatible Type

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

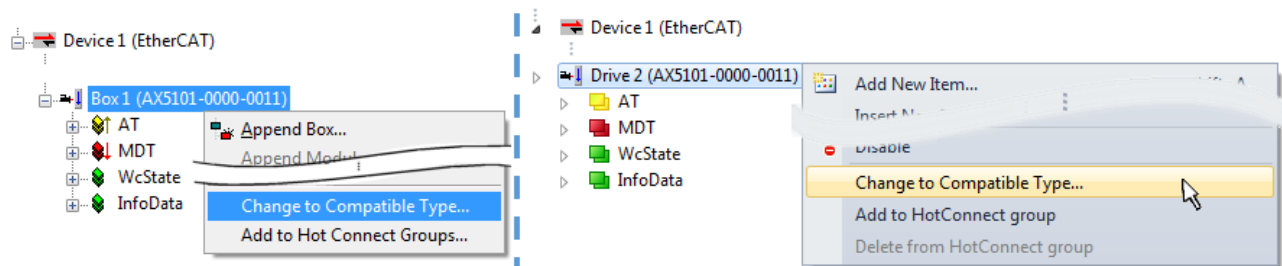


Fig. 119: Dialog “Change to Compatible Type...” (left: TwinCAT 2; right: TwinCAT 3)

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

- Physics (e.g. RJ45, Ebus...)
- FMMU (additional ones are allowed)
- SyncManager (SM, additional ones are allowed)
- EoE (attributes MAC, IP)
- CoE (attributes SdoInfo, PdoAssign, PdoConfig, PdoUpload, CompleteAccess)
- FoE
- PDO (process data: Sequence, SyncUnit SU, SyncManager SM, EntryCount, Entry.Datatype)

This function is preferably to be used on AX5000 devices.

Change to Alternative Type

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

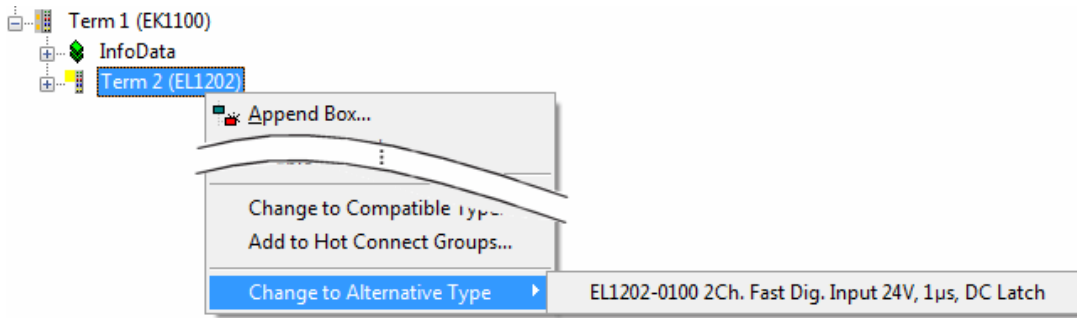


Fig. 120: TwinCAT 2 Dialog Change to Alternative Type

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

5.2.6 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).

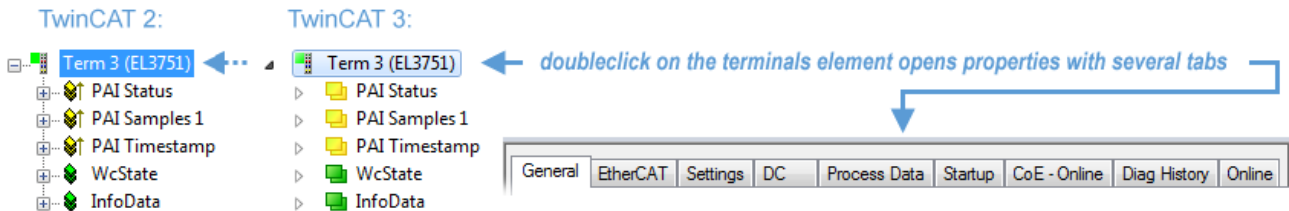


Fig. 121: Branch element as terminal EL3751

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

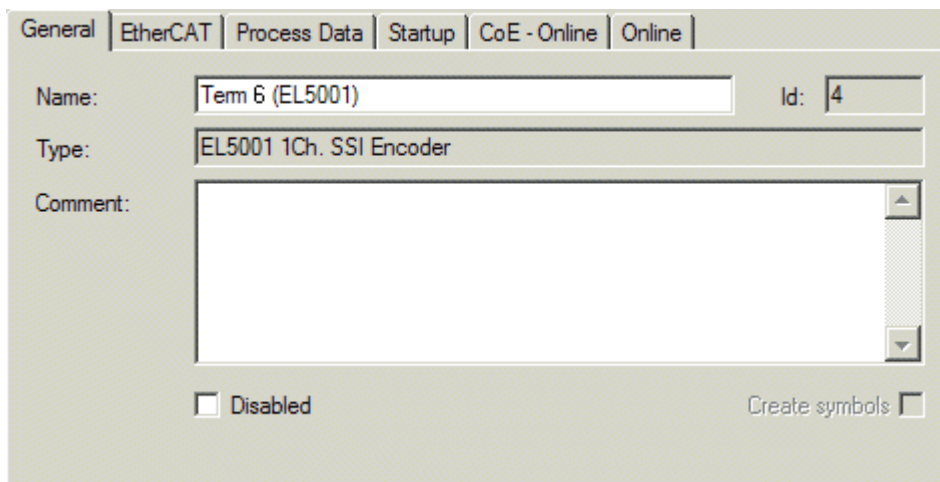


Fig. 122: “General” tab

| | |
|-----------------------|---|
| 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

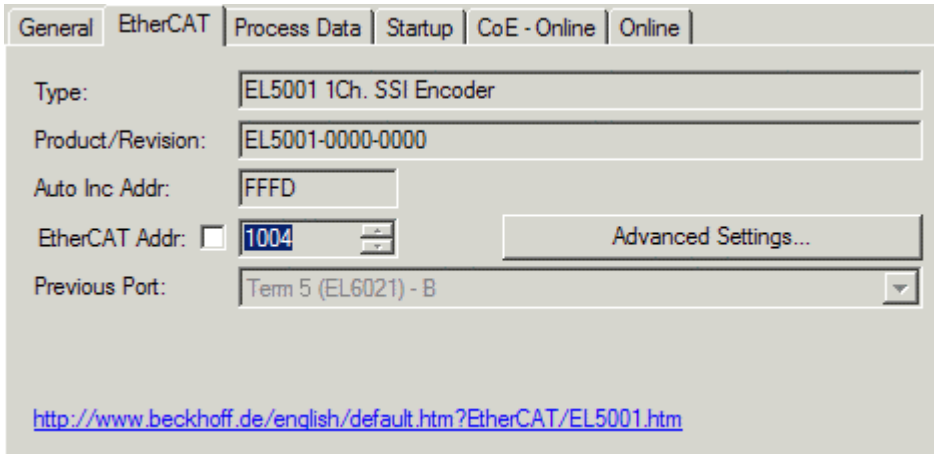


Fig. 123: “EtherCAT” tab

| | |
|--------------------------|---|
| 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 (**Process Data Objects, 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.

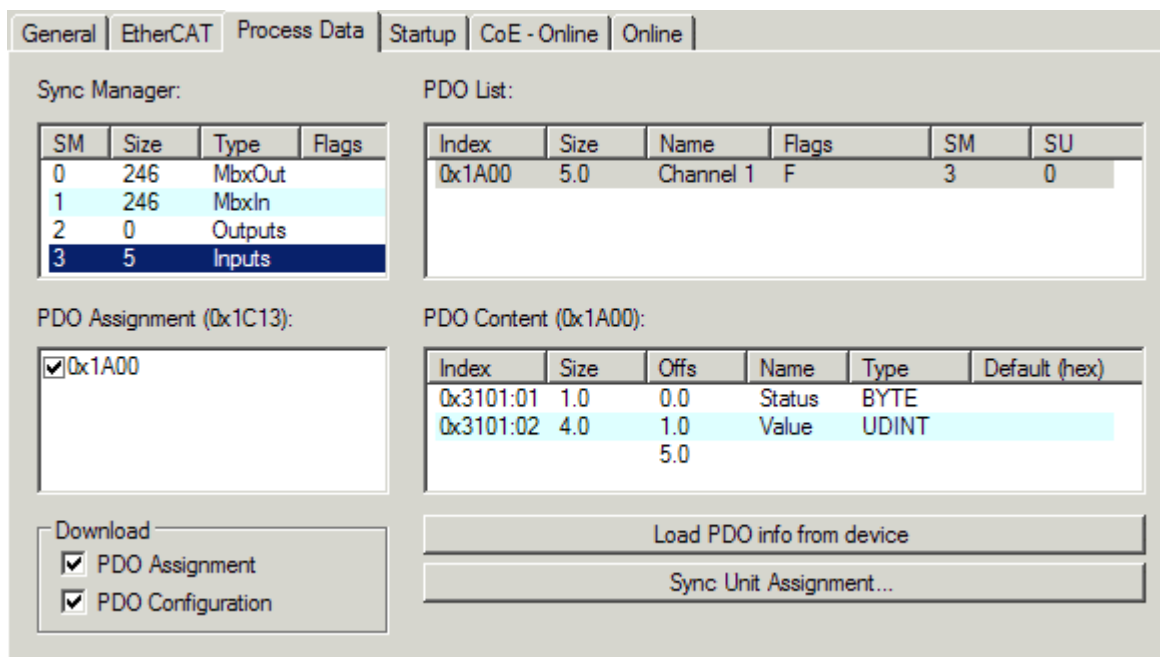


Fig. 124: "Process Data" tab

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").

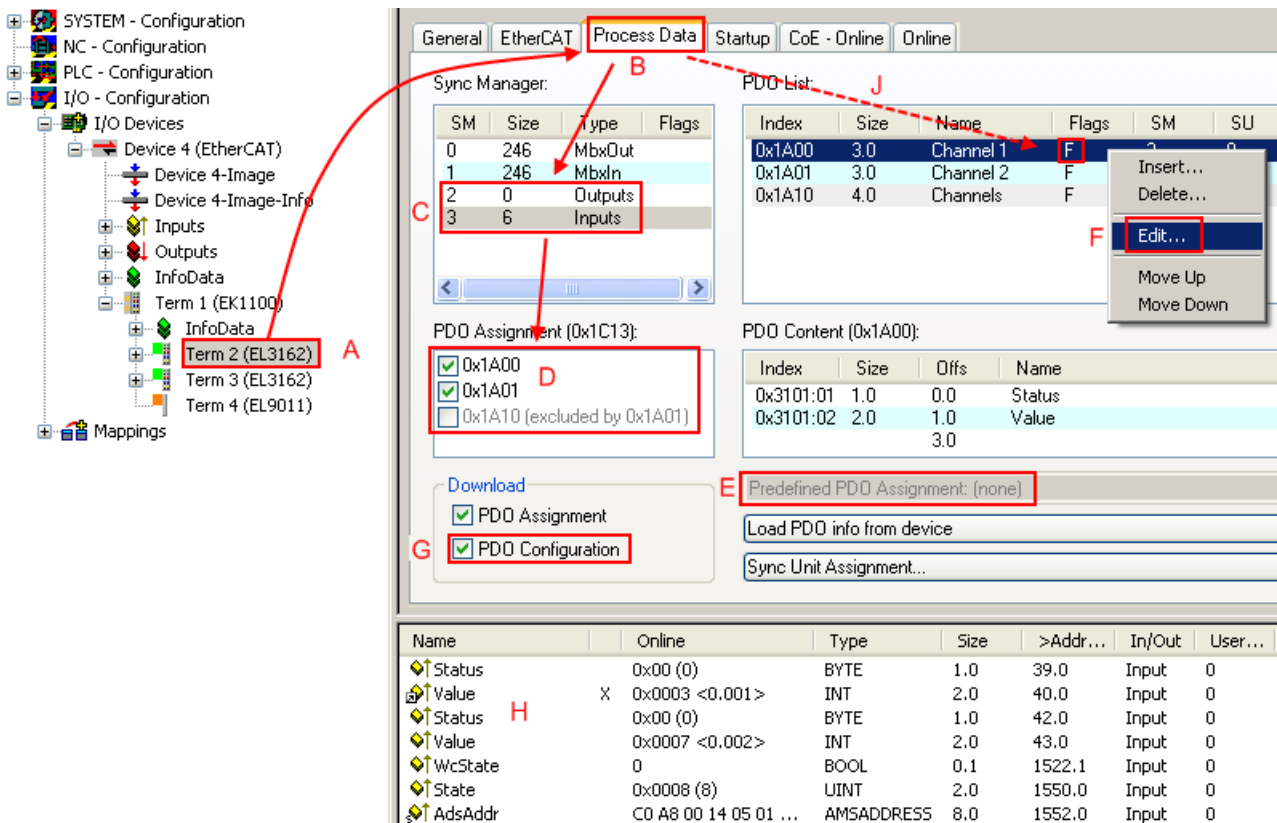


Fig. 125: Configuring the process data

i Manual modification of the process data

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

A detailed description [▶ 106] 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.

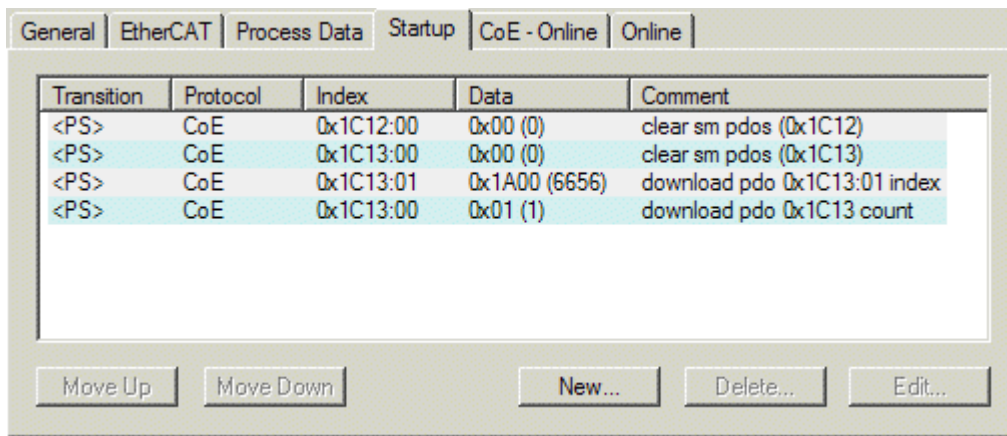


Fig. 126: "Startup" tab

| Column | Description |
|------------|--|
| Transition | Transition to which the request is sent. This can either be <ul style="list-style-type: none"> 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.

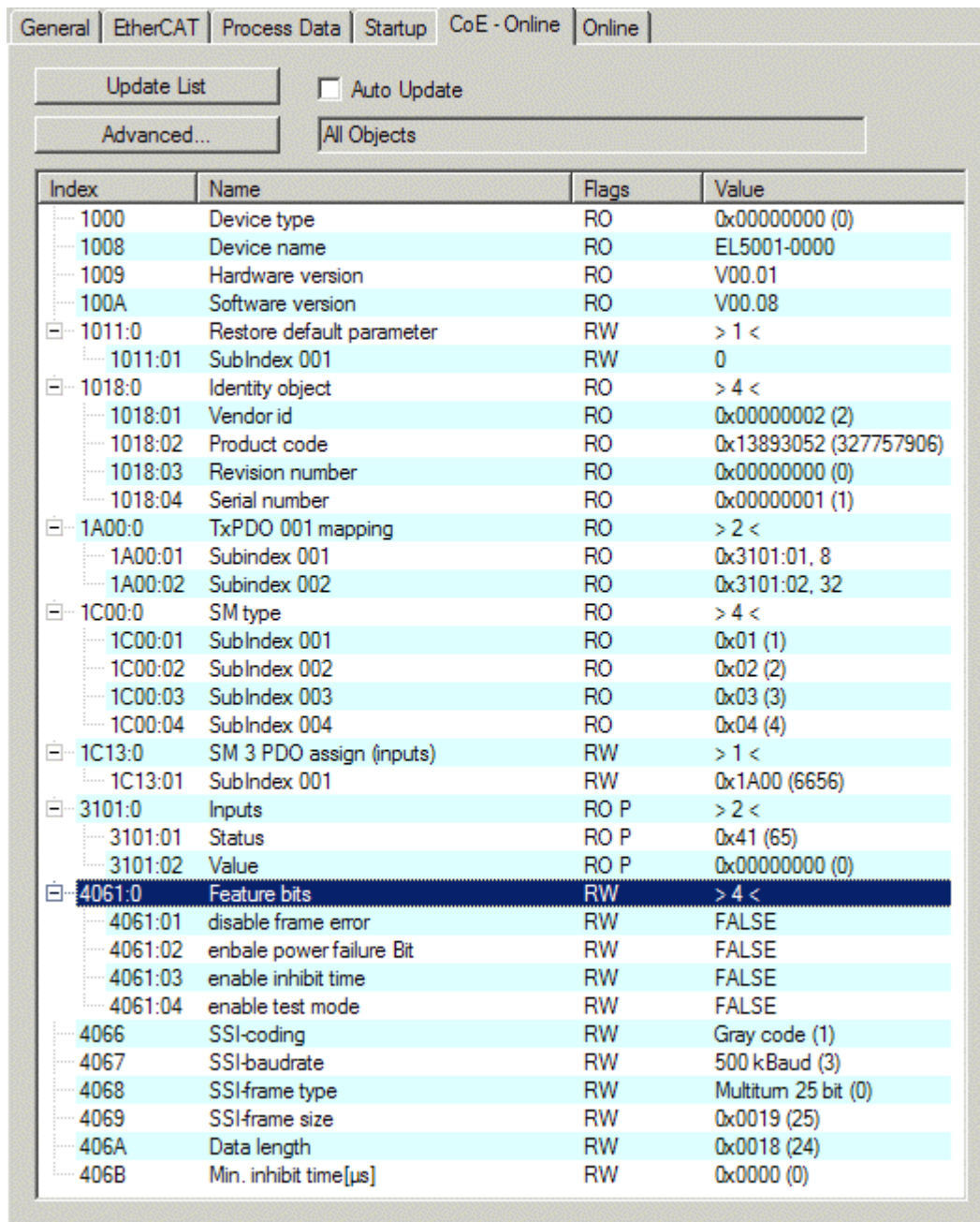


Fig. 127: "CoE - Online" tab

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.

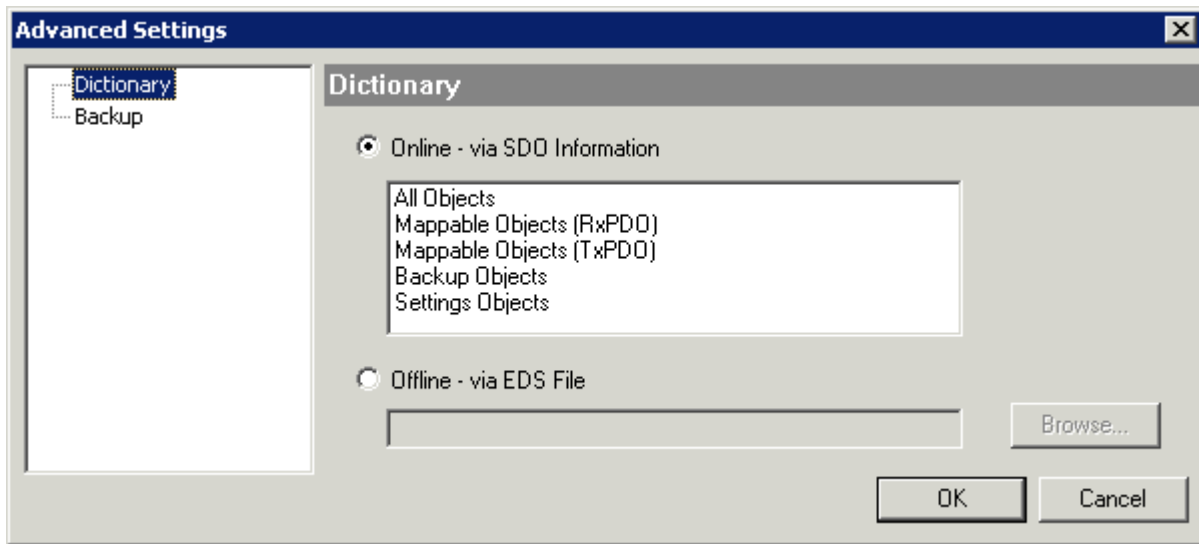


Fig. 128: Dialog “Advanced settings”

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

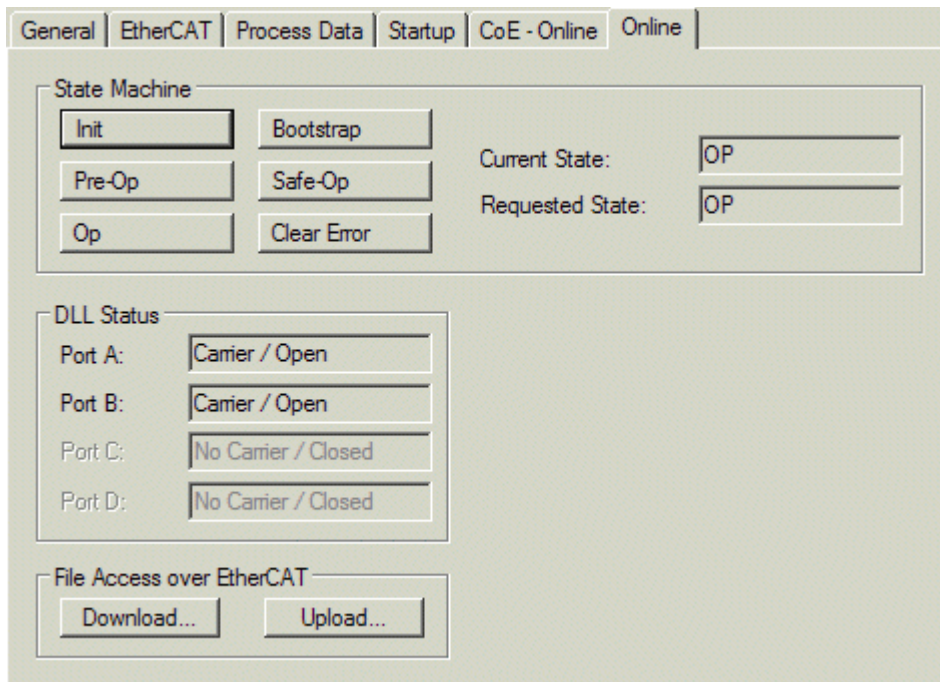


Fig. 129: “Online” tab

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)

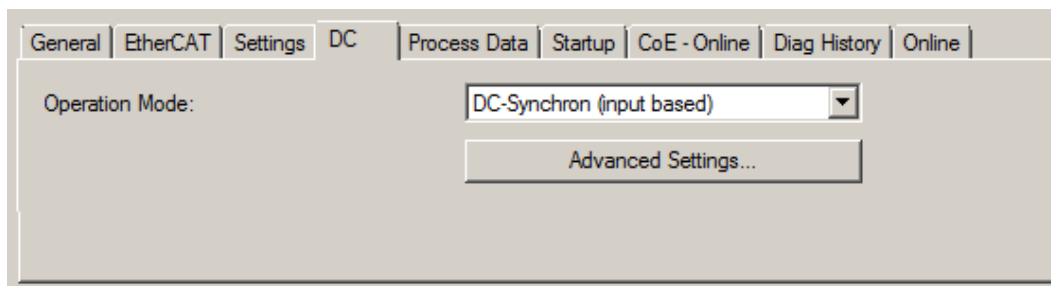


Fig. 130: “DC” tab (Distributed Clocks)

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

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

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

5.2.6.1 Detailed description of Process Data tab

Sync Manager

Lists the configuration of the Sync Manager (SM).

If the EtherCAT device has a mailbox, SM0 is used for the mailbox output (MbxOut) and SM1 for the mailbox input (MbxIn).

SM2 is used for the output process data (outputs) and SM3 (inputs) for the input process data.

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

PDO Assignment



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

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

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

i Activation of PDO assignment

- ✓ If you have changed the PDO assignment, in order to activate the new PDO assignment,
 - a) the EtherCAT slave has to run through the PS status transition cycle (from pre-operational to safe-operational) once (see [Online tab \[▶ 104\]](#)),
 - 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 <i>PDO Assignment</i> list |
| SM | Sync Manager to which this PDO is assigned. If this entry is empty, this PDO does not take part in the process data traffic. | |
| SU | Sync unit to which this PDO is assigned. | |

PDO Content

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

Download

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

PDO Assignment

If this check box is selected, the PDO assignment that is configured in the PDO Assignment list is downloaded to the device on startup. The required commands to be sent to the device can be viewed in the Startup [▶ 101] tab.

PDO Configuration

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

5.3 General Commissioning Instructions for an EtherCAT Slave

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

Diagnosis in real time: WorkingCounter, EtherCAT State and Status

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

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

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

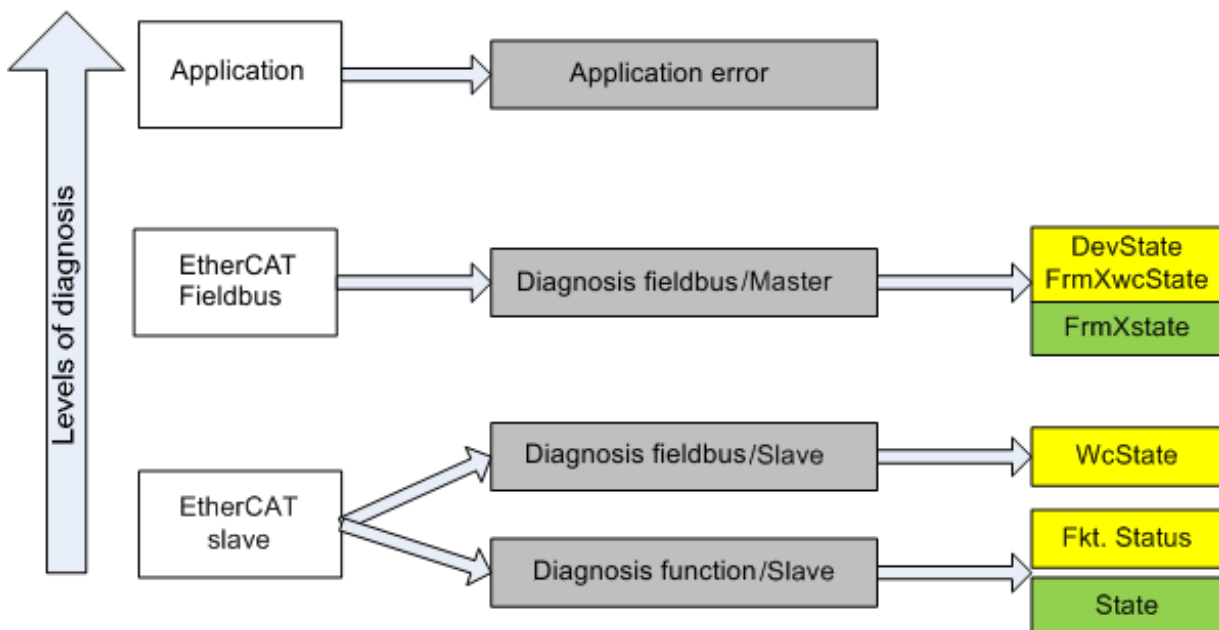


Fig. 131: Selection of the diagnostic information of an EtherCAT Slave

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.

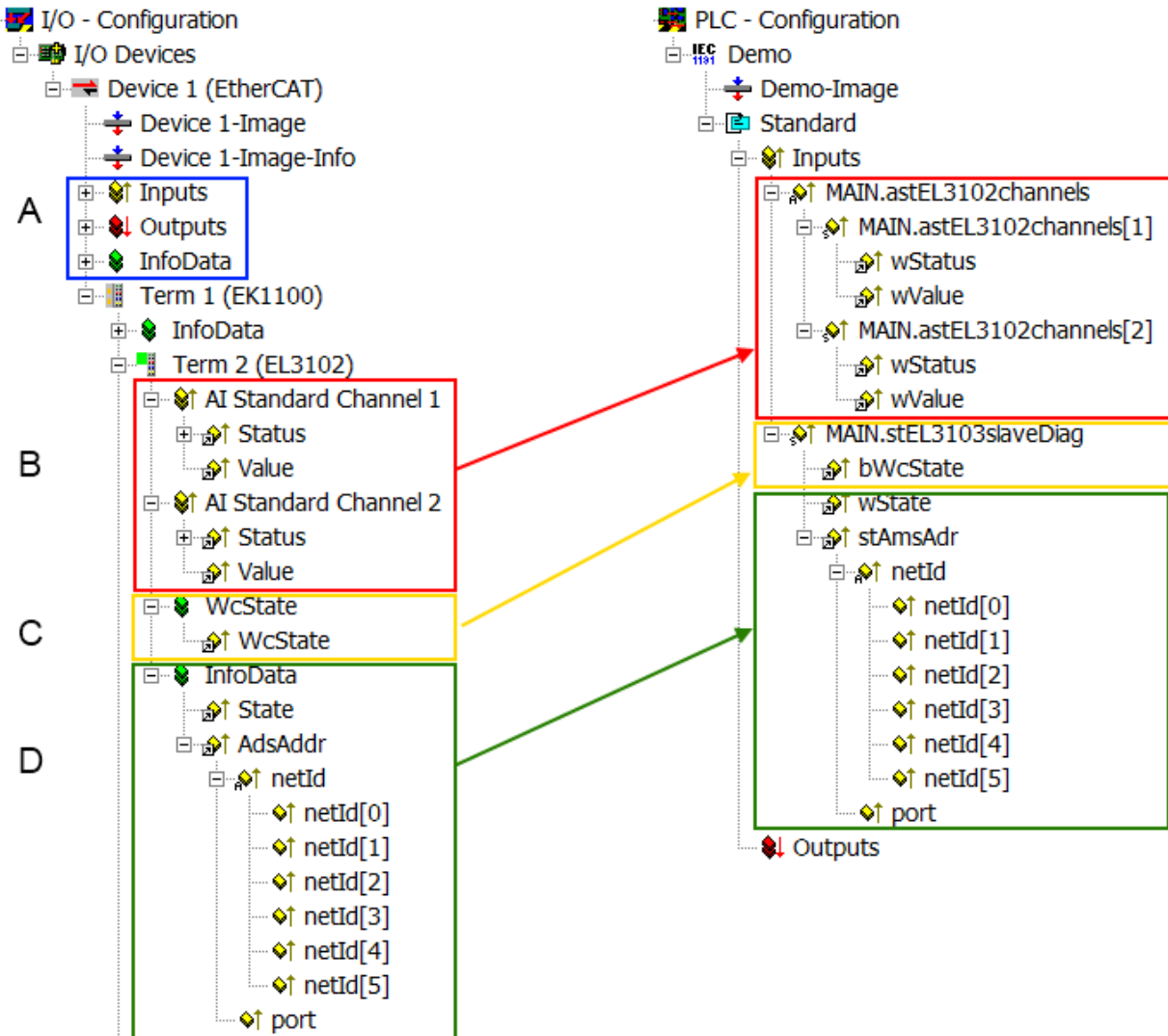


Fig. 132: Basic EtherCAT Slave Diagnosis in the PLC

The following aspects are covered here:

| Code | Function | Implementation | Application/evaluation |
|------|---|---|---|
| A | The EtherCAT Master's diagnostic information updated acyclically (yellow) or provided acyclically (green). | | At least the DevState is to be evaluated for the most recent cycle in the PLC. The EtherCAT Master's diagnostic information offers many more possibilities than are treated in the EtherCAT System Documentation. A few keywords: <ul style="list-style-type: none"> • CoE in the Master for communication with/through the Slaves • Functions from <i>TcEtherCAT.lib</i> • 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 <ul style="list-style-type: none"> • 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 <ol style="list-style-type: none"> 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 <ul style="list-style-type: none"> • 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. <i>AdsAddr</i> The ADS address is useful for communicating from the PLC/task via ADS with the EtherCAT Slave, e.g. for reading/writing to the CoE. The AMS-NetID of a slave corresponds to the AMS-NetID of the EtherCAT Master; communication with the individual Slave is possible via the <i>port</i> (= EtherCAT address). | Information variables for the EtherCAT Master that are updated acyclically. This means that it is possible that in any particular cycle they do not represent the latest possible status. It is therefore possible to read such variables through ADS. |

NOTICE

Diagnostic information

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

CoE Parameter Directory

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

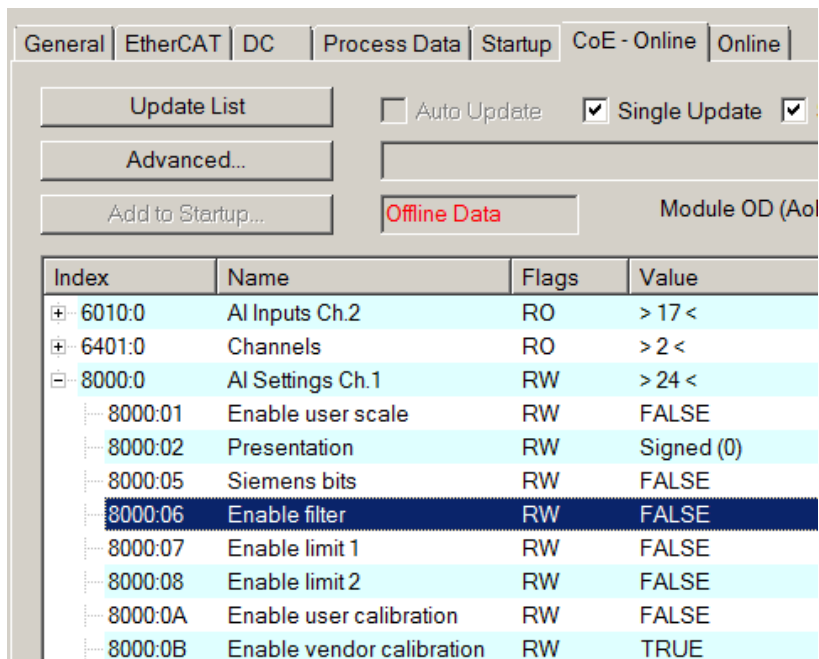


Fig. 133: EL3102, CoE directory

i 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.

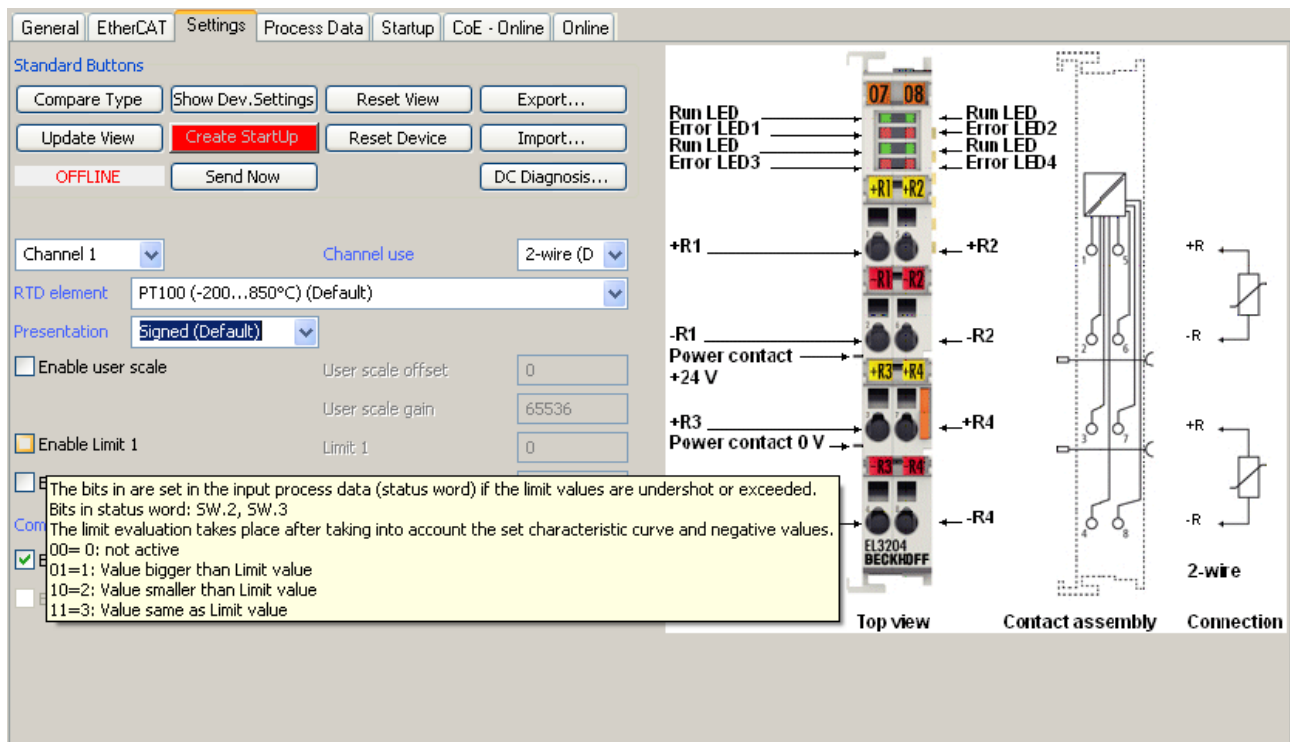


Fig. 134: Example of commissioning aid for a EL3204

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 \[► 40\]](#)" 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.

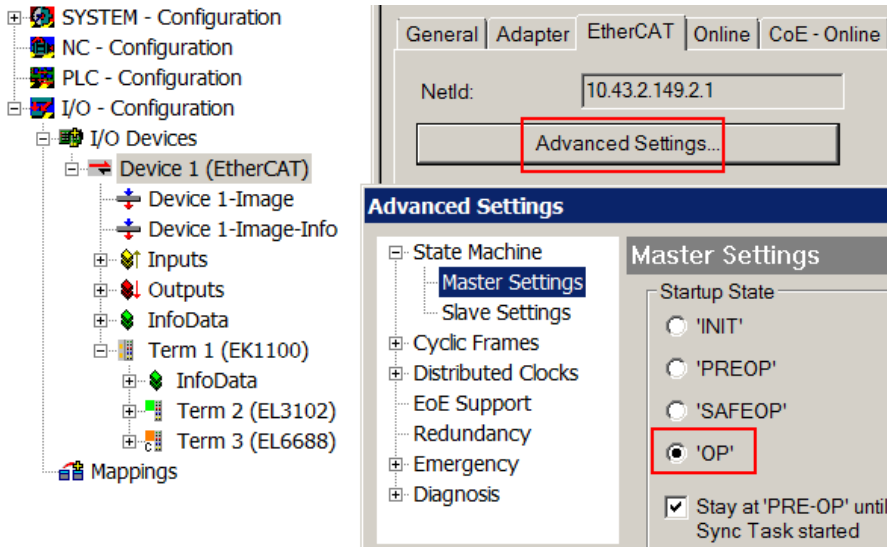


Fig. 135: Default behaviour of the System Manager

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

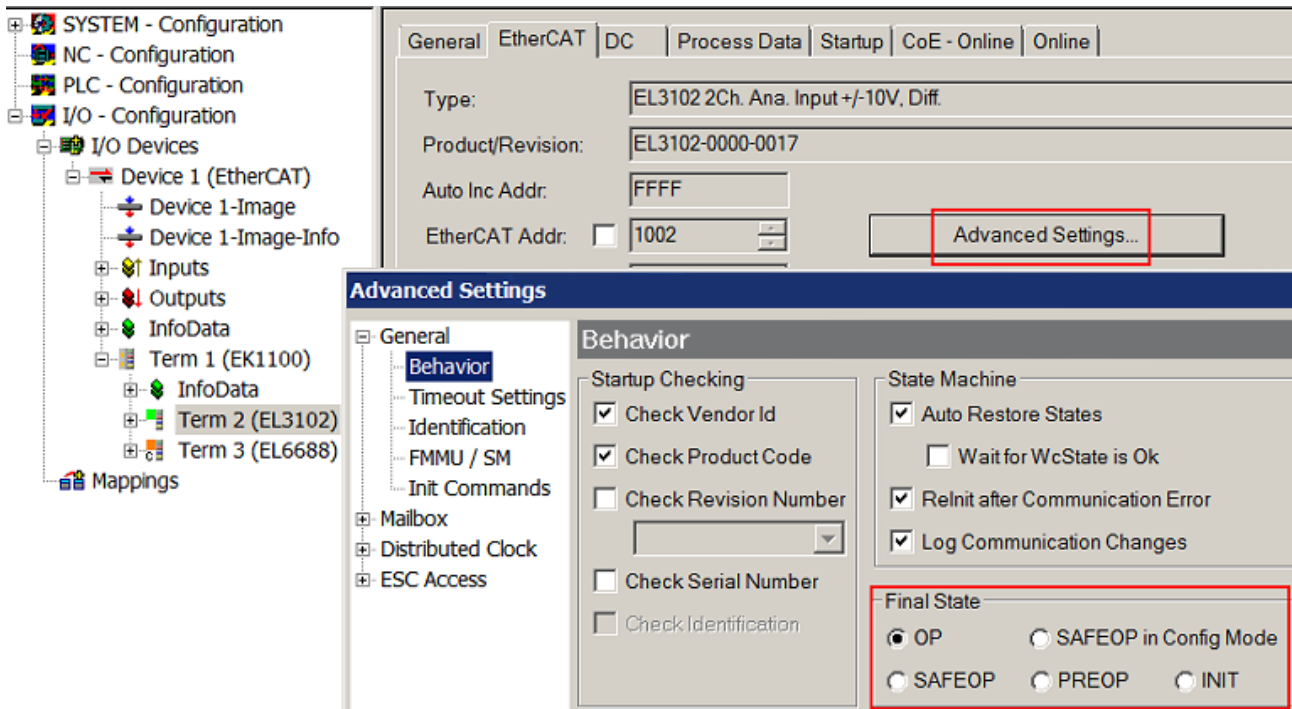


Fig. 136: Default target state in the Slave

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.

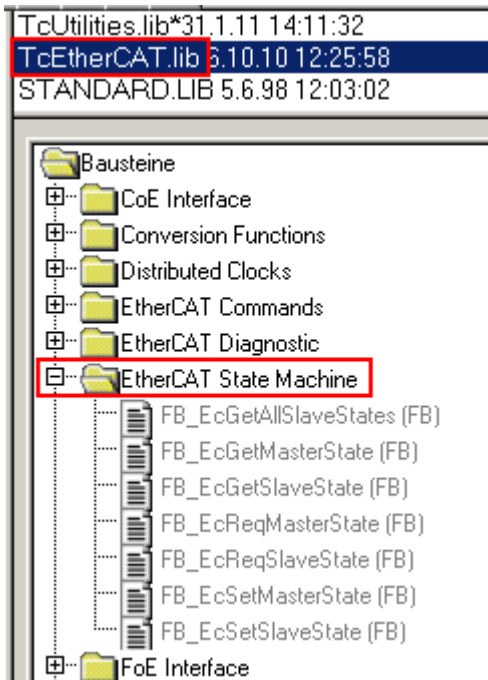


Fig. 137: PLC function blocks

Note regarding E-Bus current

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

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

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

Fig. 138: 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:

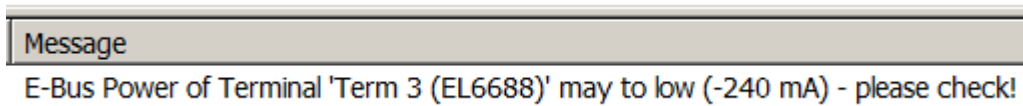


Fig. 139: Warning message for exceeding E-Bus current

NOTICE

Caution! Malfunction possible!

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

5.4 Object description and parameterization

i EtherCAT XML Device Description

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

i Parameterization via the CoE list (CAN over EtherCAT)

The EtherCAT device is parameterized via the CoE-Online tab [► 102] (double-click on the respective object) or via the Process Data tab [► 99](allocation of PDOs). Please note the following general CoE notes [► 43] when using/manipulating the CoE parameters:

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

5.4.1 Standard objects

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 (5002). | UINT32 | RO | 0x0000138A (5002 _{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 | EL6740-0010-0000 |

Index 1009 Hardware version

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

Index 100A Software version

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

Index 1010 Store Interbus Configuration (from FW 02)

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|------------------|---|-----------|-------|--------------------------------|
| 1010:0 | Store Parameters | Saving the current Interbus configuration | UINT8 | RW | 0x01 (1 _{dec}) |
| 1010:01 | Codeword | The current Interbus configuration from object <u>0xF800:02</u> [► 118] is stored in the EL6740-0010 when 0x65766173 ("evas") is written to index 1010:01. To this end the terminal must be in OP or SAFE-OP state. Once these settings have been saved, other Interbus configurations are rejected with abort code 0x0026 during transition to SAFE-OP. | UINT32 | RW | 0x00000000 (0 _{dec}) |

Index 1011 Restore Interbus Configuration (ab FW 02)

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|--------------------|---|-----------|-------|--------------------------------|
| 1011:0 | Restore Parameters | Resetting the stored Interbus configuration | UINT8 | RW | 0x01 (1 _{dec}) |
| 1011:01 | Codeword | The Interbus configuration stored in the EL6740-0010 is deleted when 0x64616F6C ("daol") is written to index 0x1011:01. To this end the terminal must be in OP or SAFE-TOP 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 _{dez}) |
| 1018:01 | Vendor ID | Vendor ID of the EtherCAT slave | UINT32 | RO | 0x00000002 (2 _{dez}) |
| 1018:02 | Product code | Product code of the EtherCAT slave | UINT32 | RO | 0x1A543052 (441725010 _{dez}) |
| 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 | 0x0000000A (10 _{dez}) |
| 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 _{dez}) |

Index 1400 RxPDO parameter

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-----------------|---|-----------|-------|--------------------------|
| 1400:0 | RxPDO parameter | Max. Subindex | UINT8 | RO P | 0x07 (7 _{dez}) |
| 1400:07 | Status | Status Rx PDO [► 18] TRUE: SUPI station Error; No data have been sent to the Interbus Value: 0x2000 (8192 _{dez}) | BOOL | RO P | TRUE |

Index 1600 RxPDO Mapping Box 001

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-------------------------|-------------------------|-----------|-------|--------------------------|
| 1600:0 | RxPDO Mapping Box 001 | Max. Subindex | UINT8 | RW | 0x03 (3 _{dez}) |
| 1600:01 | Output Mapping Area 001 | PDO mapping range 001** | UINT32 | RW | 0x7000:01, nnn** |
| ... | ... | ... | ... | . | . |
| 1600:03 | Output Mapping Area 003 | PDO mapping range 003** | UINT32 | RW | 0x0000:00, nnn** |

**dynamically generated object, size depends on process variable

Index 1601 PDO control

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-------------|---------------|-----------|-------|--------------------------|
| 1601:0 | PDO control | Max. Subindex | UINT8 | RW | 0x01 (1 _{dec}) |
| 1601:01 | | Dummy | UINT32 | RW | 0x0000:00, 16 |

Index 1800 TxPDO parameter

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-----------------|--|-----------|-------|--------------------------|
| 1800:0 | TxPDO parameter | Max. Subindex | UINT8 | RO P | 0x07 (7 _{dec}) |
| 1800:07 | Status | Status Tx PDO [► 18] TRUE: SUPI invalid data received from Interbus Invalid data receives from the Interbus Value: 0x4000 (16384 _{dec}) | BOOL | RO P | TRUE |

Index 1A00 TxPDO Mapping Box 001

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|------------------------|------------------------|-----------|-------|--------------------------|
| 1A00:0 | TxPDO Mapping Box 001 | Max. Subindex | UINT8 | RW | 0x03 (3 _{dec}) |
| 1A00:01 | Input Mapping Area 001 | PDO mapping area 001** | UINT32 | RW | 0x6000:01, nnn** |
| ... | ... | ... | ... | . | . |
| 1A00:03 | Input Mapping Area 003 | PDO mapping area 003** | UINT32 | RW | 0x0000:00, nnn** |

**dynamically generated object, size depends on process variable

Index 1A01 PDO status

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|---------------|-----------------|-----------|-------|--------------------------|
| 1A01:0 | PDO status | Max. Subindex | UINT8 | RW | 0x04 (4 _{dec}) |
| 1A01:01 | [Alignment] | Alignment dummy | UINT32 | RW | 0x0000:00, 13 |
| 1A01:02 | Rx PDO status | Status Rx PDO | UINT32 | RW | 0x1400:07, 1 [►_116] |
| 1A01:03 | Tx PDO status | Status Tx PDO | UINT32 | RW | 0x1800:07, 1 [►_116] |
| 1A01:04 | [Alignment] | Alignment-Dummy | UINT32 | RW | 0x0000:00, 1 |

Index 1C00 Sync manager type

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-------------------|--|-----------|-------|--------------------------|
| 1C00:0 | Sync manager type | Using the sync managers | UINT8 | RO | 0x02 (2 _{dec}) |
| 1C00:01 | SubIndex 001 | Sync-Manager Type Channel 1: Mailbox Write | UINT8 | RO | 0x01 (1 _{dez}) |
| 1C00:02 | SubIndex 002 | Sync-Manager Type Channel 2: Mailbox Read | UINT8 | RO | 0x02 (2 _{dec}) |

Index 1C12 RxPDO assign

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|--------------|--|-----------|-------|-------------------------------|
| 1C12:0 | RxPDO assign | PDO Assign Outputs | UINT8 | RW | 0x02 (2 _{dec}) |
| 1C12:01 | SubIndex 001 | 1. allocated RxPDO (contains the index of the associated RxPDO mapping object) | UINT16 | RW | 0x1601 (5633 _{dez}) |
| 1C12:02 | SubIndex 002 | 2. allocated RxPDO (contains the index of the associated RxPDO mapping object) | UINT16 | RW | 0x1600 (5632 _{dez}) |

Index 1C13 TxPDO assign

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|--------------|--|-----------|-------|-------------------------------|
| 1C13:0 | TxPDO assign | PDO Assign Inputs | UINT8 | RW | 0x02 (2 _{dez}) |
| 1C13:01 | SubIndex 001 | 1. allocated TxPDO (contains the index of the associated TxPDO mapping object) | UINT16 | RW | 0x1A01 (6657 _{dez}) |
| 1C13:02 | SubIndex 002 | 2. allocated TxPDO (contains the index of the associated TxPDO mapping object) | UINT16 | RW | 0x1A00 (6656 _{dez}) |

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

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

Index 6000 IBS Inputs

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|------------|--------------------|-----------|-------|---|
| 6000:0 | IBS Inputs | Max. Subindex | UINT8 | RW | 0x01 (1 _{dez}) |
| 6000:01 | Data | Input process data | RECORD | RW | 00 |

Index 7000 IBS Inputs

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|------------|---------------------|-----------|-------|---|
| 7000:0 | IBS Inputs | Max. Subindex | UINT8 | RW | 0x01 (1 _{dez}) |
| 7000:01 | Data | Output process data | RECORD | RW | 00 |

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 | 0x04 (4 _{dez}) |
| F000:01 | Module index distance | Index distance of the objects of the individual channels | UINT16 | RO | 0x0001 (1 _{dez}) |
| F000:02 | Maximum number of modules | Number of channels | UINT16 | RO | 0x0000 (1 _{dez}) |
| F000:03 | Standard Entries in Object 0x8yy0 | Standard Entries in the Objects 0x8yy0 | UINT32 | RO | 0x00000000 (0 _{dez}) |
| F000:04 | Standard Entries in Object 0x9yy0 | Standard Entries in the Objects 0x8yy0 | UINT32 | RO | 0x00000000 (0 _{dez}) |

Index F100 Status PDO

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|---------------|---------------|-----------|-------|---------------------------|
| F100:0 | Status PDO | Max. Subindex | UINT8 | RO P | 0x0F (15 _{dez}) |
| F100:0E | Rx PDO Status | Rx PDO Status | UINT8 | RO P | TRUE |
| F100:0F | Tx PDO Status | Tx PDO Status | UINT16 | RO P | TRUE |

Index F800 IBS config

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|------------|-----------------------|--------------------------------------|-------|--------------------------|
| F800:0 | IBS config | Max. Subindex | UINT8 | RW | 0x01 (1 _{dez}) |
| F800:01 | 2 Mbit | FALSE | Interbus baud rate [► 20]: 500 kbaud | RW | TRUE |
| | | TRUE | | | |
| F800:02 | IBS ID | Interbus ID (ab FW02) | UNIT16 | RW | - |

Index F801 IBS Info

| Index (hex) | Name | Meaning | Data type | Flags | Default |
|-------------|-----------------|--|-----------|-------|----------------------------|
| F801:0 | IBS Info | Max. Subindex | UINT8 | RO | 0x08 (8 _{dez}) |
| F801:01 | ECAT to IBS | EtherCAT output counters were transferred to the IBS | UINT16 | RO | 0x0000 (0 _{dez}) |
| F801:02 | IBS to ECAT | IBS output counters were transferred to the EL6740-0010 | UINT16 | RO | 0x0000 (0 _{dez}) |
| F801:03 | IBS inactive | IBS counter not active (SUPI 3, bit 0 of the interrupt event register) | UINT16 | RO | 0x0000 (0 _{dez}) |
| F801:04 | IBS reset | IBS counter reset triggered active (SUPI 3, bit 4 of the interrupt event register) | UINT16 | RO | 0x0000 (0 _{dez}) |
| F801:05 | ECAT wdg | ECAT watchdog counter triggered | UINT16 | RO | 0x0000 (0 _{dez}) |
| F801:06 | IBS wdg | IBS watchdog counter triggered | UINT16 | RO | 0x0000 (0 _{dec}) |
| F801:07 | IBS Int pending | A new request is pending while an IBS cycle is processed. Possible cause: IBS cycle too fast. | UINT16 | RO | 0x0000 (0 _{dez}) |
| F801:08 | SUPI IB state | Dump des SUPI IB-state Registers. This is used for mapping various internal operational states (see also <u>IBS SUPI 3</u> documentation) | UINT16 | RO | 0x0000 (0 _{dez}) |

6 Appendix

6.1 EtherCAT AL Status Codes

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

6.2 Firmware compatibility

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

Note

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

NOTICE

Risk of damage to the device!

Pay attention to the instructions for firmware updates on the [separate page \[▶ 121\]](#). 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!

| EL6740-0010 | | | |
|---------------|----------|------------------|--------------|
| Hardware (HW) | Firmware | Revision no. | Release date |
| 00 - 03 | 01 | EL6740-0010-0000 | 2008/01 |
| | 02 | | 2008/10 |
| | 03 | | 2009/01 |
| | 04 | | 2009/06 |
| | 05 | | 2009/07 |
| | 06 | | 2010/05 |
| 04 | 07 | | 2010/05 |
| | 08 | | 2010/07 |
| 05 - 12* | 09 | | 2010/08 |
| | 10 | | 2010/09 |
| | 11 | | 2012/04 |
| | | EL6740-0010-0002 | 2013/04 |
| 12* | | 2013/11 | |

*) 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.

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

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

NOTICE

Only use TwinCAT 3 software!

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

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

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

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

Storage locations

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

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

NOTICE

Application-specific writing of the ESI-EEPROM

The ESI is developed by the device manufacturer according to ETG standard and released for the corresponding product.

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

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

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

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

Simplified update by bundle firmware

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

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

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

Following the update, its success should be verified

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

NOTICE

Risk of damage to the device!

✓ Note the following when downloading new device files

- 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.

6.3.1 Device description ESI file/XML

NOTICE

Attention regarding update of the ESI description/EEPROM

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

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

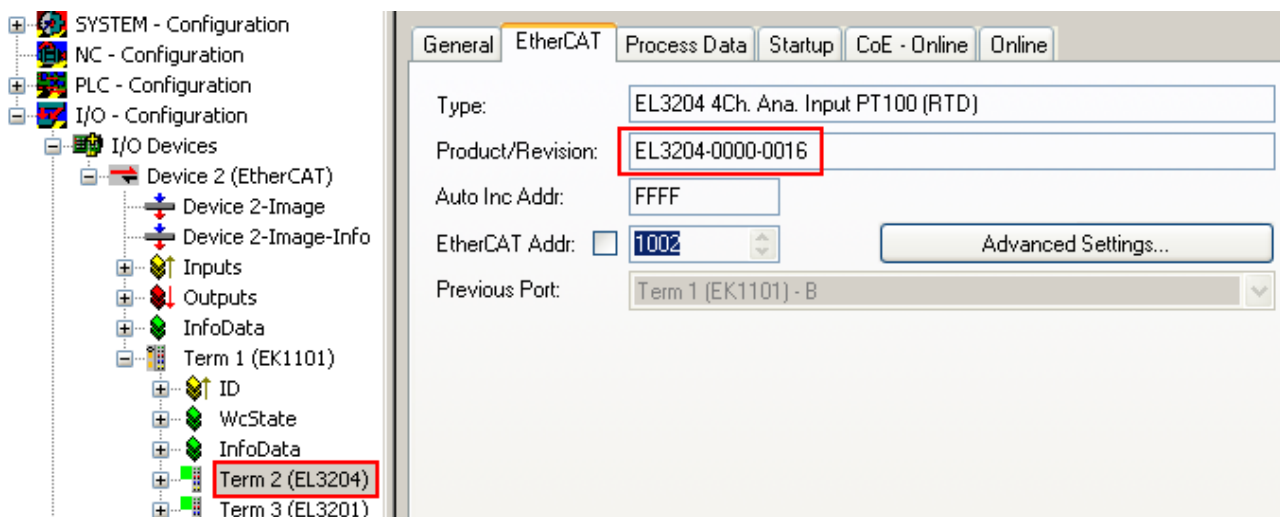


Fig. 140: Device identifier consisting of name EL3204-0000 and revision -0016

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](#).

i 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:

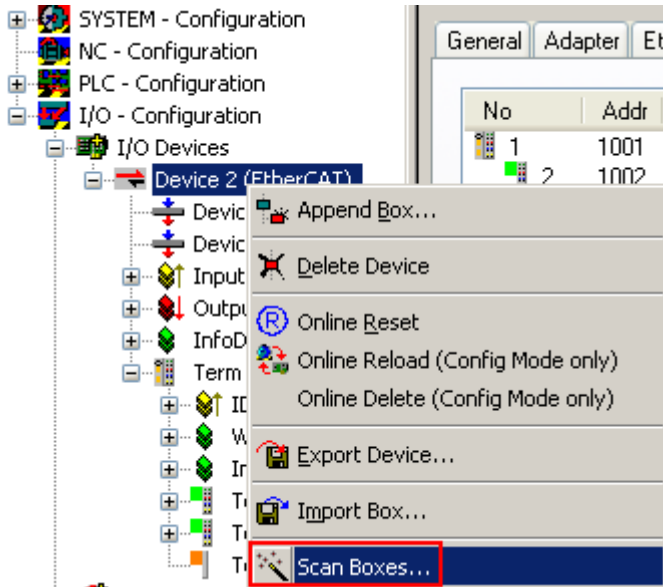


Fig. 141: Scan the subordinate field by right-clicking on the EtherCAT device

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



Fig. 142: Configuration is identical

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

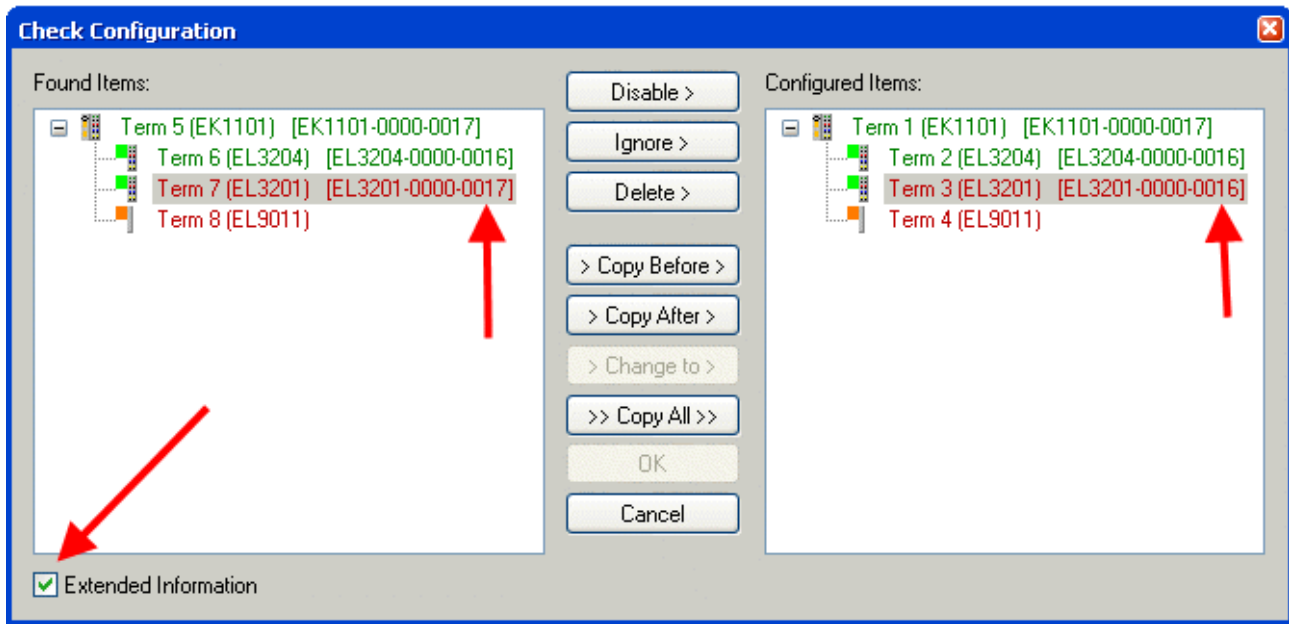


Fig. 143: Change dialog

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*

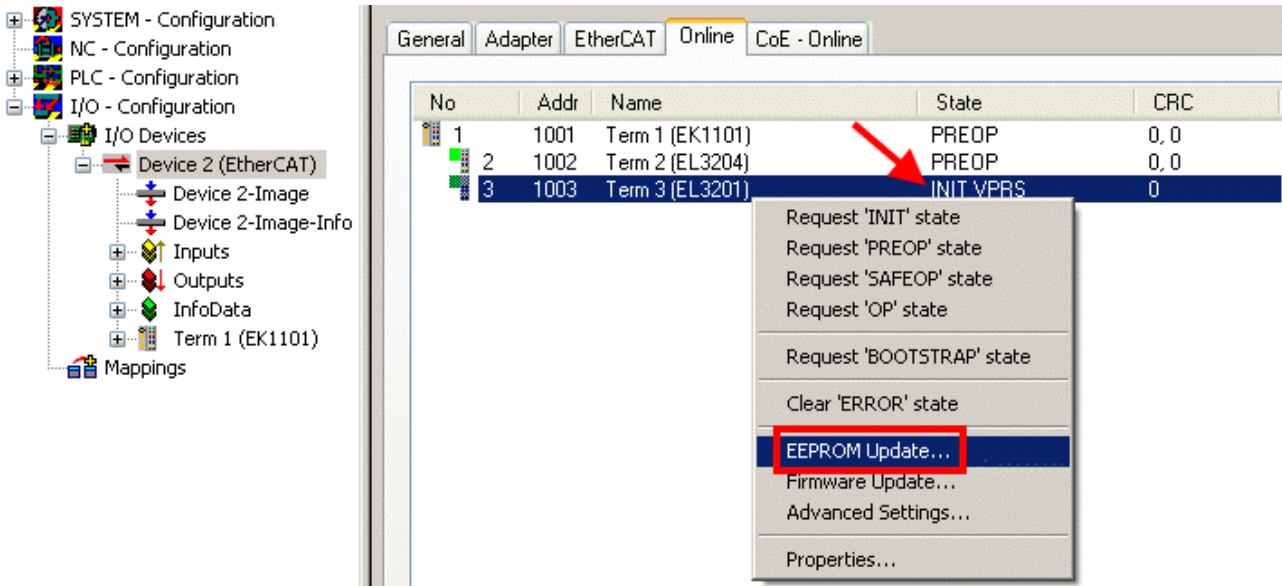


Fig. 144: EEPROM Update

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.

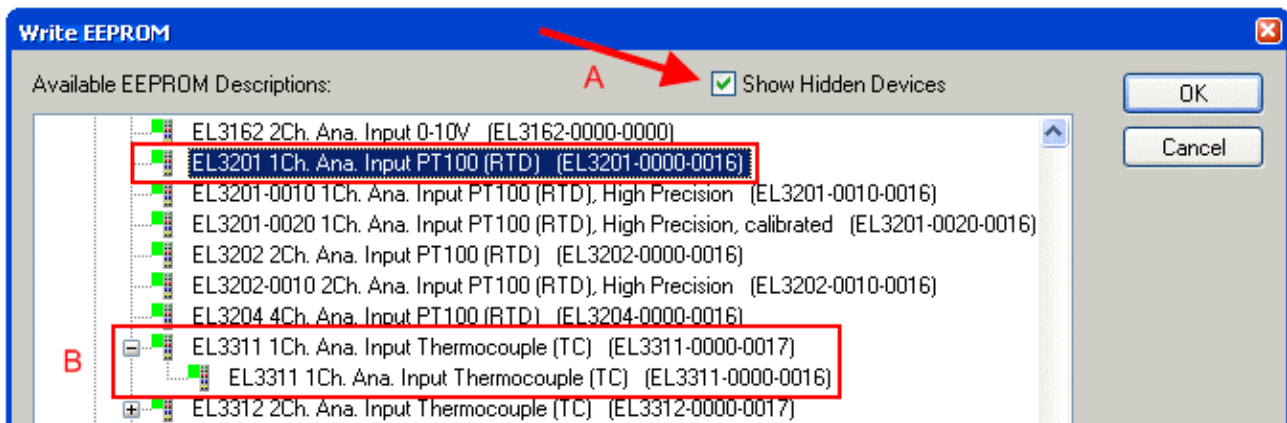


Fig. 145: Selecting the new ESI

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

i **The change only takes effect after a restart.**

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

6.3.2 Firmware explanation

Determining the firmware version

Determining the version via the System Manager

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

i **CoE Online and Offline CoE**

Two CoE directories are available:

- **online:** This is offered in the EtherCAT slave by the controller, if the EtherCAT slave supports this. This CoE directory can only be displayed if a slave is connected and operational.
- **offline:** The EtherCAT Slave Information ESI/XML may contain the default content of the CoE. This CoE directory can only be displayed if it is included in the ESI (e.g. "Beckhoff EL5xxx.xml").

The Advanced button must be used for switching between the two views.

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

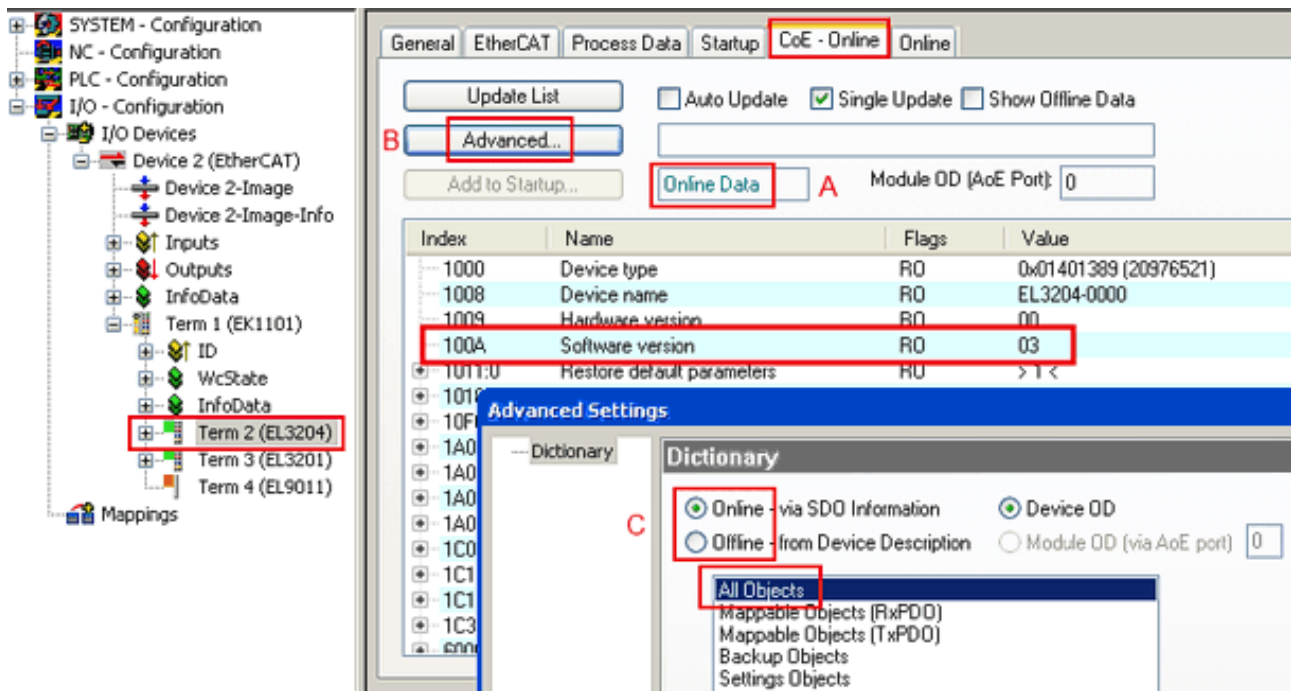


Fig. 146: Display of EL3204 firmware version

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

6.3.3 Updating controller firmware *.efw

● CoE directory

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

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

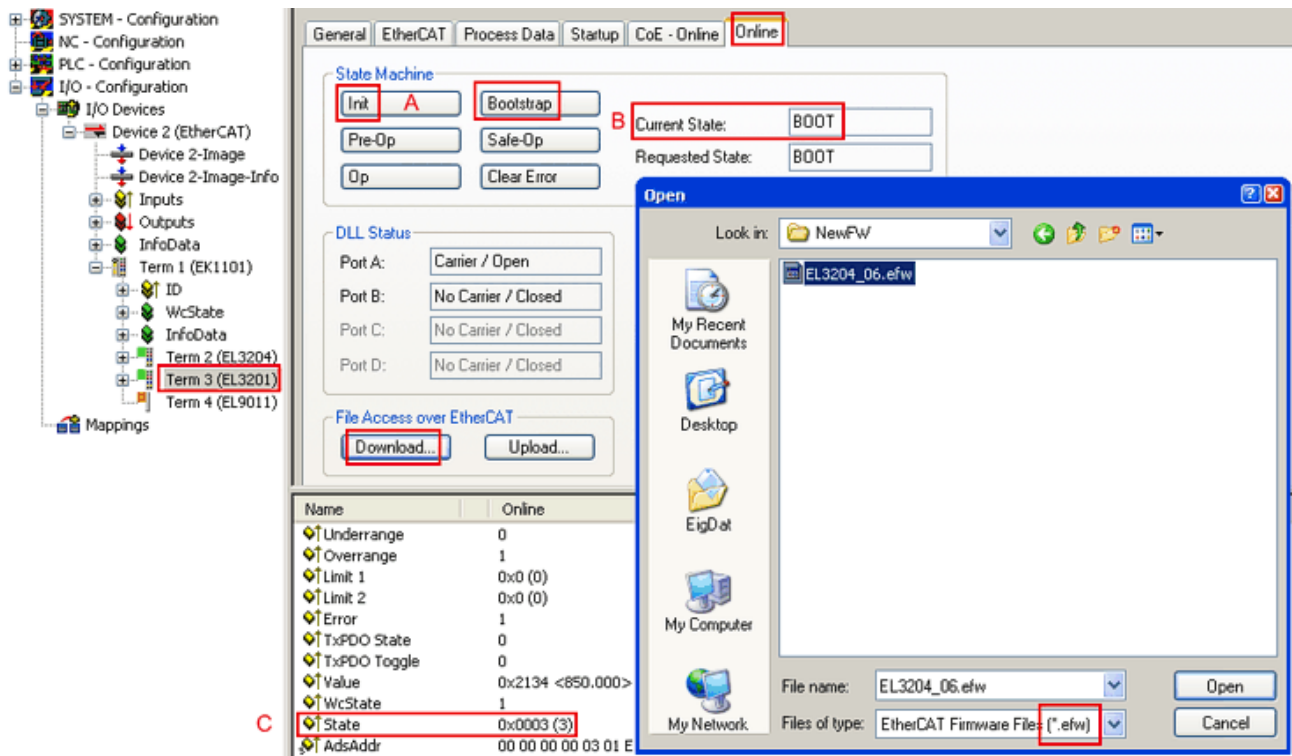
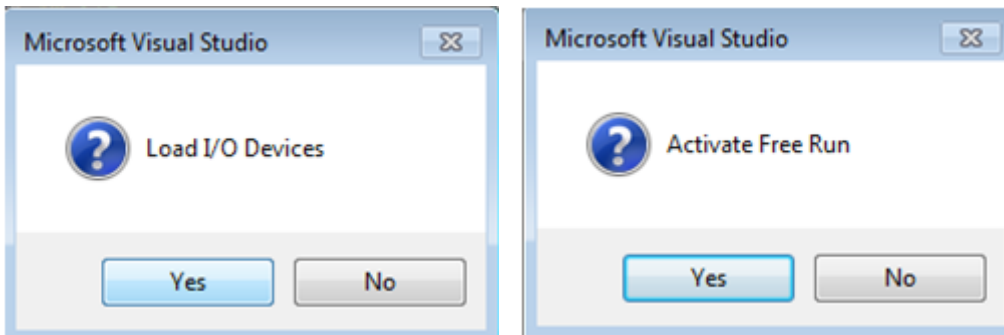


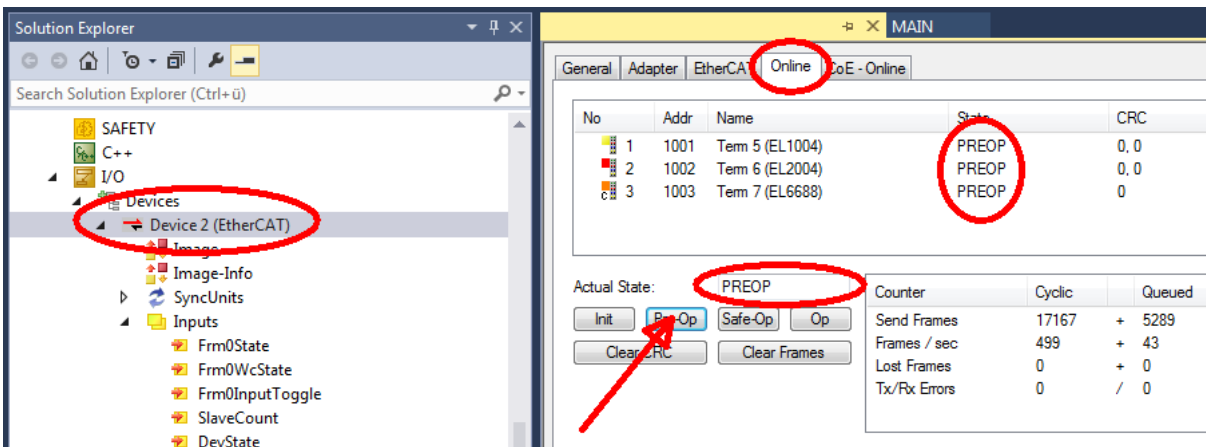
Fig. 147: Firmware Update

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

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

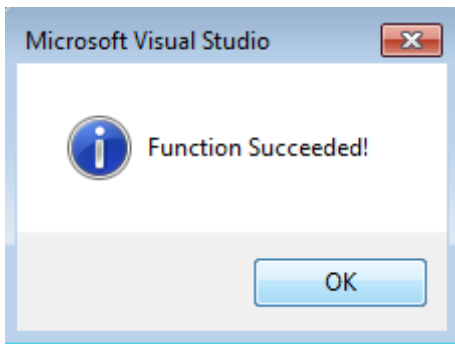


- Switch EtherCAT Master to PreOP



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

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



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

6.3.4 FPGA firmware *.rbf

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

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

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

Determining the version via the 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.

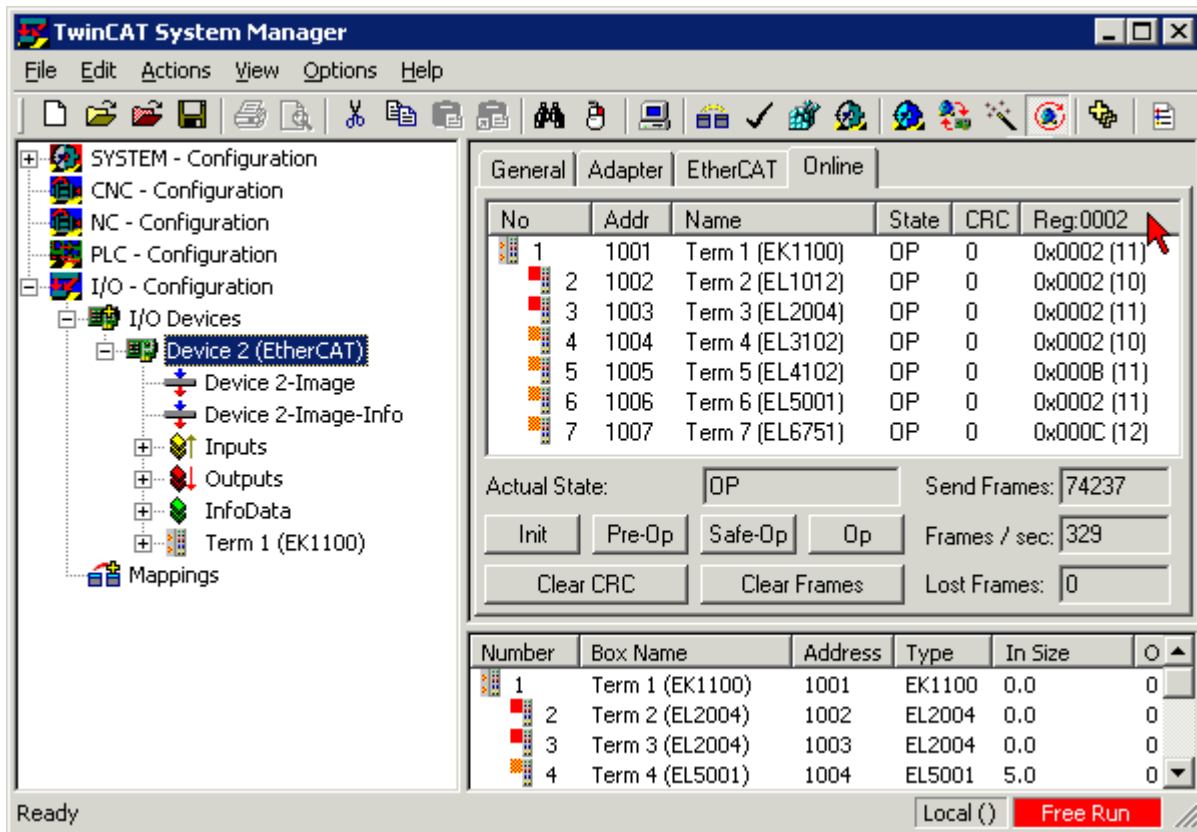


Fig. 148: FPGA firmware version definition

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

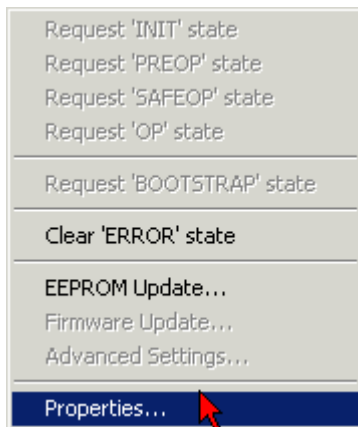


Fig. 149: Context menu *Properties*

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.

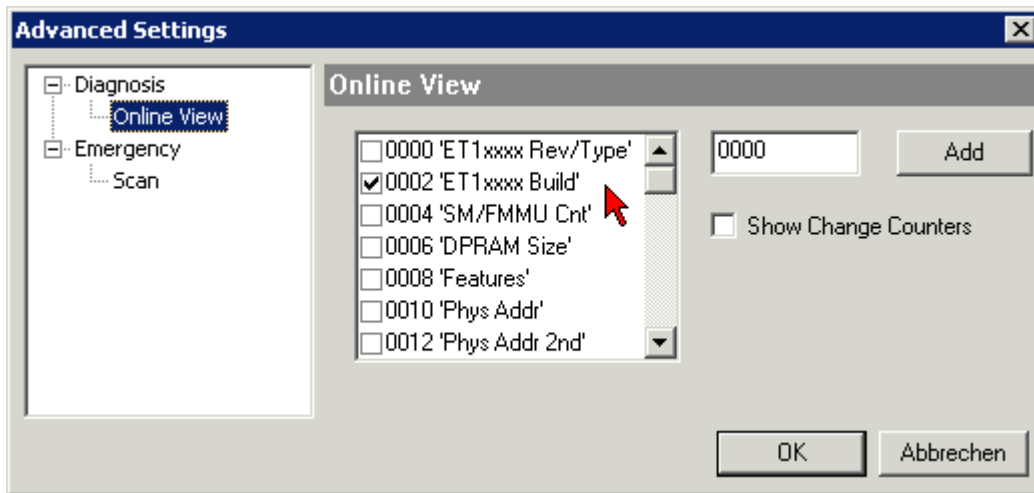


Fig. 150: Dialog *Advanced Settings*

Update

For updating the FPGA firmware

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

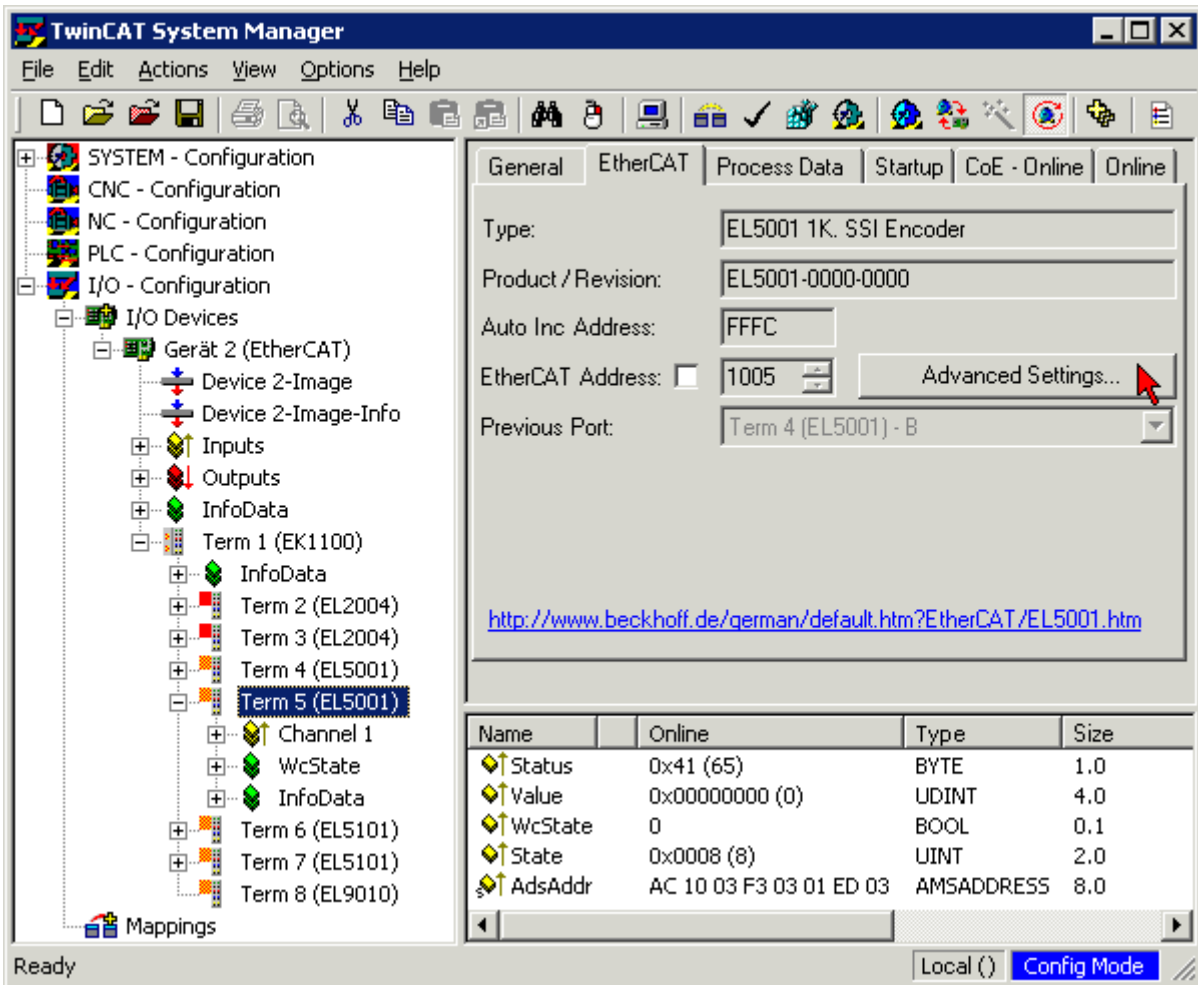
Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

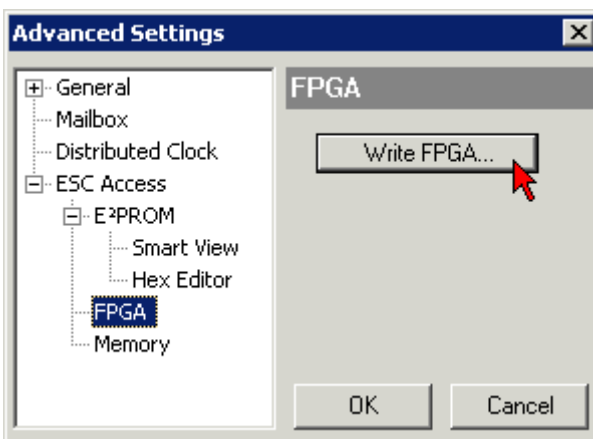
The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

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

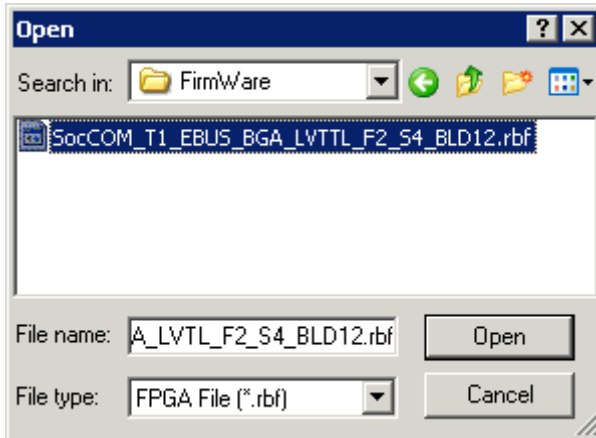
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and click the *Advanced Settings* button in the *EtherCAT* tab:



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



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



- Wait until download ends
- Switch slave current less for a short time (don't pull under voltage!). In order to activate the new FPGA firmware a restart (switching the power supply off and on again) of the EtherCAT device is required.
- Check the new FPGA status

NOTICE

Risk of damage to the device!

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

6.3.5 Simultaneous updating of several EtherCAT devices

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

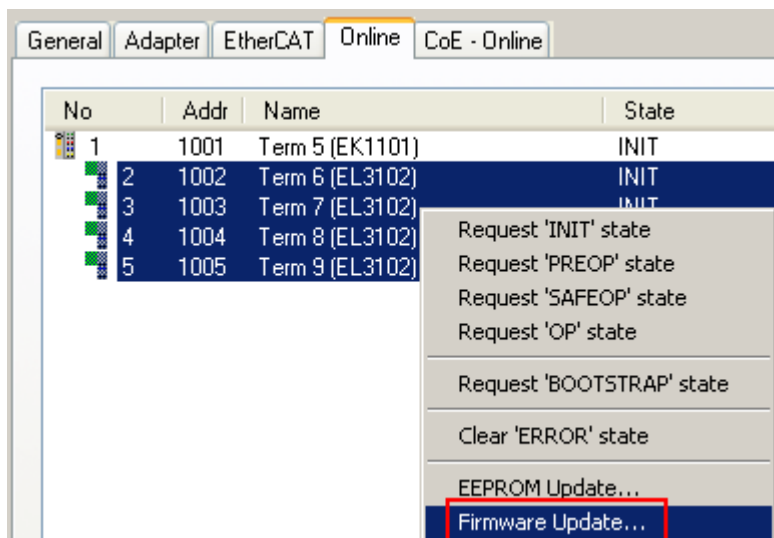


Fig. 151: Multiple selection and firmware update

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

6.4 Support and Service

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

Beckhoff's branch offices and representatives

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

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

You will also find further documentation for Beckhoff components there.

Support

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

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

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

Service

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

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

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

Headquarters Germany

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Germany

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

More Information:
www.beckhoff.com/EL6740-0010

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

