BECKHOFF New Automation Technology

Documentation | EN

EL5072

2 Channel LVDT Interface

2022-10-17 | Version: 1.1

Table of contents

1 Foreword

1.1 Notes on the documentation

Intended audience

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

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

Disclaimer

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

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

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

Trademarks

Beckhoff®, TwinCAT®, 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

1.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Description of instructions

In this documentation the following instructions are used.
These instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow this safety instruction directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow this safety instruction endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow this safety instruction can lead to injuries to persons.

NOTE

Damage to environment/equipment or data loss

Failure to follow this instruction can lead to environmental damage, equipment damage or data loss.

Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3 Documentation issue status

Version	Comment
1.1	• Update chapter “Version identification of EtherCAT devices” • Update chapter “Introduction” • Update chapter “Technical data” • Update chapter “Mounting and wiring” • Chapter “Setting recommendations of common sensor series” added • Update revision status • Update structure
1.0	• First release
0.2	• Modifications
0.1	• Provisional documentation for EL5072

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**	Example with serial number 12 06 3A 02:
KK - week of production (CW, calendar week)	12 - production week 12
YY - year of production	06 - production year 2006
FF - firmware version	3A - firmware version 3A
HH - hardware version	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.

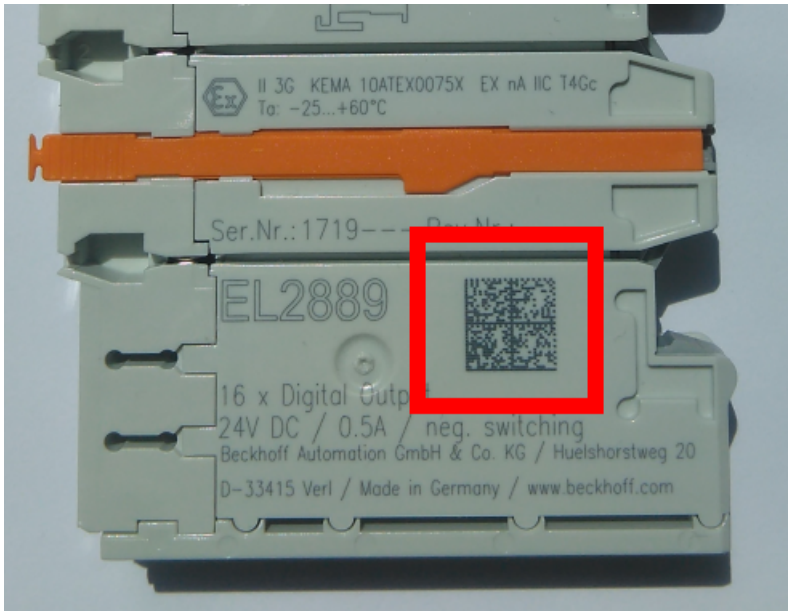


Fig. 2: BIC as data matrix code (DMC, code scheme ECC200)

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

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

Structure of the BIC

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

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

Accordingly as DMC:

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

BTN

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

NOTE

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.

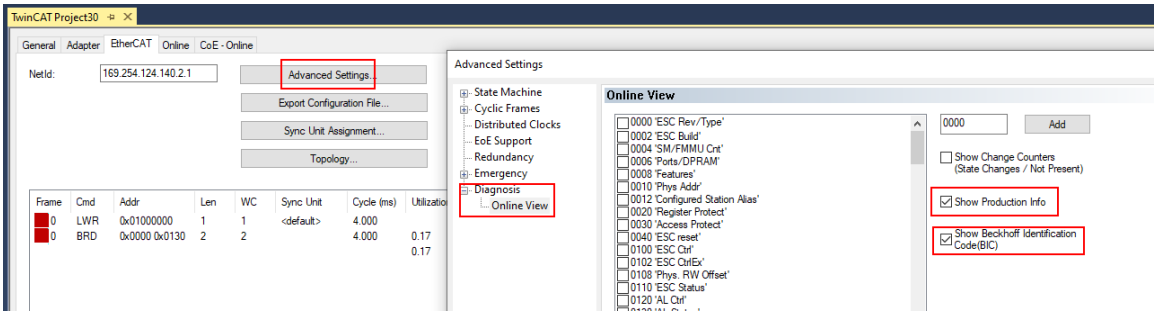
EtherCAT devices (IP20, IP67)

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.

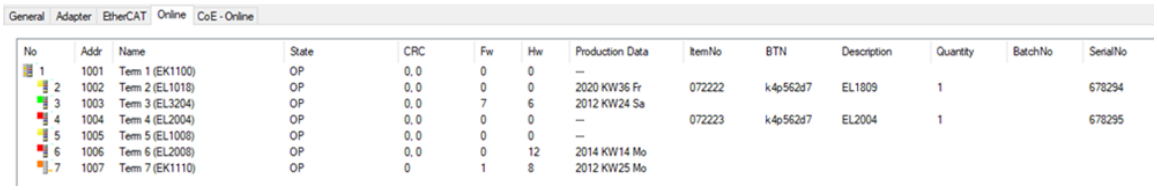
The eBIC is also stored in the ESI-EEPROM. The eBIC was introduced into the Beckhoff I/O production (terminals, box modules) from 2020; widespread implementation is expected in 2021.

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:



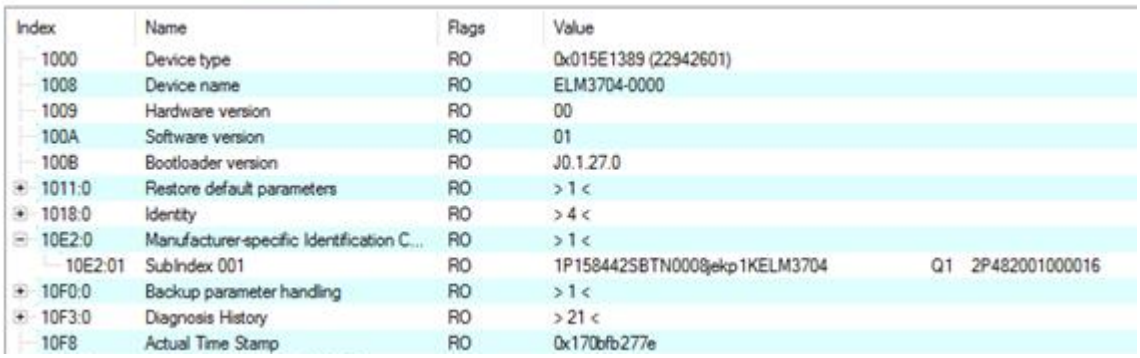
  - The BTN and its contents are then displayed:



  - 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".

- From TwinCAT 3.1. build 4024.24 the functions *FB_EcReadBIC* and *FB_EcReadBTN* for reading into the PLC and further eBIC auxiliary functions are available in the Tc2_EtherCAT Library from v3.3.19.0.

- In the case of EtherCAT devices with CoE directory, the object 0x10E2:01 can additionally by used to display the device's own eBIC; the PLC can also simply access the information here:
  - The device must be in PREOP/SAFEOP/OP for access:



Index	Name	Flags	Value
1000	Device type	RO	0x015E1389 (22942601)
1008	Device name	RO	ELM3704-0000
1009	Hardware version	RO	00
100A	Software version	RO	01
100B	Bootloader version	RO	J0.1.27.0
1011:0	Restore default parameters	RO	> 1 <
1018:0	Identity	RO	> 4 <
10E2:0	Manufacturer-specific Identification C...	RO	> 1 <
10E2:01	SubIndex 001	RO	1P158442SBTN0008jekp1KELM3704 Q1 2P482001000016
10F0:0	Backup parameter handling	RO	> 1 <
10F3:0	Diagnosis History	RO	> 21 <
10F8	Actual Time Stamp	RO	0x170bfb277e

  - the object 0x10E2 will be introduced into stock products in the course of a necessary firmware revision.
  - From TwinCAT 3.1. build 4024.24 the functions *FB_EcCoEReadBIC* and *FB_EcCoEReadBTN* for reading into the PLC and further eBIC auxiliary functions are available in the Tc2_EtherCAT Library from v3.3.19.0.
- 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

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

2 Product description

2.1 Introduction

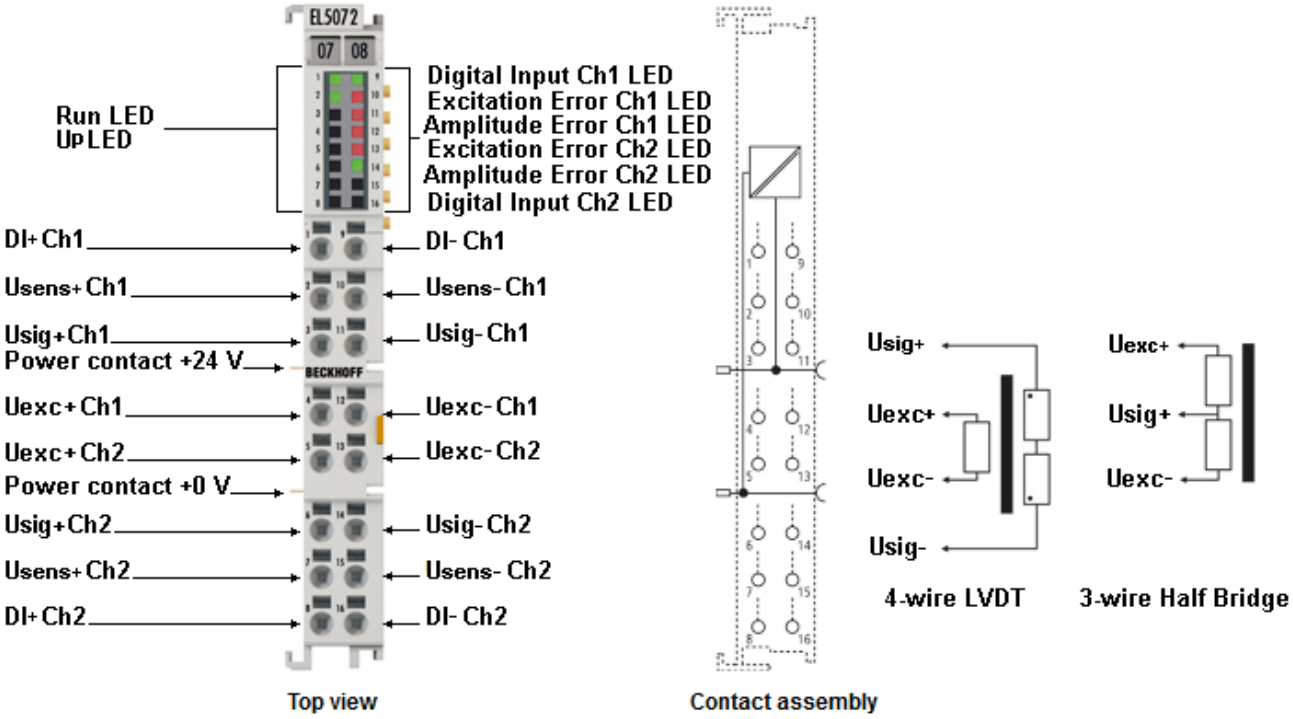


Fig. 4: EL5072

2-channel inductive displacement sensor interface (LVDT, half bridge, RVDT)

The EL5072 EtherCAT Terminal is used for direct connection of up to two inductive displacement sensors, including measuring probes in LVDT and half bridge design or inductive angular position sensors in RVDT design. The integrated excitation source provides a wide range with different adjustable excitation frequencies and voltages. Through internal switching and switchable input impedances, commercially available inductive sensors such as LVDT in 4-, 5-, 6-wire design or half bridge in 3- and 5-wire design can be connected and evaluated.

All sensor parameters such as excitation frequency, excitation voltage, measuring range and sensitivity are set directly in the terminal. The measuring signal range is automatically adjusted; the measured value is directly output as a 32-bit position value. Inversion of the measuring signal and a reference position can be parameterized via the process data. Furthermore, short circuit or overload of the excitation source, as well as amplitude errors of the input signal are diagnosed for each channel and indicated via signal LEDs.

Precision measuring tasks in the area of position and distance measurement with inductive measuring probes can be successfully solved in this way with the EL5072.

Quick links

- Basic Function Principles [▶ 15]
- EL5072 - Connection [▶ 39]
- EL5072 - LEDs [▶ 45]
- EL5072 - Functions [▶ 101]

2.2 Technical data

Technical data	EL5072
Technology	Inductive displacement sensor interface
Input connections	LVDT (ratiometric and differential), inductive half bridge, RVDT 2 x digital input (5 V_{DC} to 24 V_{DC}, 2-wire, switching threshold typically 5 V at 3 mA)
Number of channels	2
Distributed clocks	yes, timestamp for position value can be saved via digital input
Excitation voltage U_{exc}	optionally 0.5 V_{rms} to 7 V_{rms} (1.5 V_{PP} to 20 V_{PP}), common for both channels
Total excitation current	max. 50 mA_{rms} (140 mA_{PP})
Excitation frequency (sine)	1 kHz to 20 kHz, adjustable, common for both channels
Resolution	24 bit, 32 bit representation
Measuring signal range U_{SIG}	max. 7 V_{rms} (automatic setting)
Conversion time	100 µs/10 kSps at max. 13 kHz
Electrical isolation	500 V (E-bus/field voltage)
Current consumption power contacts	40 mA typ.+ load
Current consumption via E-bus	200 mA typ.
Special features	Short circuit and overload detection, amplitude error per channel, set and save digital input for position value
Weight	app. 60 g
Permissible ambient temperature range during operation	0°C ... +55°C
Permissible ambient temperature range during storage	-25°C ... +85°C
Permissible relative air humidity	95%, no condensation
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)
Mounting	on 35 mm support rail according to EN 60715
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approvals / markings*	CE, EAC, UKCA

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

2.3 Basics of inductive measuring probes

Inductive displacement sensors are transformers with a special design that are used for displacement recording / path length measurement. They are passive components that are available with different circuit options. A basic common feature is that an alternating voltage excites a coil system. A moving ferromagnetic core affects the inductance in the coils. The inductance change is proportional to the movement of the core and can be evaluated.

The EL5072 can evaluate the following inductive displacement sensors:

- Inductive half bridge [▶ 15]
- LVDT (Linear Variable Differential Transformer) [▶ 16]
- RVDT (Rotary Variable Differential Transformer) [▶ 16]

Operating principle of inductive half bridge position transducers

Electrically, inductive half bridges (differential chokes) represent a Wheatstone half bridge with variable, complex resistances. This consists of:

- two measuring coils
- a movable ferromagnetic core that moves inside the coils

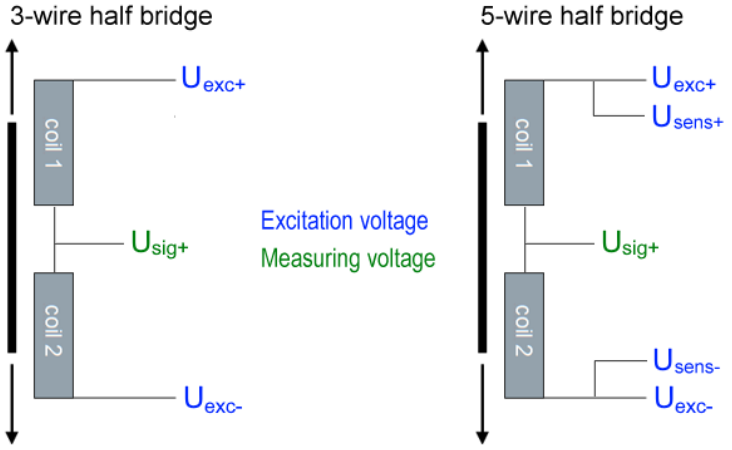


Fig. 5: Inductive half bridge - operating principle

Principle of operation:

An alternating voltage is applied to the two coils connected in series. The ferromagnetic core changes the inductance of the coils when the measuring probe is deflected. In a symmetrical configuration and in the zero position of the position transducer, the impedance of the two coils is the same. If the core is moved from its center position, the impedance in the two coils changes in opposite directions. This results in a linear and absolute displacement signal, which can be measured with the EL5072.

Operating principle of Linear Variable Differential Transformers (LVDTs)

inductive LVDT displacement sensors generally consist of:

- a primary coil used for excitation
- two secondary coils, which are arranged in phase opposition to each other.
- a movable ferromagnetic core, which serves to couple the primary and secondary coils

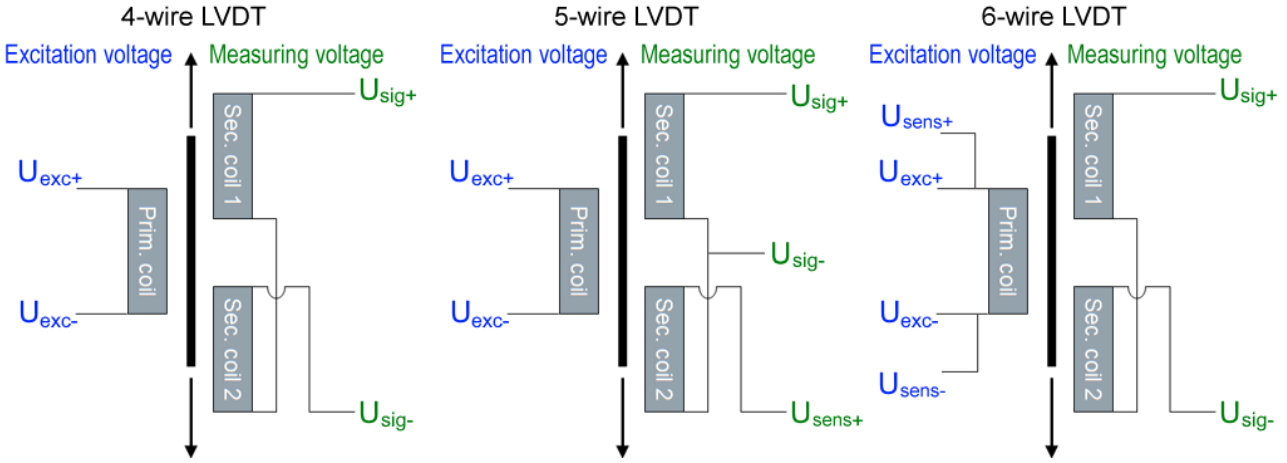


Fig. 6: LVDT - Operating principle

Principle of operation:

The primary coil fed with an alternating voltage induces a secondary voltage in the secondary winding. In a symmetrical configuration, the secondary voltages are equal in magnitude in the zero position of the position transducer but phase-inverted. The resulting signal voltage is zero. If the core is deflected, the induced voltage increases in one secondary coil and decreases in the other. This results in a linear and absolute displacement signal, which can be measured with the EL5072.

The deflection direction of the inductive measuring probe is determined by the movement of the core and the resulting phase shift between the excitation voltage U_{exc} and the measured signal voltage U_{sig}. The general rule is:

- Negative deflection direction of the core relative to the zero position:
  excitation voltage U_{exc} and signal voltage U_{sig} are in phase
- Positive deflection direction of the core relative to the zero position:
  excitation voltage U_{exc} and signal voltage U_{sig} are phase-shifted by 180°
- Zero position of the core:
  phase jump between excitation voltage U_{exc} and signal voltage U_{sig}

In addition to the usual 4-wire connection, LVDT probes are also available as 5- or 6-wire versions. The 5-wire version allows ratiometric measurement on the secondary side.

With the 6-wire version, the excitation voltage fed in is measured back from the sensor, thus minimizing influences on the voltage measurement which could be caused by a voltage drop along the supply lines.

Operating principle of Rotary Variable Differential Transformers (RVDT)

An RVDT rotary encoder represents a special design of the LVDT measuring principle. The main difference is that the LVDT uses a linear displacement of the core, whereas the RVDT uses a cam-shaped rotating core to measure the angular displacement.

To ensure correct output of the measured value, special Notes for RVDTs [▶ 115] must be observed.

3 Basics communication

3.1 EtherCAT basics

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

3.2 EtherCAT cabling – wire-bound

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

Cables and connectors

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

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

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

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

Recommended cables

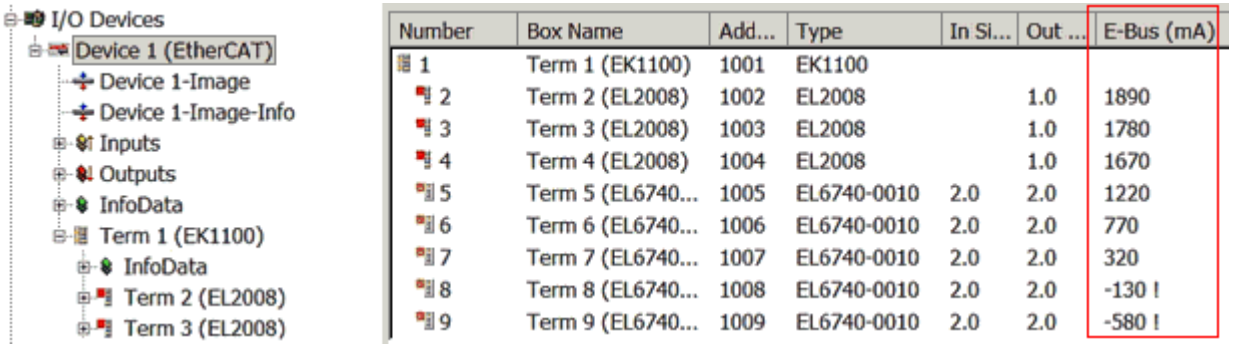
It is recommended to use the appropriate Beckhoff components e.g.
- cable sets ZK1090-9191-xxxx respectively
- RJ45 connector, field assembly ZS1090-0005
- EtherCAT cable, field assembly ZB9010, ZB9020

Suitable cables for the connection of EtherCAT devices can be found on the Beckhoff website!

E-Bus supply

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

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



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

Fig. 7: System manager current calculation

NOTE

Malfunction possible!

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

3.3 General notes for setting the watchdog

ELxxxx terminals are equipped with a safety feature (watchdog) that switches off the outputs after a specifiable time e.g. in the event of an interruption of the process data traffic, depending on the device and settings, e.g. in OFF state.

The EtherCAT slave controller (ESC) features two watchdogs:

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

SM watchdog (SyncManager Watchdog)

The SyncManager watchdog is reset after each successful EtherCAT process data communication with the terminal. If no EtherCAT process data communication takes place with the terminal for longer than the set and activated SM watchdog time, e.g. in the event of a line interruption, the watchdog is triggered and the outputs are set to FALSE. The OP state of the terminal is unaffected. The watchdog is only reset after a successful EtherCAT process data access. Set the monitoring time as described below.

The SyncManager watchdog monitors correct and timely process data communication with the ESC from the EtherCAT side.

PDI watchdog (Process Data Watchdog)

If no PDI communication with the EtherCAT slave controller (ESC) takes place for longer than the set and activated PDI watchdog time, this watchdog is triggered.
PDI (Process Data Interface) is the internal interface between the ESC and local processors in the EtherCAT slave, for example. The PDI watchdog can be used to monitor this communication for failure.

The PDI watchdog monitors correct and timely process data communication with the ESC from the application side.

The settings of the SM- and PDI-watchdog must be done for each slave separately in the TwinCAT System Manager.

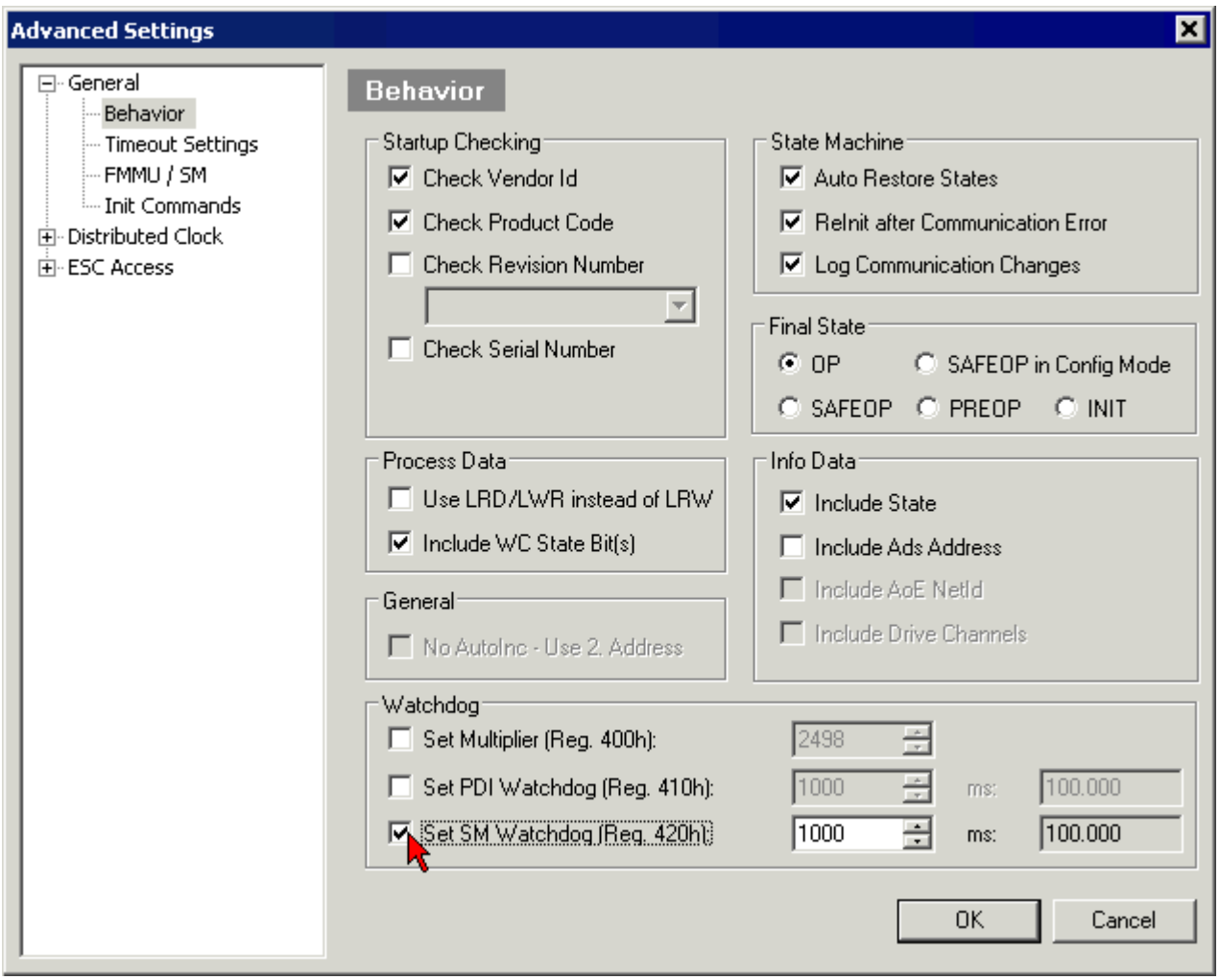


Fig. 8: EtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

- the multiplier is valid for both watchdogs.
- each watchdog has its own timer setting, the outcome of this in summary with the multiplier is a resulting time.
- Important: the multiplier/timer setting is only loaded into the slave at the start up, if the checkbox is activated.
  If the checkbox is not activated, nothing is downloaded and the ESC settings remain unchanged.

Multiplier

Both watchdogs receive their pulses from the local terminal cycle, divided by the watchdog multiplier:

1/25 MHz * (watchdog multiplier + 2) = 100 µs (for default setting of 2498 for the multiplier)

The standard setting of 1000 for the SM watchdog corresponds to a release time of 100 ms.

The value in multiplier + 2 corresponds to the number of basic 40 ns ticks representing a watchdog tick. The multiplier can be modified in order to adjust the watchdog time over a larger range.

Example “Set SM watchdog”

This checkbox enables manual setting of the watchdog times. If the outputs are set and the EtherCAT communication is interrupted, the SM watchdog is triggered after the set time and the outputs are erased. This setting can be used for adapting a terminal to a slower EtherCAT master or long cycle times. The default SM watchdog setting is 100 ms. The setting range is 0...65535. Together with a multiplier with a range of 1...65535 this covers a watchdog period between 0...~170 seconds.

Calculation

Multiplier = 2498 → watchdog base time = 1 / 25 MHz * (2498 + 2) = 0.0001 seconds = 100 µs
SM watchdog = 10000 → 10000 * 100 µs = 1 second watchdog monitoring time

⚠ CAUTION

Undefined state possible!

The function for switching off of 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.

3.4 EtherCAT State Machine

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

A distinction is made between the following states:

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

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

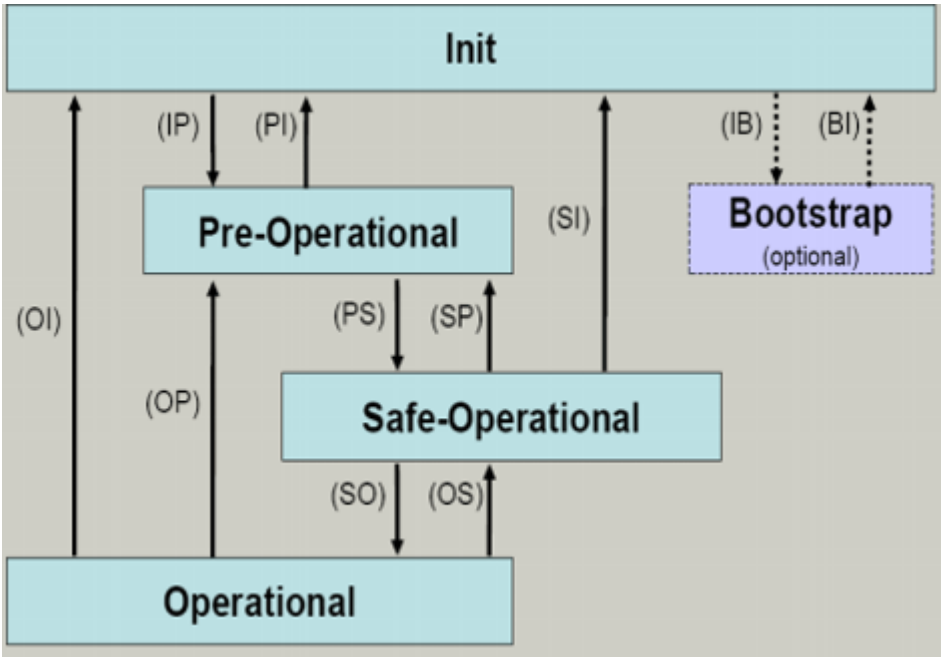


Fig. 9: States of the EtherCAT State Machine

Init

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

Pre-Operational (Pre-Op)

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

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

Safe-Operational (Safe-Op)

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

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

Outputs in SAFEOP state

The default set watchdog [▶ 18] monitoring sets the outputs of the module in a safe state - depending on the settings in SAFEOP and OP - e.g. in OFF state. If this is prevented by deactivation of the watchdog monitoring in the module, the outputs can be switched or set also in the SAFEOP state.

Operational (Op)

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

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

Boot

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

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

3.5 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.

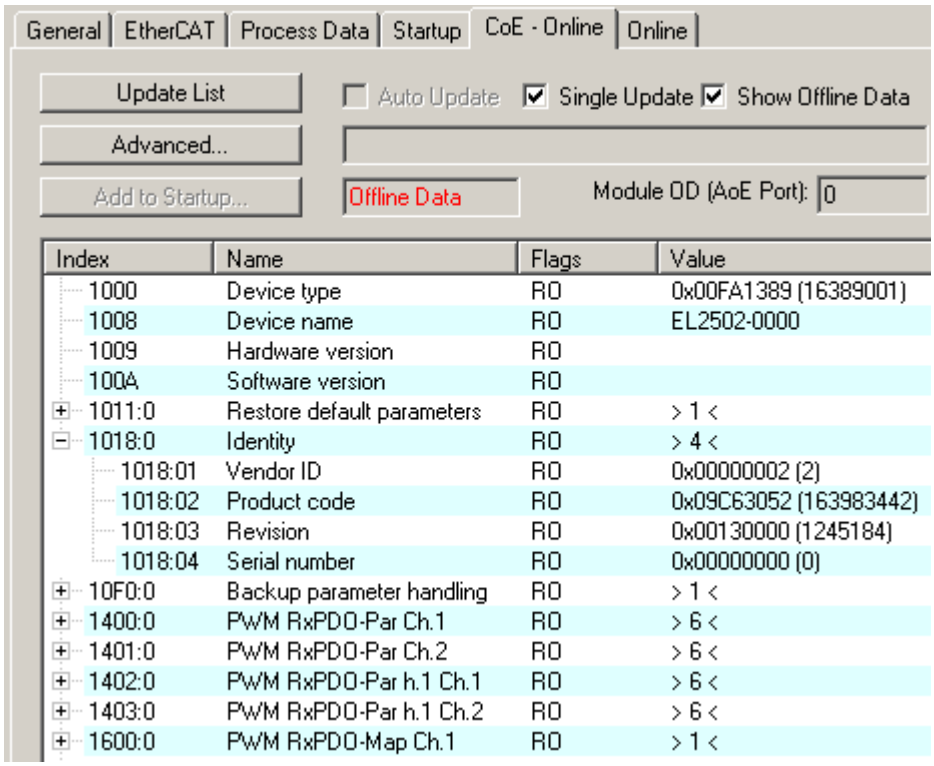
Other important ranges are:

- 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:



Index	Name	Flags	Value
1000	Device type	RO	0x00FA1389 (16389001)
1008	Device name	RO	EL2502-0000
1009	Hardware version	RO	
100A	Software version	RO	
1011:0	Restore default parameters	RO	> 1 <
1018:0	Identity	RO	> 4 <
1018:01	Vendor ID	RO	0x00000002 (2)
1018:02	Product code	RO	0x09C63052 (163983442)
1018:03	Revision	RO	0x00130000 (1245184)
1018:04	Serial number	RO	0x00000000 (0)
10F0:0	Backup parameter handling	RO	> 1 <
1400:0	PWM RxPDO-Par Ch.1	RO	> 6 <
1401:0	PWM RxPDO-Par Ch.2	RO	> 6 <
1402:0	PWM RxPDO-Par h.1 Ch.1	RO	> 6 <
1403:0	PWM RxPDO-Par h.1 Ch.2	RO	> 6 <
1600:0	PWM RxPDO-Map Ch.1	RO	> 1 <

Fig. 10: “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.

NOTE

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 Bibliothek: Tc2_EtherCAT and Example programm R/W CoE)

Data management and function “NoCoeStorage”

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

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

Data management

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

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

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

Startup list

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

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

Recommended approach for manual modification of CoE parameters

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

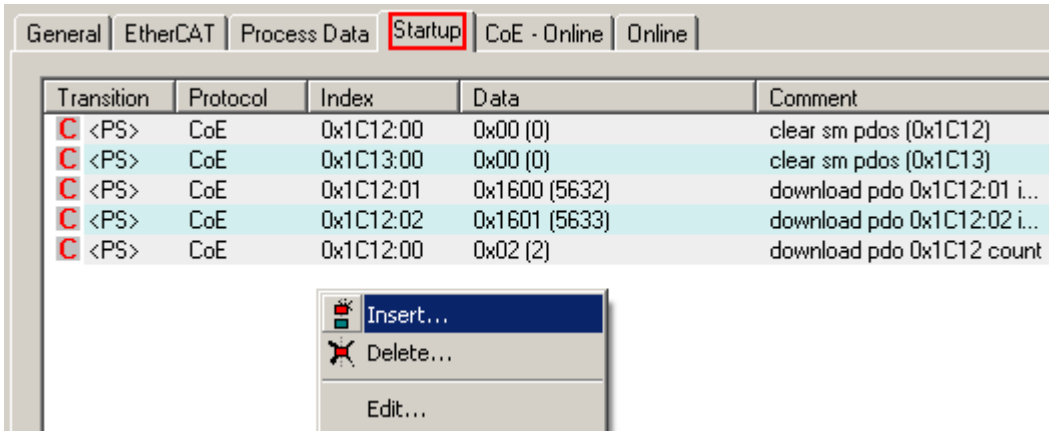


Fig. 11: Startup list in the TwinCAT System Manager

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

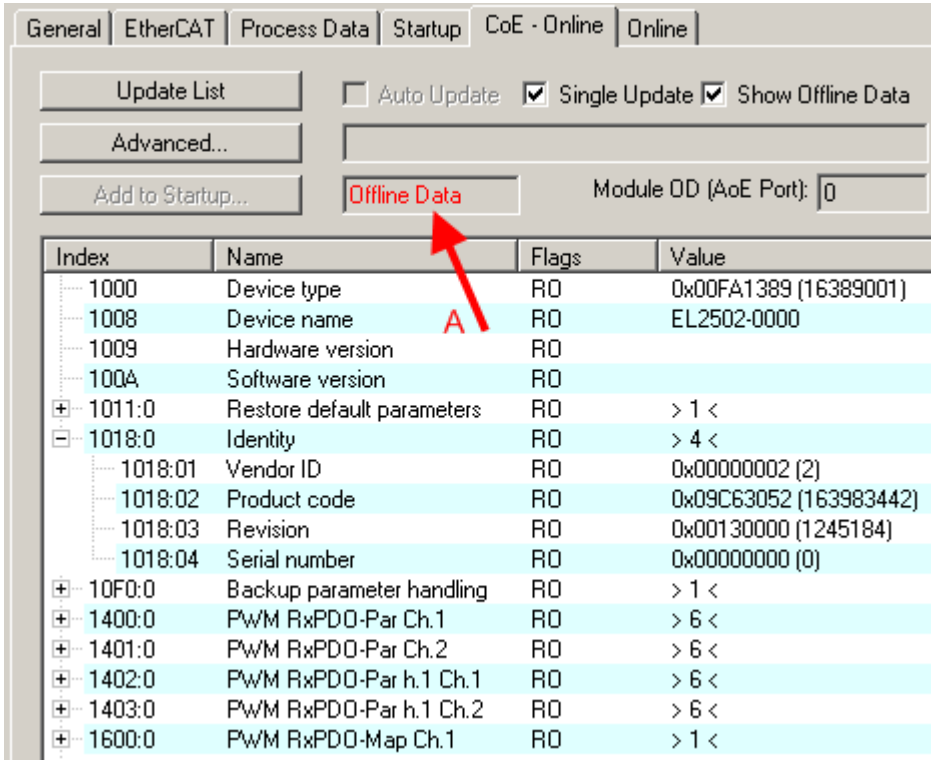


Fig. 12: Offline list

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

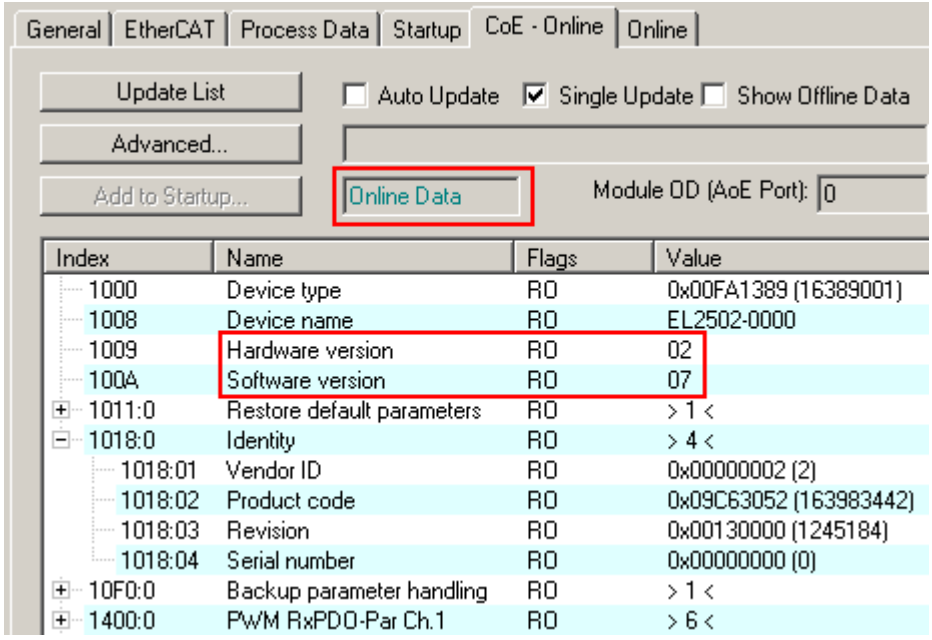


Fig. 13: Online list

Channel-based order

The CoE list is available in EtherCAT devices that usually feature several functionally equivalent channels. For example, a 4-channel analog 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.

3.6 Distributed Clock

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

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

For detailed information please refer to the EtherCAT system description.

4 Mounting and wiring

4.1 Instructions for ESD protection

NOTE

Destruction of the devices by electrostatic discharge possible!

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

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

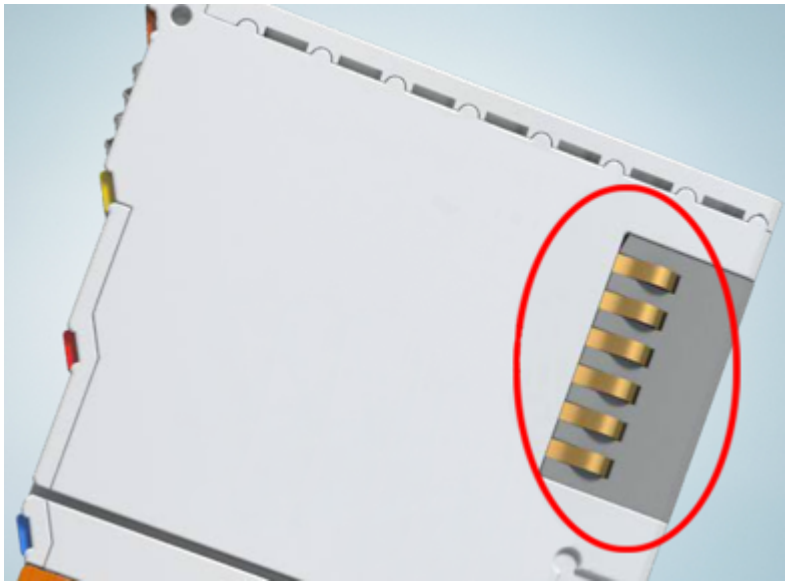


Fig. 14: Spring contacts of the Beckhoff I/O components

4.2 Installation on mounting rails

⚠ 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!

Assembly

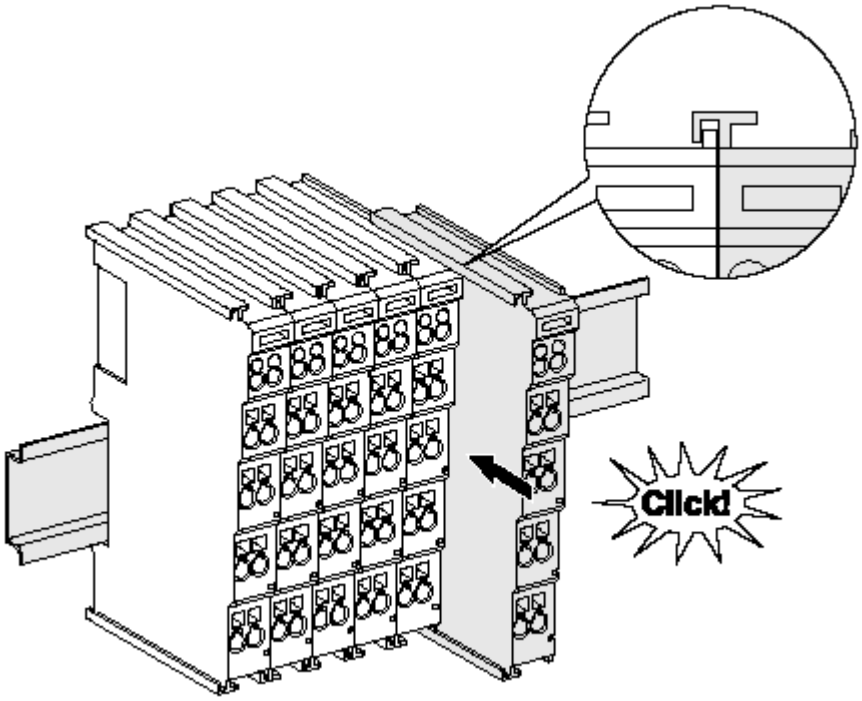


Fig. 15: Attaching on mounting rail

The bus coupler and bus terminals are attached to commercially available 35 mm mounting rails (DIN rails according to EN 60715) by applying slight pressure:

1. First attach the fieldbus coupler to the mounting rail.
2. The bus terminals are now attached on the right-hand side of the fieldbus coupler. Join the components with tongue and groove and push the terminals against the mounting rail, until the lock clicks onto the mounting rail.
   If the terminals are clipped onto the mounting rail first and then pushed together without tongue and groove, the connection will not be operational! When correctly assembled, no significant gap should be visible between the housings.

Fixing of mounting rails

The locking mechanism of the terminals and couplers extends to the profile of the mounting rail. At the installation, the locking mechanism of the components must not come into conflict with the fixing bolts of the mounting rail. To mount the mounting rails with a height of 7.5 mm under the terminals and couplers, you should use flat mounting connections (e.g. countersunk screws or blind rivets).

Disassembly

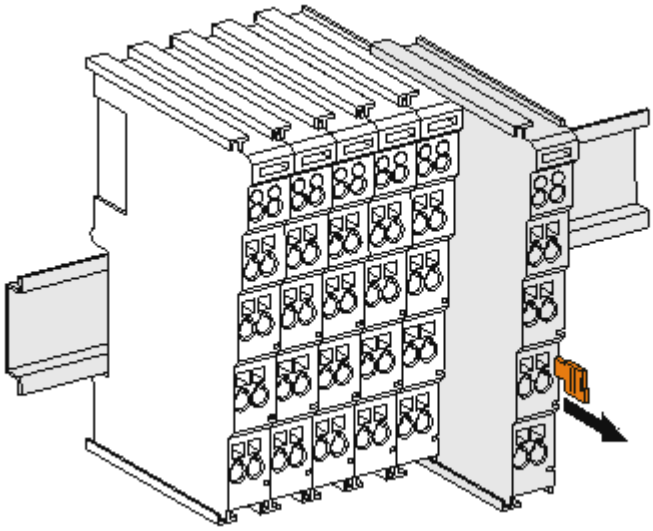


Fig. 16: Disassembling of terminal

Each terminal is secured by a lock on the mounting rail, which must be released for disassembly:

1. Pull the terminal by its orange-colored lugs approximately 1 cm away from the mounting rail. In doing so for this terminal the mounting rail lock is released automatically and you can pull the terminal out of the bus terminal block easily without excessive force.
2. Grasp the released terminal with thumb and index finger simultaneous at the upper and lower grooved housing surfaces and pull the terminal out of the bus terminal block.

Connections within a bus terminal block

The electric connections between the Bus Coupler and the Bus Terminals are automatically realized by joining the components:

- The six spring contacts of the K-Bus/E-Bus deal with the transfer of the data and the supply of the Bus Terminal electronics.
- The power contacts deal with the supply for the field electronics and thus represent a supply rail within the bus terminal block. The power contacts are supplied via terminals on the Bus Coupler (up to 24 V) or for higher voltages via power feed terminals.

Power Contacts

During the design of a bus terminal block, the pin assignment of the individual Bus Terminals must be taken account of, since some types (e.g. analog Bus Terminals or digital 4-channel Bus Terminals) do not or not fully loop through the power contacts. Power Feed Terminals (KL91xx, KL92xx or EL91xx, EL92xx) interrupt the power contacts and thus represent the start of a new supply rail.

PE power contact

The power contact labeled PE can be used as a protective earth. For safety reasons this contact mates first when plugging together, and can ground short-circuit currents of up to 125 A.

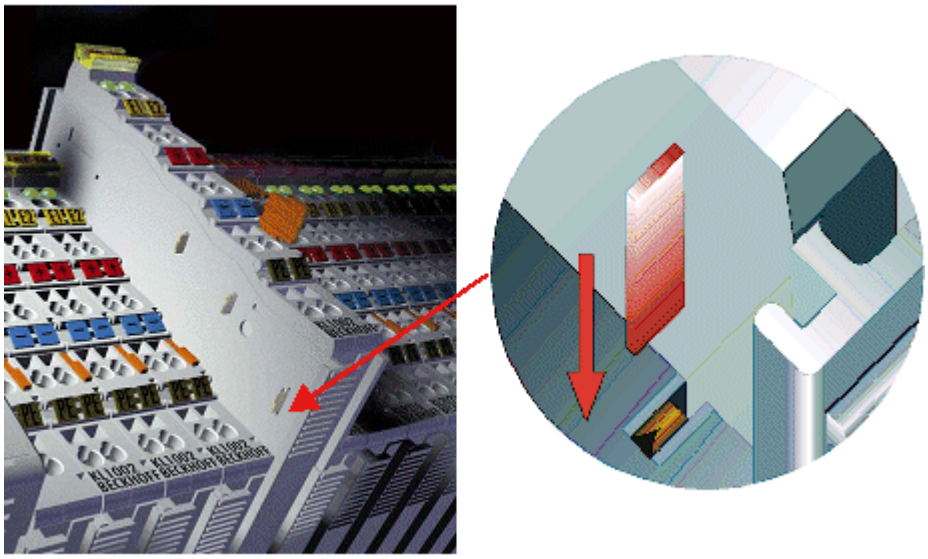


Fig. 17: Power contact on left side

NOTE

Possible damage of the device

Note that, for reasons of electromagnetic compatibility, the PE contacts are capacitatively coupled to the mounting rail. This may lead to incorrect results during insulation testing or to damage on the terminal (e.g. disruptive discharge to the PE line during insulation testing of a consumer with a nominal voltage of 230 V). For insulation testing, disconnect the PE supply line at the Bus Coupler or the Power Feed Terminal! In order to decouple further feed points for testing, these Power Feed Terminals can be released and pulled at least 10 mm from the group of terminals.

⚠ WARNING

Risk of electric shock!

The PE power contact must not be used for other potentials!

4.3 Connection

4.3.1 Connection system

⚠ 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!

Overview

The bus terminal system offers different connection options for optimum adaptation to the respective application:

- The terminals of ELxxxx and KLxxxx series with standard wiring include electronics and connection level in a single enclosure.
- The terminals of ESxxxx and KSxxxx series feature a pluggable connection level and enable steady wiring while replacing.
- The High Density Terminals (HD Terminals) include electronics and connection level in a single enclosure and have advanced packaging density.

Standard wiring (ELxxxx / KLxxxx)

Fig. 18: Standard wiring

The terminals of ELxxxx and KLxxxx series have been tried and tested for years.
They feature integrated screwless spring force technology for fast and simple assembly.

Pluggable wiring (ESxxxx / KSxxxx)

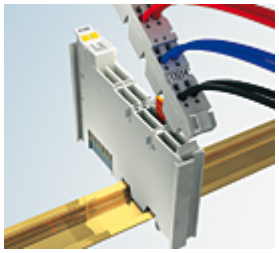


Fig. 19: Pluggable wiring

The terminals of ESxxxx and KSxxxx series feature a pluggable connection level.
The assembly and wiring procedure is the same as for the ELxxxx and KLxxxx series.
The pluggable connection level enables the complete wiring to be removed as a plug connector from the top of the housing for servicing.
The lower section can be removed from the terminal block by pulling the unlocking tab.
Insert the new component and plug in the connector with the wiring. This reduces the installation time and eliminates the risk of wires being mixed up.

The familiar dimensions of the terminal only had to be changed slightly. The new connector adds about 3 mm. The maximum height of the terminal remains unchanged.

A tab for strain relief of the cable simplifies assembly in many applications and prevents tangling of individual connection wires when the connector is removed.

Conductor cross sections between 0.08 mm² and 2.5 mm² can continue to be used with the proven spring force technology.

The overview and nomenclature of the product names for ESxxxx and KSxxxx series has been retained as known from ELxxxx and KLxxxx series.

High Density Terminals (HD Terminals)

Fig. 20: High Density Terminals

The terminals from these series with 16 terminal points are distinguished by a particularly compact design, as the packaging density is twice as large as that of the standard 12 mm bus terminals. Massive conductors and conductors with a wire end sleeve can be inserted directly into the spring loaded terminal point without tools.

Wiring HD Terminals

The High Density Terminals of the ELx8xx and KLx8xx series doesn't support pluggable wiring.

Ultrasonically “bonded” (ultrasonically welded) conductors

Ultrasonically “bonded” conductors

It is also possible to connect the Standard and High Density Terminals with ultrasonically “bonded” (ultrasonically welded) conductors. In this case, please note the tables concerning the wire-size width [▶ 34]!

4.3.2 Wiring

⚠ 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!

Terminals for standard wiring ELxxxx/KLxxxx and for pluggable wiring ESxxxx/KSxxxx

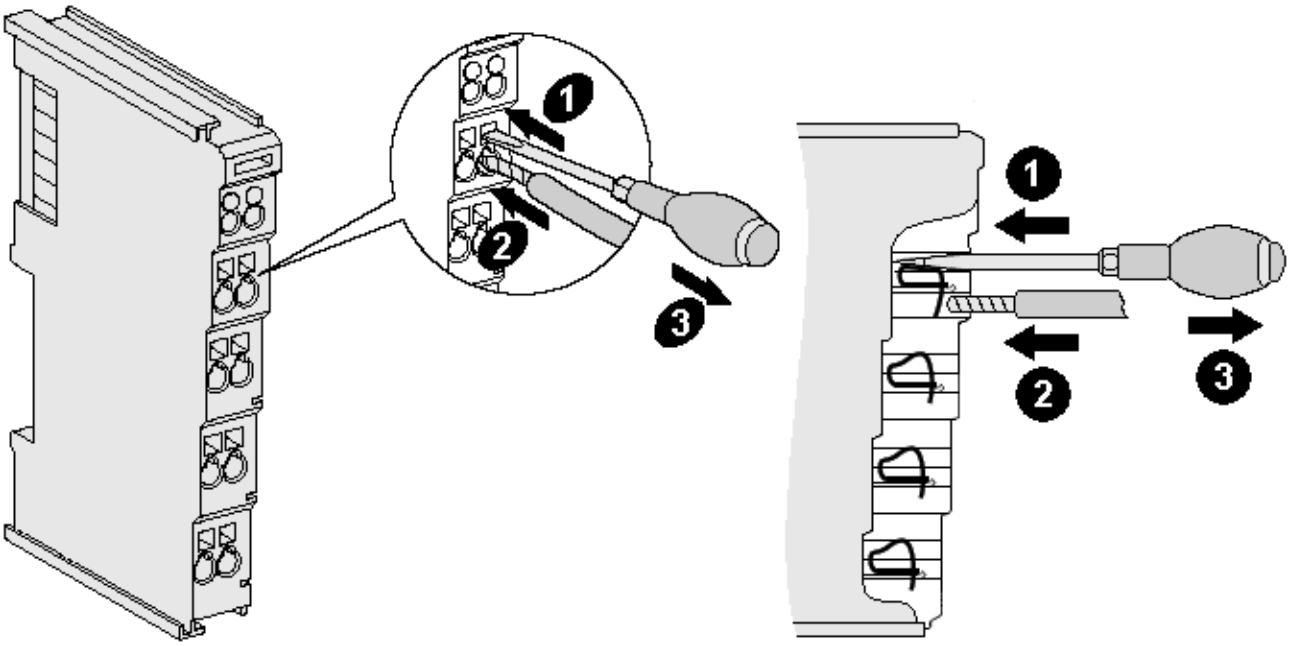


Fig. 21: Connecting a cable on a terminal point

Up to eight terminal points enable the connection of solid or finely stranded cables to the bus terminal. The terminal points are implemented in spring force technology. Connect the cables as follows:

1. Open a terminal point by pushing a screwdriver straight against the stop into the square opening above the terminal point. Do not turn the screwdriver or move it alternately (don't toggle).
2. The wire can now be inserted into the round terminal opening without any force.
3. The terminal point closes automatically when the pressure is released, holding the wire securely and permanently.

See the following table for the suitable wire size width.

Terminal housing	ELxxxx, KLxxxx	ESxxxx, KSxxxx
Wire size width (single core wires)	0.08 ... 2.5 mm²	0.08 ... 2.5 mm²
Wire size width (fine-wire conductors)	0.08 ... 2.5 mm²	0.08 ... 2.5 mm²
Wire size width (conductors with a wire end sleeve)	0.14 ... 1.5 mm²	0.14 ... 1.5 mm²
Wire stripping length	8 ... 9 mm	9 ... 10 mm

High Density Terminals (HD Terminals [▶ 33]) with 16 terminal points

The conductors of the HD Terminals are connected without tools for single-wire conductors using the direct plug-in technique, i.e. after stripping the wire is simply plugged into the terminal point. The cables are released, as usual, using the contact release with the aid of a screwdriver. See the following table for the suitable wire size width.

Terminal housing	High Density Housing
Wire size width (single core wires)	0.08 ... 1.5 mm²
Wire size width (fine-wire conductors)	0.25 ... 1.5 mm²
Wire size width (conductors with a wire end sleeve)	0.14 ... 0.75 mm²
Wire size width (ultrasonically “bonded" conductors)	only 1.5 mm² (see notice [▶ 33])
Wire stripping length	8 ... 9 mm

4.3.3 Shielding

Shielding

Encoder, analog sensors and actors should always be connected with shielded, twisted paired wires.

4.4 Note - Power supply

⚠ WARNING

Power supply from SELV/PELV power supply unit!

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

Notes:

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

4.5 Installation positions

NOTE

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.

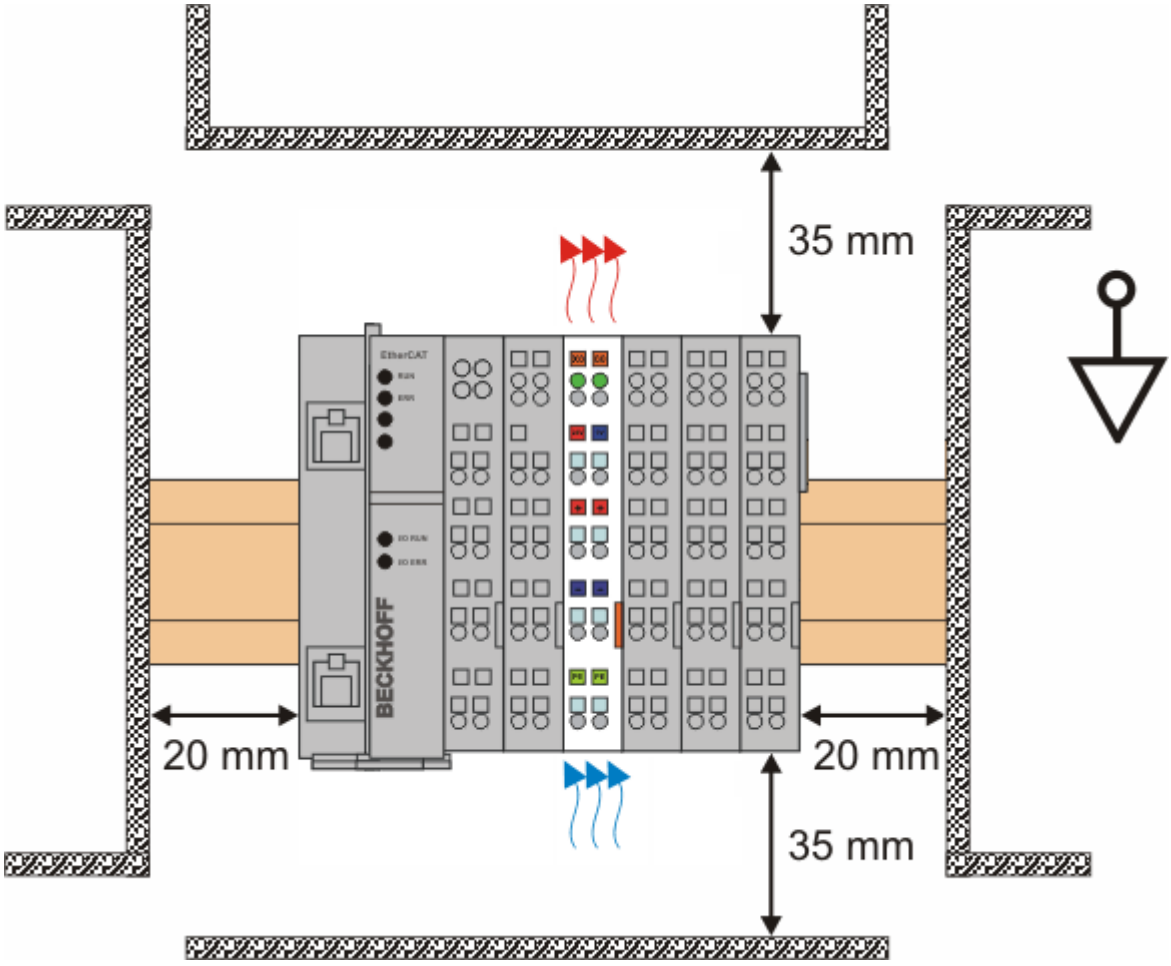


Fig. 22: Recommended distances for standard installation position

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.

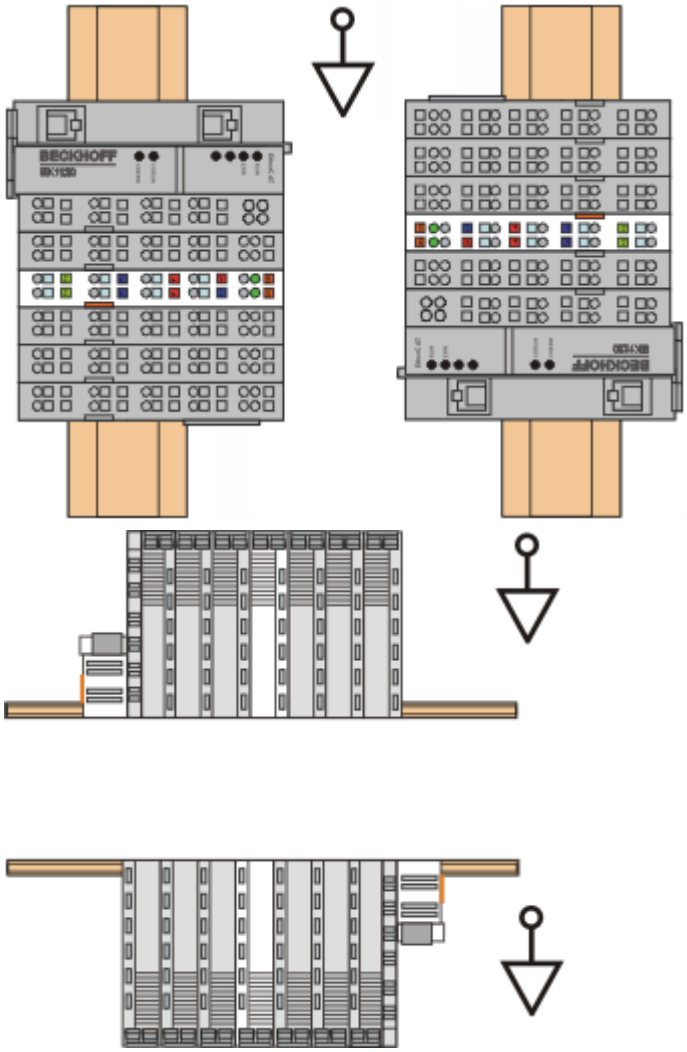


Fig. 23: Other installation positions

4.6 Positioning of passive Terminals

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)

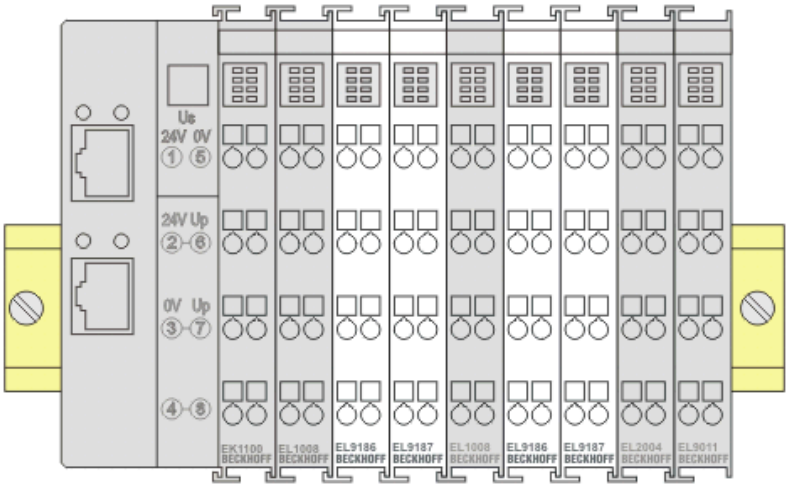


Fig. 24: Correct positioning

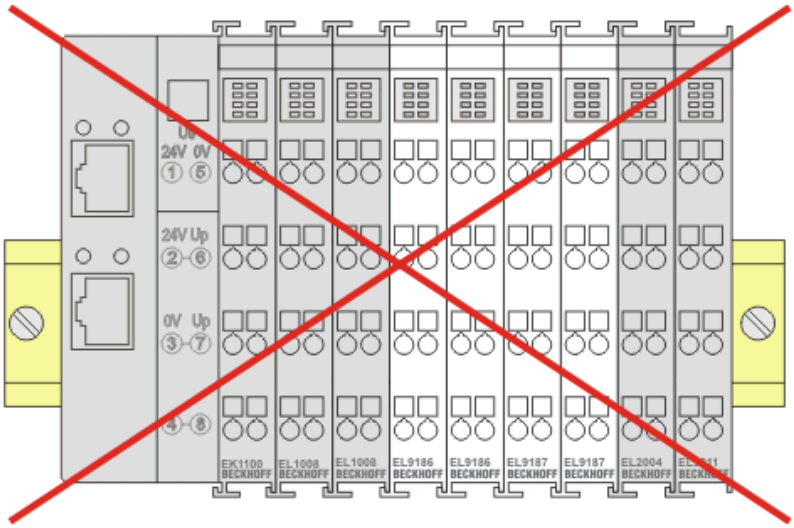


Fig. 25: Incorrect positioning

4.7 EL5072 - Connection

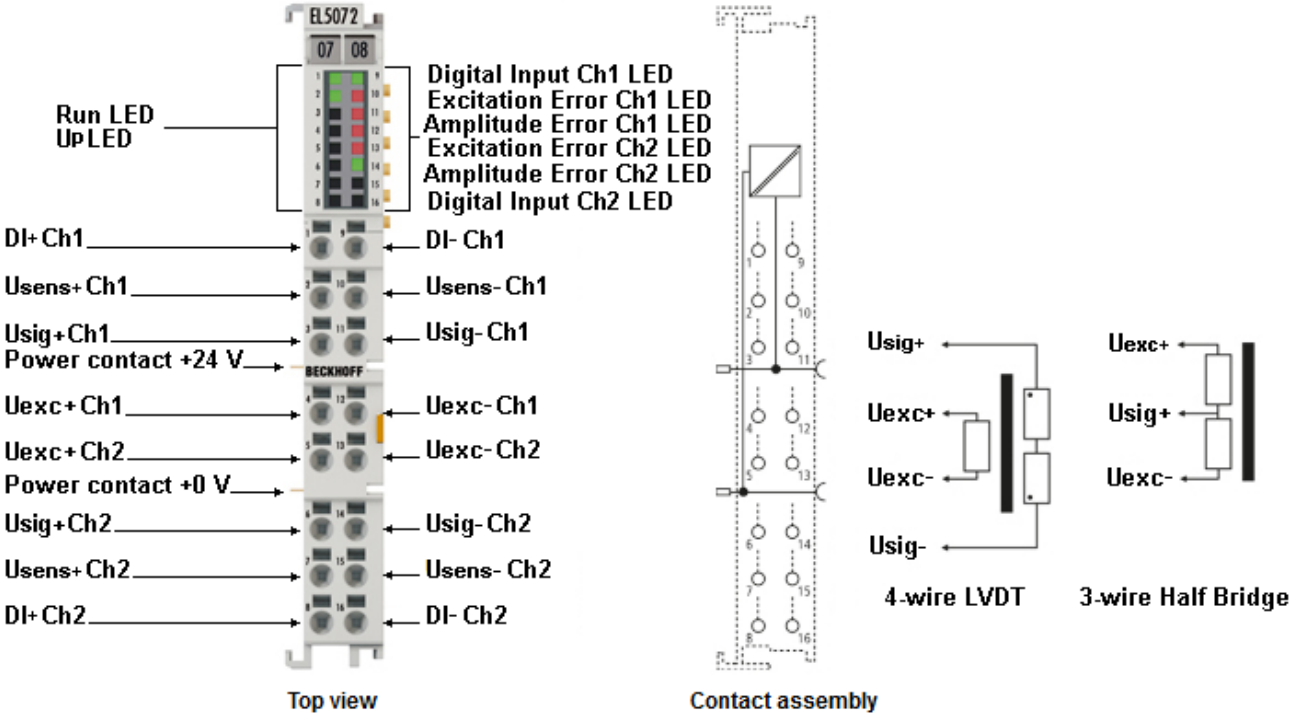


Fig. 26: EL5072

Terminal point	No.	Comment
DI+ Ch 1	1	+ digital input 24 V_{DC} channel 1
DI- Ch 1	9	- digital input channel 1
U_{sens+} Ch 1	2	+ input reference voltage channel 1
U_{sens-} Ch 1	10	- input reference voltage channel 1
U_{sig+} Ch 1	3	+ input measuring voltage channel 1
U_{sig-} Ch 1	11	- input measuring voltage channel 1
U_{exc+} Ch 1	4	+ input excitation voltage channel 1
U_{exc-} Ch 1	12	- input excitation voltage channel 1
U_{exc+} Ch 2	5	+ input excitation voltage channel 2
U_{exc-} Ch 2	13	- input excitation voltage channel 2
U_{sig+} Ch 2	6	+ input measuring voltage channel 2
U_{sig-} Ch 2	14	- input measuring voltage channel 2
U_{sens+} Ch 2	7	+ input reference voltage channel 2
U_{sens-} Ch 2	15	- input reference voltage channel 2
DI+ Ch 2	8	+ digital input 24 V_{DC} channel 2
DI- Ch 2	16	- digital input channel 2

Automatic switching of the bridges

All necessary bridges are automatically switched in the terminal. The user therefore only has to connect the lines shown in the following diagrams and make the corresponding settings in the CoE configuration data (index 0x80n1:12 "Connection type").

NOTE

Setting and activating the excitation frequency and excitation voltage

- Check the supply to the power contacts (see chapter EL5072 – Diagnostics [▶ 126]).
- Set the excitation frequency index 0x8001:14 "Excitation frequency" and excitation voltage index 0x8001:15 "Excitation voltage" centrally via the first channel. These settings are then valid for both channels.
- Before switching, make sure that both sensors support the set range!
- The excitation voltage is switched off in the delivery state and must be switched on by setting index 0x8000:08 "Enable excitation" to TRUE.

NOTE

Wiring the digital input

To ensure correct function of the digital input, in addition to the 24 V signal at connection point 1 for DI+ Ch1 or connection point 8 for DI+ Ch2, the corresponding ground connection must also be connected to connection point 10 for DI- Ch1 or connection point 16 for DI- Ch2.

4.7.1 Notes on the electrical connection of inductive measuring probes

Observe the following instructions to achieve an optimum measurement result:

- The use of suitable low-capacitance cables is recommended. Depending on the measuring method, parasitic capacitances of the individual cables have a direct influence on the accuracy of the measurement result
- If separate cables are used for the secondary coils, it is recommended to connect two cables to a center tap as close as possible to the measuring probe, so that a 5-wire LVDT signal can be evaluated at the EL5072.
- For medium cable lengths it is usually sufficient to shield only the signal line of the secondary winding.
- For longer cable lengths (> 20 m) and in cases with strong interference, the supply cable for excitation of the primary coil and the cable of the secondary winding for measurement of the signal should be routed in separate shields
- As a general rule, the supply line and the signal line should not be routed in a multi-core, jointly shielded line.
- The shield should be placed over a large area using a shielding bracket.

NOTE

Application notes on measurement deviation

In order to achieve an optimum measurement result observe the Application notes on measurement deviations [▶ 119]!

4.7.2 LVDT connection

4-wire LVDT connection

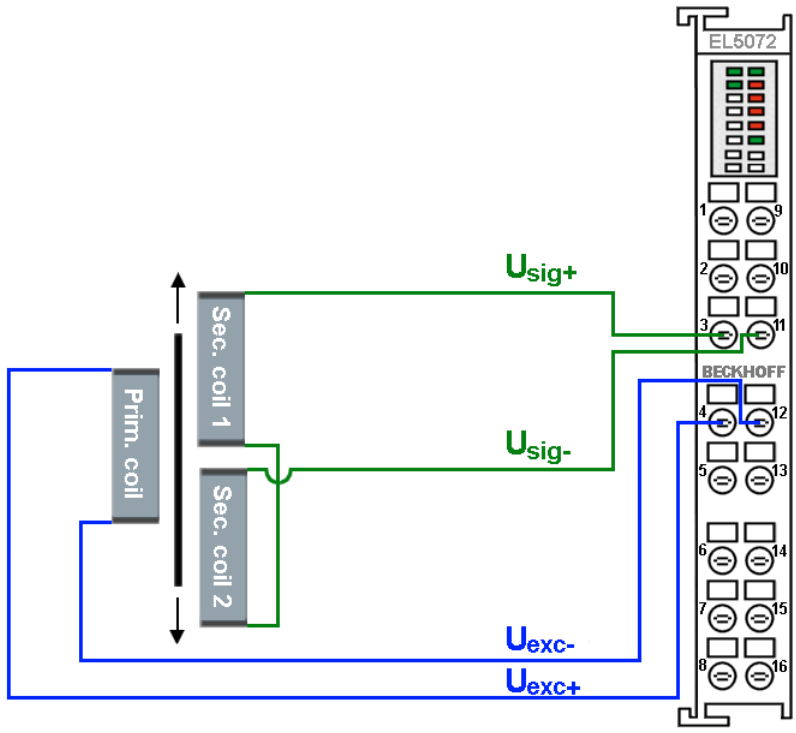


Fig. 27: Connection 4-wire LVDT

5-wire LVDT connection

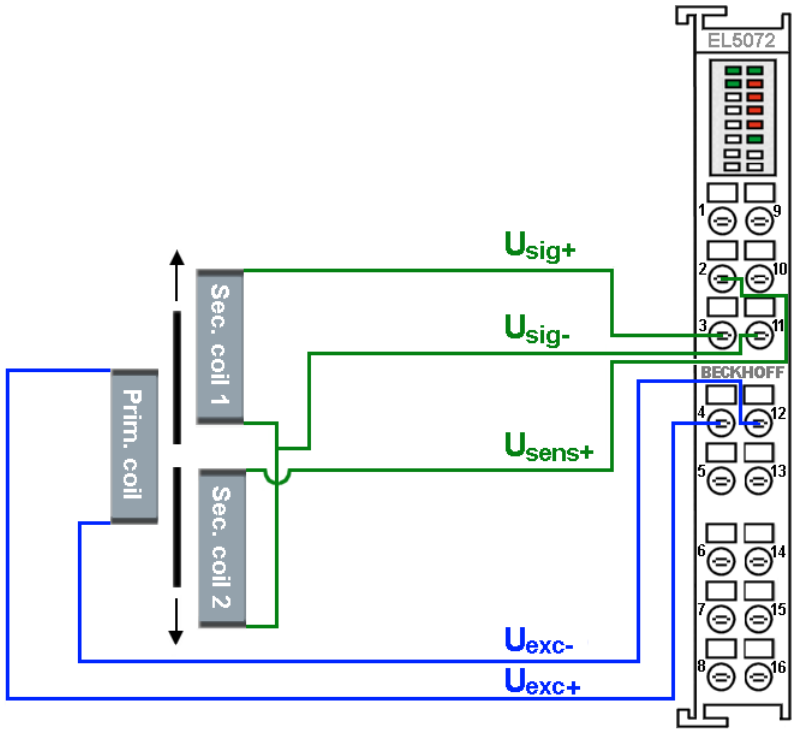


Fig. 28: Connection 5-wire LVDT

6-wire LVDT connection

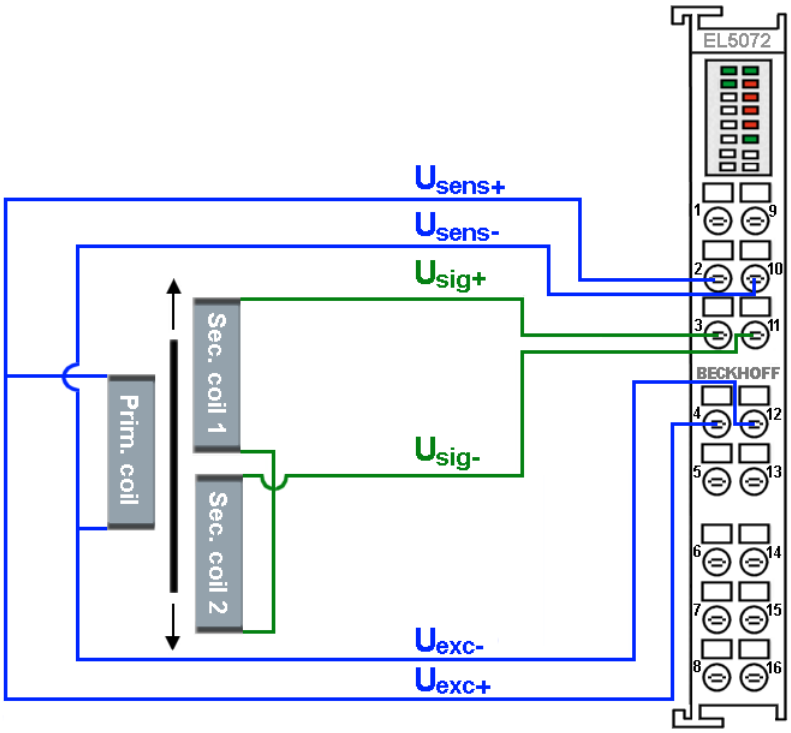


Fig. 29: Connection 6-wire LVDT

5-wire LVDT Mahr connection

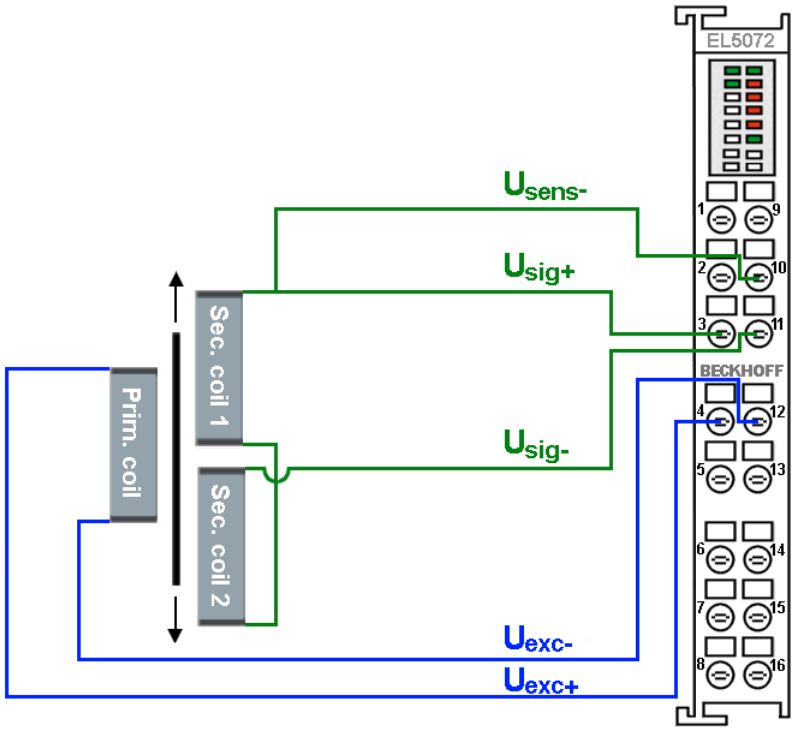


Fig. 30: Connection for 5-wire LVDT Mahr circuit

NOTE

Setting for Mahr®-compatible sensors

Mahr®-compatible sensors are connected using 5 wires. Since the fifth line is not a signal line but a virtual ground, use the 4-wire LVDT mode in index 0x80n1:12 "Connection type".

4.7.3 Half bridge - Connection

3-wire half bridge connection

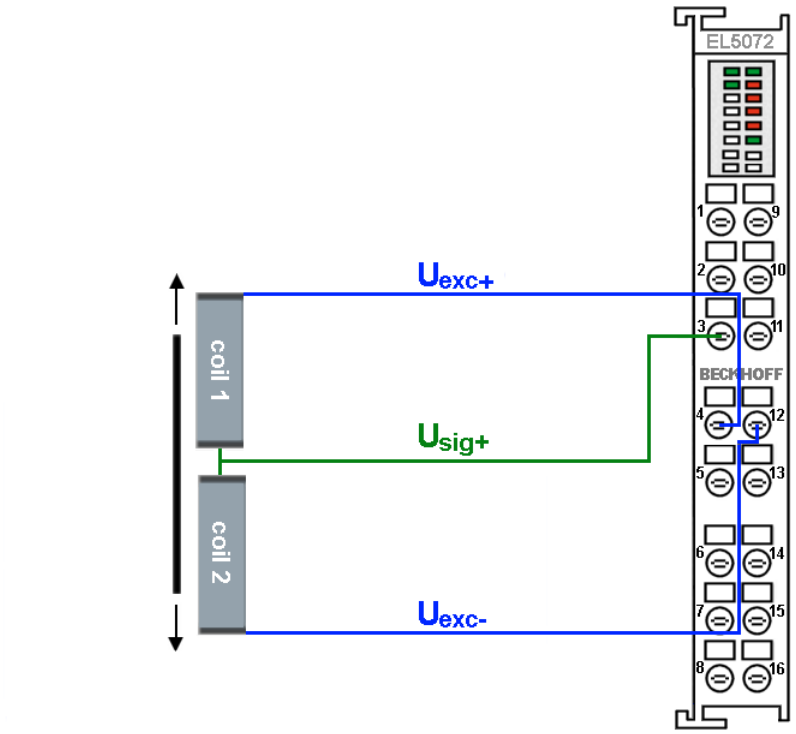


Fig. 31: Connection for 3-wire inductive half bridge

5-wire half bridge connection

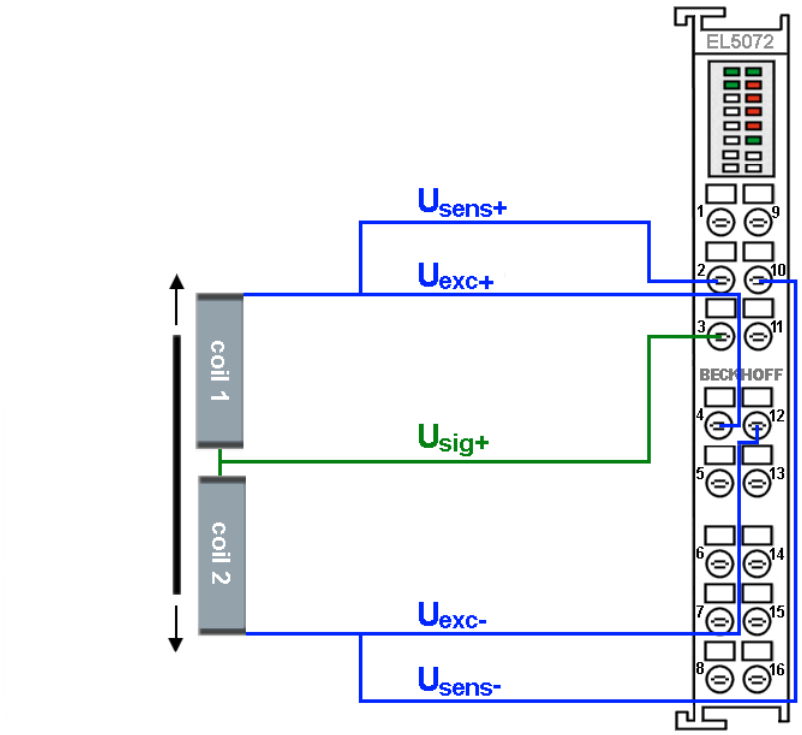


Fig. 32: Connection for 5-wire inductive half bridge

4.7.4 Variable input impedances

Different input impedances are required, depending on the sensor type and manufacturer. This information can be found in the respective sensor data sheet or can be obtained directly from the manufacturer.

The EL5072 provides three different input impedances for connecting various sensors. In each case the selection is made via the CoE configuration data (0x80n1:13 Sensor Impedance). The bridges are switched automatically.

The following diagrams show the terminal designations of the EL5072 as an example for channel 1.

High impedance / Mahrposs® impedance

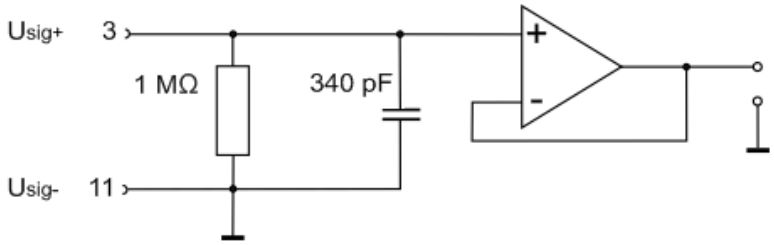


Fig. 33: Block diagram of high input impedance / Mahrposs® (channel1)

Tesa® impedance

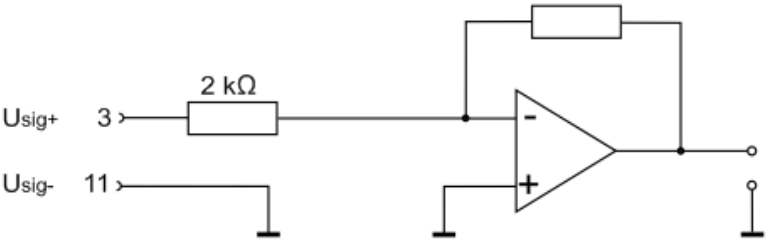


Fig. 34: Block diagram for Tesa® input impedance (channel 1)

Mahr® impedance

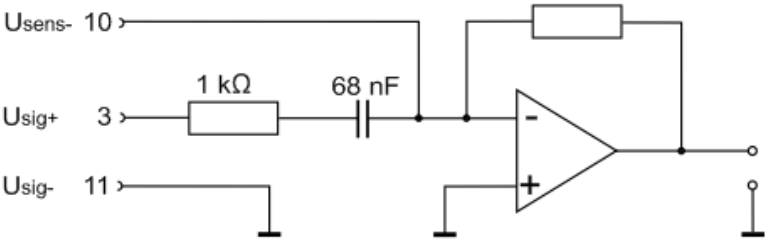


Fig. 35: Block diagram for Mahr® input impedance (channel 1)

4.8 EL5072 - LEDs

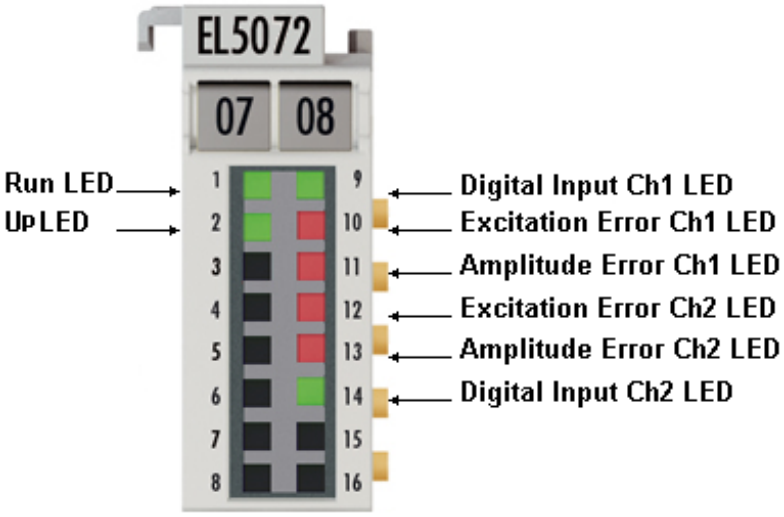


Fig. 36: EL5072 - LEDs

LED	No.	Color	Description	
Run LED	1	green	This LED indicates the terminal's operating state:	
			Off	State of the EtherCAT State Machine: **INIT** = initialization of the terminal
			Flashing	State of the EtherCAT State Machine: **PREOP =** function for mailbox communication and different default settings set
			Single flash	State of the EtherCAT State Machine: **SAFEOP** = verification of the sync manager channels and the distributed clocks. Outputs remain in safe state
			On	State of the EtherCAT State Machine: **OP =** normal operating state; mailbox and process data communication is possible
			flickering	State of the EtherCAT State Machine: **BOOTSTRAP** = function for terminal firmware updates
Up LED	2	green	Off	Supply voltage Up not present
			On	Supply voltage Up is present, excitation voltage U_{exc} is switched on
			Flashing	Supply voltage Up is present, excitation voltage U_{exc} is switched off
Digital Input Ch1 LED	9	green	On	Digital input is active
Excitation Error Ch1 LED	10	red	On	Short circuit on primary side on channel 1 or overload of excitation source was detected
Amplitude Error Ch1 LED	11	red	On	Amplitude error on secondary side on channel 1 was detected
Excitation Error Ch 2 LED	12	red	On	Short circuit on primary side on channel 2 or overload of excitation source was detected
Amplitude Error Ch 2 LED	13	red	On	Amplitude error on secondary side on channel 2 was detected
Digital Input Ch 2 LED	14	green	On	Digital input is active

4.9 Disposal

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

5 Commissioning

5.1 TwinCAT 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-Operatingsystem
  - Automatic code generation and project creation with the TwinCAT Automation Interface
  - More...

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

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

5.1.1 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.

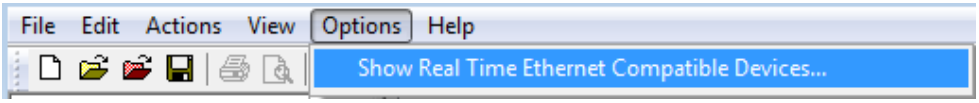


Fig. 37: System Manager “Options” (TwinCAT 2)

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

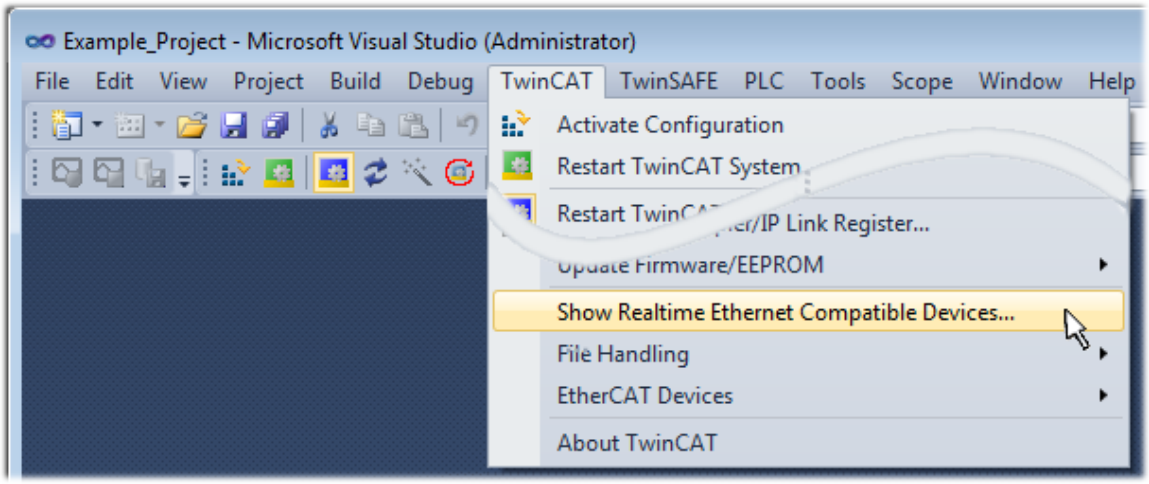


Fig. 38: Call up under VS Shell (TwinCAT 3)

B: Via TcRteInstall.exe in the TwinCAT directory

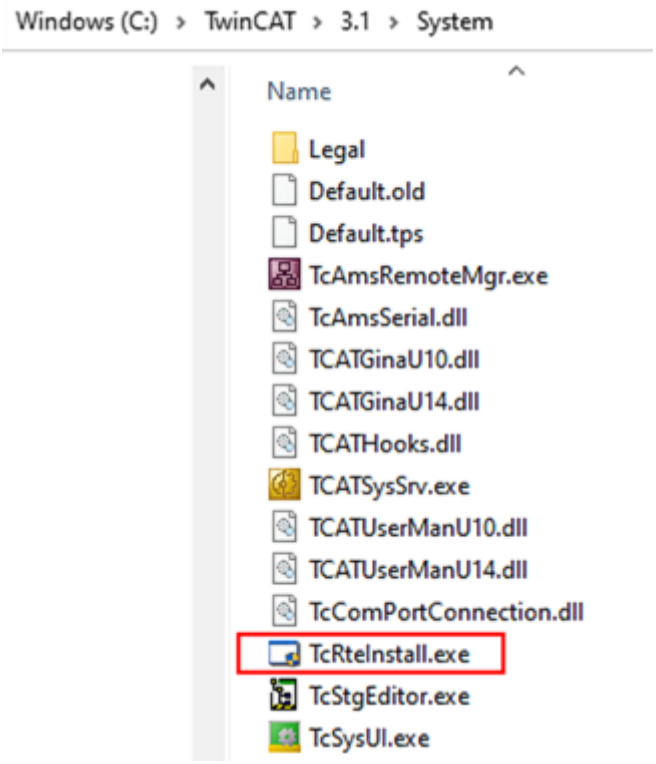


Fig. 39: TcRteInstall in the TwinCAT directory

In both cases, the following dialog appears:

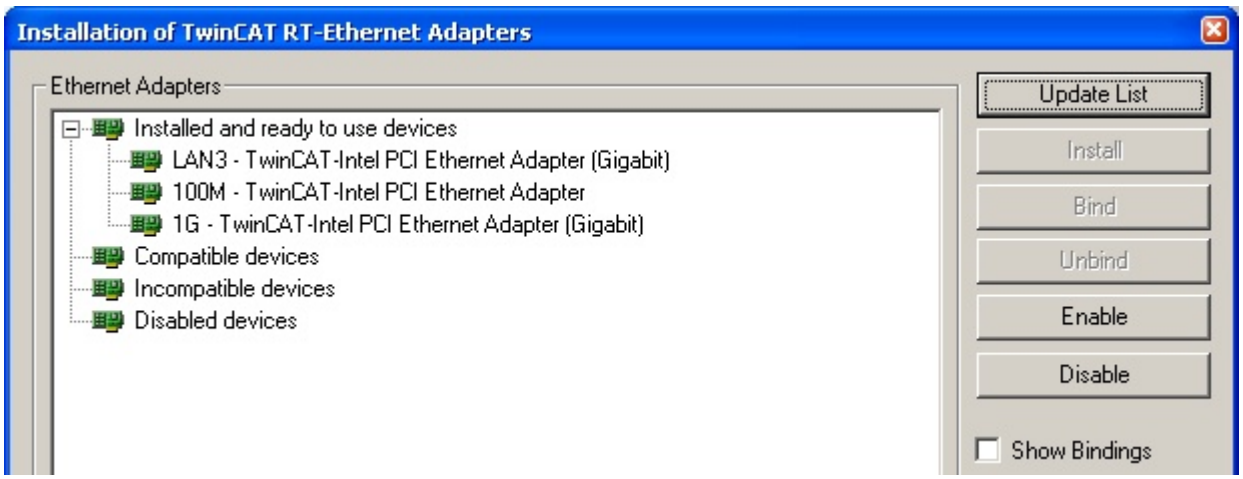


Fig. 40: Overview of network interfaces

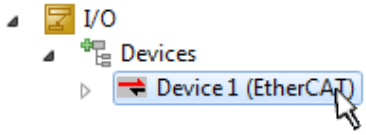
Interfaces listed under “Compatible devices” can be assigned a driver via the “Install” button. A driver should only be installed on compatible devices.

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

Fig. 41: 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)

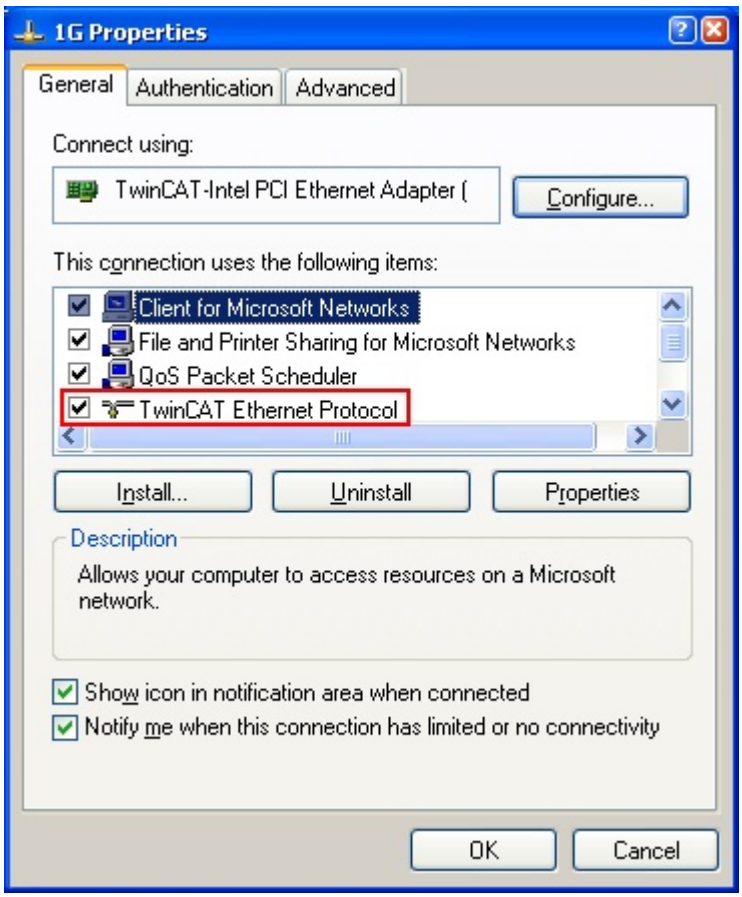


Fig. 42: Windows properties of the network interface

A correct setting of the driver could be:

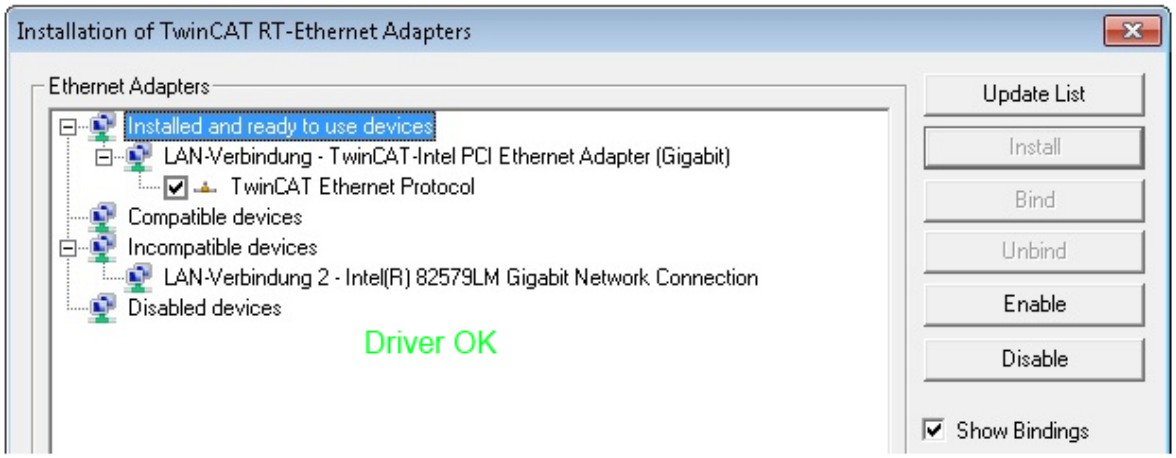


Fig. 43: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

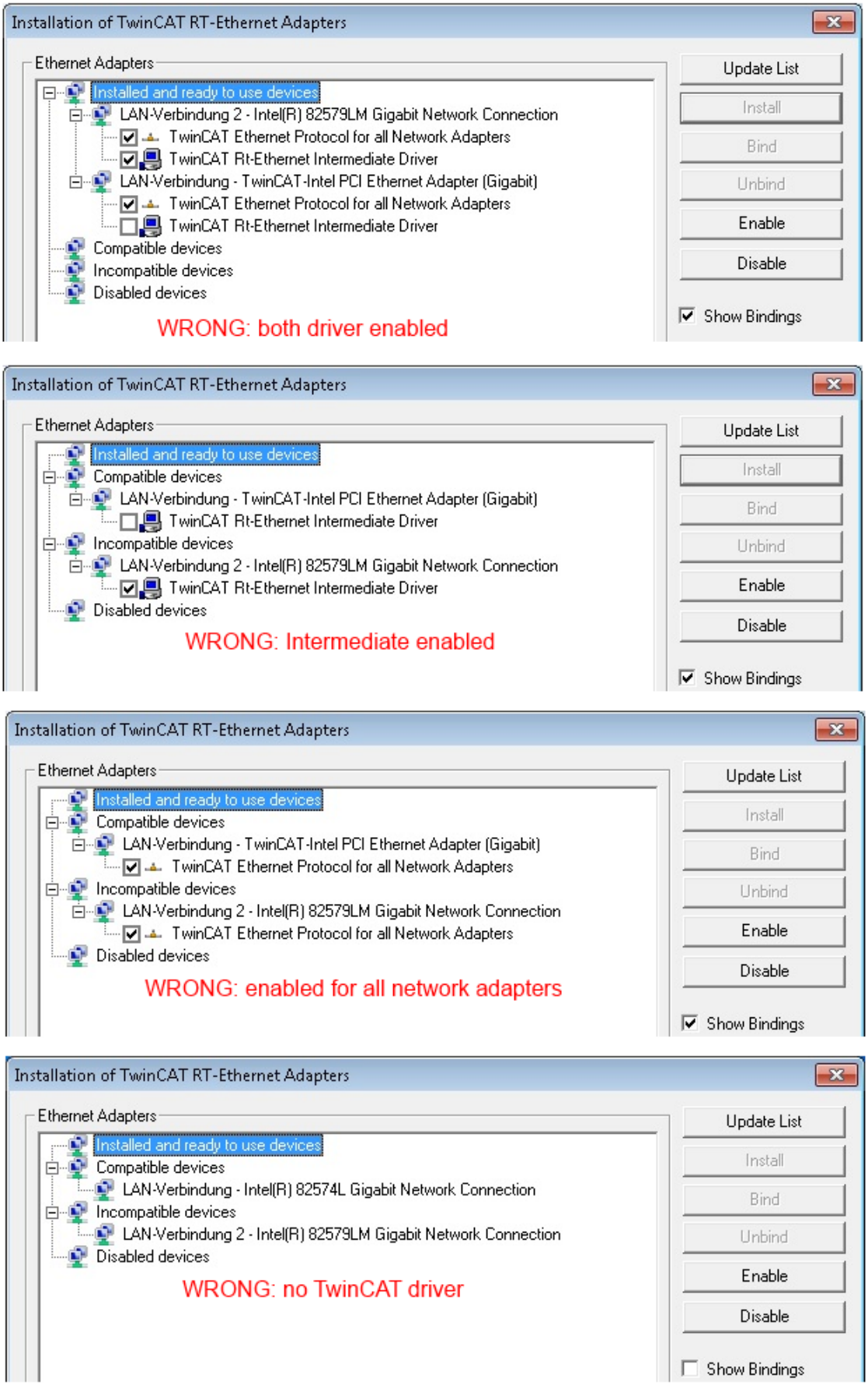


Fig. 44: Incorrect driver settings for the Ethernet port

IP address of the port used

IP address/DHCP

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

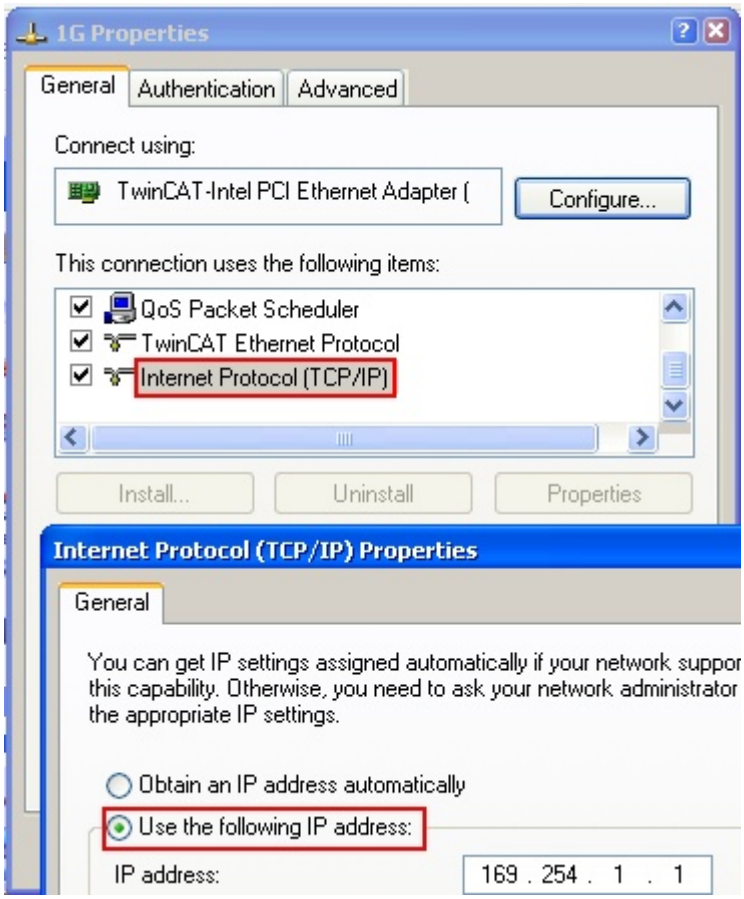


Fig. 45: TCP/IP setting for the Ethernet port

5.1.2 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 [▶ 57] 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”

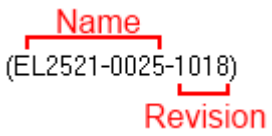


Fig. 46: Identifier structure

The order identifier consisting of name + type (here: EL2521-0010) describes the device function. The revision indicates the technical progress and is managed by Beckhoff. In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation. Each revision has its own ESI description. See further notes [▶ 7].

Online description

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

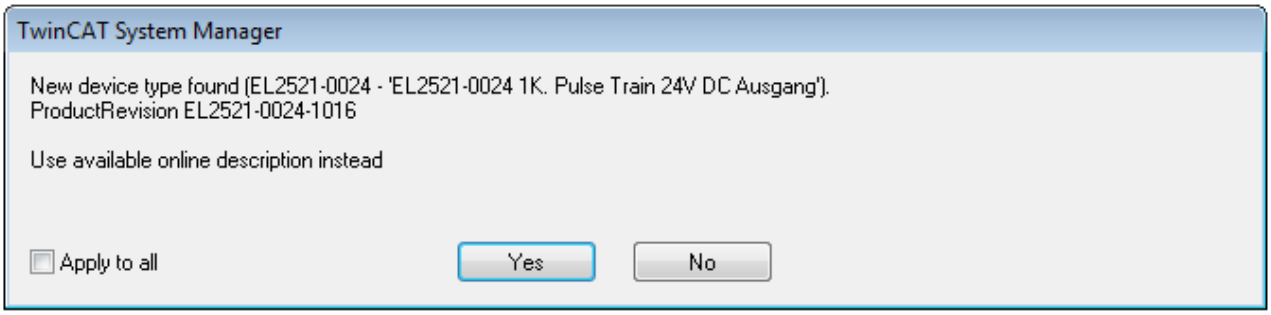


Fig. 47: OnlineDescription information window (TwinCAT 2)

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

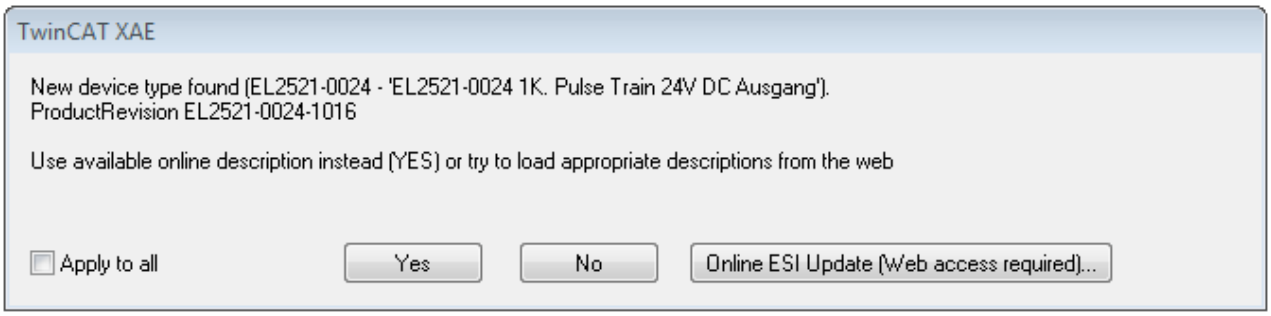


Fig. 48: Information window OnlineDescription (TwinCAT 3)

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.

NOTE

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 [▶ 58]".

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

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

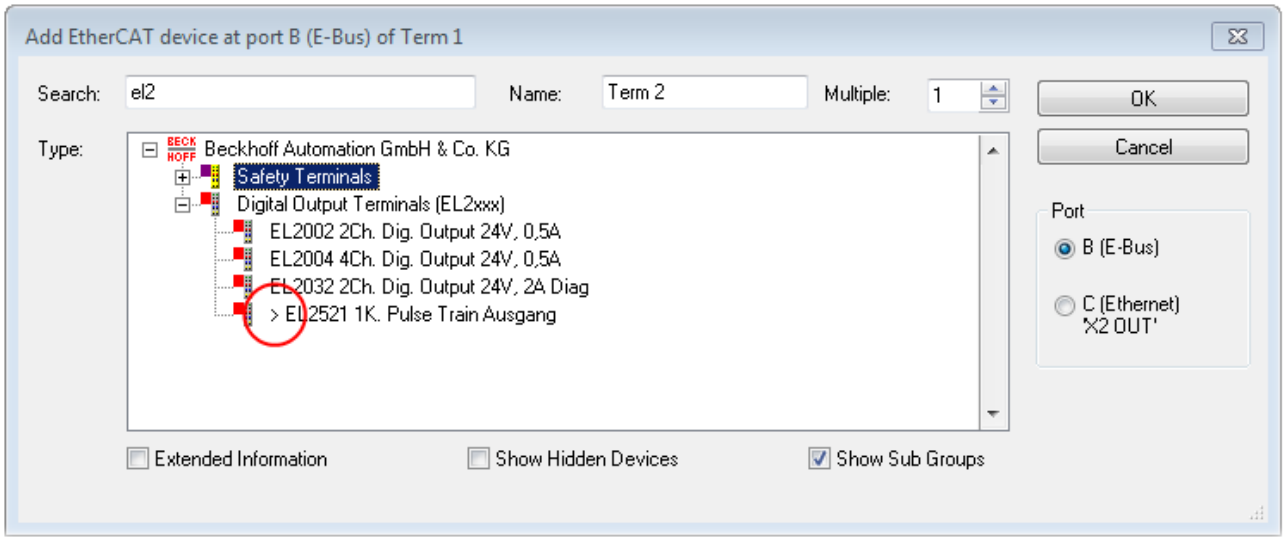


Fig. 50: Indication of an online recorded ESI of EL2521 as an example

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

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

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

OnlineDescription for TwinCAT 3.x

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

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

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

Faulty ESI file

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

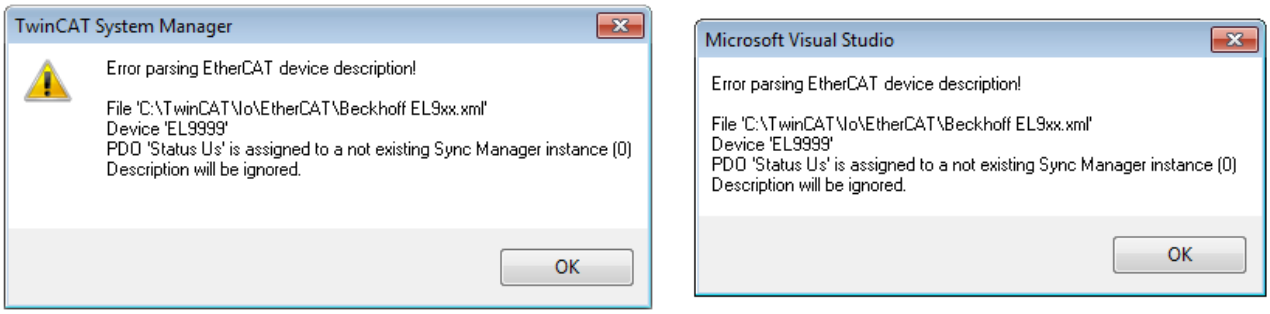


Fig. 51: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

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

5.1.3 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:

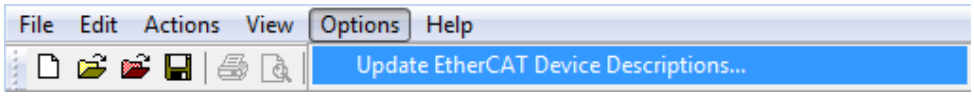


Fig. 52: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:

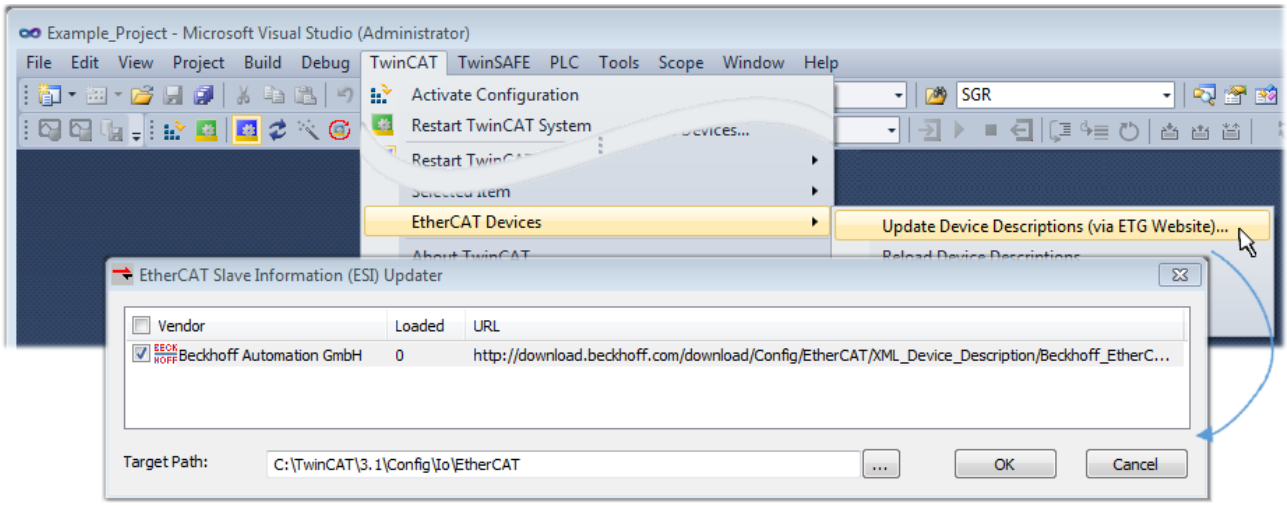


Fig. 53: Using the ESI Updater (TwinCAT 3)

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)…”.

5.1.4 Distinction between Online and Offline

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

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

In any case, during each startup the EtherCAT master checks whether the slaves it finds match the configuration. This test can be parameterised in the extended slave settings. Refer to note “Installation of the latest ESI-XML device description” [▶ 53].

For preparation of a configuration:

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

- the devices/modules be connected to the power supply and ready for communication
- TwinCAT must be in CONFIG mode on the target system.

The online scan process consists of:

- detecting the EtherCAT device [▶ 63] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 64]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 67]

The scan with existing configuration [▶ 68] can also be carried out for comparison.

5.1.5 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

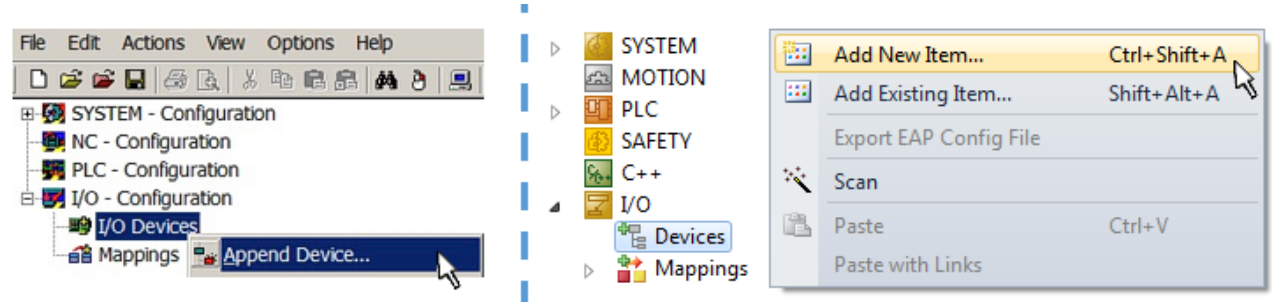


Fig. 54: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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”.

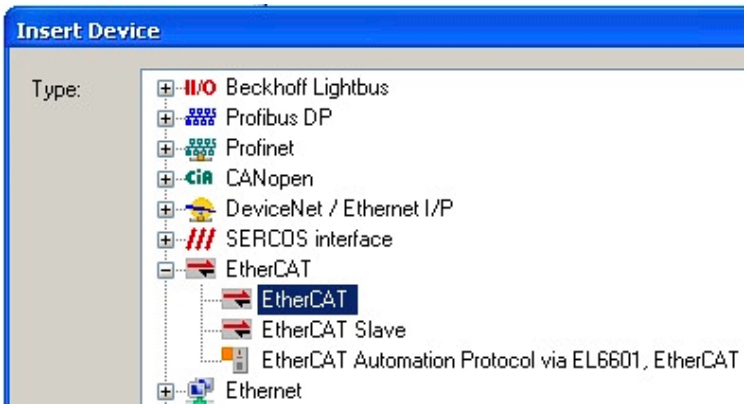


Fig. 55: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

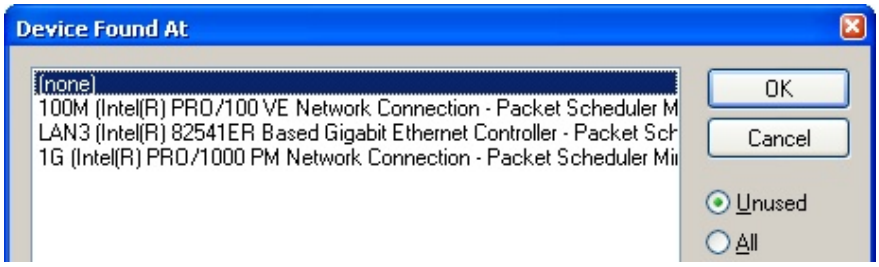


Fig. 56: Selecting the Ethernet port

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)”.

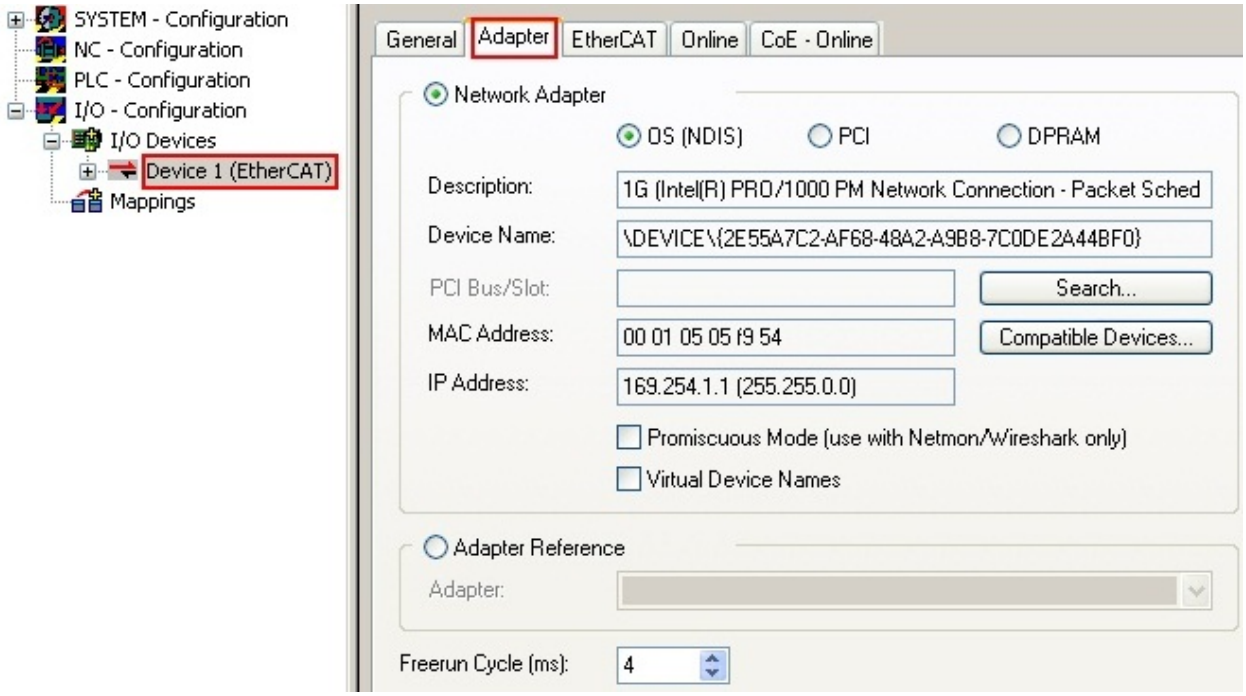
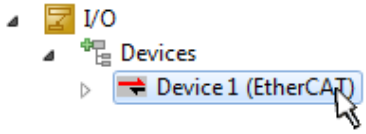


Fig. 57: EtherCAT device properties (TwinCAT 2)

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



Selecting the Ethernet port

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 [▶ 47].

Defining EtherCAT slaves

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

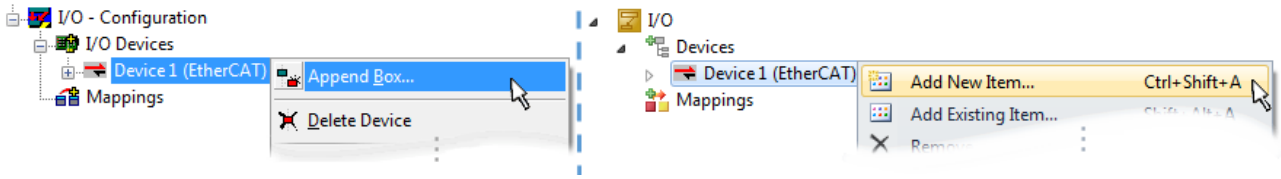


Fig. 58: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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).

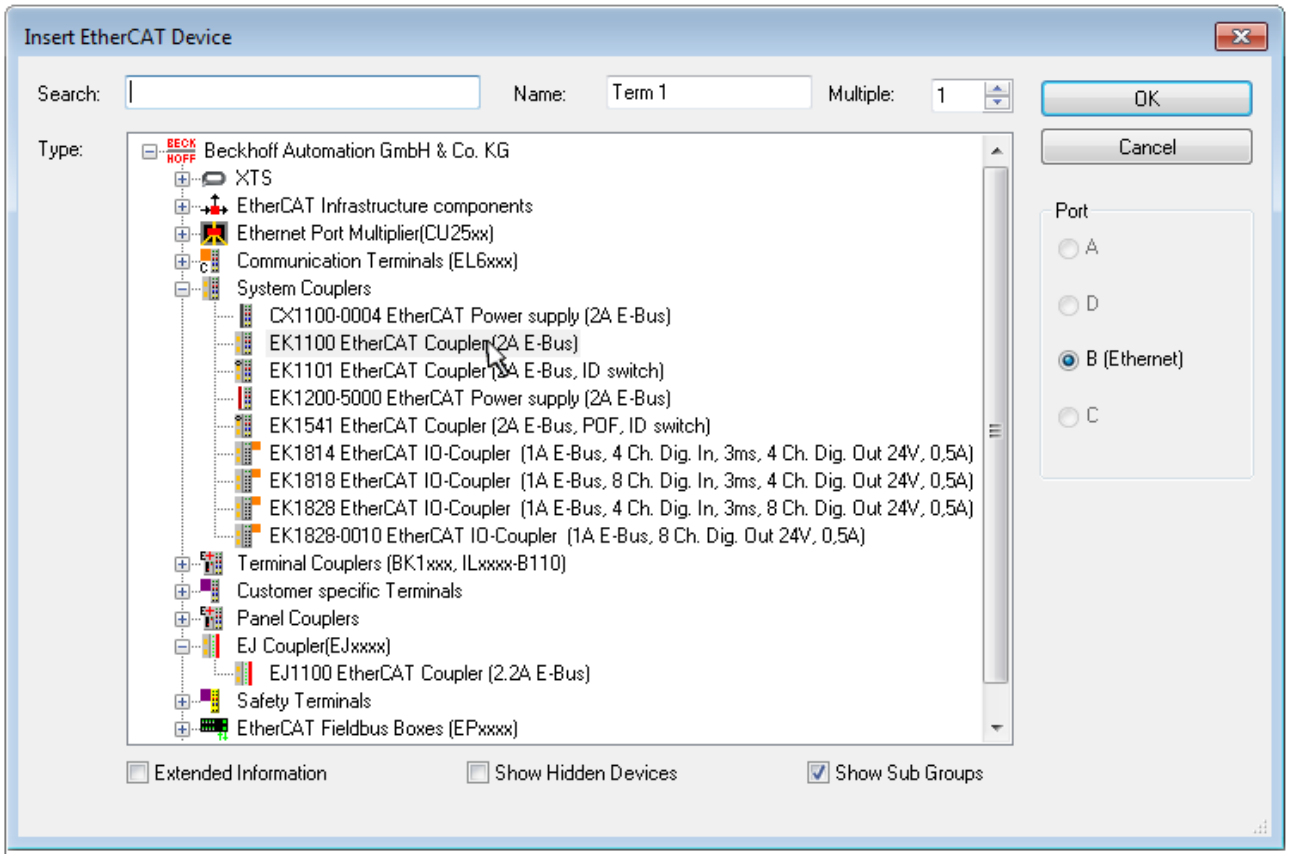


Fig. 59: Selection dialog for new EtherCAT device

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”.

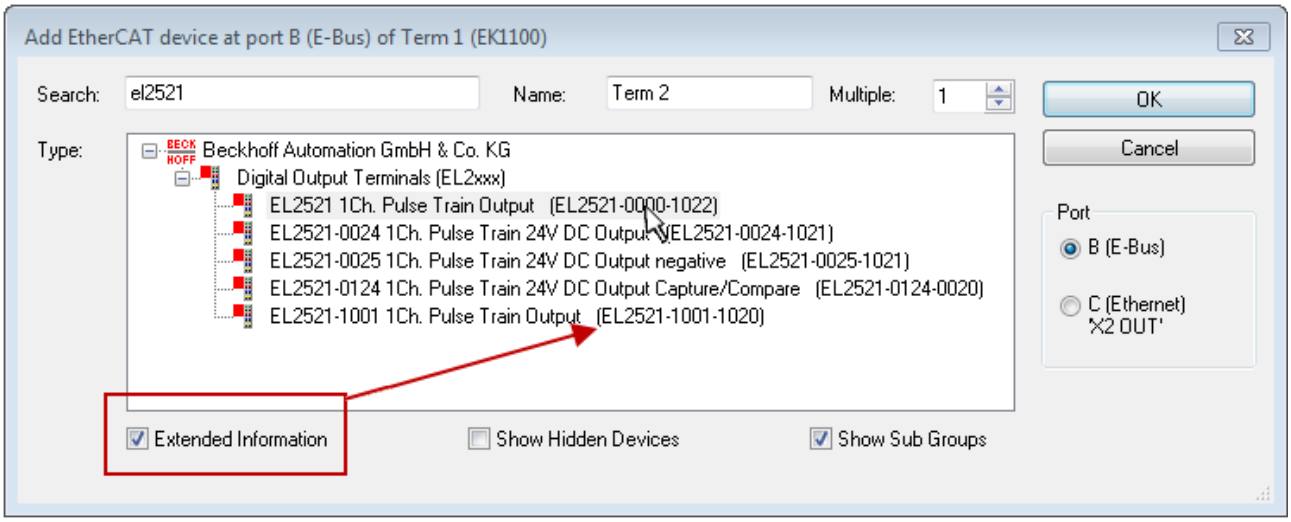


Fig. 60: Display of device revision

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”.

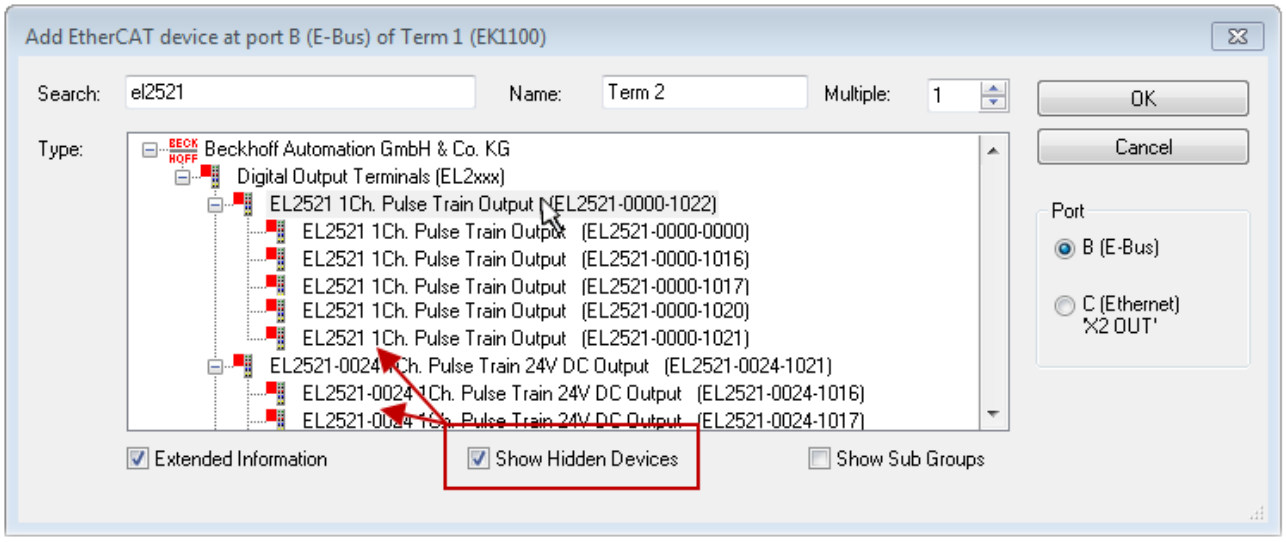


Fig. 61: Display of previous revisions

Device selection based on revision, compatibility

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

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

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

Example

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

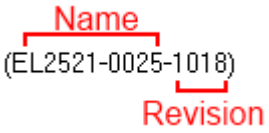


Fig. 62: Name/revision of the terminal

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, ...

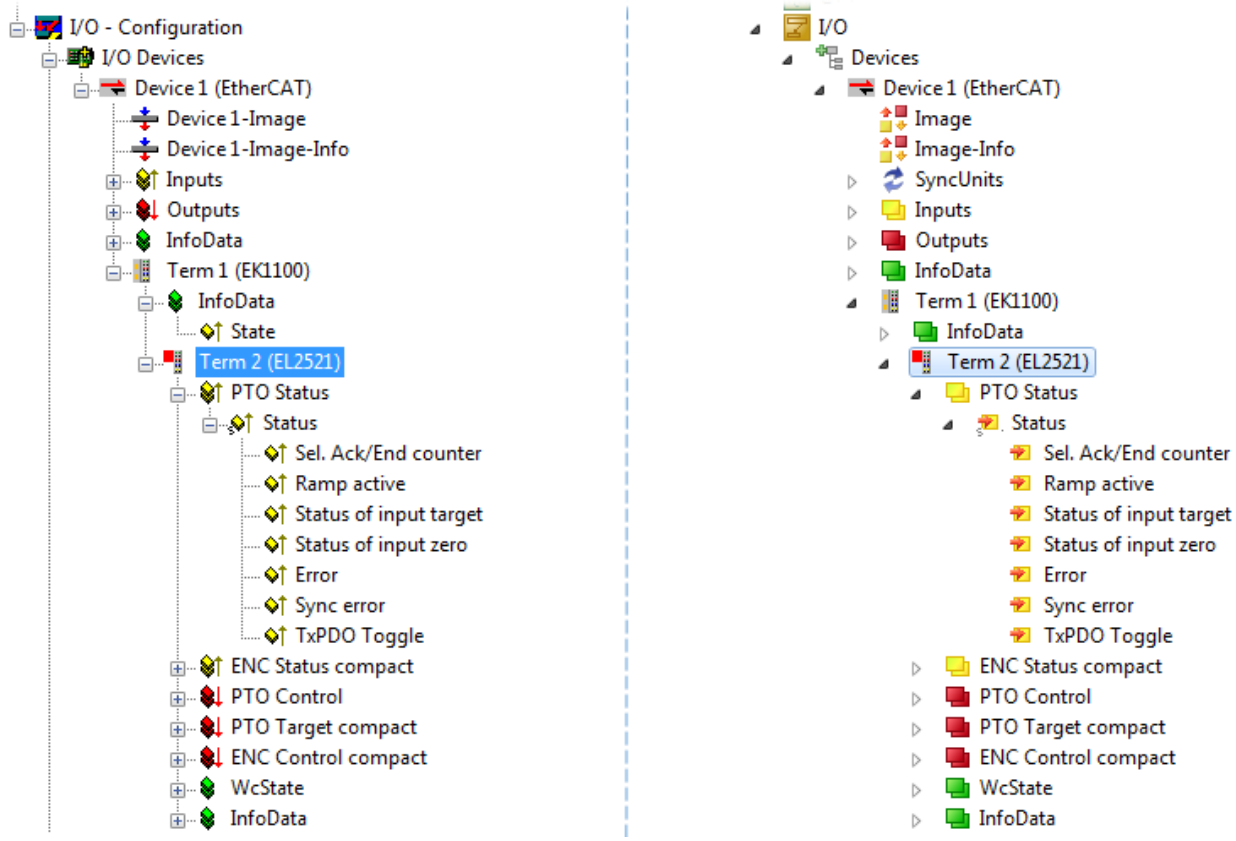


Fig. 63: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)

5.1.6 ONLINE configuration creation

Detecting/scanning of the EtherCAT device

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

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

TwinCAT can be set into this mode:

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

Online scanning in Config mode

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

The TwinCAT 2 icon ( ) or TwinCAT 3 icon ( ) within the Windows-Taskbar always shows the TwinCAT mode of the local IPC. Compared to that, the System Manager window of TwinCAT 2 or the user interface of TwinCAT 3 indicates the state of the target system.

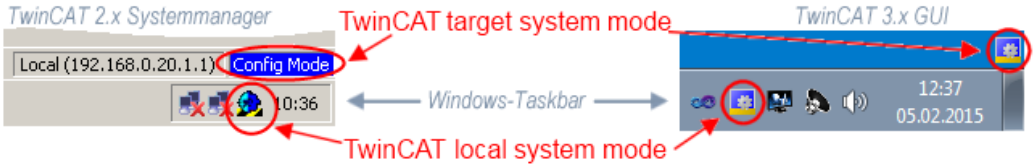


Fig. 64: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

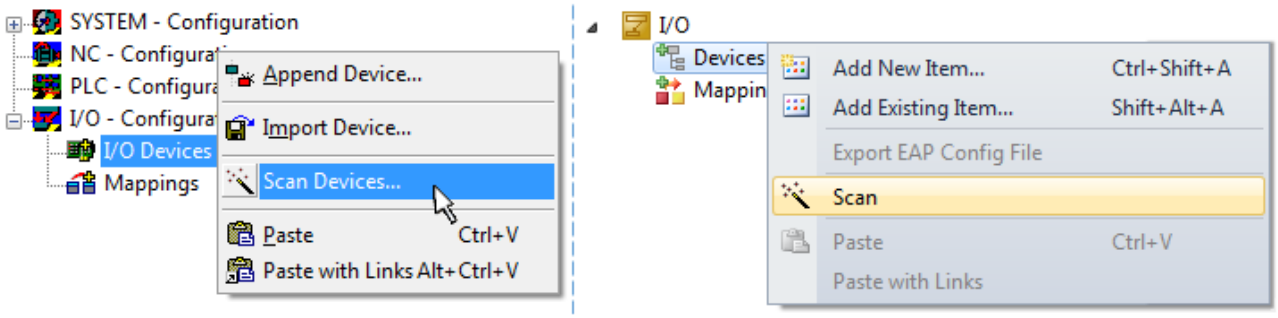


Fig. 65: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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.

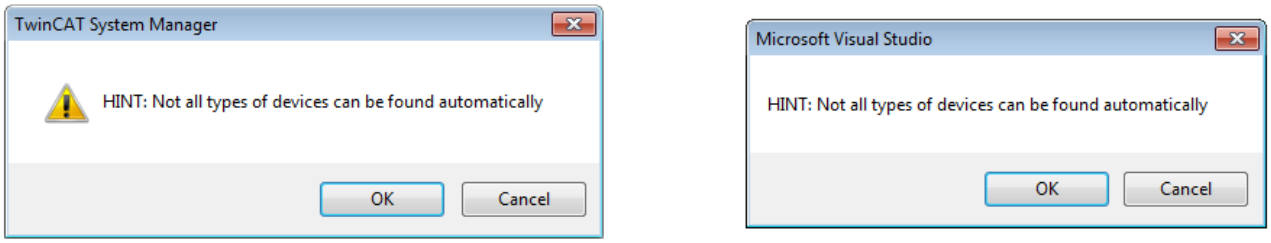


Fig. 66: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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” .

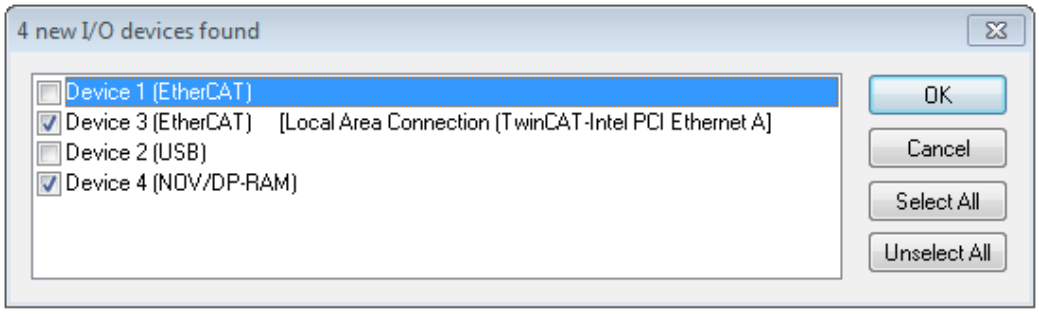


Fig. 67: Detected Ethernet devices

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 [▶ 47].

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.

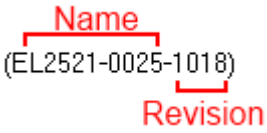


Fig. 68: Example default state

NOTE

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 [▶ 68] 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:

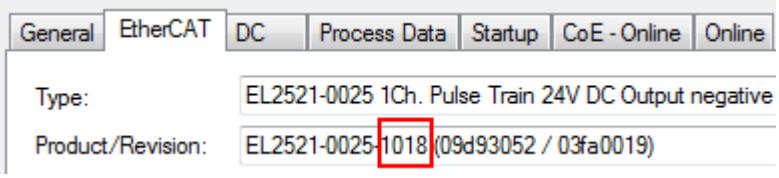


Fig. 69: Installing EthetCAT terminal with revision -1018

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

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

After some time Beckhoff extends the EL2521-0025 by a new feature C. Therefore the FW is changed, outwardly recognizable by a higher FW version and **a new revision -1019**. Nevertheless the new device naturally supports functions and interfaces of the predecessor version(s); an adaptation of “B.tsm” or even “B.pro” is therefore unnecessary. The series-produced machines can continue to be built with “B.tsm” and “B.pro”; it makes sense to perform a comparative scan [▶ 68] 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:

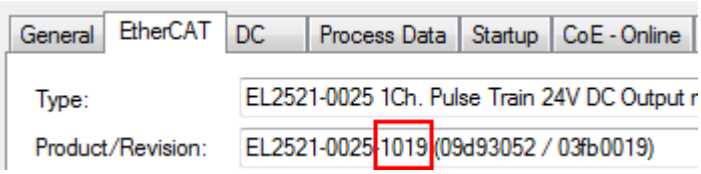


Fig. 70: Detection of EtherCAT terminal with revision -1019

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

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

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

Fig. 71: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

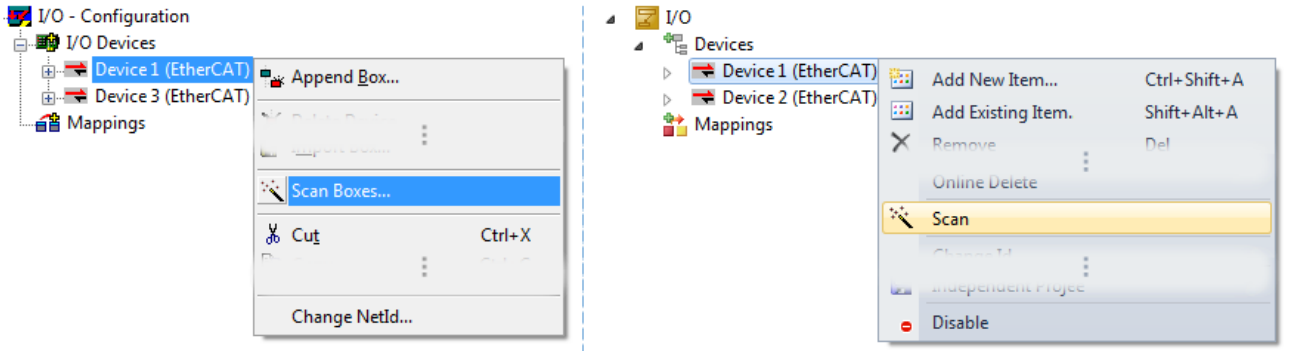


Fig. 72: Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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

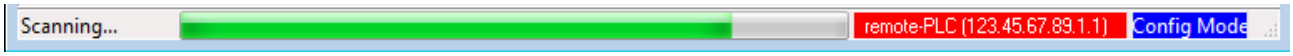


Fig. 73: *Scan progressexemplary by TwinCAT 2*

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

Fig. 74: 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).

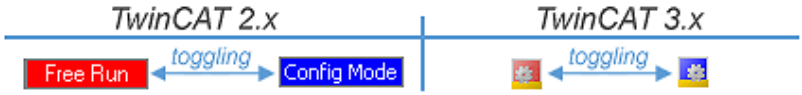


Fig. 75: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar

Fig. 76: 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*.

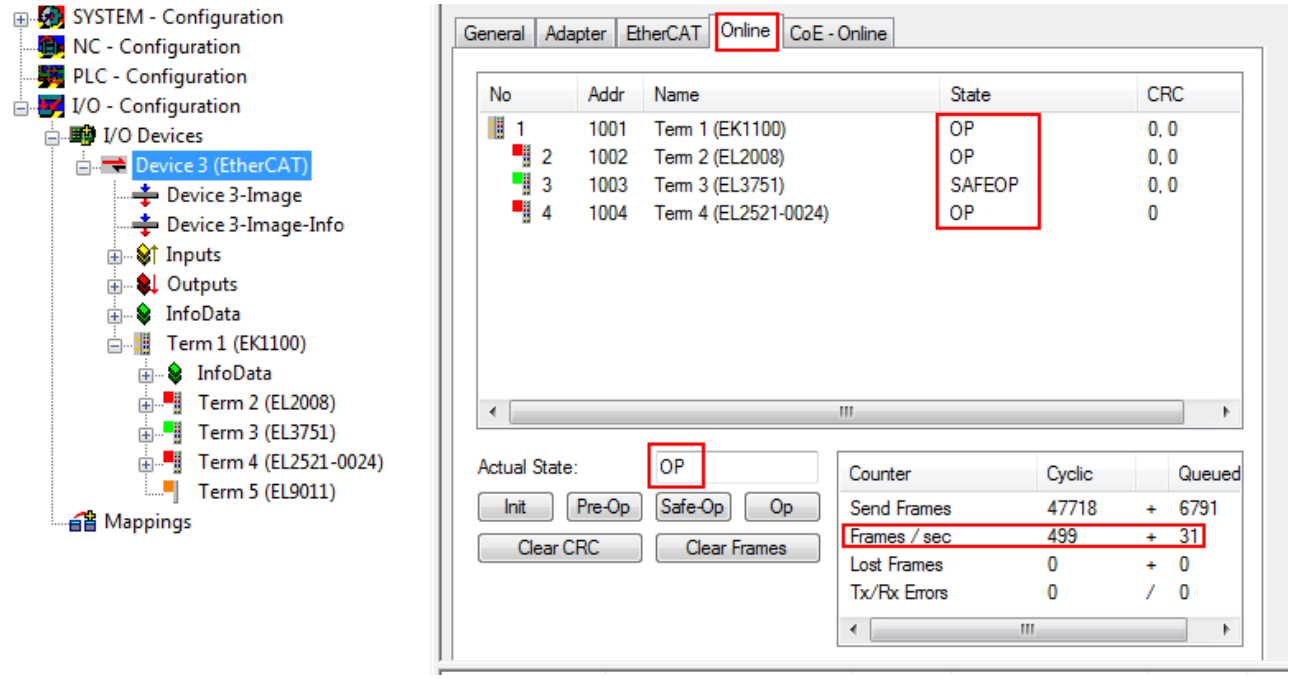


Fig. 77: Online display example

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 [▶ 58].

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.

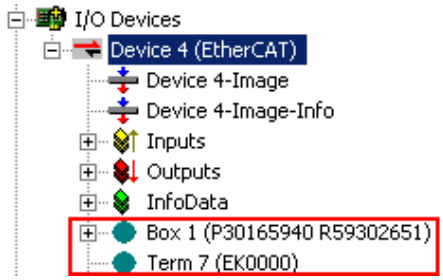


Fig. 78: Faulty identification

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

NOTE

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. 79: 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.

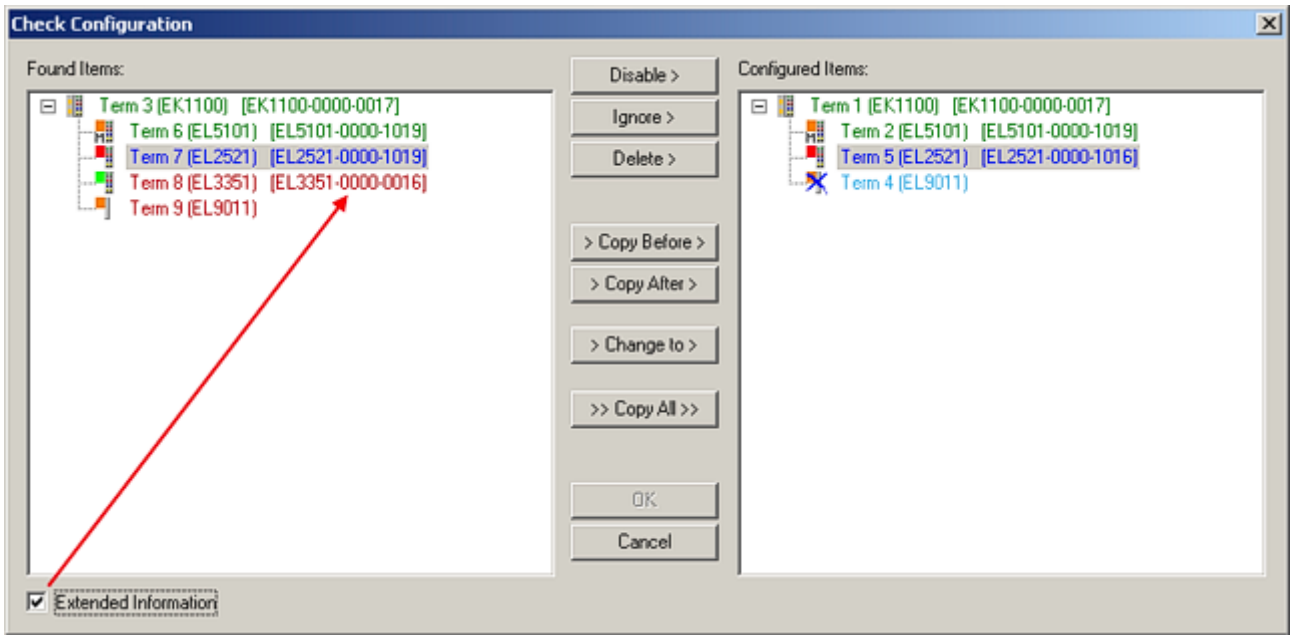


Fig. 80: Correction dialog

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

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

Device selection based on revision, compatibility

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

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

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

Example

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

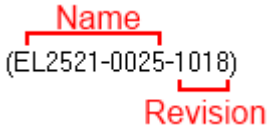


Fig. 81: Name/revision of the terminal

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, ...

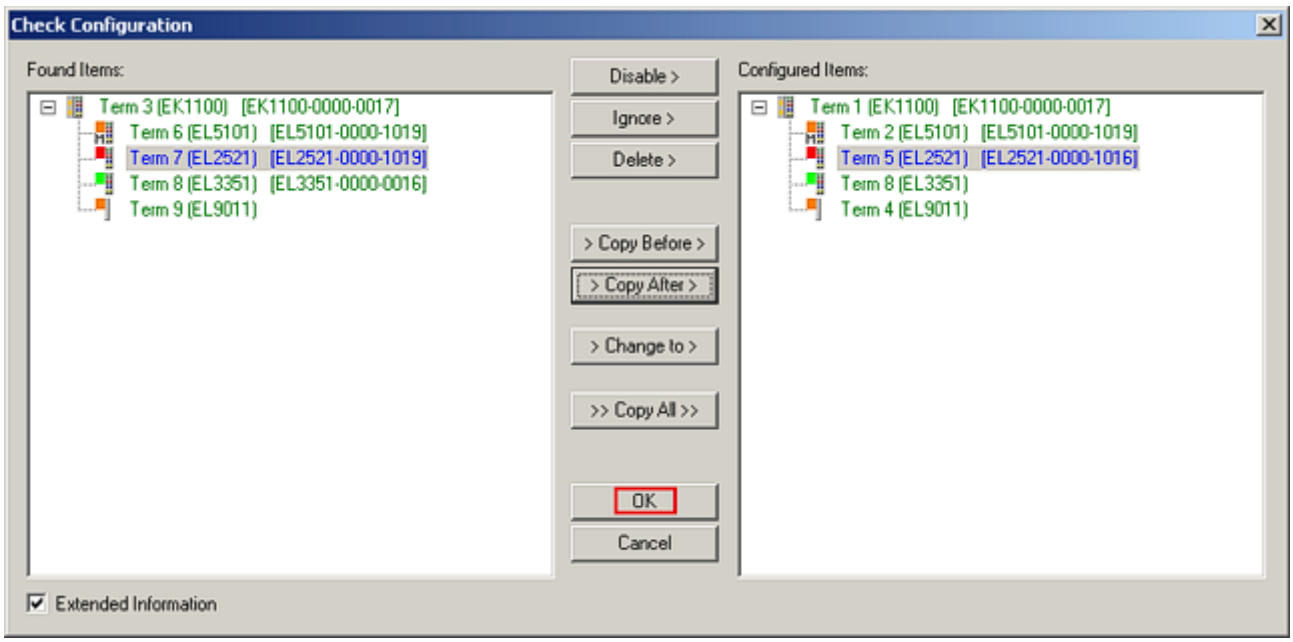


Fig. 82: Correction dialog with modifications

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.

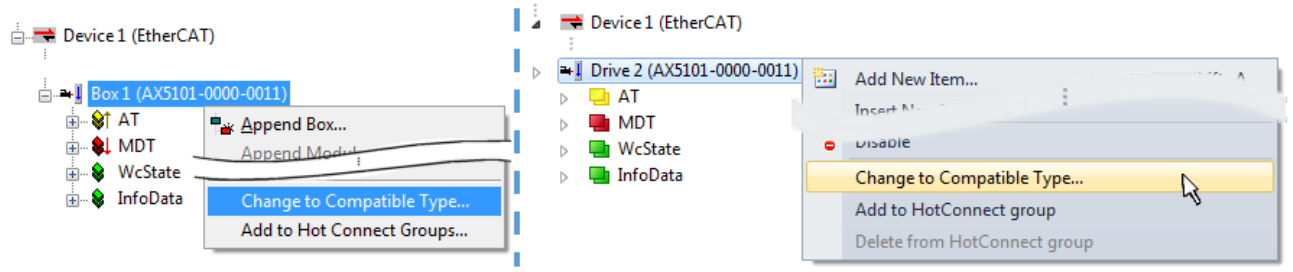


Fig. 83: Dialog “Change to Compatible Type…” (left: TwinCAT 2; right: TwinCAT 3)

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

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

- FMMU (additional ones are allowed)

- SyncManager (SM, additional ones are allowed)

- EoE (attributes MAC, IP)

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

- FoE

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

This function is preferably to be used on AX5000 devices.

Change to Alternative Type

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

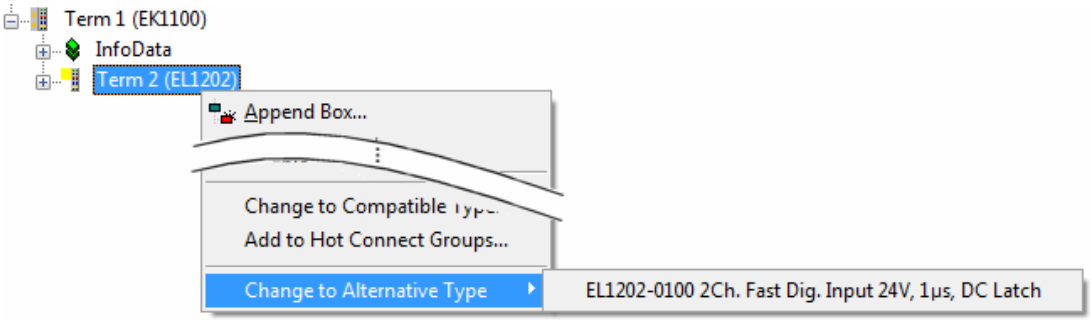


Fig. 84: TwinCAT 2 Dialog Change to Alternative Type

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).

5.1.7 EtherCAT subscriber configuration

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

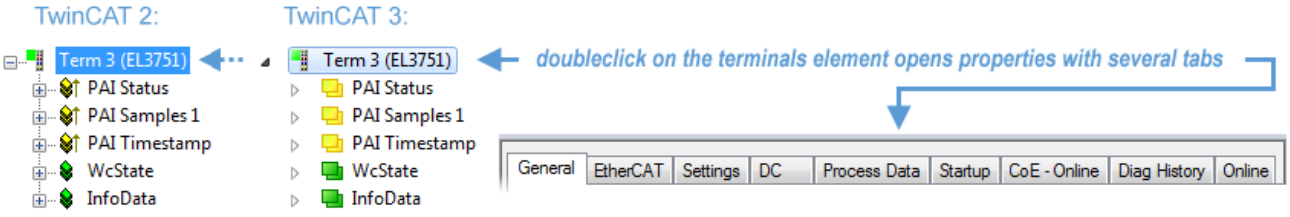


Fig. 85: Branch element as terminal EL3751

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

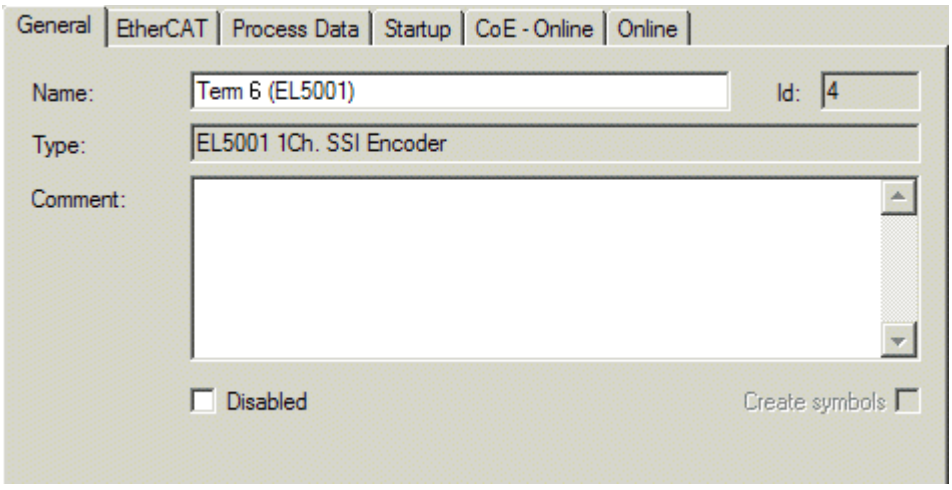


Fig. 86: “General” tab

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

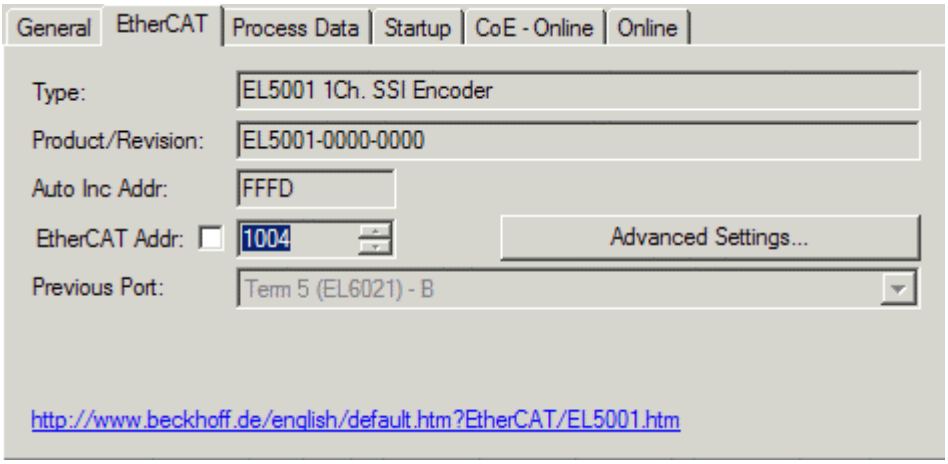


Fig. 87: "EtherCAT" tab

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

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

"Process Data" tab

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

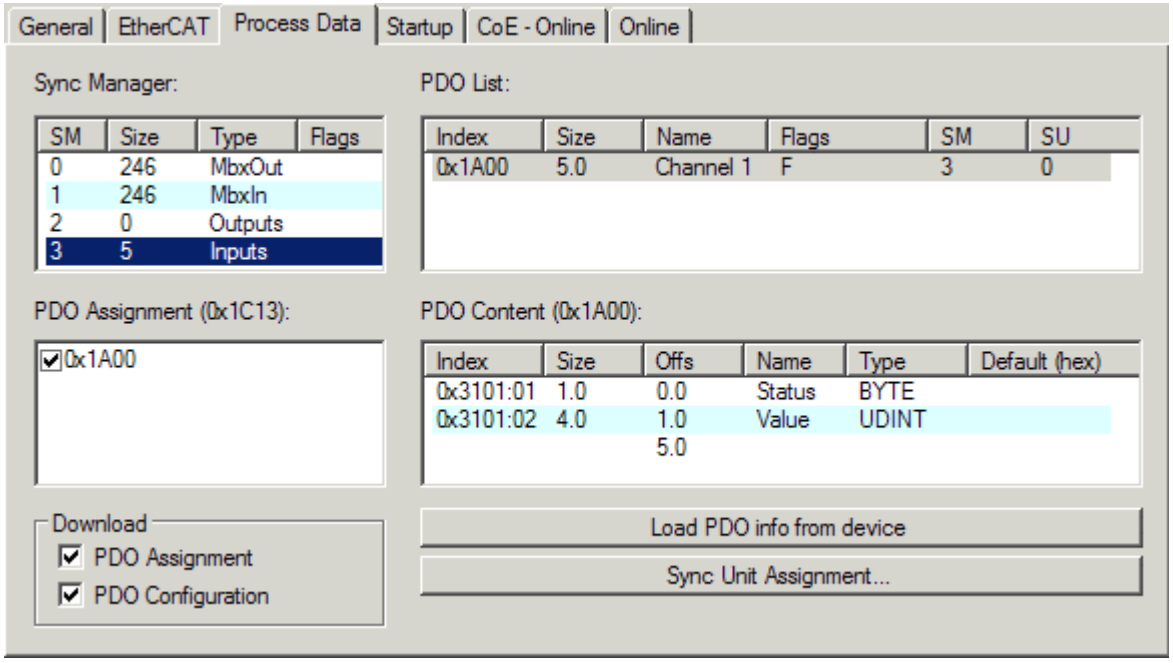


Fig. 88: “Process Data” tab

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”).

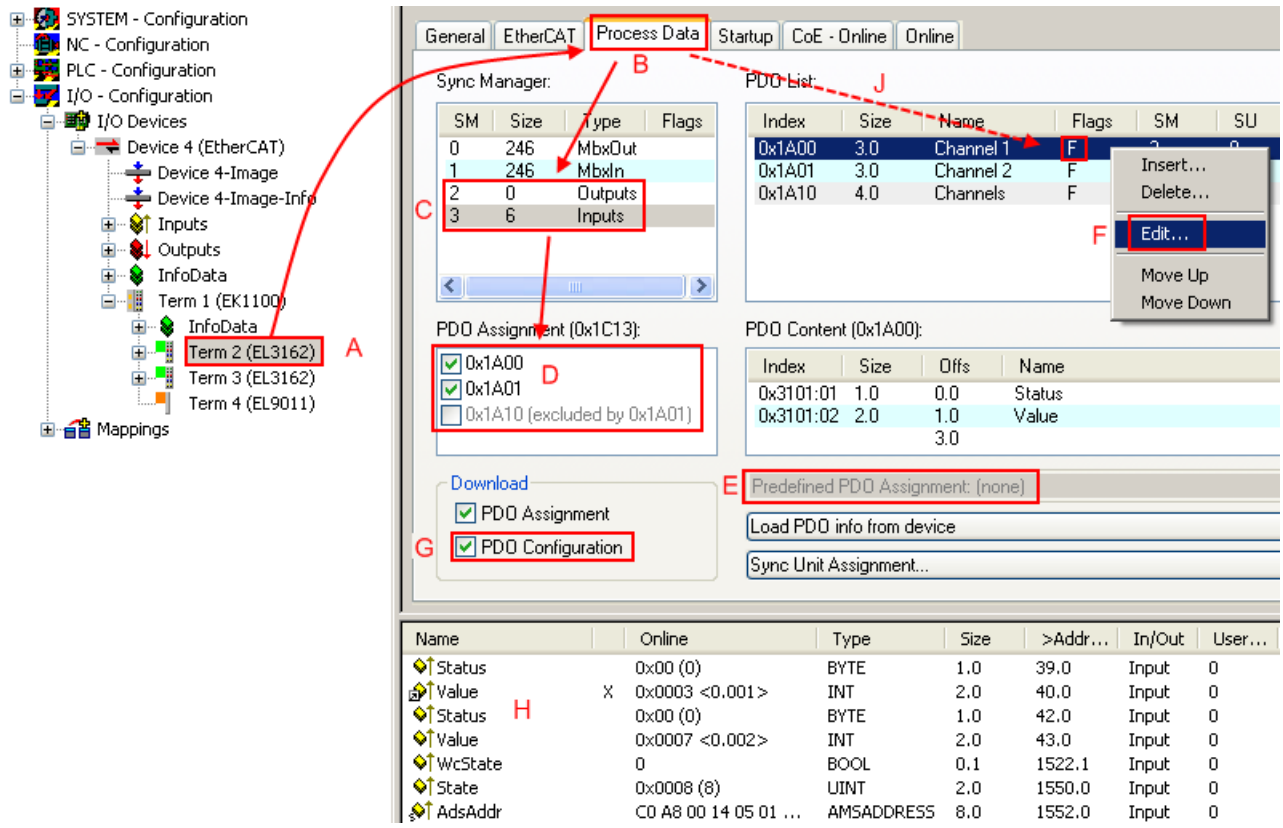


Fig. 89: Configuring the process data

Manual modification of the process data

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

A detailed description [▶ 79] 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.

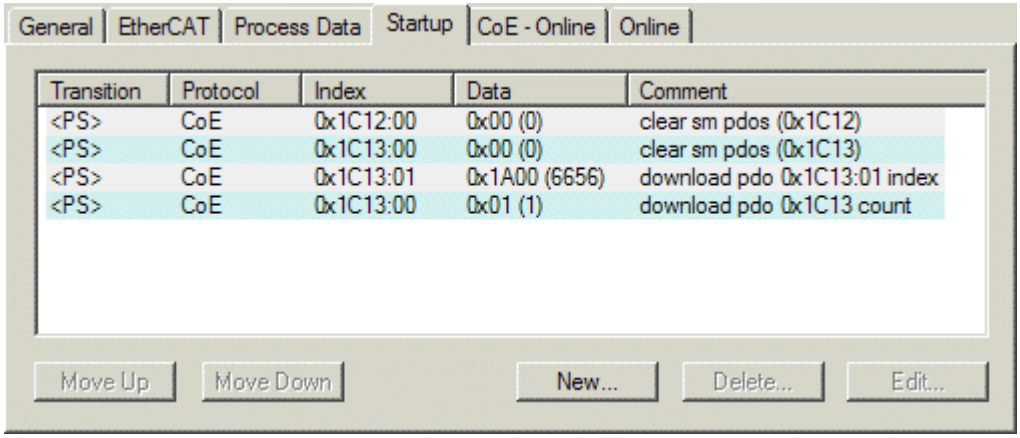


Fig. 90: “Startup” tab

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

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

“CoE - Online” tab

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

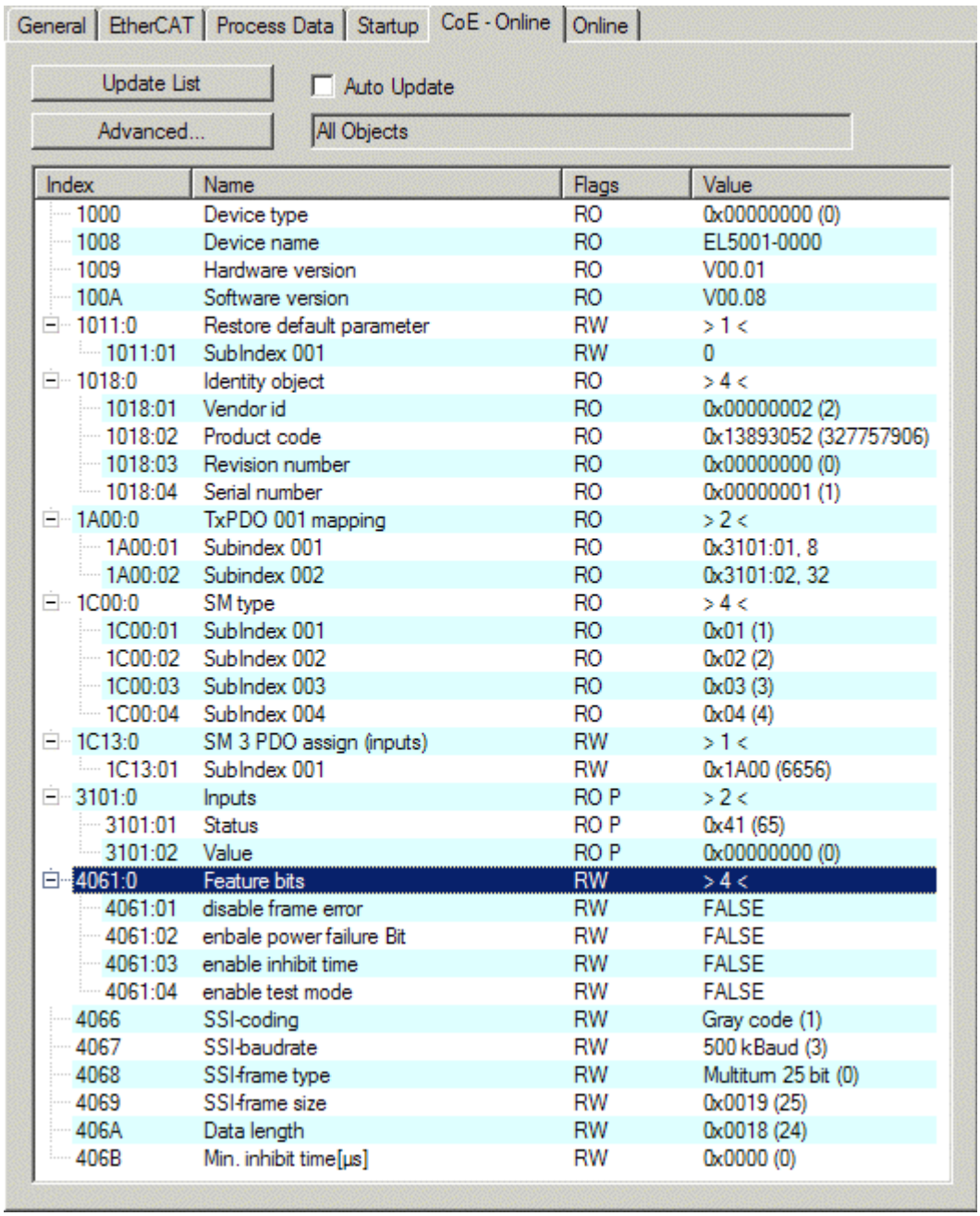


Fig. 91: “CoE - Online” tab

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.

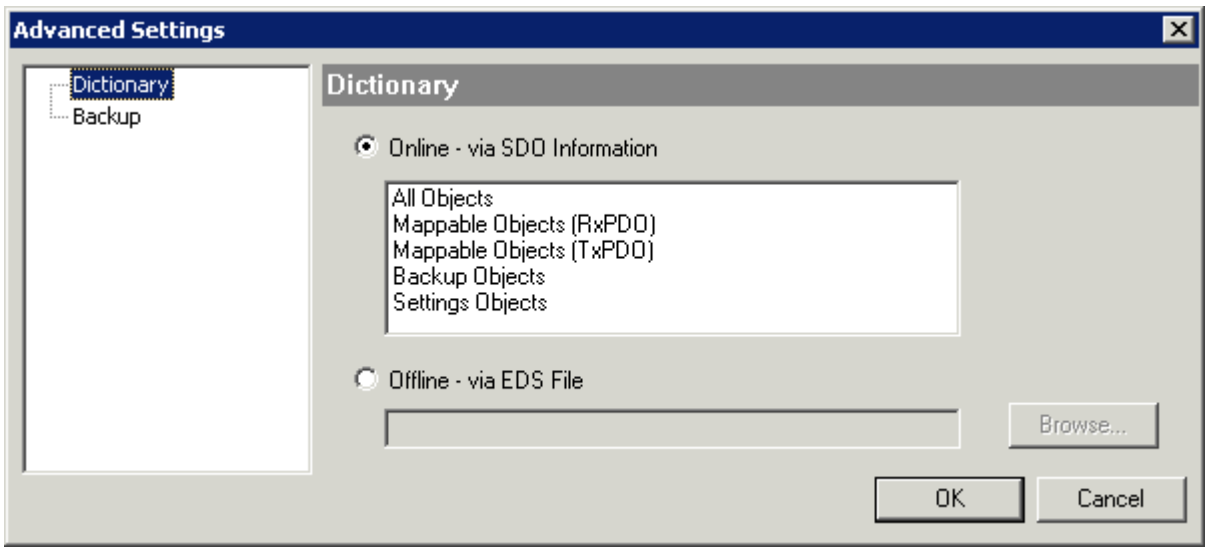


Fig. 92: Dialog “Advanced settings”

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

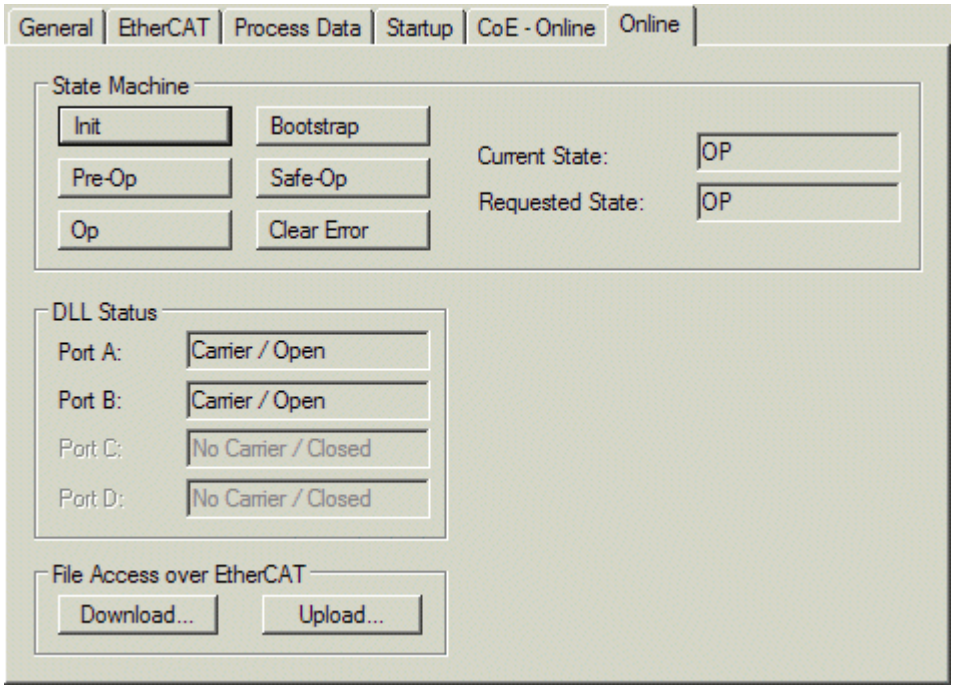


Fig. 93: “Online” tab

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)

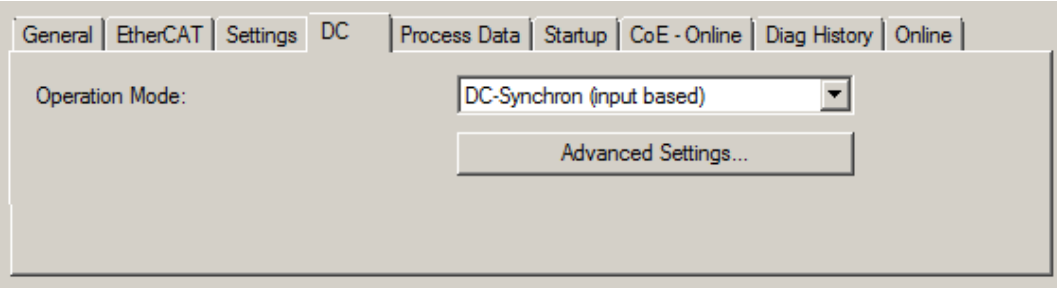


Fig. 94: “DC” tab (Distributed Clocks)

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

5.1.7.1 Detailed description of Process Data tab

Sync Manager

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

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

PDO Assignment

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

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

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

Activation of PDO assignment

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

a) the EtherCAT slave has to run through the PS status transition cycle (from pre-operational to safe-operational) once (see Online tab [▶ 77]),

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

( button for TwinCAT 2 or  button for TwinCAT 3)

PDO list

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

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

PDO Content

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

Download

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

PDO Assignment

If this check box is selected, the PDO assignment that is configured in the PDO Assignment list is downloaded to the device on startup. The required commands to be sent to the device can be viewed in the Startup [▶ 74] 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.

5.1.8 Import/Export of EtherCAT devices with SCI and XTI

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

5.1.8.1 Basic principles

An EtherCAT slave is basically parameterized through the following elements:

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

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

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

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

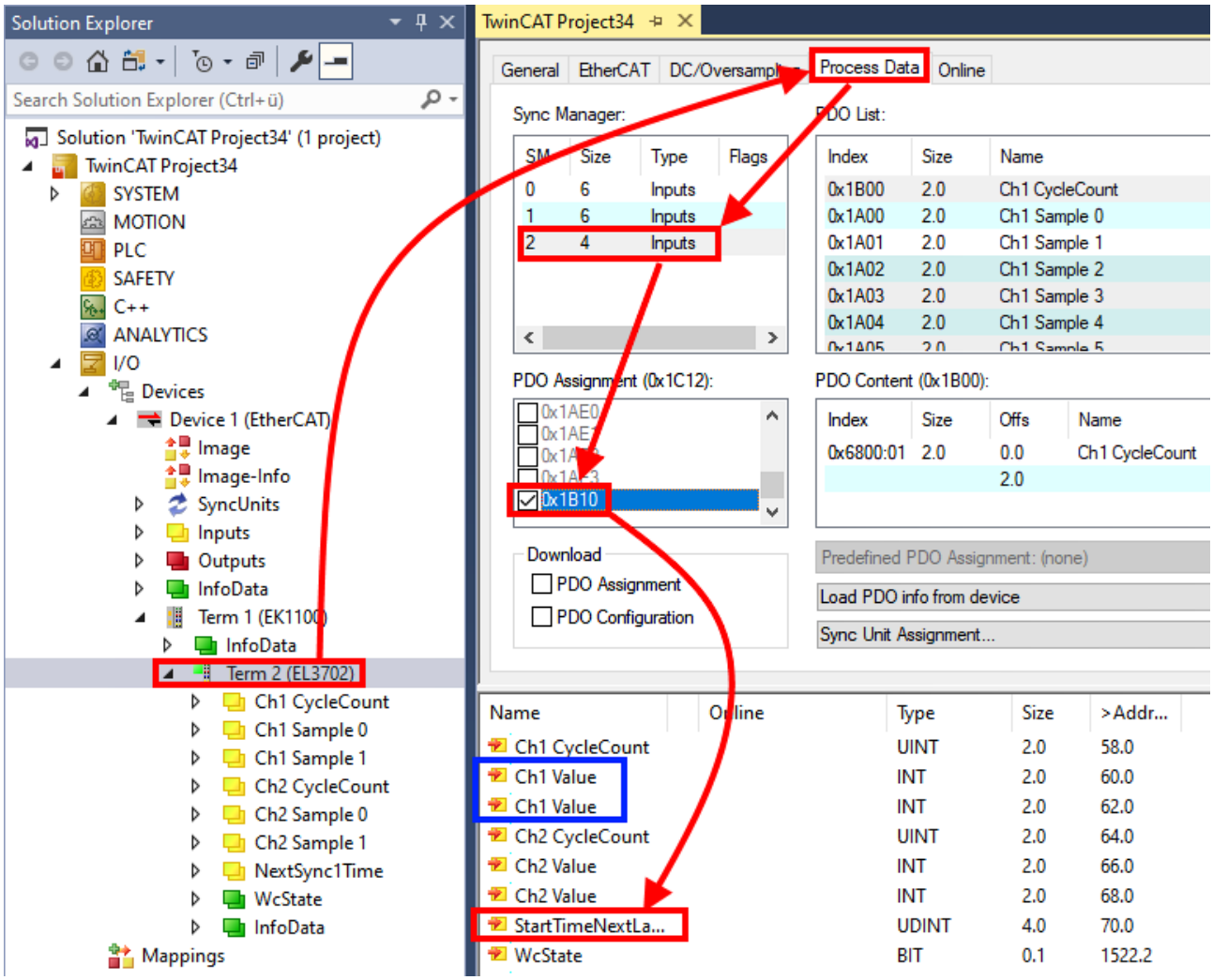
It may become necessary to reuse the modified slave in other projects in this way, without having to make equivalent configuration changes to the slave again. To accomplish this, proceed as follows:

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

TwinCAT offers two methods for this purpose:

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

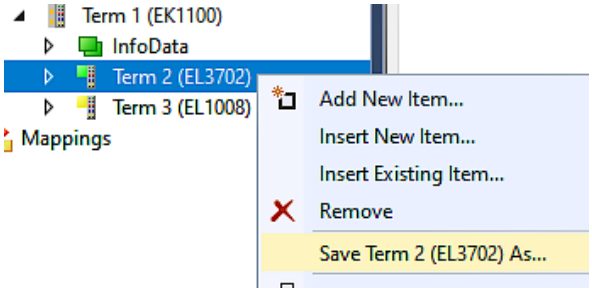
An example is provided below for illustration purposes: an EL3702 terminal with standard setting is switched to 2-fold oversampling (blue) and the optional PDO "StartTimeNextLatch" is added (red):



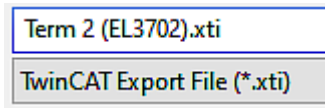
The two methods for exporting and importing the modified terminal referred to above are demonstrated below.

5.1.8.2 Procedure within TwinCAT with xti files

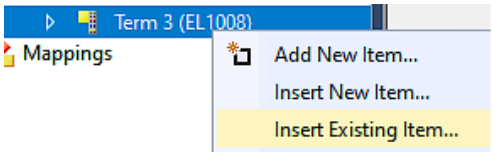
Each IO device can be exported/saved individually:



The xti file can be stored:



and imported again in another TwinCAT system via "Insert Existing item":



5.1.8.3 Procedure within and outside TwinCAT with sci file

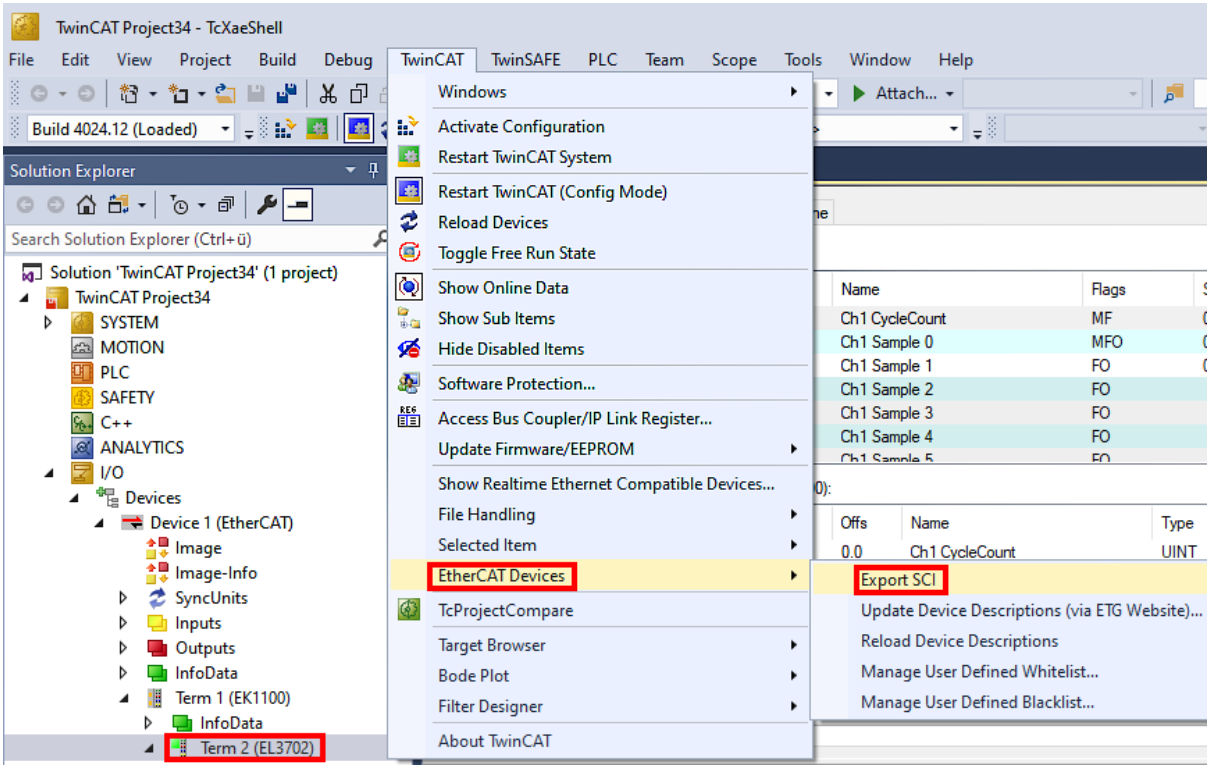
Note regarding availability (2021/01)

The SCI method is available from TwinCAT 3.1 build 4024.14.

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

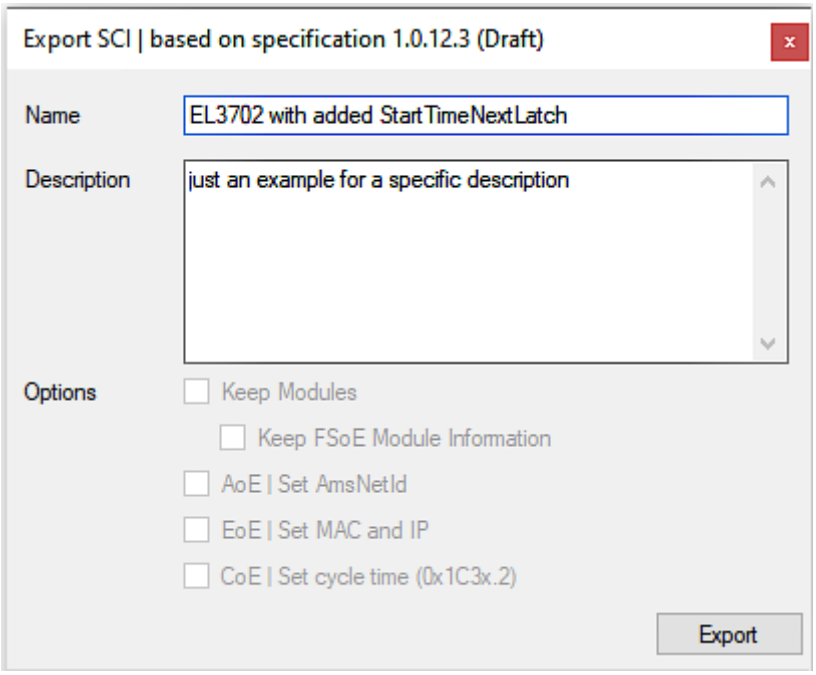
Export:

- select a single device via the menu (multiple selection is also possible):
  TwinCAT → EtherCAT Devices → Export SCI.



- If TwinCAT is offline (i.e. if there is no connection to an actual running controller) a warning message may appear, because after executing the function the system attempts to reload the EtherCAT segment. However, in this case this is not relevant for the result and can be acknowledged by clicking OK:

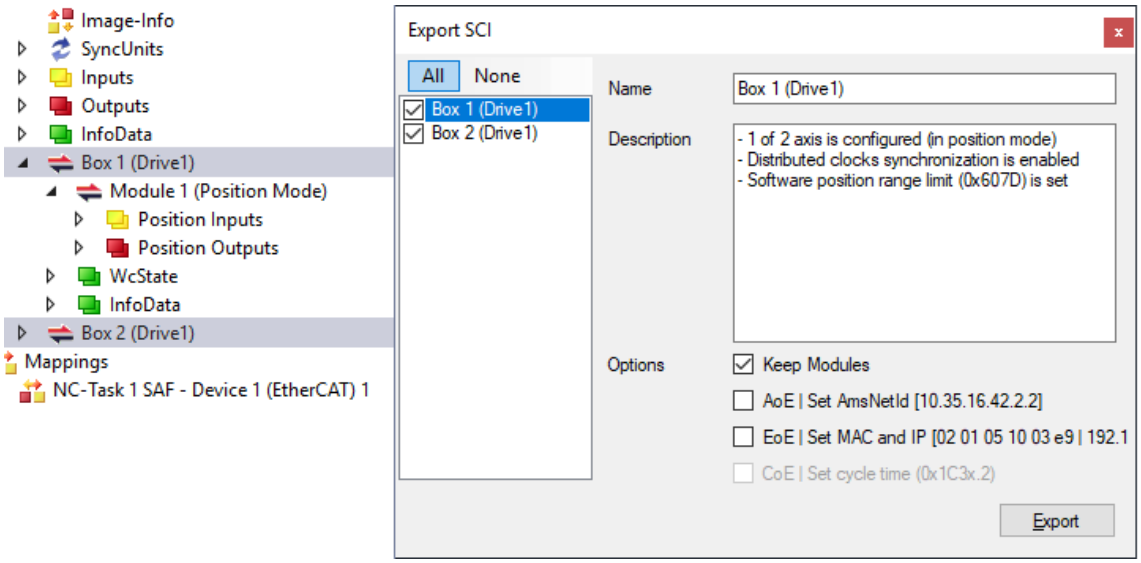
- A description may also be provided:



- Explanation of the dialog box:

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

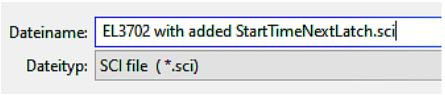
- A list view is available for multiple selections (*Export multiple SCI files)*:



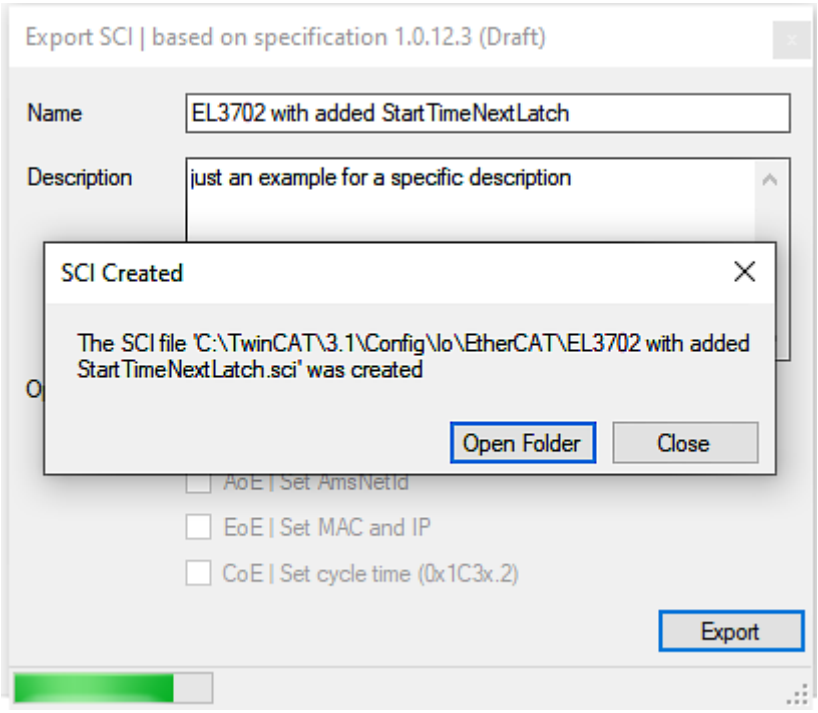
- Selection of the slaves to be exported:
  - All:
    All slaves are selected for export.

- None:
  All slaves are deselected.

- The sci file can be saved locally:

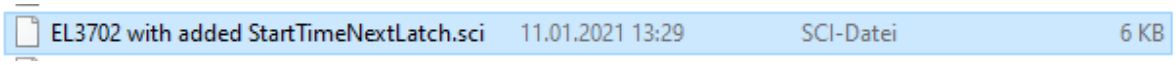


- The export takes place:

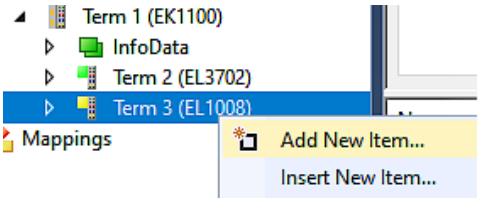


Import

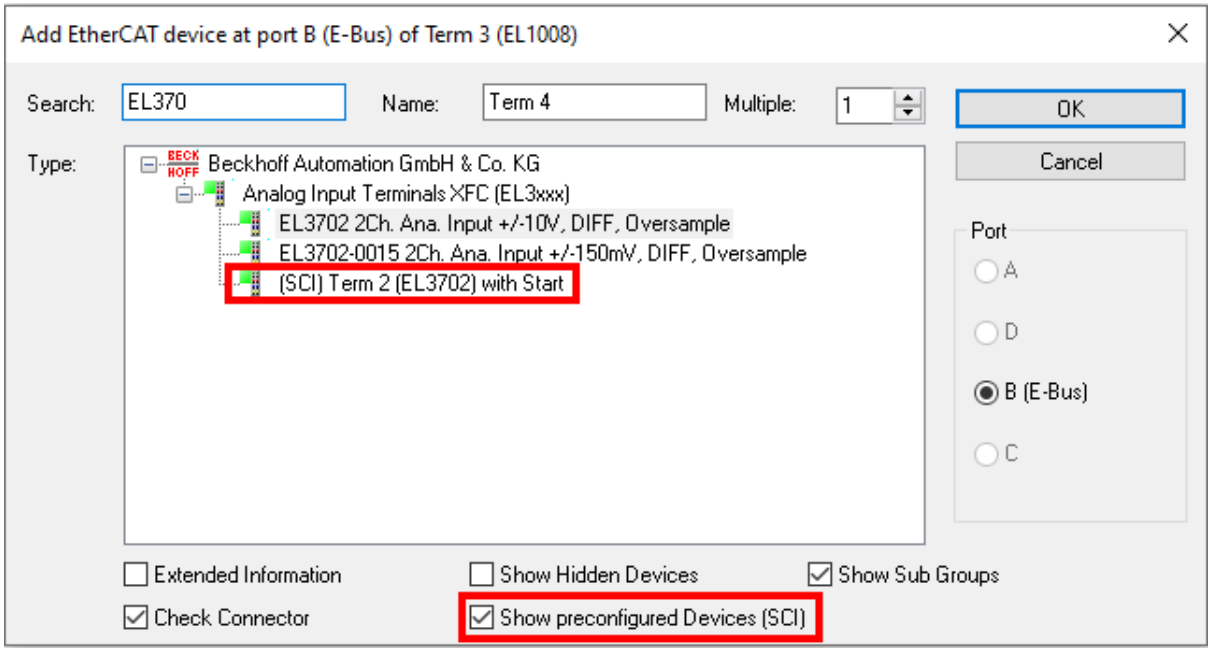
- An sci description can be inserted manually into the TwinCAT configuration like any normal Beckhoff device description.
- The sci file must be located in the TwinCAT ESI path, usually under:
  C:\TwinCAT\3.1\Config\Io\EtherCAT



- Open the selection dialog:

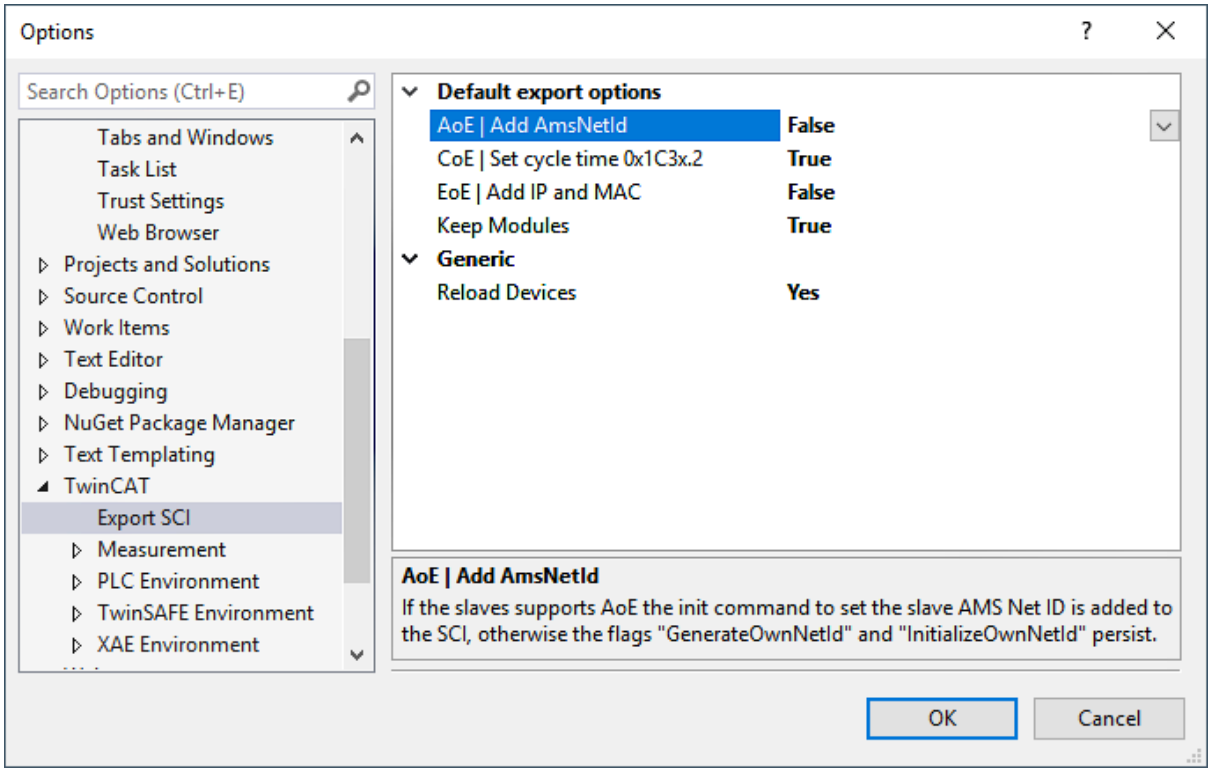


- Display SCI devices and select and insert the desired device:



Additional Notes

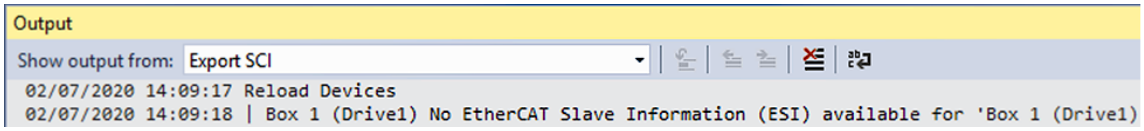
- Settings for the SCI function can be made via the general Options dialog (Tools → Options → TwinCAT → Export SCI):



Explanation of the settings:

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

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



5.2 General Notes - EtherCAT Slave Application

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

Diagnosis in real time: WorkingCounter, EtherCAT State and Status

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

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

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

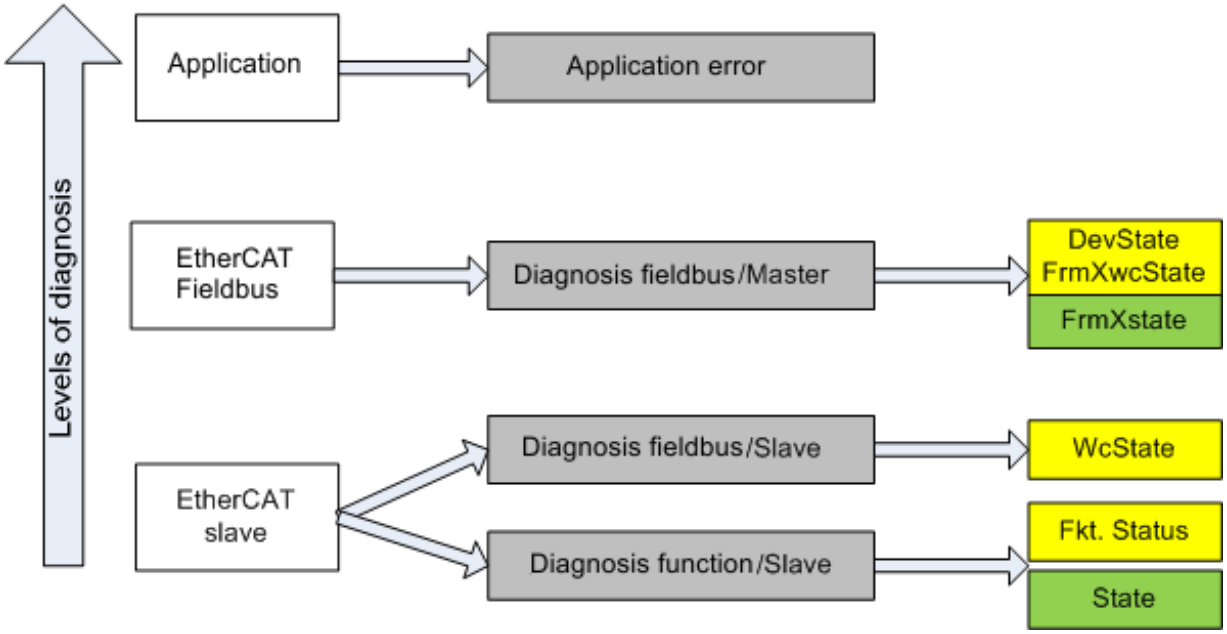


Fig. 95: Selection of the diagnostic information of an EtherCAT Slave

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.

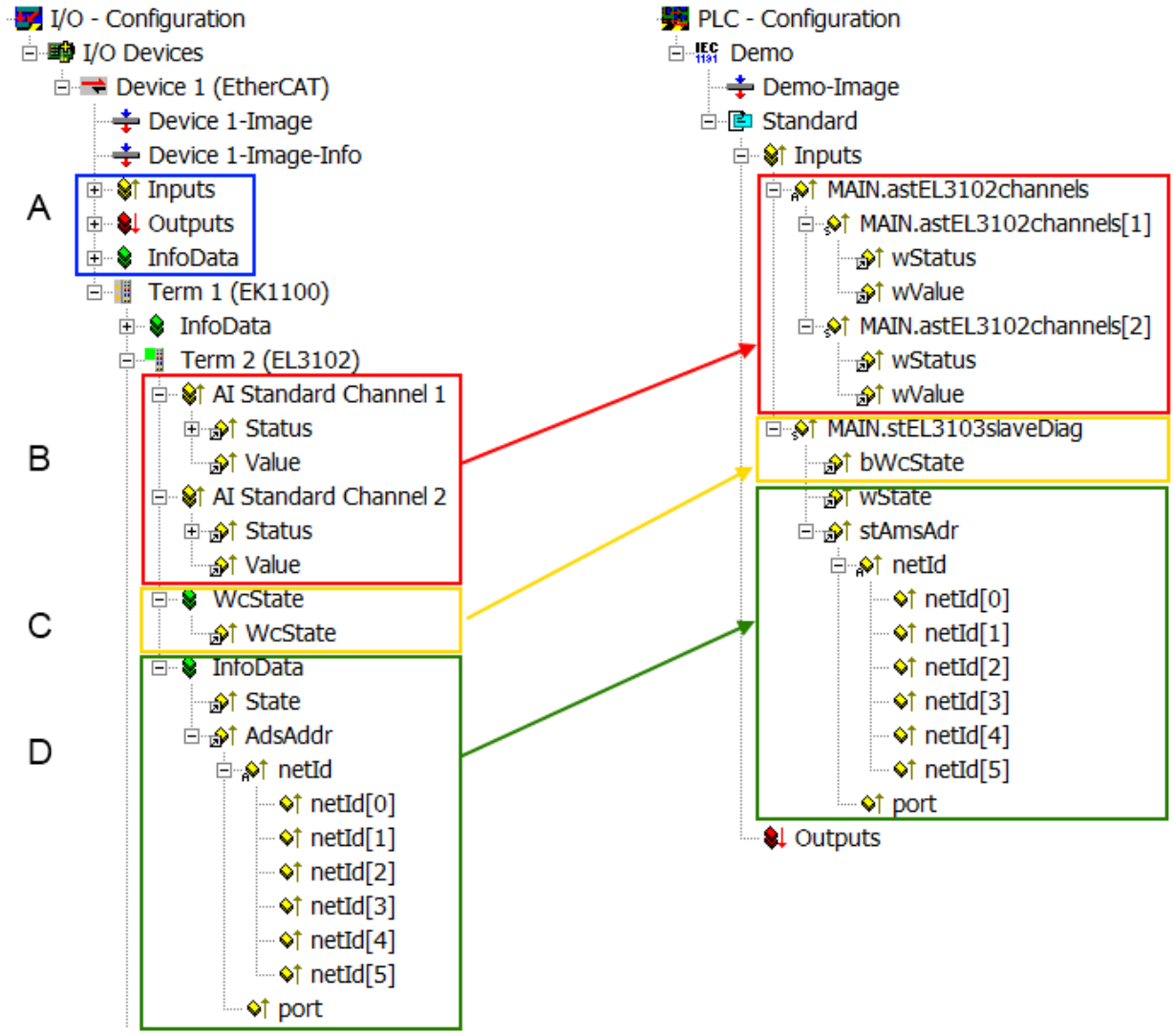


Fig. 96: Basic EtherCAT Slave Diagnosis in the PLC

The following aspects are covered here:

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

NOTE

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*:

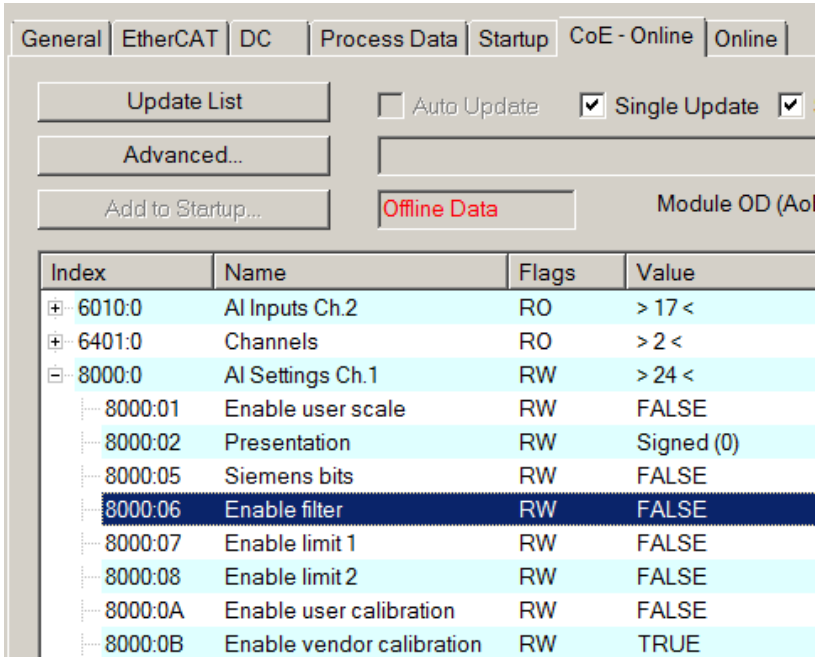


Fig. 97: EL3102, CoE directory

EtherCAT System Documentation

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

A few brief extracts:

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

Commissioning aid in the TwinCAT System Manager

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

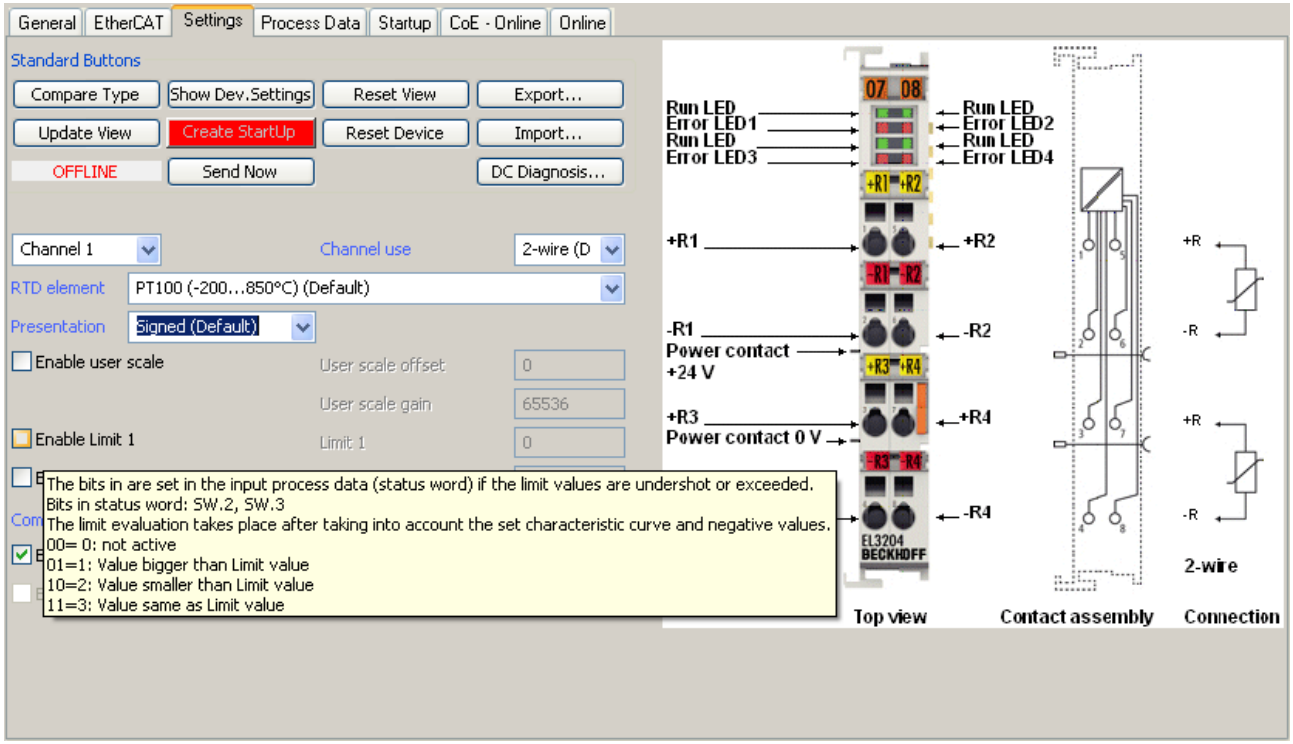


Fig. 98: Example of commissioning aid for a EL3204

This commissioning process simultaneously manages

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

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

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

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

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

- INIT
- PREOP
- SAFEOP
- OP

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

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

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

Standard setting

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

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

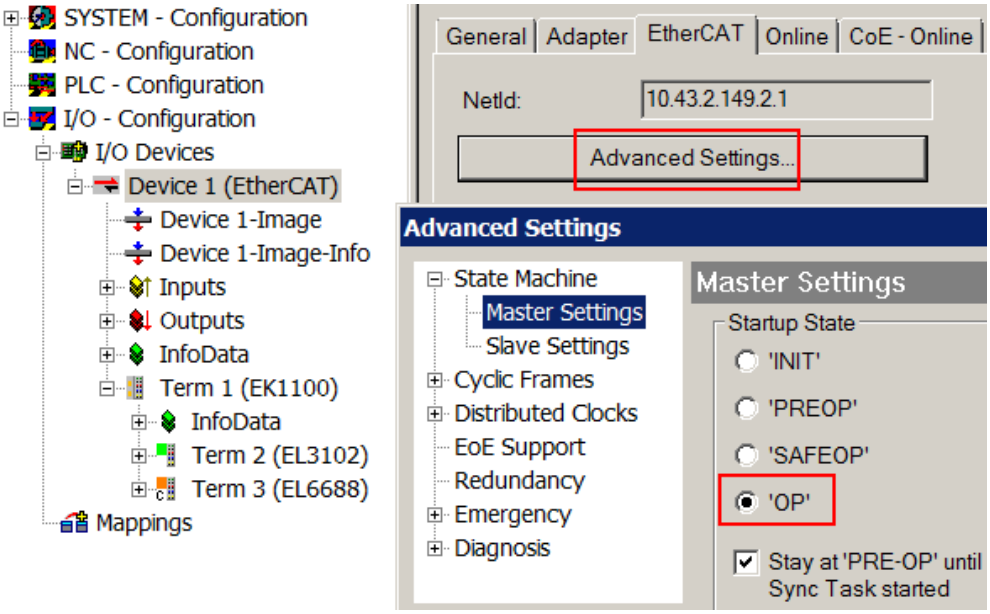


Fig. 99: Default behaviour of the System Manager

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

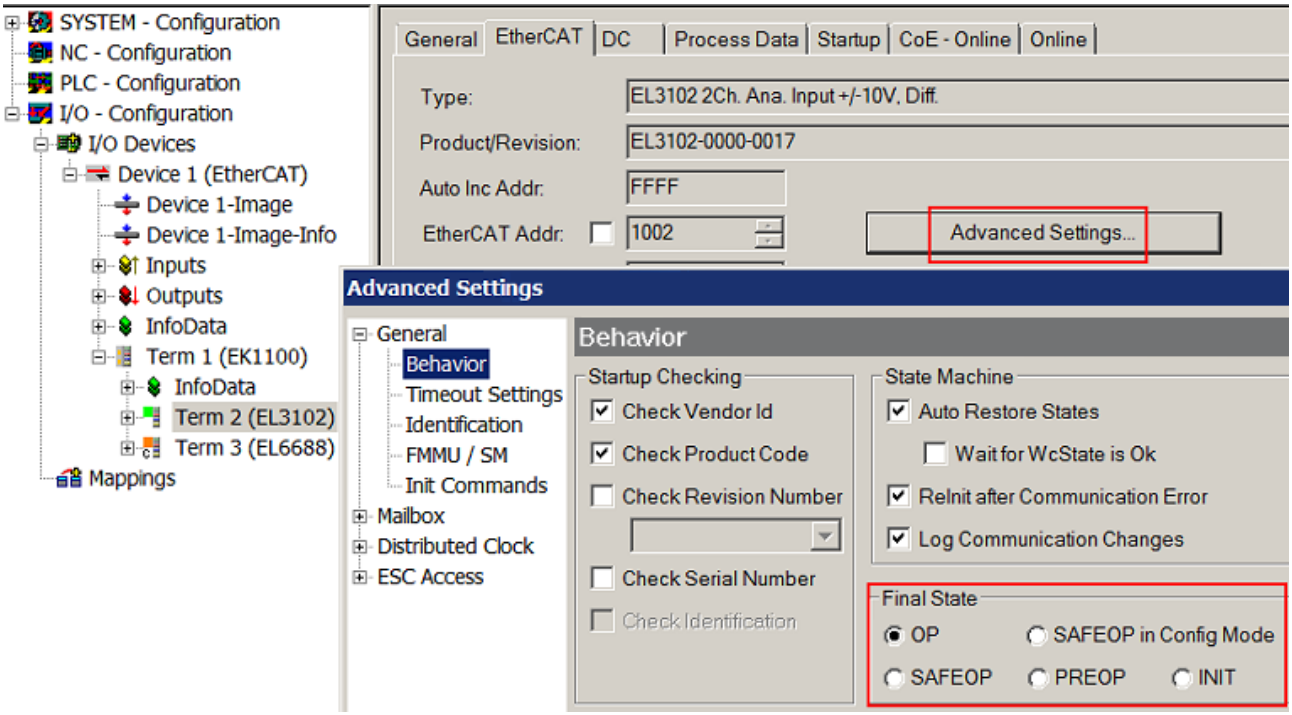


Fig. 100: Default target state in the Slave

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.

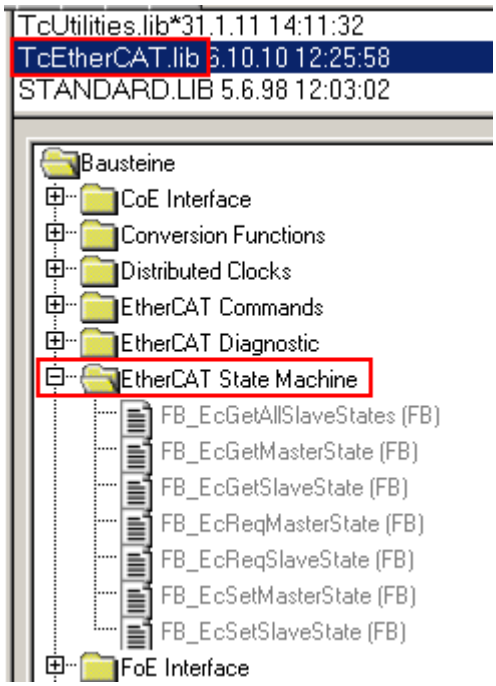
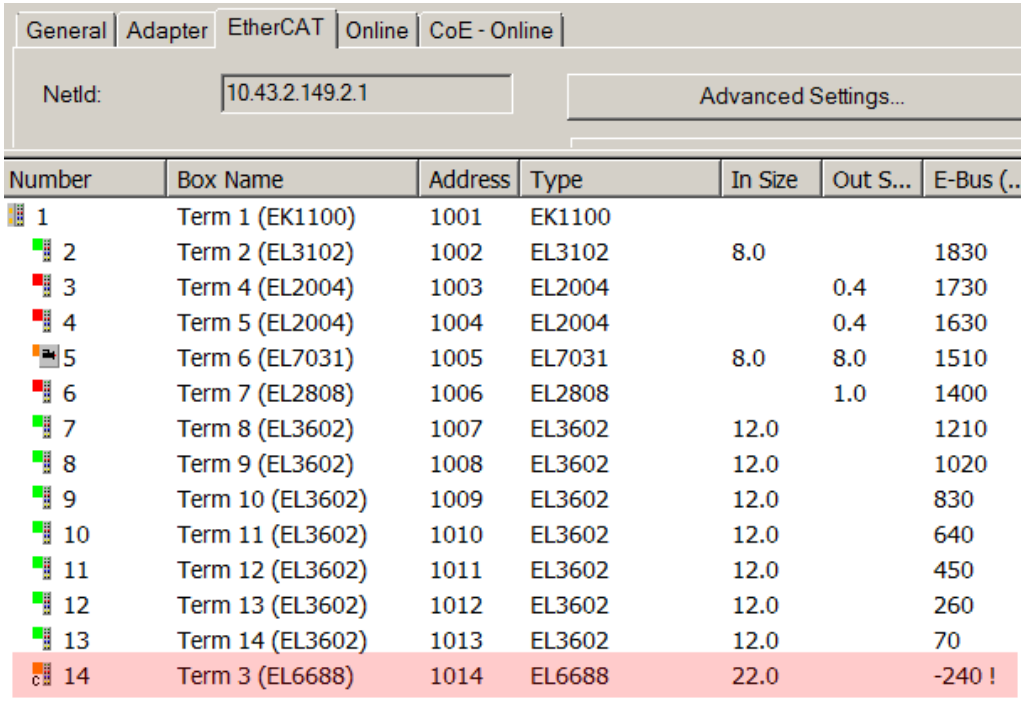


Fig. 101: PLC function blocks

Note regarding E-Bus current

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

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



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

Fig. 102: 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:

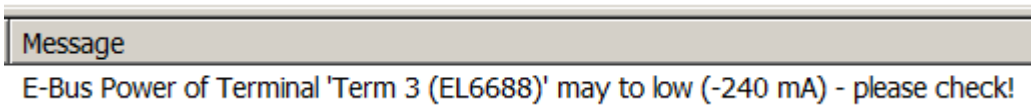


Fig. 103: Warning message for exceeding E-Bus current

NOTE

Caution! Malfunction possible!

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

6 EL5072 - Commissioning

6.1 Overview of functions

The functionality of the EL5072 is summarized in the following table. A detailed description can be found in the individual chapters.

Function	Description
Parameterization for evaluation of the measuring probe [▶ 103]	LVDT, inductive half bridges and RVDT sensors can be connected and evaluated.
Filter function for mechanical cut-off frequency [▶ 116]	The input signal can be filtered by specifying a mechanical cut-off frequency.
Direction inversion [▶ 118]	The position direction can be adapted to the application.
Position value output [▶ 119]	The measured value is output as a position value.
Position value overflow / underflow [▶ 119]	Exceeding and falling below the maximum counter depth are displayed in a separate process data.
Set position value [▶ 120]	The position value can be set to a specified value at runtime via the process data or the digital input.
Save position value [▶ 122]	The current position value can be saved, independent of the cycle time, in a separate process data via an edge at the digital input.
User calibration [▶ 117]	It is possible to parameterize a user calibration via offset and gain values or via a lookup table.
Diagnostic data [▶ 126]	Different diagnostic data are available. In this way, a short circuit on the primary side and an amplitude error on the secondary side can be detected.

Process data	Description
Operation modes [▶ 98]	The scope of the process data can be selected via "Predefined PDO Assignment".
Synchronicity mode [▶ 99]	Frame-triggered operating mode (SM mode) and time-synchronous recording of the latch timestamp signal (DC latch active) are available.

Digital input	Description
Digital input [▶ 124]	The digital input can be used to set and save the position value.

6.2 Process data

6.2.1 Sync Manager (SM)

The scope of process data offered can be modified via the "Process Data" tab (see Fig. *Process Data tab SM3, EL5112 (default)* below).

A detailed description for setting the process data can be found in chapter EtherCAT subscriber configuration [▶ 72].

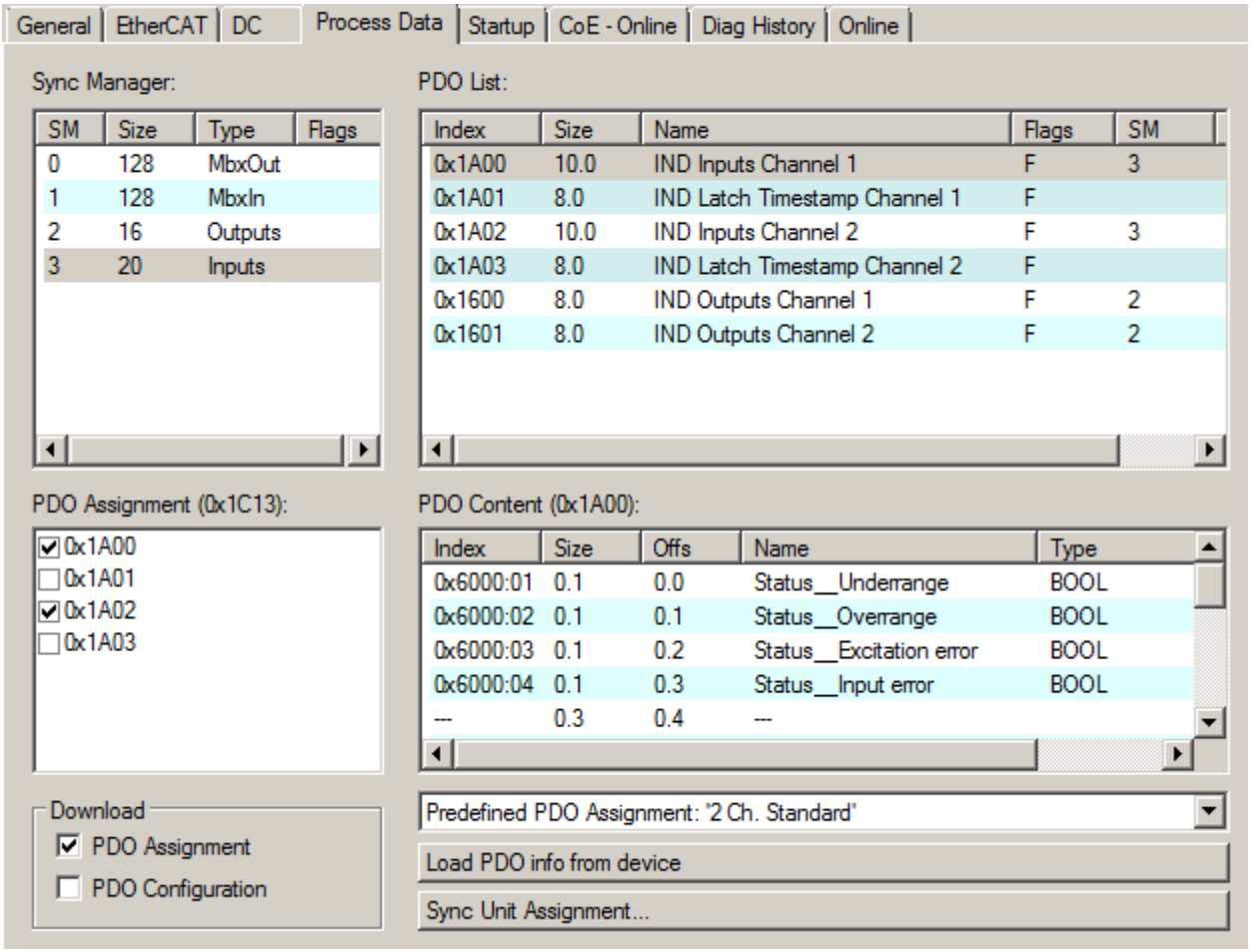


Fig. 104: EL5072 - Process data tab SM3 (default)

6.2.2 PDO assignment

6.2.2.1 SM3 - Inputs (0x1A00 - 0x1A03)

0x1A00 - IND Inputs Channel 1 (10.0)	
Contents **Index - name size (byte.bit)**	**Excluded PDOs** **Index - name size (byte.bit)**
0x6000:01 [▶ 141] - Status_Underrange (0.1) 0x6000:02 [▶ 141] - Status_Overrange (0.1) 0x6000:03 [▶ 141] - Status_Excitation error (0.1) 0x6000:04 [▶ 141] - Status_Input error (0.1) 0x6000:08 [▶ 141] - Status_Digital input (0.1) 0x6000:0A [▶ 141] - Status_Latch extern valid (0.1) 0x6000:0B [▶ 141] - Status_Set position done (0.1) 0x6000:0D [▶ 141] - Status_Diag (0.1) 0x6000:0E [▶ 141] - Status_TxPDO State (0.1) 0x6000:0F [▶ 141] - Status_Input cycle counter (0.2) 0x6001:01 [▶ 141] - Position (4.0) 0x6001:02 [▶ 141] - Latch value (4.0)	-

0x1A01 - IND Latch Timestamp Channel 1 (8.0)	
Contents **Index - name size (byte.bit)**	**Excluded PDOs** **Index - name size (byte.bit)**
0x6002:01 [▶ 141] - Latch event Timestamp (8.0)	-

0x1A02 - IND Inputs Channel 2 (10.0)	
Contents **Index - name size (byte.bit)**	**Excluded PDOs** **Index - name size (byte.bit)**
0x6010:01 [▶ 141] - Status_Underrange (0.1) 0x6010:02 [▶ 141] - Status_Overrange (0.1) 0x6010:03 [▶ 141] - Status_Excitation error (0.1) 0x6010:04 [▶ 141] - Status_Input error (0.1) 0x6010:08 [▶ 141] - Status_Digital input (0.1) 0x6010:0A [▶ 141] - Status_Latch extern valid (0.1) 0x6010:0B [▶ 141] - Status_Set position done (0.1) 0x6010:0D [▶ 141] - Status_Diag (0.1) 0x6010:0E [▶ 141] - Status_TxPDO State (0.1) 0x6010:0F [▶ 141] - Status_Input cycle counter (0.2) 0x6011:01 [▶ 141] - Position (4.0) 0x6011:02 [▶ 141] - Latch value (4.0)	-

0x1A03 - IND Latch Timestamp Channel 2 (8.0)	
Contents **Index - name size (byte.bit)**	**Excluded PDOs** **Index - name size (byte.bit)**
0x6012:01 [▶ 141] - Latch event Timestamp (8.0)	-

6.2.2.2 SM2 - Outputs (0x1600, 0x1601)

0x1600 - IND Outputs Channel 1 (8.0)	
Contents **Index - name size (byte.bit)**	**Excluded PDOs** **Index - name size (byte.bit)**
0x7000:01 [▶ 142] - Set position (0.1) 0x7000:02 [▶ 142] - Set position on digital input (0.1) 0x7000:03 [▶ 142] - Enable latch on digital input (0.1) 0x7000:11 [▶ 142] - Set position value (4.0)	-

0x1601 - IND Outputs Channel 2 (8.0)	
Contents **Index - name size (byte.bit)**	**Excluded PDOs** **Index - name size (byte.bit)**
0x7010:01 [▶ 142] - Set position (0.1) 0x7010:02 [▶ 142] - Set position on digital input (0.1) 0x7010:03 [▶ 142] - Enable latch on digital input (0.1) 0x7010:11 [▶ 142] - Set position value (4.0)	-

6.2.3 Predefined PDO Assignment

The "Predefined PDO Assignment" enables a simplified selection of the process data. The desired function is selected on the lower part of the Process Data tab. As a result, all necessary PDOs are automatically enabled and the unnecessary PDOs are disabled.

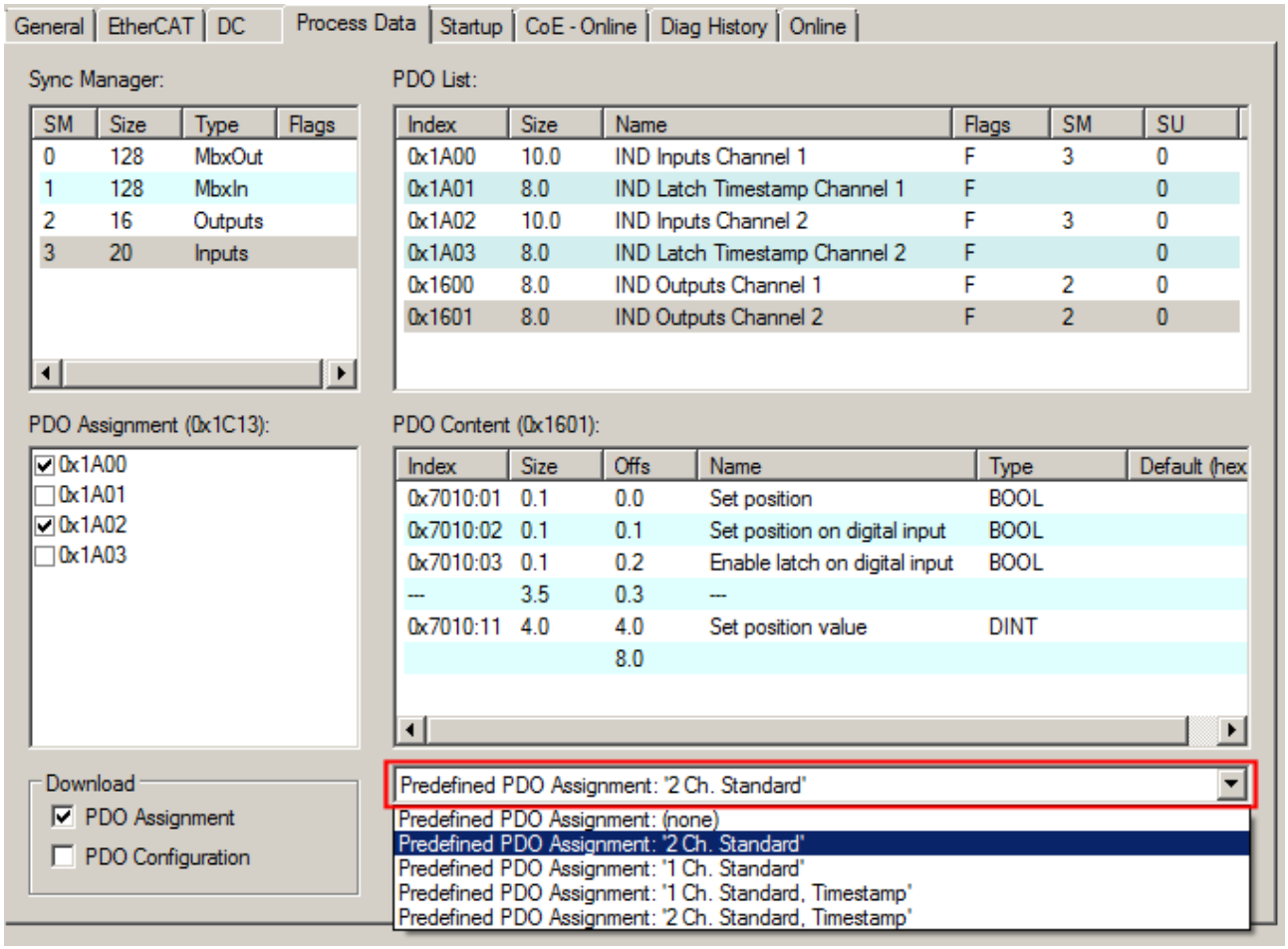


Fig. 105: EL5072 - Process data, Predefined PDO (default: 2 Ch. Standard)

Four PDO assignments are available for selection:

Predefined PDO Assignment	PDO assignment (SM3)	PDO assignment (SM2)
1 Channel	0x1A00 [▶ 144] - IND Inputs Channel 1 (10.0)	0x1600 [▶ 143] - IND Outputs Channel 1 (8.0)
2 Channel	0x1A00 [▶ 144] - IND Inputs Channel 1 (10.0) 0x1A02 [▶ 145] - IND Inputs Channel 2 (10.0)	0x1600 [▶ 143] - IND Outputs Channel 1 (8.0) 0x1601 [▶ 144] - IND Outputs Channel 2 (8.0)
1 Channel, Timestamp	0x1A00 [▶ 144] - IND Inputs Channel 1 (10.0) 0x1A01 [▶ 144] - IND Latch Timestamp Channel 1 (8.0)	0x1600 [▶ 143] - IND Outputs Channel 1 (8.0)
2 Channel, Timestamp	0x1A00 [▶ 144] - IND Inputs Channel 1 (10.0) 0x1A01 [▶ 144] - IND Latch Timestamp Channel 1 (8.0) 0x1A02 [▶ 145] - IND Inputs Channel 2 (10.0) 0x1A03 [▶ 145] - IND Latch Timestamp Channel 2 (8.0)	0x1600 [▶ 143] - IND Outputs Channel 1 (8.0) 0x1601 [▶ 144] - IND Outputs Channel 2 (8.0)

6.2.4 Synchronicity mode

The terminal can be operated in two different operation modes. Further information can be found in the EtherCAT system documentation in chapter Distributed Clocks -> Basics.

The following operating modes are available for selection in the "DC" tab:

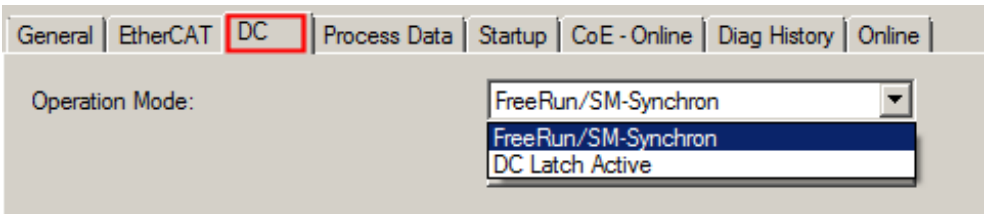


Fig. 106: EL5072 - "DC" tab

Operation mode	Description
FreeRun/SM Synchron	Cyclic, frame-triggered exchange of process data. An Ethernet frame triggers the process data provision for the next retrieving frame.
DC Latch Active	In addition to the cyclic, frame-triggered exchange of process data, a timestamp is provided for saving the position value (latch event timestamp). An Ethernet frame triggers the process data provision for the next retrieving frame. The timestamp is determined cyclically by the integrated distributed clocks unit.

6.2.5 EtherCAT cycle time

The EtherCAT cycle time depends on the selection of the process data to be transmitted and the set excitation frequency 0x8001:14 "Excitation frequency".

The following table provides an overview of the recommended cycle time, depending on the "Predefined PDO Assignment" and the excitation frequency.

- The specifications refer to a multiple of the "Base Time" to be set via the TwinCAT Master.
- If a faster cycle time is used, the process data 0x60n0:0F "Input Cycle Counter" must be used to monitor when new process data are delivered.

EL5072 – Predefined PDO Assignment	0x8001:14 "Excitation frequency"	Min. EtherCAT cycle time
2. Ch. Standard	1 kHz	typical 66.6 µs
	2 kHz	typical 66.6 µs
	2.5 kHz	typical 66.6 µs
	4 kHz	typical 66.6 µs
	5 kHz	typical 71.4 µs
	7.5 kHz	typical 71.4 µs
	10 kHz	typical 76.9 µs
	12.5 kHz	typical 83.3 µs
	13 kHz	typical 83.3 µs
	15 kHz	typical 100 µs
	19.4 kHz	typical 200 µs
	20 kHz	typical 230.7 µs
2. Ch. Standard, Timestamp	1 kHz	typical 66.6 µs
	2 kHz	typical 71.4 µs
	2.5 kHz	typical 71.4 µs
	4 kHz	typical 76.9 µs
	5 kHz	typical 83.3 µs
	7.5 kHz	typical 83.3 µs
	10 kHz	typical 100 µs
	12.5 kHz	typical 100 µs
	13 kHz	typical 100 µs
	15 kHz	typical 125 µs
	19.4 kHz	typical 214.2 µs
	20 kHz	typical 285.6 µs

6.3 Functions

Basics of data acquisition and evaluation

The measuring functions of the EL5072 can be described as follows:

1. **Feeding the alternating voltage into the measuring probe**
   - The alternating voltage is fed into the measuring probe via the connection points U_{exc+} and U_{exc-}.
     - The excitation frequency (0x8001:14 [▶ 139] "Excitation frequency") and the excitation voltage (0x8001:15 [▶ 139] "Excitation voltage") can be set via the CoE objects. The setting simultaneously applies to both channels.
     - First switch on the supply voltage (0x8000:08 [▶ 138] "Enable excitation"). It is switched off in the delivery state.
2. **Measurement**
   - In the EL5072 the measuring principle is defined via:
     - the selection of the connection (0x80n1:12 [▶ 139] "Connection type") and the
     - input impedance for the sensor (0x80n1:13 [▶ 139] "Sensor impedance").
   - For the maximum input voltage to be measured, the internal gain factors are determined from the information provided below. This automatically adjusts the measuring signal range.
     - Excitation voltage (0x80n1:15 [▶ 139] "Excitation voltage"),
     - Sensitivity (0x80n1:16 [▶ 139] "Sensitivity"),
     - Maximum measuring range of the measuring probe (0x80n1:18 [▶ 139] "Overall sensor range")

$$\text{Max. Output signal [mV]} = \text{Excitation voltage} * \text{Sensitivity} * \text{Measuring range} \left[\text{V} * \frac{\text{mV}}{\text{V mm}} * \text{mm}\right]$$

   - Depending on the measuring probe version, the measuring voltage is measured via the connection points U_{sig} and U_{sens}.
3. **Output of the measured value**
   - The measured value is output in the process data 0x60n1:01 [▶ 141] "Position" in the unit nm.

NOTE

Observe the notes on application and parameterization

- For correct measured value output, parameterize the EL5072 according to the measuring probe used, as described in chapter Parameterization for evaluation of the measuring probe [▶ 103].
- During commissioning observe the Application notes on measurement deviations [▶ 119]!

Data flow

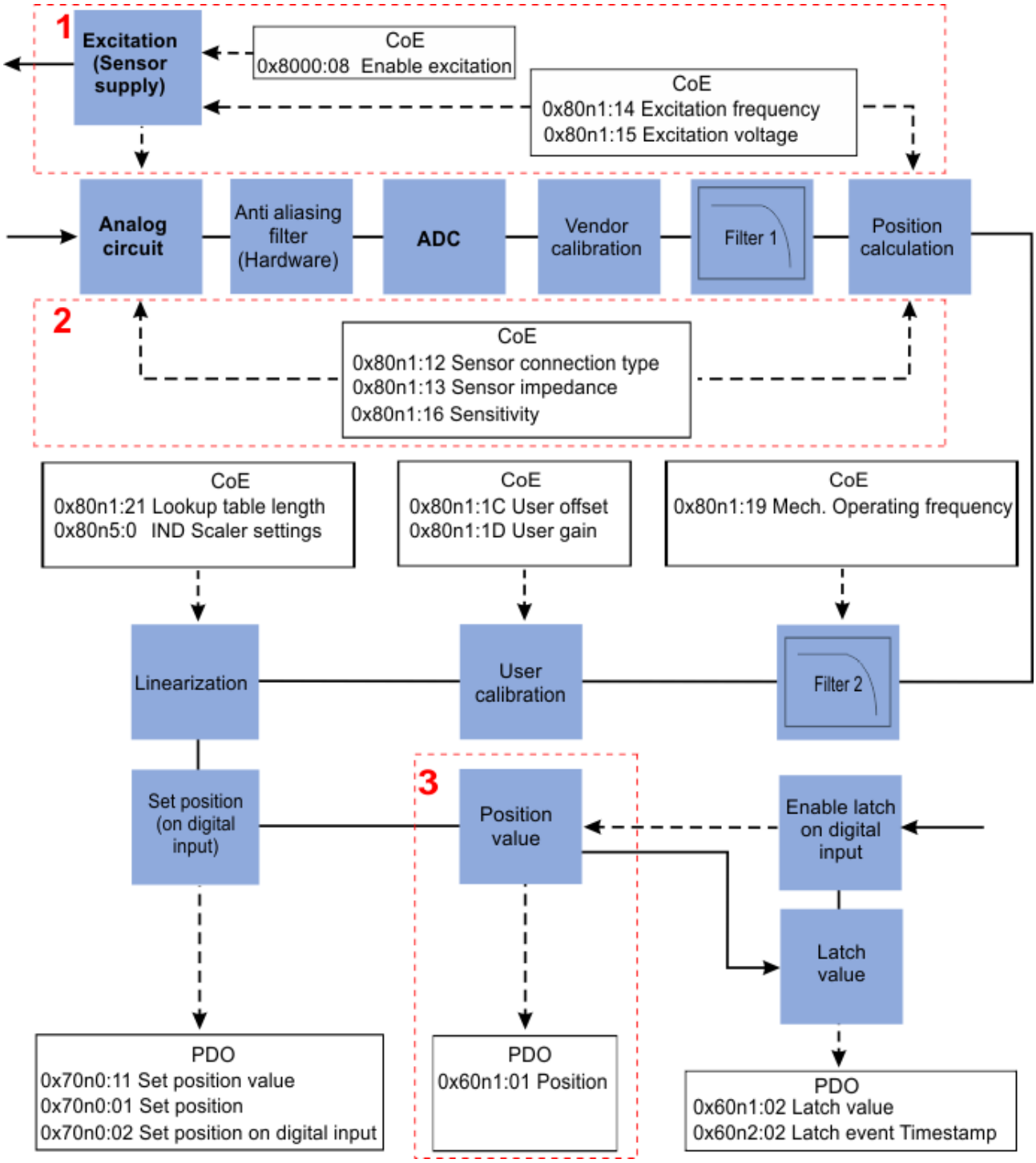


Fig. 107: EL5072 - Data flow

6.3.1 Parameterization for evaluation of the measuring probe

The EL5072 offers the option to evaluate measuring probes in LVDT and half bridge design or inductive angular position sensors in RVDT design. The measuring signal range is adjusted automatically. The measured value is output directly as a position value via the process data 0x60n01:01 [▶ 141] "Position".

For correct output of the measured value, all sensor parameters must be entered in the configuration data.

Index (hex)	Name	Meaning
80n1:12 [▶ 139]	Connection type	Selecting the connection [▶ 104]
80n1:13 [▶ 139]	Sensor impedance	Selecting the input impedance [▶ 104]
80n1:14 [▶ 139]	Excitation frequency	Excitation frequency [▶ 105]
80n1:15 [▶ 139]	Excitation voltage	Excitation voltage [▶ 105]
80n1:16 [▶ 139]	Sensitivity	Sensitivity [▶ 106]
80n1:18 [▶ 139]	Overall sensor range	Maximum measuring probe travel path [▶ 106]

Please refer to the data sheet or the specification of the measuring probe used. For the commissioning an RVDT follow the corresponding instructions [▶ 115].

Other available parameterization options: filter function; user calibration via offset and gain values or via a lookup table; reversal of the measuring direction.

Index (hex)	Name	Meaning
80n1:19	Mech. operating frequency	Selecting the mechanical cut-off frequency [▶ 116]
80n1:1C	User offset	User offset compensation of the position value [▶ 117]
80n1:1D	User gain	User gain compensation of the position value [▶ 117]
80n5:0	IND Scaler Settings Ch.n	Optional lookup table for user calibration [▶ 118]
80n0:05	Sign inversion	Selection for direction inversion [▶ 118]

6.3.1.1 Selecting the sensor connection, index 0x80n12 (Connection type)

The following connection options are available:

Index 0x80n1:12 [▶ 139] "Connection type"	
Setting	**Meaning**
4_{dec}: 4-wire LVDT	LVDT connection [▶ 41]
5_{dec}: 5-wire LVDT	LVDT connection [▶ 41]
6_{dec}: 6-wire LVDT	LVDT connection [▶ 41]
13_{dec}: 3-wire Half Bridge	Half bridge - Connection [▶ 43]
15_{dec}: 5-wire Half Bridge	Half bridge - Connection [▶ 43]

The correct connections of the EL5072 are described in chapter EL5072 - Connection [▶ 39]. All necessary bridges are automatically switched in the terminal. See chapter Setting recommendations of common sensor series [▶ 107]

NOTE

Setting for Mahr®-compatible sensors

Mahr®-compatible sensors are connected using 5 wires. Since the fifth line is not a signal line but a virtual ground, use the 4-wire LVDT mode in index 0x80n1:12 "Connection type".

6.3.1.2 Sensor impedance, index 0x80n1:13

The input impedance for the inductive measuring probe in the EL5072 can be selected from the following values:

Index 0x80n1:13 [▶ 139] "Sensor impedance"	
Setting	**Meaning**
0_{dec}: High impedance / Marposs ®	High impedance input circuit, suitable for Marposs®-compatible measuring probes, for example
1_{dec} : Mahr ® impedance	Input circuit suitable for Mahr®-compatible measuring probes, for example
2_{dec} : Tesa ® impedance	Input circuit suitable for Tesa®-compatible measuring probes, for example

The input circuits are described in chapter Variable input impedances [▶ 44]. See chapter Setting recommendations of common sensor series [▶ 107]

6.3.1.3 Excitation frequency (0x80n1:14), excitation voltage (0x80n1:15)

NOTE

Setting and activating the excitation frequency and excitation voltage

- Check the supply to the power contacts (see chapter EL5072 – Diagnostics [▶ 126]).
- Set the excitation frequency index 0x8001:14 "Excitation frequency" and excitation voltage index 0x8001:15 "Excitation voltage" centrally via the first channel. These settings are then valid for both channels.
- Before switching, make sure that both sensors support the set range!
- The excitation voltage is switched off in the delivery state and must be switched on by setting index 0x8000:08 "Enable excitation" to TRUE.

Excitation frequency (index 0x8001:14 [▶ 139])

The EL5072 provides a high-frequency alternating voltage for sensor excitation. Predefined excitation or carrier frequencies are provided for this purpose. Since it is a joint excitation source for both channels, the settings must be made in channel 1.

Index 0x8001:14 "Excitation frequency"	
Setting	**Meaning**
1000_{dec}	1 kHz
2000_{dec}	2 kHz
2500_{dec}	2.5 kHz
4000_{dec}	4 kHz
5000_{dec}	5 kHz
7500_{dec}	7.5 kHz
10000_{dec}	10 kHz
12500_{dec}	12.5 kHz
13000_{dec}	13 kHz
15000_{dec}	15 kHz
19400_{dec}	19.4 kHz
20000_{dec}	20 kHz

Excitation voltage (index 0x8001:15 [▶ 139])

The following excitation voltages are provided as RMS values:

Index 0x8001:15 "Excitation voltage"	
Setting	**Meaning**
500_{dec}	0.5 V_{rms}
1000_{dec}	1 V_{rms}
1500_{dec}	1.5 V_{rms}
2000_{dec}	2 V_{rms}
2500_{dec}	2.5 V_{rms}
3000_{dec}	3 V_{rms}
3500_{dec}	3.5 V_{rms}
4000_{dec}	4 V_{rms}
4500_{dec}	4.5 V_{rms}
5000_{dec}	5 V_{rms}
5500_{dec}	5.5 V_{rms}
6000_{dec}	6 V_{rms}
6500_{dec}	6.5 V_{rms}
7000_{dec}	7 V_{rms}

6.3.1.4 Sensitivity (0x80n1:16) and maximum travel path (0x80n1:18)

Sensitivity (index 0x80n1:16 [▶ 139])

The measurement output signal of a sensor varies depending on the core displacement and the excitation voltage. Sensitivity is the ratio of these quantities and is expressed in:

Sensitivity [mV_{output} / V_{supply} / $mm_{measuring\ path}$]

The sensitivity is usually specified for the calibrated measuring range. Outside the measuring range linearity errors must be taken into account. Please refer to the information provided by the manufacturer.

Maximum travel path ("Overall sensor range", index 0x80n1:18 [▶ 139])

An essential selection criterion for inductive measuring probes is the measuring path of the sensor. A distinction is made between:

- **Measuring range:**
  in this range the inductive measuring probe usually has a high linearity and accuracy. This range usually also corresponds to the measuring range calibrated by the manufacturer, for which the sensitivity is valid.
- **Maximum mechanical travel path (overall sensor range)**:
  this is the maximum mechanical travel path or maximum mechanical stroke of the inductive measuring probe. This value is always greater than the calibrated measuring range.

The maximum mechanical travel path is specified via index 0x80n1:18 "Overall sensor range". The travel path must be entered symmetrically with "+/-" in the unit mm.

This information is required to set internal gain factors in such a way that a valid position value is still measured even at maximum travel path. If the value entered is too low, amplitude errors may occur in the limit range. These are displayed via the process data 0x60n0:04 [▶ 141] "Input error".

6.3.1.5 Setting recommendations of common sensor series

In the following chapters, you will find, sorted by manufacturer, recommendations for setting

- the “Connection type” (index 0x80n1:12) and
- the “Sensor impedance” (index 0x80n1:13).

For excitation frequency, excitation voltage, sensitivity and maximum travel path please refer to the manufacturer`s data sheet.

NOTE

Observe current documentation of the manufacturer!

The following setting recommendations are based on the manufacturer's documentation valid at the time the documentation was prepared.

The user is responsible to check whether the following recommendations are suitable for the application. We accept no responsibility for the completeness and correctness of the information. We reserve the right to modify the content of this document at any time and accept no responsibility for errors and missing information.

- Read the currently valid manufacturer documentation before connecting or commissioning the device and check whether the stated recommendations are suitable for your application!

6.3.1.5.1 Mahr® sensors

Series: PS2000		**EL5072 Setting recommendations**	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
P2004 M	Mahr	4_{dec}: 4-wire LVDT	1_{dec}: Mahr ® impedance
P2004 MA			
P2004 MB			
P2004 U	Marposs	4_{dec}: 4-wire LVDT	0_{dec}: High impedance / Marposs ®
P2004 UA			
P2004 UB			
P2004 T	Tesa	13_{dec}: 3-wire Half Bridge	2_{dec}: Tesa ® impedance
P2004 TA			
P2004 TB			

6.3.1.5.2 TESA® sensors

Series: TESA ® Standard version probes		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
GT 21	Tesa	13_{dec}: 3-wire Half Bridge	2_{dec}: Tesa ® impedance
GT 22			
GTL 21			
GTL 22			
GT 27			
GT 271			
GT 28			
GT 61			
GT 611			
GT 62			
GTL 212			
GTL 222			
GT 272			
GT 282			
GT 612			
GT 622			
GT 41			
GT 42			
GT 43			
GT 44			

Series: TESA ® lever probes		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
GT 31	Tesa	13_{dec}: 3-wire Half Bridge	2_{dec}: Tesa ® impedance

Series: TESA ® Universal measuring probe		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
FMS100	Tesa	13_{dec}: 3-wire Half Bridge	2_{dec}: Tesa ® impedance
FMS 102			
FMS 130			
FMS 132			

6.3.1.5.3 Peter Hirt GmbH sensors

Series: Probes T101 / T102		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
T101F	Half Bridge	13_{dec}: 3-wire Half Bridge	2_{dec}: Tesa ® impedance
T102F			
T101V			
T102V			
T101P			
T102P			
T101L			
T102L			

Series: Probes T151 / T152		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
T151F	Full Bridge (LVDT)	5_{dec}: 5-wire LVDT	0_{dec}: High impedance / Marposs ®
T152F			
T151V			
T152V			
T151P			
T152P			
T151L			
T152L			

Series: Probes T151 / T152		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
T151F-024	Marposs	5_{dec}: 5-wire LVDT	0_{dec}: High impedance / Marposs ®
T152F-024			
T151V-024			
T152V-024			
T151P-024			
T152P-024			
T151L-024			
T152L-024			

Series: Probes T161 / T162		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
T161F	Mahr	4_{dec}: 4-wire LVDT	1_{dec}: Mahr ® impedance
T162F			
T161V			
T162V			
T161P			
T162P			
T161L			
T162L			

Series: Probes T201 / T202		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
T201F	Half Bridge	13_{dec}: 3-wire Half Bridge	2_{dec}: Tesa ® impedance
T202F			
T202V			
T202P			
T202L			

Series: Probes T301 / T302		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
T301F	Half Bridge	13_{dec}: 3-wire Half Bridge	2_{dec}: Tesa ® impedance
T302F			
T301V			
T302V			
T301P			
T302P			
T301L			
T302L			

Series: Probes T401 / T402, T451 / T452		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
T401	Half Bridge	13_{dec}: 3-wire Half Bridge	2_{dec}: Tesa ® impedance
T402			
T451	Full Bridge (LVDT)	5_{dec}: 5-wire LVDT	0_{dec}: High impedance / Marposs ®
T452			

Series: Probes T501 / T502		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
T501F	Half Bridge	13_{dec}: 3-wire Half Bridge	2_{dec}: Tesa ® impedance
T502F			
T501V			
T502V			
T501P			
T502P			
T501L			
T502L			

Series: Probes T521 / T522		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
T521F	Half Bridge	13_{dec}: 3-wire Half Bridge	2_{dec}: Tesa ® impedance
T522F			
T521V			
T522V			
T521P			
T522P			
T521L			
T522L			

Series: Probes T523 / T524		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
T523F	Half Bridge	13_{dec}: 3-wire Half Bridge	2_{dec}: Tesa ® impedance
T524F			
T523V			
T524V			
T523P			
T524P			
T523L			
T524L			

Series: Probes T801 / T802		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
T801	Half Bridge	13_{dec}: 3-wire Half Bridge	2_{dec}: Tesa ® impedance
T802			

Series: Probes T851 / T852		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
T851	Full Bridge (LVDT)	5_{dec}: 5-wire LVDT	0_{dec}: High impedance / Marposs ®
T852			

Series: Probes 10P0 / 10P1 / 1P0		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
10P0	Tesa	13_{dec}: 3-wire Half Bridge	2_{dec}: Tesa ® impedance
10P1			
1P00			

6.3.1.5.4 Marposs® sensors

Series: Red Crown2 - LVDT Marposs		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
F05	Marposs	5_{dec}: 5-wire LVDT	0_{dec}: High impedance / Marposs ®
FR05			
F10			
FR10			
FPA10			
FP10			
FVA10			
FV10			
F21			
FR21			
FPA21			
FP21			
FVA21			
FV21			

Series: Red Crown2 HBT Marposs		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
H05	Half Bridge	13_{dec}: 3-wire Half Bridge	2_{dec}: Tesa ® impedance
HR05			
H10			
HR10			
HPA10			
HP10			
HVA10			
HV10			
H21			
HR21			
HPA21			
HP21			
HVA21			
HV21			

6.3.1.5.5 Solatron sensors

Series: AX		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
AX/1.0/S	LVDT	5_{dec}: 5-wire LVDT	0_{dec}: High impedance / Marposs ®
AX/1.5/S			
AX/2.5/S			
AX/5.0/S			
AX/10.0/S			
AX/1.0/SH	Half Bridge	13_{dec}: 3-wire Half Bridge	2_{dec}: Tesa ® impedance
AX/1.5/SH			
AX/2.5/SH			
AX/5.0/SH			
AX/10.0/SH			

6.3.1.5.6 Messotron sensors

Series: Displacement transducers DF		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
DF 6	LVDT	4_{dec}: 4-wire LVDT	0_{dec}: High impedance / Marposs ®
DF 12			
DF 20			
DF 25			
DF 50			
DF 80			
DF 100			
DF 130			

6.3.1.5.7 MEAS / Lucas Schaevitz® sensors

Series: HR		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
050 HR	LVDT	4_{dec}: 4-wire LVDT (*)	0_{dec}: High impedance / Marposs ®
100 HR			
200 HR			
300 HR			
500 HR			

Series: GCA		EL5072 Setting recommendations	
Manufacturer designation	**Compatibility**	**0x80n1:12 „Connection type“**	**0x80n1:13 „Sensor impedance“**
GCA050	LVDT	4_{dec}: 4-wire LVDT (*)	0_{dec}: High impedance / Marposs ®
GCA125			
GCA250			
GCA500			

(*) Separate cables are used for the secondary coil. It is recommended to connect two cables as close as possible to the measuring probe, so that a 4-wire LVDT signal can be evaluated at the EL5072.

6.3.1.6 Notes for RVDTs

The following must be observed when commissioning RVDTs:

- An RVDT should be parameterized like a corresponding LVDT sensor.
- The following information is to be converted and entered specific to the RVDT:
  - RVDT sensitivity (0x80n1:16 "Sensitivity")
  - RVDT maximum travel path (0x80n1:18 "Overall sensor range")

RVDT sensitivity specification ("Sensitivity", index 0x80n1:16)

The sensitivity of an RVDT sensor indicates the ratio of the measurement output voltage as a function of the core displacement and the excitation voltage and is often referred to as the scale factor

Scale factor [$V_{output}/V_{supply}/°_{measuring\ path}$]

Conversion formula:

Sensitivity [mV/V/mm] = 1 / (Scale Factor [V/V/deg])

RVDT specification of maximum travel path ("Overall sensor range", index 0x80n1:18)

For RVDTs the maximum travel path or maximum angle of rotation is not specified in mm but in degrees [°]. This information should be entered directly in index 0x80n1:18 "Overall sensor range", symmetrically with "+/-".

If the maximum angle of rotation is not specified, it can be calculated using the following information. This information is required to set internal gain factors so that a valid value is still measured at the maximum angle of rotation. If the value entered is too low, amplitude errors may occur in the limit range. These are displayed via the process data 0x60n0:04 "Input error".

Maximum angle of rotation = Maximum signal voltage Usig / ScaleFactor / Excitation voltage Uexc

6.3.1.7 Mechanical cut-off frequency (0x80n1:19 “Mech. operating frequency”)

The input of the mechanical cut-off frequency via index 0x80n1:19 "Mech. operating frequency" offers the option of signal filtering. The following filter frequencies are available:

Index 0x80n1:16 [▶ 139] "Mech. operating Frequency"	
Setting	**Meaning**
Filter disabled	Additional cut-off frequency filter is inactive, only internal filter 1 is active
20_{dec}	20 Hz
30_{dec}	30 Hz
40_{dec}	40 Hz
50_{dec}	50 Hz
60_{dec}	60 Hz
80_{dec}	80 Hz
100_{dec}	100 Hz
200_{dec}	200 Hz

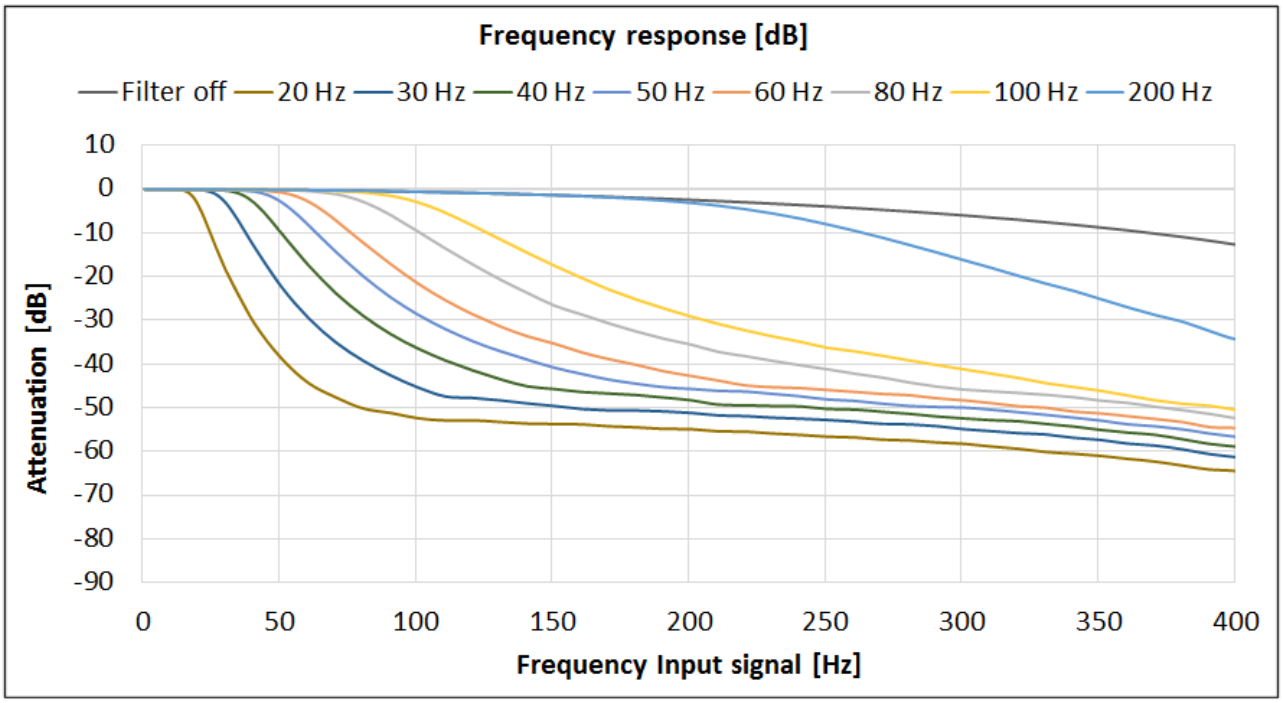


Fig. 108: Filter characteristic

6.3.1.8 User calibration via offset (0x80n1:1C) and gain (0x80n01:1D)

An offset and a gain object are available for user calibration of the position value. The new position value is calculated according to the following formula:
New position value = (position value * user gain) + user offset

Index (hex)	Name	Meaning
0x80n0:0A [▶ 138]	Enable user calibration	Enables user calibration
0x80n1:1C [▶ 139]	User offset	User offset compensation of the position value, default 0
0x80n1:1D [▶ 139]	User gain	User gain compensation of the position value, default 1

NOTE

Changes are not applied when password protection is enabled

The entries for user calibration can be protected with a password via index 0xF009 (see chapter Password protection for user data [▶ 117]). If password protection is enabled, any changes in the protected objects are not applied. There is **no** error message!

- Check whether the password protection has been enabled via index 0xF009 [▶ 140]!

6.3.1.9 Password protection for user data

Some user data are protected against unwanted or inadvertent writing by an additional password to be entered in CoE 0xF009:

- CoE write accesses by the user, PLC or startup entries in *Single*- or *CompleteAccess* mode
- Overwrite the values by *RestoreDefaultParameter* Access to 0x80n0 and 0x80n5.

Use of CoE 0xF009

- Entering 0x12345678 enables password protection → Object displays "1" (enabled).
  Protected objects can now no longer be changed. A write access attempt results in an error message, e.g.:
  CoE ('InitDown' 0x8011:1d) - SDO Abort ('Data cannot be transferred or stored to the application because of local control.', 0x08000021).
- Entering 0x11223344 disables password protection → Object displays "0" (disabled).

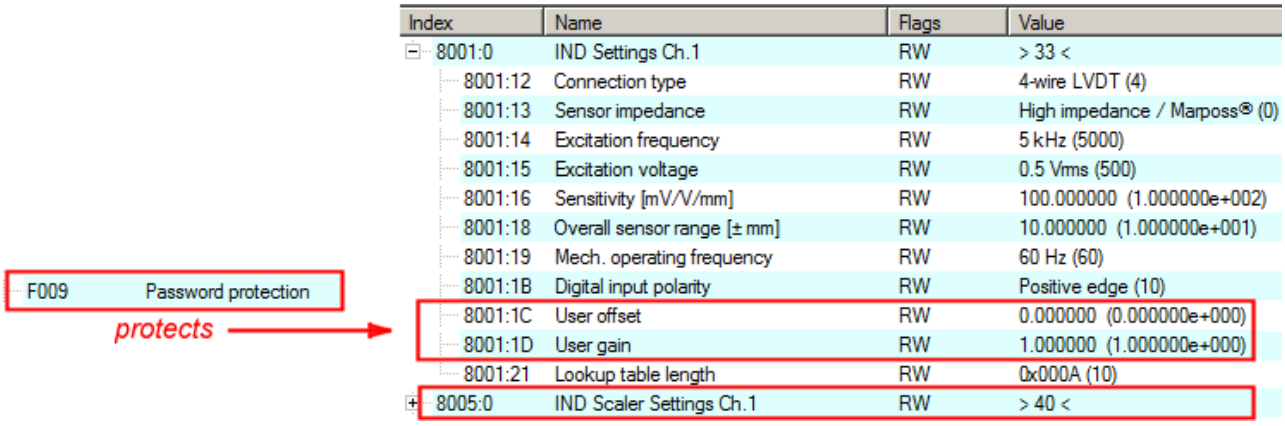


Fig. 109: Password protection for EL5072 using channel 1 as an example

The password protection applies to the following EL5072 user data:

- 0x80n1:1C User Offset
- 0x80n0:1D User Gain
- 0x80n5:0 IND Scaler Settings Ch.n

6.3.1.10 Lookup table (0x80n1:21 “lookup table length”)

Using the lookup table offers the option to perform a transformation of the input values within the terminal. In the task cycle the terminal uses interpolation points to return a corresponding predefined output value for each input value. This allows reconciliation of non-linearities in the signal sequence.

- The lookup table is enabled via index 0x80n0:0B "Enable lookup table"
- The total length of the interpolation points to be used is entered in index 0x80n1:21 "Lookup table length". A maximum of 20 X/Y value pairs can be entered, which corresponds to a maximum lookup table capacity of 40 values.
- The respective X and Y value pairs are to be entered in object 0x80n5 "IND Scaler Settings Ch.n". Note the following regarding the entries in the lookup table:
  - The odd subindices (1, 3, 5, ...) refer to the corresponding X-values, i.e. the values measured by the sensor
  - The even subindices (2, 4, 6, ...) refer to the corresponding Y-values, i.e. the values to be mapped in each case
  - Each X-value must be accompanied by a corresponding Y-value. Within the interpolation points the signal is interpolated; above/below it is extrapolated with the last/first available slope.

Index (hex)	Name	Meaning
0x80n0:0B [▶ 138]	Enable user lookup table	Enable lookup table
0x80n1:21 [▶ 139]	Lookup table length	Number of x/y value pairs in the lookup table
0x80n5:01 [▶ 140]	Scaler Value 1	Interpolation point / lookup x-value 1
0x80n5:02	Scaler Value 2	Interpolation point / lookup y-value 1
…		
0x80n5:27	Scaler Value 39	Interpolation point / lookup x-value 20
0x80n5:28	Scaler Value 40	Interpolation point / lookup y-value 20

6.3.1.11 Direction inversion, index 0x80n0:05 (Sign inversion)

To adapt the direction of deflection to the application, the direction can be inverted by setting the bit in index 0x80n0:05 [▶ 138] "Sign inversion".

6.3.2 Position value output

The position value:

- is output in index 0x60n1:01 "Position"
- as signed integer data type DINT (lower limit: -2147483648, upper limit: 2147483648)
- in the unit nm

Position value overflow / underflow

An overflow or underflow of the position value is indicated by the process data in index 0x60n0:01 [▶ 141] "Underrange" or 0x60n0:02 [▶ 141] "Overrange". The position value is frozen.

- "Underrange", index 0x60n0:01
  - TRUE: The lower position limit (-2147483648) has been reached.
  - FALSE: The lower position limit (-2147483648) has not been reached.
- "Overrange", index 0x60n0:02
  - TRUE: The upper position limit (2147483648) has been reached.
  - FALSE: The upper position limit (2147483648) has been not reached.

6.3.3 Application notes on measurement deviations

Shifting of the zero point and gain of the measuring probe

- The zero point and the gain of the measuring probe can shift with longer cable lengths between the probe and the EL5072. It is therefore recommended to install the measuring probe with the minimum necessary cable length between the probe and the EL5072.
- Due to slightly different coil characteristics and influences from other components used, the lower range value may vary slightly. Please follow the manufacturer's instructions to find the zero position of the measuring probe.
- In order to achieve an optimum measuring result, it is recommended to carry out a user calibration via gain and offset [▶ 117] over the entire measuring distance.

Possible causes of measurement deviations

- Mechanical components of the measuring probe (larger clearance of the mechanical components, e. g. ball head, bent plunger, …)
- Unsuitable or unshielded long cables
- Resistance changes of the supply lines due to temperature changes

NOTE	
	**Electrical connection of inductive measuring probes** Observe the Notes on the electrical connection of inductive measuring probes [▶ 40]!

6.3.4 Set position value

The position value (0x60n1:01 [▶ 141] "Position") can be set to a specified position value at runtime. The default can be enabled as follows:

- PLC variable: [▶ 120] the position can be set from the PLC application ("Set position value", index 0x70n0:11 [▶ 142])
- Digital input: [▶ 120] via a positive or negative edge at the digital input ("Set position on digital input", index 0x70n0:02 [▶ 142])

Setting the position value via a PLC variable ("Set position value", index 0x70n0:11)

The position value can be set to a specified value at runtime via the process data. In the PLC this bit can be linked to a digital input, for example, or used directly as a variable.

- The position value is specified via index 0x70n0:11 [▶ 142] "Set position value".
- Activation of the position value specification via index 0x70n0:01 [▶ 142] "Set position" = TRUE"
- For confirmation the "Set position done" bit in index 0x60n0:0B [▶ 141] is set to TRUE
- The position specification cannot be reactivated until index 0x70n0:01 [▶ 142] has been set to FALSE.

Set position value via digital input ("Set position on digital input", index 0x70n0:02)

- Index 0x80n1:1B [▶ 139] "Digital input polarity" can be used to define the level at which the position value is to be set to a specified value:
  - 0: input disabled
  - 10: positive edge
  - 11: negative edge
- Position value specification via 0x70n0:11 [▶ 142] "Set position value".
- **Index** 0x70n0:02 [▶ 142] **"Set position on digital input" = TRUE**
- With the first parameterized edge (positive or negative) at the digital input the current position value (0x60n1:01 [▶ 141]) is set to the preset position value (0x70n0:11 [▶ 142]).
- For confirmation the "Set position done" bit in index 0x60n0:0B is set to TRUE.
- The position specification cannot be reactivated until index 0x70n0:02 [▶ 142] has been set to FALSE.
- Index 0x80n0:06 [▶ 138] "Enable continuous digital input" can be used to specify whether it is necessary to reactivate the command:
  - FALSE:
    The following pulses at the digital input have no influence on the position value in index 0x60n1:01 [▶ 141] "Position" if the bit in index 0x70n0:02 [▶ 142] is set.
    A new position value can be written when index 0x60n0:0B [▶ 141] "Set position done" is FALSE.
  - TRUE:
    The position value 0x60n1:01 [▶ 141] "Position" is set to the specified position value (0x70n0:11 [▶ 142]) with each parameterized edge at the digital input.
    Renewed activation of the index 0x70n0:02 [▶ 142] is not necessary.
    The index 0x60n0:0B [▶ 141] "Set position done" is set to TRUE on a registered edge for a PLC cycle.
- The status of the digital input can be recorded via index 0x60n0:08 [▶ 141] "Digital input".

Parameterization of the digital input

The parameterization of the digital input via index 0x80n1:1B [▶ 139] "Digital input latch polarity" and index 0x80n0:06 [▶ 138] "Enable continuous digital input" applies simultaneously to the functions "Set position on digital input" index 0x70n0:02 [▶ 142] and "Enable latch on digital input" index 0x70n0:03 [▶ 142].

Saving the offset value ("Set Position Offset", index 0x90n0:11 [▶ 142]

An inductive measuring probe provides an absolute measured value, i.e. the position value does not have to be referenced after a voltage interruption.

- If a position specification is carried out via the process data 0x70n0:01 [▶ 142] "Set position" or 0x70n0:02 [▶ 142] "Set position on digital input", the offset value is stored in index 0x90n0:11 "Set Position Offset".
- The offset value can be reset to zero in the FB command object 0xFB00 [▶ 140] "Command" by the following commands.

Command	Description
0x8010	Deleting the offset value for both position values
0x8011	Deleting the offset value for position value of channel 1
0x8012	Deleting the offset value for position value of channel 2

Command entry via the System Manager

If the command input is written via the System Manager, the value must be entered in the "Binary" field (LOWBYTE first). If the command is initiated via the PLC, then this takes place automatically.

6.3.5 Save position value

The latch function enables the current position value (0x60n1:01 "Position") to be stored in a separate process data, independent of the cycle time. This function can be triggered via:

- Digital input: [▶ 122] via a positive or negative edge at the digital input

Save position value via digital input ("Enable latch on digital input", index 0x70n0:03 [▶ 142]

- Index 0x80n1:1B [▶ 139] "Digital input latch polarity" can be used to specify the level at which the position value is to be stored in a separate process data.
  - 0: input disabled
  - 10: positive edge
  - 11: negative edge
- **Index 0x70n0:03 "Enable latch on digital input" = TRUE**
- At the first parameterized edge at the digital input the current position value is stored in index 0x60n1:02 [▶ 141] "Latch value".
- For confirmation the "Latch extern valid" bit in index 0x60n0:0A [▶ 141] is set to TRUE.
- A renewed activation for saving the position value can only take place when index 0x70n0:03 has been set to FALSE.
- Index 0x80n0:06 [▶ 138] "Enable continuous digital input" can be used to specify whether it is necessary to reactivate the command
  - FALSE:
    The following pulses at the digital input have no influence on the latch value in index 0x60n1:02 [▶ 141] "Latch value" if the bit in index 0x70n0:03 [▶ 142] is set.
    A new position value can only be written to index 0x60n1:02 [▶ 141] "Latch value" when index 0x60n1:0A "Latch extern valid" is FALSE.
  - TRUE:
    The position value 0x60n1:01 [▶ 141] "Position" is written to index 0x60n1:02 [▶ 141] "Latch value" for each parameterized edge at the digital input.
    Renewed activation of the index 0x70n0:03 [▶ 142] is not necessary. The index 0x60n0:0A [▶ 141] "Latch extern valid" is set to TRUE on a registered edge for a PLC cycle.
- The status of the digital input can be recorded via index 0x60n0:08 [▶ 141] "Digital input"

6.3.6 Timestamp for the stored position value

Note on using the timestamp function

To be able to use the timestamp function, the EL5072 must be operated in "DC Latch active" mode.

The timestamp is made possible by the distributed clocks technology in the EtherCAT system. This local clock can be used to synchronize the data acquisition.

The EL5072 provides a timestamp on the stored position value.

The terminal provides the timestamp with an extent of 64 bits. The timestamps are contained in the following "Predefined PDO Assignment" [▶ 98]:

Predefined PDO Assignment
1. Ch. Standard, Timestamp
2. Ch. Standard, Timestamp

They can be selected or deselected as optional PDOs.

PDO Assignment	Name	Description
0x1A01 [▶ 96]	IND Latch Timestamp Channel 1	Contains timestamps in the 64 bit range for channel 1
0x1A03 [▶ 96]	IND Latch Timestamp Channel 2	Contains timestamps in the 64 bit range for channel 2

Timestamp on the stored position value

The process data index 0x60n2:01 "Latch event Timestamp" indicates the timestamp of the position value that was stored via the last positive edge of digital input.

Parameterization of the Latch input		0x60n2:01 "Latch event Timestamp" provides the timestamp for:
0x70n0:03	Enable Latch on digital input	the last rising edge at the digital input at which the counter value was stored in index 0x60n1:02 "Latch value"

6.3.7 Digital input

NOTE

Wiring the digital input

To ensure correct function of the digital input, in addition to the 24 V signal at connection point 1 for DI+ Ch1 or connection point 8 for DI+ Ch2, the corresponding ground connection must also be connected to connection point 10 for DI- Ch1 or connection point 16 for DI- Ch2.

The EL5072 has two digital inputs for voltages from 5 V_{DC} to 24 V_{DC}. One digital input per channel can be used as follows

- Set position value [▶ 124]: The position can be set to a specified position value via an edge ("Set position on digital input", index 0x70n0:02)
- Save position value [▶ 125]: The position value can be stored in a separate process data via an edge ("Enable latch on digital input", index 0x70n0:03)

Index 0x80n0:06 [▶ 138] "Enable continuous digital input" can be used to parameterize whether the function is executed at every parameterized edge or only once after every activation.

Furthermore, it is possible to disable the digital input.

Set position value via digital input ("Set position on digital input", index 0x70n0:02)

- Index 0x80n1:1B [▶ 139] "Digital input polarity" can be used to define the level at which the position value is to be set to a specified value:
  - 0: input disabled
  - 10: positive edge
  - 11: negative edge
- Position value specification via 0x70n0:11 [▶ 142] "Set position value".
- **Index 0x70n0:02 [▶ 142] "Set position on digital input" = TRUE**
- With the first parameterized edge (positive or negative) at the digital input the current position value (0x60n1:01 [▶ 141]) is set to the preset position value (0x70n0:11 [▶ 142]).
- For confirmation the "Set position done" bit in index 0x60n0:0B is set to TRUE.
- The position specification cannot be reactivated until index 0x70n0:02 [▶ 142] has been set to FALSE.
- Index 0x80n0:06 [▶ 138] "Enable continuous digital input" can be used to specify whether it is necessary to reactivate the command:
  - FALSE:
    The following pulses at the digital input have no influence on the position value in index 0x60n1:01 [▶ 141] "Position" if the bit in index 0x70n0:02 [▶ 142] is set.
    A new position value can be written when index 0x60n0:0B [▶ 141] "Set position done" is FALSE.
  - TRUE:
    The position value 0x60n1:01 [▶ 141] "Position" is set to the specified position value (0x70n0:11 [▶ 142]) with each parameterized edge at the digital input.
    Renewed activation of the index 0x70n0:02 [▶ 142] is not necessary.
    The index 0x60n0:0B [▶ 141] "Set position done" is set to TRUE on a registered edge for a PLC cycle.
- The status of the digital input can be recorded via index 0x60n0:08 [▶ 141] "Digital input".

Saving the offset value ("Set Position Offset", index 0x90n0:11 [▶ 142]

An inductive measuring probe provides an absolute measured value, i.e. the position value does not have to be referenced after a voltage interruption.

- If a position specification is carried out via the process data 0x70n0:01 [▶ 142] "Set position" or 0x70n0:02 [▶ 142] "Set position on digital input", the offset value is stored in index 0x90n0:11 "Set Position Offset".
- The offset value can be reset to zero in the FB command object 0xFB00 [▶ 140] "Command" by the following commands.

Command	Description
0x8010	Deleting the offset value for both position values
0x8011	Deleting the offset value for position value of channel 1
0x8012	Deleting the offset value for position value of channel 2

Command entry via the System Manager

If the command input is written via the System Manager, the value must be entered in the "Binary" field (LOWBYTE first). If the command is initiated via the PLC, then this takes place automatically.

Save position value via digital input ("Enable latch on digital input", index 0x70n0:03 [▶ 142]

- Index 0x80n1:1B [▶ 139] "Digital input latch polarity" can be used to specify the level at which the position value is to be stored in a separate process data.
  - 0: input disabled
  - 10: positive edge
  - 11: negative edge
- **Index 0x70n0:03 "Enable latch on digital input" = TRUE**
- At the first parameterized edge at the digital input the current position value is stored in index 0x60n1:02 [▶ 141] "Latch value".
- For confirmation the "Latch extern valid" bit in index 0x60n0:0A [▶ 141] is set to TRUE.
- A renewed activation for saving the position value can only take place when index 0x70n0:03 has been set to FALSE.
- Index 0x80n0:06 [▶ 138] "Enable continuous digital input" can be used to specify whether it is necessary to reactivate the command
  - FALSE:
    The following pulses at the digital input have no influence on the latch value in index 0x60n1:02 [▶ 141] "Latch value" if the bit in index 0x70n0:03 [▶ 142] is set.
    A new position value can only be written to index 0x60n1:02 [▶ 141] "Latch value" when index 0x60n1:0A "Latch extern valid" is FALSE.
  - TRUE:
    The position value 0x60n1:01 [▶ 141] "Position" is written to index 0x60n1:02 [▶ 141] "Latch value" for each parameterized edge at the digital input.
    Renewed activation of the index 0x70n0:03 [▶ 142] is not necessary. The index 0x60n0:0A [▶ 141] "Latch extern valid" is set to TRUE on a registered edge for a PLC cycle.
- The status of the digital input can be recorded via index 0x60n0:08 [▶ 141] "Digital input"

Parameterization of the digital input

The parameterization of the digital input via index 0x80n1:1B [▶ 139] "Digital input latch polarity" and index 0x80n0:06 [▶ 138] "Enable continuous digital input" applies simultaneously to the functions "Set position on digital input" index 0x70n0:02 [▶ 142] and "Enable latch on digital input" index 0x70n0:03 [▶ 142].

7 Diagnostics

7.1 EL5072 - Diagnostics

The integrated excitation source provides a wide range with different adjustable excitation frequencies and voltages. The following diagnostic options for the excitation source are available:

- Missing supply voltage of the excitation source [▶ 126]
- Excitation source not activated [▶ 126]
- Short circuit or overload of the excitation source [▶ 126]

The remeasured input signal is checked for:

- Amplitude error of the measuring signal [▶ 127]

Furthermore, an increased temperature in the terminal can lead to shutdown. The temperature overshoot can be caused by:

- Exceeding of the total excitation current [▶ 127]
- Insufficient ventilation of the terminal [▶ 127] (see chapter Installation positions [▶ 36])

Missing supply voltage of the excitation source

The excitation voltage is generated from the 24 V_{DC} supply of the power contacts. If the supply to the power contacts Up is missing or too low, no voltage can be provided. This is indicated by the following messages.

Error diagnosis	Display	Description
Up LED	Off	Supply voltage Up missing or too low of the power contacts
DiagMessage, Type "Error", text ID:	0x810B	Supply voltage Up too low or missing

Excitation source not activated

The excitation source is switched off in the delivery state and must be switched on by setting index 0x800:08 "Enable excitation" to TRUE.

If there is a short circuit or an overload, the excitation source is also switched off until the cause has been eliminated.

A disabled excitation source is indicated by:

Error diagnosis	Display	Description
Up LED	Flashing green	Supply voltage Up of the power contacts is present but excitation voltage U_{exc} has been disabled

Short circuit or overload of the excitation source

A short circuit or an overload of the excitation source is indicated by:

Error diagnosis	Display	Description
LED Excitation Error Ch.n	Red	There is a short circuit on the primary side on channel n, or an overload of the excitation source has been detected.
0x60n0:03 Excitation Error	TRUE	There is a short circuit on the primary side on channel n, or an overload of the excitation source has been detected.
DiagMessage, Type "Error", text ID	0x8624	Overcurrent detected Ch 1 + Ch 2, excitation voltage switched off

If a short circuit or an overload is detected, the excitation source is switched off to prevent a malfunction of the terminal. Checks are carried out at regular intervals to ascertain whether the short circuit or overload is still present. If this is no longer the case, the excitation source is switched on again.

The currently measured currents can be read out from the terminal. If the excitation source is switched off, only a leakage current flows.

Index	Name	Meaning
0x9000:19	Excitation current	Current measured on channel 1
0x9010:19	Excitation current	Current measured on channel 2
0xF900:19	Excitation current (sum)	Currently measured sum current on channel 1 + channel 2

Amplitude error of the measuring signal

In order to optimally resolve the input measuring signal, the internal gain factors are determined using the following parameters, among others:

- 0x80n1:15 "Excitation voltage"
- 0x80n1:16 "Sensitivity"
- 0x80n1:18 "Overall sensor range"

If these specifications are not selected correctly, it can lead to an input measuring signal that is too high (amplitude error). This is indicated by:

Error diagnosis	Display	Description
LED Amplitude Error Ch.n	Red	There is an amplitude error of the measuring signal on channel n
0x60n0:04 Input Error	TRUE	There is an amplitude error of the measuring signal on channel n
DiagMessage, Type "Error", text ID	0x8706	Channel n in saturation

Exceeding of the total excitation current

The maximum sum current of the excitation source is 50 mA_{rms} (140 mA_{pp}). If the excitation source is permanently subjected to this or higher loads, this can lead to an increased temperature inside the terminal. In order to avoid a malfunction of the terminal, a warning is issued first and, if the temperature continues to be exceeded, an error is issued which leads to the excitation source being switched off.

Error diagnosis	Display	Description
DiagMessage, Type "Warning", text ID	0x470e	Warning: Temperature has exceeded the limit
DiagMessage, Type "Error", text ID	0x8104	Error: Permissible temperature exceeded

The currently measured currents can be read out from the terminal. If the excitation source is switched off, only a leakage current flows.

Index	Name	Meaning
0x9000:19	Excitation current	Current measured on channel 1
0x9010:19	Excitation current	Current measured on channel 2
0xF900:19	Excitation current (sum)	Currently measured sum current on channel 1 + channel 2

Insufficient ventilation of the terminal

A high ambient temperature or insufficient air circulation inside the terminal housing, e.g. due to the installation position, can lead to excess temperature inside the terminal. In order to avoid a malfunction of the terminal, a warning is issued first and, if the temperature continues to be exceeded, an error is issued which leads to the excitation source being switched off.

Error diagnosis	Display	Description
DiagMessage, type "Warning", text ID	0x470e	Warning: Temperature has exceeded the limit
DiagMessage, type "Error", text ID	0x8104	Error: Permissible temperature exceeded

NOTE

Using the fan cartridge ZB8610

For demanding applications with high power requirements or high ambient temperatures, the use of the ZB8610 fan cartridge can be helpful. This enforces air circulation within the terminal housing and ensures better heat dissipation from the housing.

7.2 Diagnostics - basic principles of diag messages

DiagMessages designates a system for the transmission of messages from the EtherCAT Slave to the EtherCAT Master/TwinCAT. The messages are stored by the device in its own CoE under 0x10F3 and can be read by the application or the System Manager. An error message referenced via a code is output for each event stored in the device (warning, error, status change).

Definition

The *DiagMessages* system is defined in the ETG (EtherCAT Technology Group) in the guideline ETG.1020, chapter 13 “Diagnosis handling”. It is used so that pre-defined or flexible diagnostic messages can be conveyed from the EtherCAT Slave to the Master. In accordance with the ETG, the process can therefore be implemented supplier-independently. Support is optional. The firmware can store up to 250 DiagMessages in its own CoE.

Each DiagMessage consists of

- Diag Code (4-byte)
- Flags (2-byte; info, warning or error)
- Text ID (2-byte; reference to explanatory text from the ESI/XML)
- Timestamp (8-byte, local slave time or 64-bit Distributed Clock time, if available)
- Dynamic parameters added by the firmware

The DiagMessages are explained in text form in the ESI/XML file belonging to the EtherCAT device: on the basis of the Text ID contained in the DiagMessage, the corresponding plain text message can be found in the languages contained in the ESI/XML. In the case of Beckhoff products these are usually German and English.

Via the entry *NewMessagesAvailable* the user receives information that new messages are available.

DiagMessages can be confirmed in the device: the last/latest unconfirmed message can be confirmed by the user.

In the CoE both the control entries and the history itself can be found in the CoE object 0x10F3:

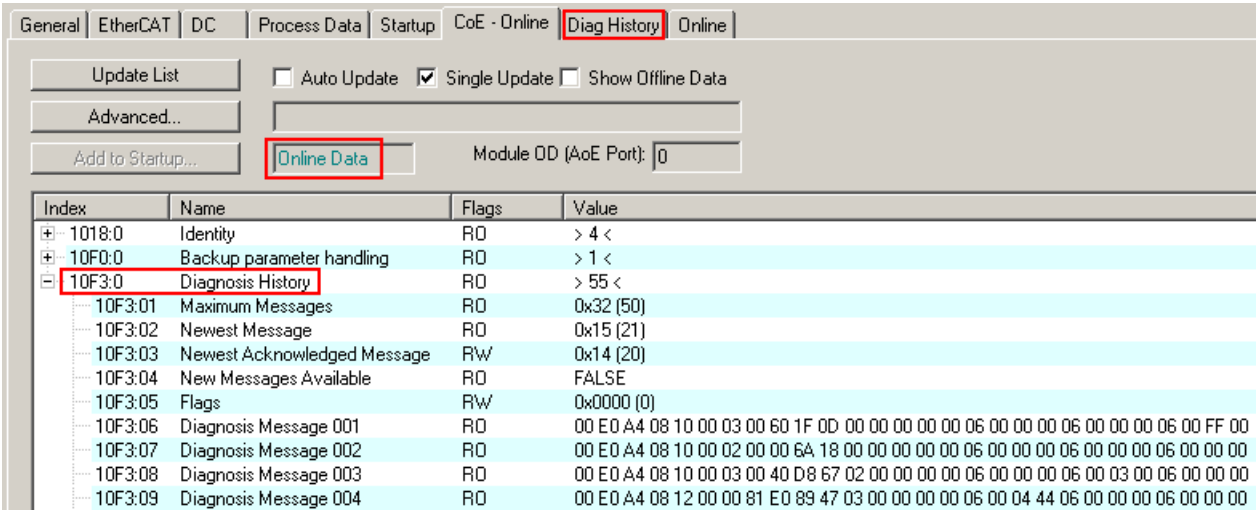


Fig. 110: DiagMessages in the CoE

The subindex of the latest *DiagMessage* can be read under 0x10F3:02.

Support for commissioning

The DiagMessages system is to be used above all during the commissioning of the plant. The diagnostic values e.g. in the StatusWord of the device (if available) are helpful for online diagnosis during the subsequent continuous operation.

TwinCAT System Manager implementation

From TwinCAT 2.11 DiagMessages, if available, are displayed in the device's own interface. Operation (collection, confirmation) also takes place via this interface.

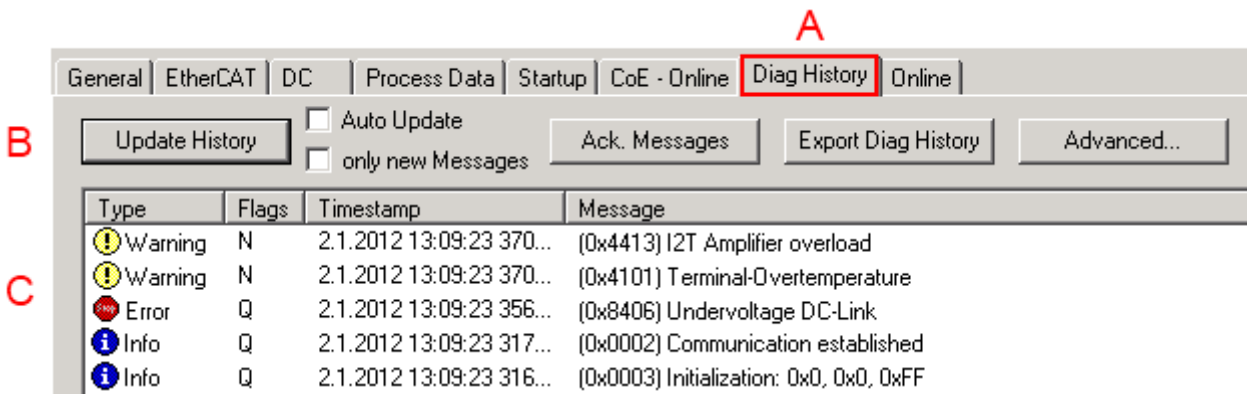


Fig. 111: Implementation of the DiagMessage system in the TwinCAT System Manager

The operating buttons (B) and the history read out (C) can be seen on the Diag History tab (A). The components of the message:

- Info/Warning/Error
- Acknowledge flag (N = unconfirmed, Q = confirmed)
- Time stamp
- Text ID
- Plain text message according to ESI/XML data

The meanings of the buttons are self-explanatory.

DiagMessages within the ADS Logger/Eventlogger

Since TwinCAT 3.1 build 4022 DiagMessages send by the terminal are shown by the TwinCAT ADS Logger. Given that DiagMessages are represented IO- comprehensive at one place, commissioning will be simplified. In addition, the logger output could be stored into a data file – hence DiagMessages are available long-term for analysis.

DiagMessages are actually only available locally in CoE 0x10F3 in the terminal and can be read out manually if required, e.g. via the DiagHistory mentioned above.

In the latest developments, the EtherCAT Terminals are set by default to report the presence of a DiagMessage as emergency via EtherCAT; the event logger can then retrieve the DiagMessage. The function is activated in the terminal via 0x10F3:05, so such terminals have the following entry in the StartUp list by default:

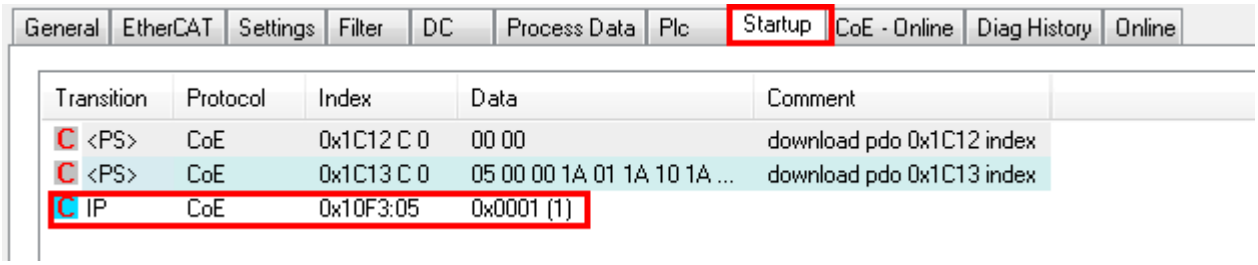


Fig. 112: Startup List

If the function is to be deactivated because, for example, many messages come in or the EventLogger is not used, the StartUp entry can be deleted or set to 0. The value can then be set back to 1 later from the PLC via CoE access if required.

Reading messages into the PLC

- In preparation -

Interpretation

Time stamp

The time stamp is obtained from the local clock of the terminal at the time of the event. The time is usually the distributed clock time (DC) from register x910.

Please note: When EtherCAT is started, the DC time in the reference clock is set to the same time as the local IPC/TwinCAT time. From this moment the DC time may differ from the IPC time, since the IPC time is not adjusted. Significant time differences may develop after several weeks of operation without a EtherCAT restart. As a remedy, external synchronization of the DC time can be used, or a manual correction calculation can be applied, as required: The current DC time can be determined via the EtherCAT master or from register x901 of the DC slave.

Structure of the Text ID

The structure of the MessageID is not subject to any standardization and can be supplier-specifically defined. In the case of Beckhoff EtherCAT devices (EL, EP) it usually reads according to **xyzz**:

x	y	zz
0: Systeminfo 2: reserved 1: Info 4: Warning 8: Error	0: System 1: General 2: Communication 3: Encoder 4: Drive 5: Inputs 6: I/O general 7: reserved	Error number

Example: Message 0x4413 --> Drive Warning Number 0x13

Overview of text IDs

Specific text IDs are listed in the device documentation.

Text ID	Type	Place	Text Message	Additional comment
0x0001	Information	System	No error	No error
0x0002	Information	System	Communication established	Connection established
0x0003	Information	System	Initialization: 0x%X, 0x%X, 0x%X	General information; parameters depend on event. See device documentation for interpretation.
0x1000	Information	System	Information: 0x%X, 0x%X, 0x%X	General information; parameters depend on event. See device documentation for interpretation.
0x1012	Information	System	EtherCAT state change Init - PreOp	
0x1021	Information	System	EtherCAT state change PreOp - Init	
0x1024	Information	System	EtherCAT state change PreOp - Safe-Op	
0x1042	Information	System	EtherCAT state change SafeOp - PreOp	
0x1048	Information	System	EtherCAT state change SafeOp - Op	
0x1084	Information	System	EtherCAT state change Op - SafeOp	
0x1100	Information	General	Detection of operation mode completed: 0x%X, %d	Detection of the mode of operation ended
0x1135	Information	General	Cycle time o.k.: %d	Cycle time OK
0x1157	Information	General	Data manually saved (Idx: 0x%X, SubIdx: 0x%X)	Data saved manually
0x1158	Information	General	Data automatically saved (Idx: 0x%X, SubIdx: 0x%X)	Data saved automatically
0x1159	Information	General	Data deleted (Idx: 0x%X, SubIdx: 0x%X)	Data deleted
0x117F	Information	General	Information: 0x%X, 0x%X, 0x%X	Information
0x1201	Information	Communication	Communication re-established	Communication to the field side restored This message appears, for example, if the voltage was removed from the power contacts and re-applied during operation.
0x1300	Information	Encoder	Position set: %d, %d	Position set - StartInputhandler
0x1303	Information	Encoder	Encoder Supply ok	Encoder power supply unit OK
0x1304	Information	Encoder	Encoder initialization successfully, channel: %X	Encoder initialization successfully completed
0x1305	Information	Encoder	Sent command encoder reset, channel: %X	Send encoder reset command
0x1400	Information	Drive	Drive is calibrated: %d, %d	Drive is calibrated
0x1401	Information	Drive	Actual drive state: 0x%X, %d	Current drive status
0x1705	Information		CPU usage returns in normal range (< 85%%)	Processor load is back in the normal range
0x1706	Information		Channel is not in saturation anymore	Channel is no longer in saturation
0x1707	Information		Channel is not in overload anymore	Channel is no longer overloaded
0x170A	Information		No channel range error anymore	A measuring range error is no longer active
0x170C	Information		Calibration data saved	Calibration data were saved
0x170D	Information		Calibration data will be applied and saved after sending the command “0x5AFE”	Calibration data are not applied and saved until the command "0x5AFE" is sent.

Text ID	Type	Place	Text Message	Additional comment
0x2000	Information	System	%s: %s	
0x2001	Information	System	%s: Network link lost	Network connection lost
0x2002	Information	System	%s: Network link detected	Network connection found
0x2003	Information	System	%s: no valid IP Configuration - Dhcp client started	Invalid IP configuration
0x2004	Information	System	%s: valid IP Configuration (IP: %d.%d.%d.%d) assigned by Dhcp server %d.%d.%d.%d	Valid IP configuration, assigned by the DHCP server
0x2005	Information	System	%s: Dhcp client timed out	DHCP client timeout
0x2006	Information	System	%s: Duplicate IP Address detected (%d.%d.%d.%d)	Duplicate IP address found
0x2007	Information	System	%s: UDP handler initialized	UDP handler initialized
0x2008	Information	System	%s: TCP handler initialized	TCP handler initialized
0x2009	Information	System	%s: No more free TCP sockets available	No free TCP sockets available.

Text ID	Type	Place	Text Message	Additional comment
0x4000	Warning		Warning: 0x%X, 0x%X, 0x%X	General warning; parameters depend on event. See device documentation for interpretation.
0x4001	Warning	System	Warning: 0x%X, 0x%X, 0x%X	
0x4002	Warning	System	%s: %s Connection Open (IN:%d OUT:%d API:%dms) from %d.%d.%d.%d successful	
0x4003	Warning	System	%s: %s Connection Close (IN:%d OUT:%d) from %d.%d.%d.%d successful	
0x4004	Warning	System	%s: %s Connection (IN:%d OUT:%d) with %d.%d.%d.%d timed out	
0x4005	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (Error: %u)	
0x4006	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (Input Data Size expected: %d Byte(s) received: %d Byte(s))	
0x4007	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (Output Data Size expected: %d Byte(s) received: %d Byte(s))	
0x4008	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (RPI:%dms not supported -> API:%dms)	
0x4101	Warning	General	Terminal-Overtemperature	Overtemperature. The internal temperature of the terminal exceeds the parameterized warning threshold.
0x4102	Warning	General	Discrepancy in the PDO-Configuration	The selected PDOs do not match the set operating mode. Sample: Drive operates in velocity mode, but the velocity PDO is but not mapped in the PDOs.
0x417F	Warning	General	Warning: 0x%X, 0x%X, 0x%X	
0x428D	Warning	General	Challenge is not Random	
0x4300	Warning	Encoder	Subincrements deactivated: %d, %d	Sub-increments deactivated (despite activated configuration)
0x4301	Warning	Encoder	Encoder-Warning	General encoder error
0x4302	Warning	Encoder	Maximum frequency of the input signal is nearly reached (channel %d)	
0x4303	Warning	Encoder	Limit counter value was reduced because of the PDO configuration (channel %d)	
0x4304	Warning	Encoder	Reset counter value was reduced because of the PDO configuration (channel %d)	
0x4400	Warning	Drive	Drive is not calibrated: %d, %d	Drive is not calibrated
0x4401	Warning	Drive	Starttype not supported: 0x%X, %d	Start type is not supported
0x4402	Warning	Drive	Command rejected: %d, %d	Command rejected
0x4405	Warning	Drive	Invalid modulo subtype: %d, %d	Modulo sub-type invalid
0x4410	Warning	Drive	Target overrun: %d, %d	Target position exceeded
0x4411	Warning	Drive	DC-Link undervoltage (Warning)	The DC link voltage of the terminal is lower than the parameterized minimum voltage. Activation of the output stage is prevented.
0x4412	Warning	Drive	DC-Link overvoltage (Warning)	The DC link voltage of the terminal is higher than the parameterized maximum voltage. Activation of the output stage is prevented.
0x4413	Warning	Drive	I2T-Model Amplifier overload (Warning)	• The amplifier is being operated outside the specification. • The I2T-model of the amplifier is incorrectly parameterized.
0x4414	Warning	Drive	I2T-Model Motor overload (Warning)	• The motor is being operated outside the parameterized rated values. • The I2T-model of the motor is incorrectly parameterized.

Text ID	Type	Place	Text Message	Additional comment
0x4415	Warning	Drive	Speed limitation active	The maximum speed is limited by the parameterized objects (e.g. velocity limitation, motor speed limitation). This warning is output if the set velocity is higher than one of the parameterized limits.
0x4416	Warning	Drive	Step lost detected at position: 0x%X%X	Step loss detected
0x4417	Warning	Drive	Motor overtemperature	The internal temperature of the motor exceeds the parameterized warning threshold
0x4418	Warning	Drive	Limit: Current	Limit: current is limited
0x4419	Warning	Drive	Limit: Amplifier I2T-model exceeds 100%%	The threshold values for the maximum current were exceeded.
0x441A	Warning	Drive	Limit: Motor I2T-model exceeds 100%%	Limit: Motor I2T-model exceeds 100%
0x441B	Warning	Drive	Limit: Velocity limitation	The threshold values for the maximum speed were exceeded.
0x441C	Warning	Drive	STO while the axis was enabled	An attempt was made to activate the axis, despite the fact that no voltage is present at the STO input.
0x4600	Warning	General IO	Wrong supply voltage range	Supply voltage not in the correct range
0x4610	Warning	General IO	Wrong output voltage range	Output voltage not in the correct range
0x4705	Warning		Processor usage at %d %%	Processor load at %d %%
0x470A	Warning		EtherCAT Frame missed (change Settings or DC Operation Mode or Sync0 Shift Time)	EtherCAT frame missed (change DC Operation Mode or Sync0 Shift Time under Settings)

Text ID	Type	Place	Text Message	Additional comment
0x8000	Error	System	%s: %s	
0x8001	Error	System	Error: 0x%X, 0x%X, 0x%X	General error; parameters depend on event. See device documentation for interpretation.
0x8002	Error	System	Communication aborted	Communication aborted
0x8003	Error	System	Configuration error: 0x%X, 0x%X, 0x%X	General; parameters depend on event. See device documentation for interpretation.
0x8004	Error	System	%s: Unsuccessful FwdOpen-Response received from %d.%d.%d.%d (%s) (Error: %u)	
0x8005	Error	System	%s: FwdClose-Request sent to %d.%d.%d.%d (%s)	
0x8006	Error	System	%s: Unsuccessful FwdClose-Response received from %d.%d.%d.%d (%s) (Error: %u)	
0x8007	Error	System	%s: Connection with %d.%d.%d.%d (%s) closed	
0x8100	Error	General	Status word set: 0x%X, %d	Error bit set in the status word
0x8101	Error	General	Operation mode incompatible to PDO interface: 0x%X, %d	Mode of operation incompatible with the PDO interface
0x8102	Error	General	Invalid combination of Inputs and Outputs PDOs	Invalid combination of input and output PDOs
0x8103	Error	General	No variable linkage	No variables linked
0x8104	Error	General	Terminal-Overtemperature	The internal temperature of the terminal exceeds the parameterized error threshold. Activation of the terminal is prevented
0x8105	Error	General	PD-Watchdog	Communication between the fieldbus and the output stage is secured by a Watchdog. The axis is stopped automatically if the fieldbus communication is interrupted. • The EtherCAT connection was interrupted during operation. • The Master was switched to Config mode during operation.
0x8135	Error	General	Cycle time has to be a multiple of 125 µs	The IO or NC cycle time divided by 125 µs does not produce a whole number.
0x8136	Error	General	Configuration error: invalid sampling rate	Configuration error: Invalid sampling rate
0x8137	Error	General	Electronic type plate: CRC error	Content of the external name plate memory invalid.
0x8140	Error	General	Sync Error	Real-time violation
0x8141	Error	General	Sync%X Interrupt lost	Sync%X Interrupt lost
0x8142	Error	General	Sync Interrupt asynchronous	Sync Interrupt asynchronous
0x8143	Error	General	Jitter too big	Jitter limit violation
0x817F	Error	General	Error: 0x%X, 0x%X, 0x%X	
0x8200	Error	Communication	Write access error: %d, %d	Error while writing
0x8201	Error	Communication	No communication to field-side (Auxiliary voltage missing)	• There is no voltage applied to the power contacts. • A firmware update has failed.
0x8281	Error	Communication	Ownership failed: %X	
0x8282	Error	Communication	To many Keys founded	
0x8283	Error	Communication	Key Creation failed: %X	
0x8284	Error	Communication	Key loading failed	
0x8285	Error	Communication	Reading Public Key failed: %X	
0x8286	Error	Communication	Reading Public EK failed: %X	
0x8287	Error	Communication	Reading PCR Value failed: %X	
0x8288	Error	Communication	Reading Certificate EK failed: %X	
0x8289	Error	Communication	Challenge could not be hashed: %X	
0x828A	Error	Communication	Tickstamp Process failed	
0x828B	Error	Communication	PCR Process failed: %X	
0x828C	Error	Communication	Quote Process failed: %X	
0x82FF	Error	Communication	Bootmode not activated	Boot mode not activated
0x8300	Error	Encoder	Set position error: 0x%X, %d	Error while setting the position

Text ID	Type	Place	Text Message	Additional comment
0x8301	Error	Encoder	Encoder increments not configured: 0x%X, %d	Encoder increments not configured
0x8302	Error	Encoder	Encoder error	The amplitude of the resolver is too small
0x8303	Error	Encoder	Encoder power missing (channel %d)	
0x8304	Error	Encoder	Encoder communication error, channel: %X	Encoder communication error
0x8305	Error	Encoder	EnDat2.2 is not supported, channel: %X	EnDat2.2 is not supported
0x8306	Error	Encoder	Delay time, tolerance limit exceeded, 0x%X, channel: %X	Runtime measurement, tolerance exceeded
0x8307	Error	Encoder	Delay time, maximum value exceeded, 0x%X, channel: %X	Runtime measurement, maximum value exceeded
0x8308	Error	Encoder	Unsupported ordering designation, 0x%X, channel: %X (only 02 and 22 is supported)	Wrong EnDat order ID
0x8309	Error	Encoder	Encoder CRC error, channel: %X	Encoder CRC error
0x830A	Error	Encoder	Temperature %X could not be read, channel: %X	Temperature cannot be read
0x830C	Error	Encoder	Encoder Single-Cycle-Data Error, channel. %X	CRC error detected. Check the transmission path and the CRC polynomial
0x830D	Error	Encoder	Encoder Watchdog Error, channel. %X	The sensor has not responded within a predefined time period
0x8310	Error	Encoder	Initialisation error	
0x8311	Error	Encoder	Maximum frequency of the input signal is exceeded (channel %d)	
0x8312	Error	Encoder	Encoder plausibility error (channel %d)	
0x8313	Error	Encoder	Configuration error (channel %d)	
0x8314	Error	Encoder	Synchronisation error	
0x8315	Error	Encoder	Error status input (channel %d)	
0x8400	Error	Drive	Incorrect drive configuration: 0x%X, %d	Drive incorrectly configured
0x8401	Error	Drive	Limiting of calibration velocity: %d, %d	Limitation of the calibration velocity
0x8402	Error	Drive	Emergency stop activated: 0x%X, %d	Emergency stop activated
0x8403	Error	Drive	ADC Error	Error during current measurement in the ADC
0x8404	Error	Drive	Overcurrent	Overcurrent in phase U, V or W
0x8405	Error	Drive	Invalid modulo position: %d	Modulo position invalid
0x8406	Error	Drive	DC-Link undervoltage (Error)	The DC link voltage of the terminal is lower than the parameterized minimum voltage. Activation of the output stage is prevented.
0x8407	Error	Drive	DC-Link overvoltage (Error)	The DC link voltage of the terminal is higher than the parameterized maximum voltage. Activation of the output stage is prevented.
0x8408	Error	Drive	I2T-Model Amplifier overload (Error)	• The amplifier is being operated outside the specification. • The I2T-model of the amplifier is incorrectly parameterized.
0x8409	Error	Drive	I2T-Model motor overload (Error)	• The motor is being operated outside the parameterized rated values. • The I2T-model of the motor is incorrectly parameterized.
0x840A	Error	Drive	Overall current threshold exceeded	Total current exceeded
0x8415	Error	Drive	Invalid modulo factor: %d	Modulo factor invalid
0x8416	Error	Drive	Motor overtemperature	The internal temperature of the motor exceeds the parameterized error threshold. The motor stops immediately. Activation of the output stage is prevented.
0x8417	Error	Drive	Maximum rotating field velocity exceeded	Rotary field speed exceeds the value specified for dual use (EU 1382/2014).
0x841C	Error	Drive	STO while the axis was enabled	An attempt was made to activate the axis, despite the fact that no voltage is present at the STO input.
0x8550	Error	Inputs	Zero crossing phase %X missing	Zero crossing phase %X missing

Text ID	Type	Place	Text Message	Additional comment
0x8551	Error	Inputs	Phase sequence Error	Wrong direction of rotation
0x8552	Error	Inputs	Overcurrent phase %X	Overcurrent phase %X
0x8553	Error	Inputs	Overcurrent neutral wire	Overcurrent neutral wire
0x8581	Error	Inputs	Wire broken Ch %D	Wire broken Ch %d
0x8600	Error	General IO	Wrong supply voltage range	Supply voltage not in the correct range
0x8601	Error	General IO	Supply voltage to low	Supply voltage too low
0x8602	Error	General IO	Supply voltage to high	Supply voltage too high
0x8603	Error	General IO	Over current of supply voltage	Overcurrent of supply voltage
0x8610	Error	General IO	Wrong output voltage range	Output voltage not in the correct range
0x8611	Error	General IO	Output voltage to low	Output voltage too low
0x8612	Error	General IO	Output voltage to high	Output voltage too high
0x8613	Error	General IO	Over current of output voltage	Overcurrent of output voltage
0x8700	Error		Channel/Interface not calibrated	Channel/interface not synchronized
0x8701	Error		Operating time was manipulated	Operating time was manipulated
0x8702	Error		Oversampling setting is not possible	Oversampling setting not possible
0x8703	Error		No slave controller found	No slave controller found
0x8704	Error		Slave controller is not in Bootstrap	Slave controller is not in bootstrap
0x8705	Error		Processor usage to high (>= 100%%)	Processor load too high (>= 100%%)
0x8706	Error		Channel in saturation	Channel in saturation
0x8707	Error		Channel overload	Channel overload
0x8708	Error		Overloadtime was manipulated	Overload time was manipulated
0x8709	Error		Saturationtime was manipulated	Saturation time was manipulated
0x870A	Error		Channel range error	Measuring range error for the channel
0x870B	Error		no ADC clock	No ADC clock available
0xFFFF	Information		Debug: 0x%X, 0x%X, 0x%X	Debug: 0x%X, 0x%X, 0x%X

8 Object description and parameterization

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.

Parameterization via the CoE list (CAN over EtherCAT)

The EtherCAT device is parameterized via the CoE-Online tab [▶ 75] (double-click on the respective object) or via the Process Data tab [▶ 72](allocation of PDOs). Please note the following general CoE notes [▶ 21] when using/manipulating the CoE parameters:

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

8.1 Restore object

Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters	Restore the default settings	UINT8	RO	0x01 (1_{dec})
1011:01	SubIndex 001	If this object is set to “**0x64616F6C**” in the set value dialog, all backup objects are reset to their delivery state. Note: Some FW versions also accept the following input: **"0x6C6F6164".**	UINT32	RW	0x00000000 (0_{dec})

8.2 Configuration data

Index 80n0 IND Boolean settings Ch.n+1 (for n=0 [channel 1], n=1 [channel 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
80n0:0	IND Boolean Settings Ch.n+1	Maximum	UINT8	RO	0x0B (11_{dec})
80n0:05	Sign inversion [▶ 118]	0: Signal inversion disabled (default) 1: Signal inversion enabled	BOOLEAN	RW	0x00 (0_{dec})
80n0:06	Enable continuous digital input [▶ 120]	Determines whether reactivation of the command is required:	BOOLEAN	RW	0x00 (0_{dec})
80n0:08	Enable excitation [▶ 105]	0: Excitation voltage disabled (default) 1: Excitation voltage enabled	BOOLEAN	RW	0x00 (0_{dec})
80n0:0A	Enable user calibration [▶ 117]	0: User calibration disabled (default) 1: User calibration enabled	BOOLEAN	RW	0x00 (0_{dec})
80n0:0B	Enable lookup table [▶ 118]	0: "lookup table" disabled (default) 1: "lookup table" enabled	BOOLEAN	RW	0x00 (0_{dec})

Index 80n1 IND Settings Ch.n+1 (for n=0 [channel 1], n=1 [channel 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
80n1:0	IND Settings Ch.n+1	Maximum subindex	UINT8	RO	0x21 (33_{dec})
80n1:12	Connection type [▶ 104]	Connection type, selection options. 4_{dec}: 4-wire LVDT (default) 5_{dec}:5-wire LVDT 6_{dec}: 6-wire LVDT 13_{dec}:3-wire half bridge 15_{dec}:5-wire half bridge	UINT16	RW	0x0004 (4_{dec})
80n1:13	Sensor impedance [▶ 104]	Sensor impedance setting: 0_{dec}: High impedance / Marposs ® (default) 1_{dec}: Mahr ® impedance 2_{dec}: Tesa ® impedance	UINT16	RW	0x0000 (0_{dec})
80n1:14	Excitation frequency [▶ 105]	Excitation voltage setting 1000_{dec}: 1 kHz 2000_{dec}: 2 kHz 2500_{dec}: 2.5 kHz 4000_{dec}: 4 kHz 5000_{dec}: 5 kHz (default) 7500_{dec}: 7.5 kHz 10000_{dec}: 10 kHz 12500_{dec}: 12.5 kHz 13000_{dec}: 13 kHz 15000_{dec}: 15 kHz 19400_{dec}: 19.4 kHz 20000_{dec}: 20 kHz	UINT16	RW	0x1388 (5000_{dec})
80n1:15	Excitation voltage [▶ 105]	Sensor supply setting 500_{dec}: 0.5 V_{rms} (default) 1000_{dec}: 1 V_{rms} 1500_{dec}: 1.5 V_{rms} 2000_{dec}: 2 V_{rms} 2500_{dec}: 2.5 V_{rms} 3000_{dec}: 3 V_{rms} 3500_{dec}: 3.5 V_{rms} 4000_{dec}: 4 V_{rms} 4500_{dec}: 4.5 V_{rms} 5000_{dec}: 5 V_{rms} 5500_{dec}: 5.5 V_{rms} 6000_{dec}: 6 V_{rms} 6500_{dec}: 6.5 V_{rms} 7000_{dec}: 7 V_{rms}	UINT16	RW	0x01F4 (500_{dec})
80n1:16	Sensitivity [mV/V/mm] [▶ 106]	Specification of the sensor sensitivity, value in mV/V/mm	REAL32	RW	0x42C80000 (100_{float})
80n1:18	Overall sensor range [±mm] [▶ 106]	Maximum travel path of the sensor, symmetrical specification in ±mm	REAL32	RW	0x41200000 (10_{float})
80n1:19	Mech. Operating frequency [▶ 116]	Selection of the filter frequency 20_{dec}: 20 Hz 30_{dec}: 30 Hz 40_{dec}: 40 Hz 50_{dec}: 50 Hz 60_{dec}: 60 Hz (default) 80_{dec}: 80 Hz 100_{dec}: 100 Hz 200_{dec}: 200 Hz	UINT16	RW	0x003C (60_{dec})
80n1:1B	Digital input latch polarity [▶ 120]	Level at which the position value is to be set to a specified value: 0_{dec}: Input disabled 10_{dec}: Latch on positive edge (default) 11_{dec}: Latch on negative edge	UINT16	RW	0x000A (10_{dec})
80n1:1C	User offset [▶ 117]	User offset compensation of the position value, default: 0	REAL32	RW	0x00000000 (0_{float})
80n1:1D	User gain [▶ 117]	User gain compensation of the position value, default: 1	REAL32	RW	0x3F800000 (1_{float})
80n1:21	Lookup table length [▶ 118]	Number of x/y value pairs in the lookup table	UINT16	RW	0x000A (10_{dec})

Index 80n5 IND Scaler Settings Ch.n+1 (for n=0 [channel 1], n=1 [channel 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
80n5:0	IND Scaler Settings Ch.n+1	Lookup table [▶ 118] with 20 x/y value pairs	UINT8	RO	0x28 (40_{dec})
80n5:01	Scaler Value 1	Interpolation point / lookup x-value 1	INT32	RW	0x00000000 (0_{dec})
80n5:02	Scaler Value 2	Interpolation point / lookup y-value 1	INT32	RW	0x00000000 (0_{dec})
…					
80n5:27	Scaler Value 39	Interpolation point / lookup x-value 20	INT32	RW	0x00000000 (0_{dec})
80n5:28	Scaler Value 40	Interpolation point / lookup y-value 20	INT32	RW	0x00000000 (0_{dec})

8.3 Configuration data (vendor-specific)

Index 80nE IND Vendor tune Ch.n+1 (for n=0 [channel 1], n=1 [channel 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
80nE:0	IND Vendor tune Ch.n+1	Manufacturer-specific compensation data	UINT8	RO	0x14 (20_{dec})

Index 80nF IND Vendor data Ch.n+1 (for n=0 [channel 1], n=1 [channel 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
80nF:0	IND Vendor data Ch.n+1	Manufacturer-specific compensation data	UINT8	RO	0x40 (64_{dec})

8.4 Command object

Index FB00 command

Index (hex)	Name	Meaning		Data type	Flags	Default
FB00:0	Command	Max. Subindex		UINT8	RO	0x03 (3_{dec})
FB00:01	Request	Commands can be sent to the terminal via the request object. Note the Instructions for entering commands via the System Manager [▶ 121].		OCTET-STRING[2]	RW	{0}
		0x8010	Deleting the offset value for both position values			
		0x8011	Deleting the offset value for position value of channel1			
		0x8012	Deleting the offset value for position value of channel2			
FB00:02	Status	Status of the command currently being executed 0: Command executed without error 255: Command is being executed		UINT8	RO	0x00 (0_{dec})
FB00:03	Response	Optional return value of the command		OCTET-STRING[4]	RO	{0}

8.5 Input data

Index 60n0 IND Boolean Inputs Ch.n+1 (for n=0 [channel 1], n=1 [channel 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
60n0:0	IND Boolean Inputs Ch.n+1	Maximum subindex	UINT8	RO	0x0F (15_{dec})
60n0:01	Underrange [▶ 119]	Measurement is below range	BOOLEAN	RO	0x00 (0_{dec})
60n0:02	Overrange [▶ 119]	Measuring range exceeded	BOOLEAN	RO	0x00 (0_{dec})
60n0:03	Excitation error	Short circuit on primary side on channel n or overload of excitation source was detected	BOOLEAN	RO	0x00 (0_{dec})
60n0:04	Input error	Wire breakage on secondary side or amplitude error on channel n was detected	BOOLEAN	RO	0x00 (0_{dec})
60n0:08	Digital input [▶ 120]	Status of the digital input	BOOLEAN	RO	0x00 (0_{dec})
60n0:0A	Latch extern valid [▶ 122]	The counter value was locked via the external latch. The data in index 0x60n1:02 [▶ 141] "Latch value" corresponds to the latched value when the bit is set. To re-enable the latch input, index 0x70n0:03 [▶ 142] "Enable latch on digital input" must first be disabled and then reset.	BOOLEAN	RO	0x00 (0_{dec})
60n0:0B	Set position done [▶ 120]	0: a new position value can be written to index 0x60n1:01 [▶ 141] "Position" 1: A value was set in index 0x60n1:01 [▶ 141] "Position". A new position value can only be set after "Set position done" has been reset to 0.	BOOLEAN	RO	0x00 (0_{dec})
60n0:0D	Diag	Indicates that a new message is available in the "Diag History"	BOOLEAN	RO	0x00 (0_{dec})
60n0:0E	TxPDO State	Validity of the data of the associated TxPDO 0: valid 1: invalid	BOOLEAN	RO	0x00 (0_{dec})
60n0:0F	Input cycle counter	2-bit counter for synchronization (increments only when a new process data is ready)	BIT2	RO	0x00 (0_{dec})

Index 60n1 IND Inputs Ch. n+1 (for n=0 [channel 1], n=1 [channel 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
60n1:0	IND Inputs Ch.n+1	Maximum subindex	UINT8	RO	0x02 (2_{dec})
60n1:01	Position [▶ 120]	Position value	DINT	RO	0x00000000 (0_{dec})
60n1:02	Latch value [▶ 122]	Latch value	DINT	RO	0x00000000 (0_{dec})

Index 60n2 IND Latch Timestamp Ch. n+1 (for n=0 [channel 1], n=1 [channel 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
60n2:0	IND Latch Timestamp Ch.n+1	Maximum subindex	UINT8	RO	0x01 (1_{dec})
60n2:01	Latch event Timestamp	Timestamp of the stored position value via the digital input	ULINT64	RO	0x00000000 (0_{dec})

8.6 Output data

Index 70n0 IND Outputs Ch.n+1 (for n=0 [channel 1], n=1 [channel 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
70n0:0	IND Outputs Ch.n+1	Maximum subindex	UINT8	RO	0x11 (17_{dec})
70n0:01	Set position [▶ 120]	Activation of the position value specification	BOOLEAN	RO	0x00 (0_{dec})
70n0:02	Set position on digital input [▶ 120]	Activates setting of the position value via the digital input	BOOLEAN	RO	0x00 (0_{dec})
70n0:03	Enable latch on digital input [▶ 122]	Activates saving of the position value via the digital input	BOOLEAN	RO	0x00 (0_{dec})
70n0:11	Set position value [▶ 120]	Specification of the position value	DINT	RO	0x00000000 (0_{dec})

8.7 Information / diagnostic data (channel specific)

Index 90n0 IND Info data Ch. n (for n=0 [channel 1], n=1 [channel 2])

Index (hex)	Name	Meaning	Data type	Flags	Default
90n0:0	IND Info data Ch.n	Maximum subindex	UINT8	RO	0x19 (25_{dec})
90n0:11	SetPositionOffset [▶ 121]	The offset value is stored here if a position was specified via 0x70n0:01 [▶ 142] "Set position" or 0x70n0:02 [▶ 142] "Set position on digital input"		RO	0x00 (0_{dec})
90n0:19	Excitation current	Excitation current		RO	0x00 (0_{dec})

Index F9000 Device Info data

Index (hex)	Name	Meaning	Data type	Flags	Default
F9000:0	Device Info data	Maximum subindex	UINT8	RO	0x19 (25_{dec})
F9000:19	Excitation current (sum)	Sum of the excitation current for both channels	REAL32	RO	0.000000 (0.000000e+000)

8.8 Diagnosis History data

Index 10F3 Diagnosis History

Index (hex)	Name	Meaning	Data type	Flags	Default
10F3:0	Diagnosis History	Maximum subindex	UINT8	RO	0x15 (21_{dec})
10F3:01	Maximum Messages	Maximum number of stored messages A maximum of 16 messages can be stored.	UINT8	RO	0x00 (0_{dec})
10F3:02	Newest Messages	Subindex of the latest message	UINT8	RO	0x00 (0_{dec})
10F3:03	Newest Acknowledged Messages	Subindex of the last confirmed message	UINT8	RW	0x00 (0_{dec})
10F3:04	New Messages Available	Indicates that a new message is available	BOOLEAN	RO	0x00 (0_{dec})
10F3:05	Flags	not used	UINT16	RW	0x0000 (0_{dec})
10F3:06	Diagnosis Message 001	Message 1	OCTET-STRING[20]	RO	{0}
…	…	…	…	…	…
10F3:15	Diagnosis Message 016	Message 16	OCTET-STRING[20]	RO	{0}

8.9 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 (5001). The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	0x00001389 (5001_{dec})

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EL5072

Index 1009 Hardware version

Index (hex)	Name	Meaning	Data type	Flags	Default
1009:0	Hardware version	Hardware version of the EtherCAT slave	STRING	RO	09

Index 100A Software version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware version of the EtherCAT slave	STRING	RO	10

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Information for identifying the slave	UINT8	RO	0x04 (4_{dec})
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2_{dec})
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x13D03052 (332410962_{dec})
1018:03	Revision	Revision number of the EtherCAT slave; the Low Word (bit 0-15) indicates the special terminal number, the High Word (bit 16-31) refers to the device description	UINT32	RO	0x00000000 (0_{dec})
1018:04	Serial number	Serial number of the EtherCAT slave; the Low Byte (bit 0-7) of the Low Word contains the year of production, the High Byte (bit 8-15) of the Low Word contains the week of production, the High Word (bit 16-31) is 0	UINT32	RO	0x00000000 (0_{dec})

Index 10F0 Backup parameter handling

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

Index 10F8 Actual Time Stamp

Index (hex)	Name	Meaning	Data type	Flags	Default
10F8	Actual Time Stamp	Current time stamp in ns	ULINT64	RO	

Index 1600 IND RxPDO-Map Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1600:0	IND RxPDO Map Outputs Ch.1	PDO Mapping RxPDO 1	UINT8	RO	0x05 (5_{dec})
1600:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (IND Outputs Ch.1), entry 0x01 (Set position))	UINT32	RO	0x7000:01, 1
1600:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (IND Outputs Ch.1), entry 0x02 (Set position on digital input))	UINT32	RO	0x7000:02, 1
1600:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (IND Outputs Ch.1), entry 0x03 (Enable latch on digital input))	UINT32	RO	0x7000:03, 1
1600:04	SubIndex 004	4. PDO Mapping entry (29 bits align)	UINT32	RO	0x0000:00, 29
1600:05	SubIndex 005	5. PDO Mapping entry (object 0x7000 (IND Outputs Ch.1), entry 0x11 (Set position value))	UINT32	RO	0x7000:11, 32

Index 1601 IND RxPDO-Map Outputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1601:0	IND RxPDO Map Outputs Ch.2	PDO Mapping RxPDO 2	UINT8	RO	0x05 (5_{dec})
1601:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (IND Outputs Ch.2), entry 0x01 (Set position))	UINT32	RO	0x7010:01, 1
1601:02	SubIndex 002	2. PDO Mapping entry (object 0x7010 (IND Outputs Ch.2), entry 0x02 (Set position on digital input))	UINT32	RO	0x7010:02, 1
1601:03	SubIndex 003	3. PDO Mapping entry (object 0x7010 (IND Outputs Ch.2), entry 0x03 (Enable latch on digital input))	UINT32	RO	0x7010:03, 1
1601:04	SubIndex 004	4. PDO Mapping entry (29 bits align)	UINT32	RO	0x0000:00, 29
1601:05	SubIndex 005	5. PDO Mapping entry (object 0x7010 (IND Outputs Ch.2), entry 0x11 (Set position value))	UINT32	RO	0x7010:11, 32

Index 1A00 IND TxPDO-Map Status Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00:0	IND TxPDO-Map Status Ch.1	PDO Mapping TxPDO 1	UINT8	RO	0x0F (15_{dec})
1A00:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (IND Boolean Inputs Ch.1), entry 0x01 (Underrange))	UINT32	RO	0x6000:01, 1
1A00:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (IND Boolean Inputs Ch.1), entry 0x02 (Overrange))	UINT32	RO	0x6000:02, 1
1A00:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (IND Boolean Inputs Ch.1), entry 0x03 (Excitation error))	UINT32	RO	0x6000:03, 1
1A00:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (IND Boolean Inputs Ch.1), entry 0x04 (Input error))	UINT32	RO	0x6000:04, 1
1A00:05	SubIndex 005	5. Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1A00:06	SubIndex 006	6. PDO Mapping entry (object 0x6000 (IND Boolean Inputs Ch.1), entry 0x08 (Digital input))	UINT32	RO	0x6000:08, 1
1A00:07	SubIndex 007	7. Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A00:08	SubIndex 008	8. PDO Mapping entry (object 0x6000 (IND Boolean Inputs Ch.1), entry 0x0A (Latch extern valid))	UINT32	RO	0x6000:0A, 1
1A00:09	SubIndex 009	9. PDO Mapping entry (object 0x6000 (IND Boolean Inputs Ch.1), entry 0x0B (Set position done))	UINT32	RO	0x6000:0B, 1
1A00:0A	SubIndex 010	10. Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A00:0B	SubIndex 011	11. PDO Mapping entry (object 0x6000 (IND Boolean Inputs Ch.1), entry 0x0D (Diag))	UINT32	RO	0x6000:0D, 1
1A00:0C	SubIndex 012	12. PDO Mapping entry (object 0x6000 (ENC Inputs Ch.1), entry 0x0E (TxPDO State))	UINT32	RO	0x6000:0E, 1
1A00:0D	SubIndex 013	13. PDO Mapping entry (object 0x6000 (IND Boolean Inputs Ch.1), entry 0x0F (Input cycle counter))	UINT32	RO	0x6000:0F, 2
1A00:0E	SubIndex 014	14. PDO Mapping entry (object 0x6001 (IND Inputs Ch.1), entry 0x01 (Position))	UINT32	RO	0x6001:01, 32
1A00:0F	SubIndex 015	15. PDO Mapping entry (object 0x6001 (IND Inputs Ch.1), entry 0x0F (Latch value))	UINT32	RO	0x6001:02, 32

Index 1A01 IND TxPDO-Map Latch Timestamp Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A01:0	IND TxPDO-Map Latch Timestamp Ch.1	PDO Mapping TxPDO 2	UINT8	RO	0x01 (1_{dec})
1A01:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (IND Latch Timestamp Ch.1), entry 0x01 (Latch event Timestamp))	UINT32	RO	0x6002:01, 64

Index 1A02 IND TxPDO-Map Inputs Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A02:0	IND TxPDO-Map Status Ch.2	PDO Mapping TxPDO 3	UINT8	RO	0x0F (15_{dec})
1A02:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (IND Boolean Inputs Ch.2), entry 0x01 (Underrange))	UINT32	RO	0x6010:01, 1
1A02:02	SubIndex 002	2. PDO Mapping entry (object 0x6010 (IND Boolean Inputs Ch.2), entry 0x02 (Overrange))	UINT32	RO	0x6010:02, 1
1A02:03	SubIndex 003	3. PDO Mapping entry (object 0x6010 (IND Boolean Inputs Ch.2), entry 0x03 (Excitation error))	UINT32	RO	0x6010:03, 1
1A02:04	SubIndex 004	4. PDO Mapping entry (object 0x6010 (IND Boolean Inputs Ch.2), entry 0x04 (Input error))	UINT32	RO	0x6010:04, 1
1A02:05	SubIndex 005	5. Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1A02:06	SubIndex 006	6. PDO Mapping entry (object 0x6010 (IND Boolean Inputs Ch.2), entry 0x08 (Digital input))	UINT32	RO	0x6010:08, 1
1A02:07	SubIndex 007	7. Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A02:08	SubIndex 008	8. PDO Mapping entry (object 0x6010 (IND Boolean Inputs Ch.2), entry 0x0A (Latch extern valid))	UINT32	RO	0x6010:0A, 1
1A02:09	SubIndex 009	9. PDO Mapping entry (object 0x6010 (IND Boolean Inputs Ch.2), entry 0x0B (Set position done))	UINT32	RO	0x6010:0B, 1
1A02:0A	SubIndex 010	10. Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A02:0B	SubIndex 011	11. PDO Mapping entry (object 0x6010 (IND Boolean Inputs Ch.2), 0x0D (Diag))	UINT32	RO	0x6010:0D, 1
1A02:0C	SubIndex 012	12. PDO Mapping entry (object 0x6010 (IND Boolean Inputs Ch.2), entry 0x0E (TxPDO State))	UINT32	RO	0x6010:0E, 1
1A02:0D	SubIndex 013	13. PDO Mapping entry (object 0x6010 (IND Boolean Inputs Ch.2), entry 0x0F (Input cycle counter))	UINT32	RO	0x6010:0F, 2
1A02:0E	SubIndex 014	14. PDO Mapping entry (object 0x6011 (IND Inputs Ch.2), entry 0x01 (Position))	UINT32	RO	0x6011:01, 32
1A02:0F	SubIndex 015	15. PDO Mapping entry (object 0x6011 (IND Inputs Ch.2), entry 0x0F (Latch value))	UINT32	RO	0x6011:02, 32

Index 1A03 IND TxPDO-Map Latch Timestamp Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A03:0	IND TxPDO-Map Latch Timestamp Ch.2	PDO Mapping TxPDO 4	UINT8	RO	0x01 (1_{dec})
1A03:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (IND Latch Timestamp Ch.2), entry 0x01 (Latch event Timestamp))	UINT32	RO	0x6012:01, 64

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

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

Index 1C13 TxPDO assign

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

Index 1C32 SM output parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20 (32_{dec})
1C32:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 2 Event • 2: DC-Mode - Synchron with SYNC0 Event • 3: DC-Mode - Synchron with SYNC1 Event	UINT16	RW	0x0001 (1_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchron with SM 2 Event: Master cycle time • DC mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x000F4240 (1000000_{dec})
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0 = 1: free run is supported • Bit 1 = 1: Synchron with SM 2 Event is supported • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 10: Output Shift with SYNC1 event (only DC mode) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08)	UINT16	RO	0x0003 (3_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x000186A0 (100000_{dec})
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:07	Minimum delay time	Min. time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:08	Get Cycle Time	With this entry the real required process data provision time can be measured. • 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 0x1C32:03, 0x1C32:05, 0x1C32:06, 0x1C32:09 [▶ 146], 0x1C33.03 [▶ 147], 0x1C33.06 [▶ 147], 0x1C33.09 [▶ 147] are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RW	0x0000 (0_{dec})
1C32:09	Maximum delay time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0_{dec})
1C32:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:14	Frame repeat time		UDINT32	RW	0x00000000 (0_{dec})
1C32:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	0x00 (0_{dec})

Index 1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32_{dec})
1C33:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 3 event (no outputs available) • 2: DC - Synchron with SYNC0 Event • 3: DC - Synchron with SYNC1 Event • 34: Synchron with SM 2 event (outputs available)	UINT16	RW	0x0022 (34_{dec})
1C33:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchron with SM 2 Event: Master cycle time DC mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x000F4240 (1000000_{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: free run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron with SM 3 Event is supported (no outputs available) • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 01: Input Shift through local event (outputs available) • Bit 4-5 = 10: Input Shift with SYNC1 event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08 [▶ 146] or 0x1C33:08 [▶ 147])	UINT16	RO	0x0003 (3_{dec})
1C33:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x000186A0 (100000_{dec})
1C33:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:07	Minimum delay time	Min. time between SYNC1 event and the reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C33:08	Get Cycle time	With this entry the real required process data provision time can be measured. • 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 0x1C32:03 [▶ 146], 0x1C32:05 [▶ 146], 0x1C32:06 [▶ 146], 0x1C32:09 [▶ 146], 0x1C33:03, 0x1C33:06, 0x1C33:09 are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RW	0x0000 (0_{dec})
1C33:09	Maximum delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C33:14	Frame repeat time	/	UDINT32	RW	0x00000000 (0_{dec})
1C33:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	0x00 (0_{dec})

Index F000 Modular device profile

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

Index F008 Code word

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

Index F009 password protection

Index (hex)	Name	Meaning	Data type	Flags	Default
F009:0	Password protection [▶ 117]	Password protection user calibration	UINT32	RW	0x00000000 (0_{dec})

9 Appendix

9.1 Firmware compatibility

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

Note

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

NOTE

Risk of damage to the device!

Pay attention to the instructions for firmware updates on the Firmware Update EL/ES/EM/ELM/EPxxxx [▶ 149]. 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!

EL5072			
Hardware (HW)	**Firmware**	**Revision no.**	**Release date**
04*	01	EL5072-0000-0016	2021/01
	02	EL5072-0000-0017	2021/10
	03*	EL5072-0000-0018	2022/03

*) 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.

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

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

NOTE

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 https://www.beckhoff.com/en-us/.

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:

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

- In some EtherCAT slaves the EtherCAT communication may also be integrated in these controllers. In this case the controller is usually a so-called **FPGA** chip with *.rbf firmware.
- In addition, each EtherCAT slave has a memory chip, a so-called **ESI-EEPROM**, for storing its own device description (ESI: EtherCAT Slave Information). On power-up this description is loaded and the EtherCAT communication is set up accordingly. The device description is available from the download area of the Beckhoff website at (https://www.beckhoff.com). All ESI files are accessible there as zip files.

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

The TwinCAT System Manager offers mechanisms for programming all 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

NOTE

Risk of damage to the device!

✓ Note the following when downloading new device files

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

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

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

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

9.2.1 Device description ESI file/XML

NOTE

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:

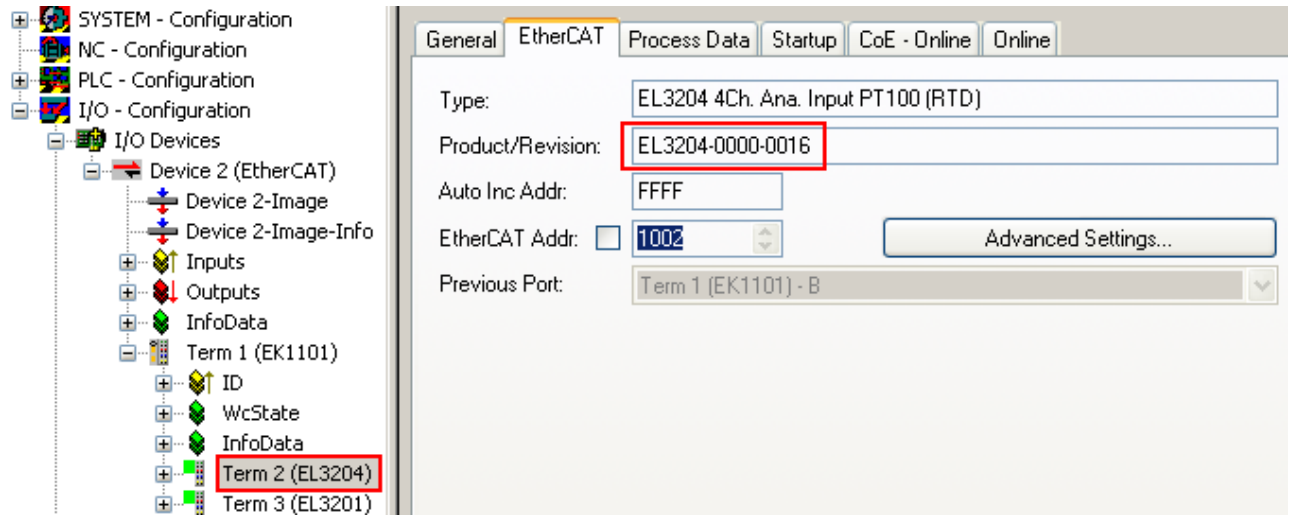


Fig. 113: Device identifier consisting of name EL3204-0000 and revision -0016

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

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

Update of XML/ESI description

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:

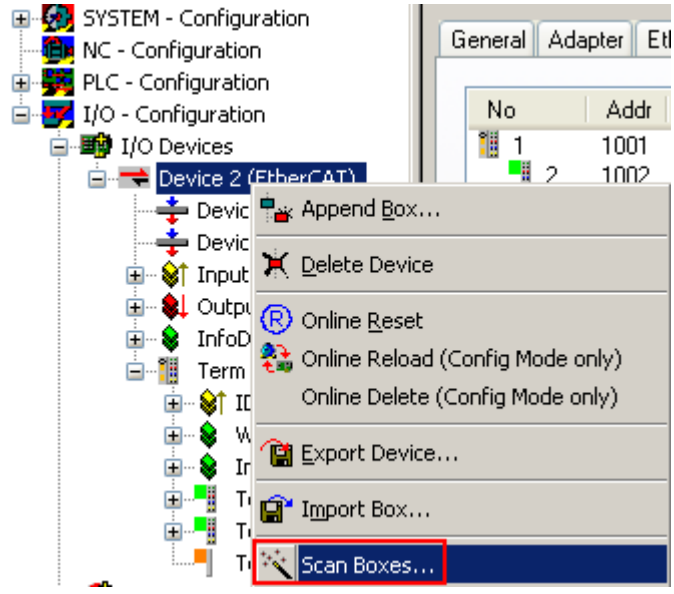


Fig. 114: Scan the subordinate field by right-clicking on the EtherCAT device

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

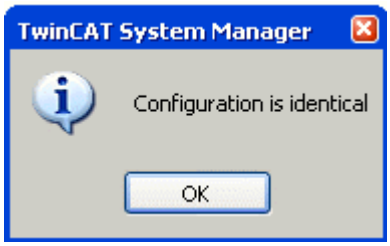


Fig. 115: Configuration is identical

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

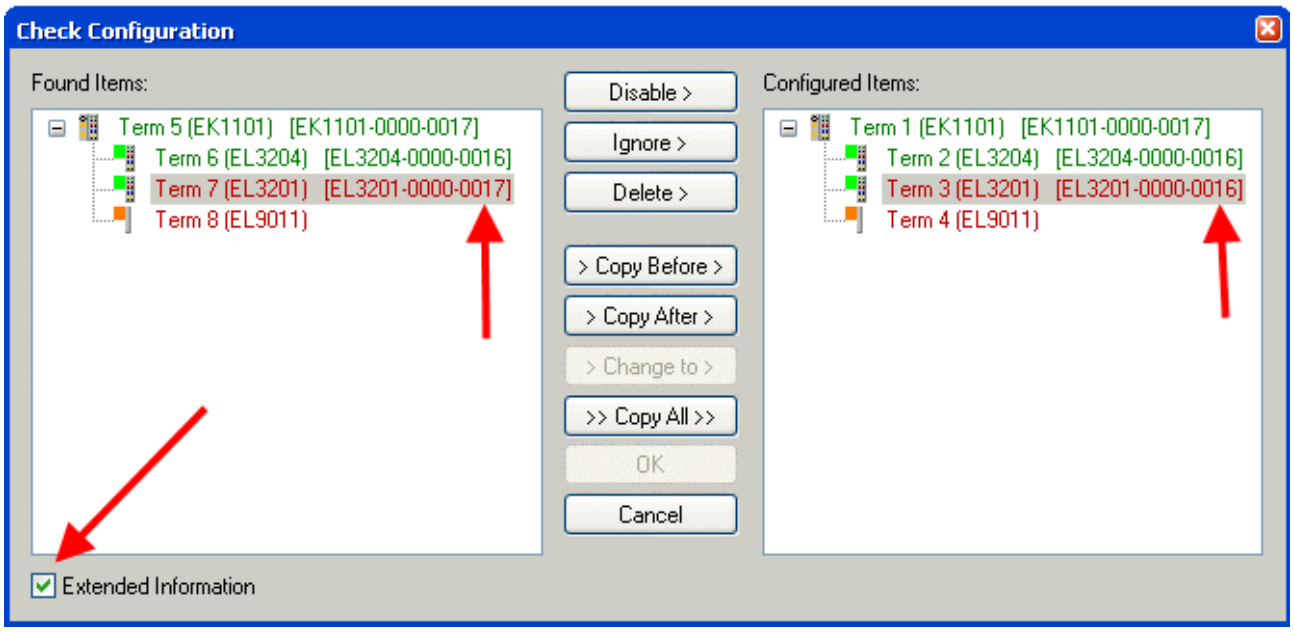


Fig. 116: Change dialog

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*

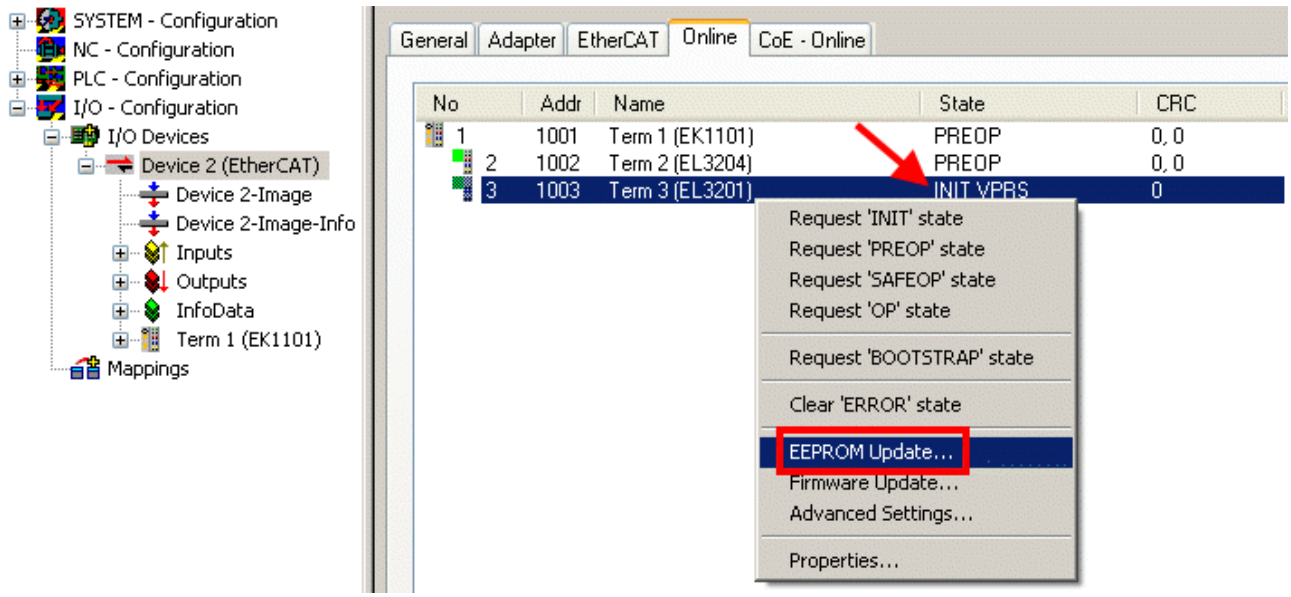


Fig. 117: EEPROM Update

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.

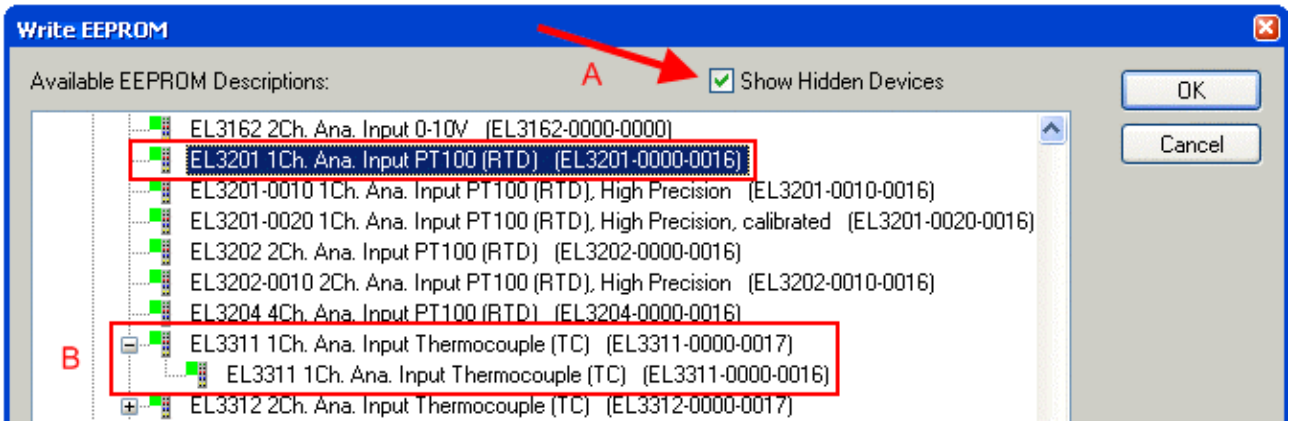


Fig. 118: Selecting the new ESI

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

The change only takes effect after a restart.

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

9.2.2 Firmware explanation

Determining the firmware version

Determining the version via the System Manager

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

CoE Online and Offline CoE

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

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

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

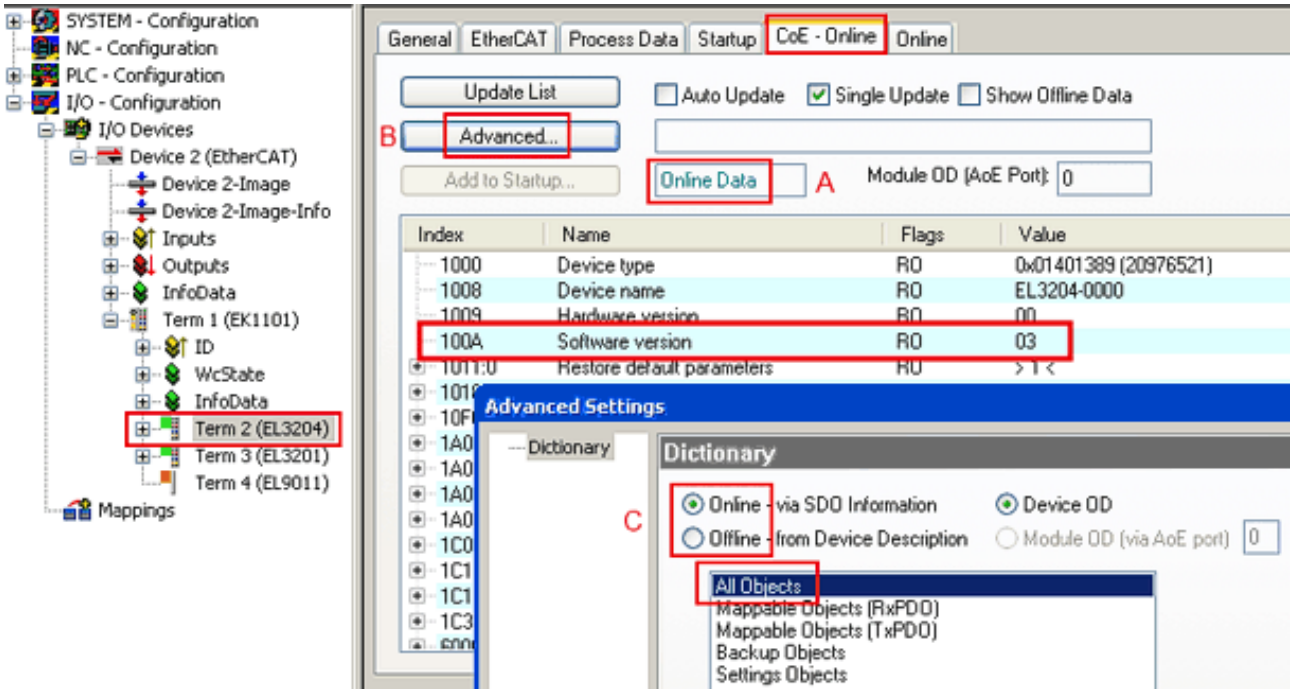


Fig. 119: Display of EL3204 firmware version

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*.

9.2.3 Updating controller firmware *.efw

CoE directory

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

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

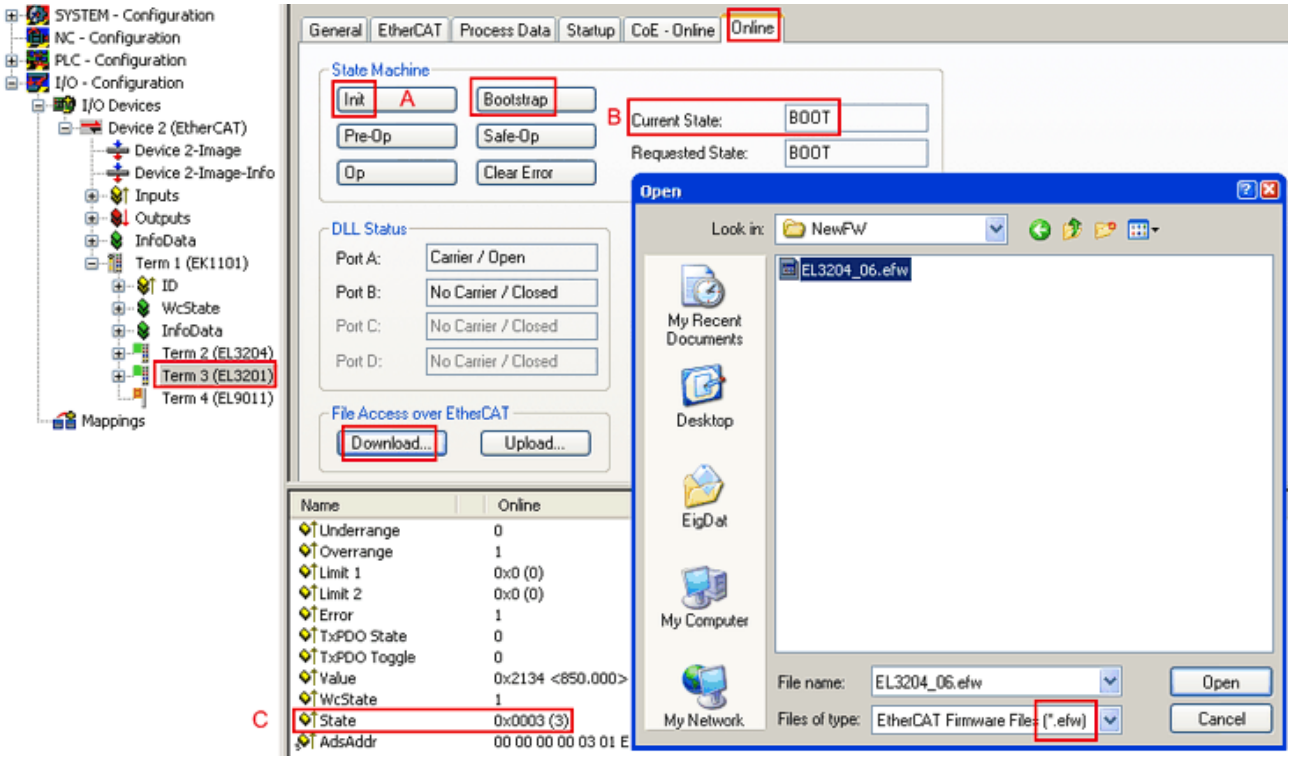
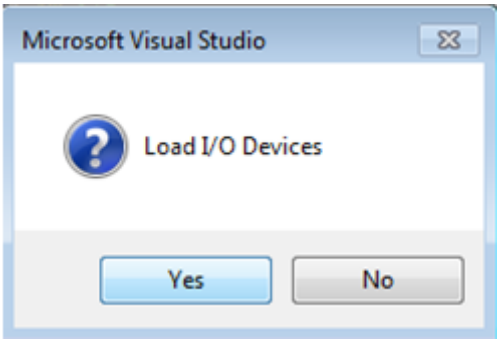
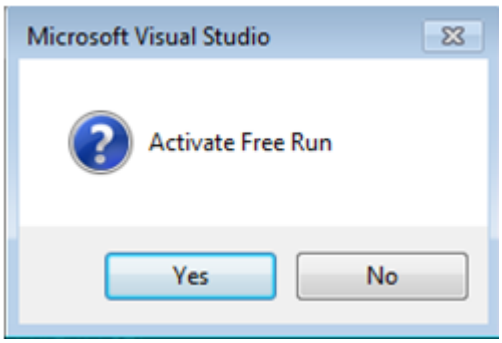


Fig. 120: Firmware Update

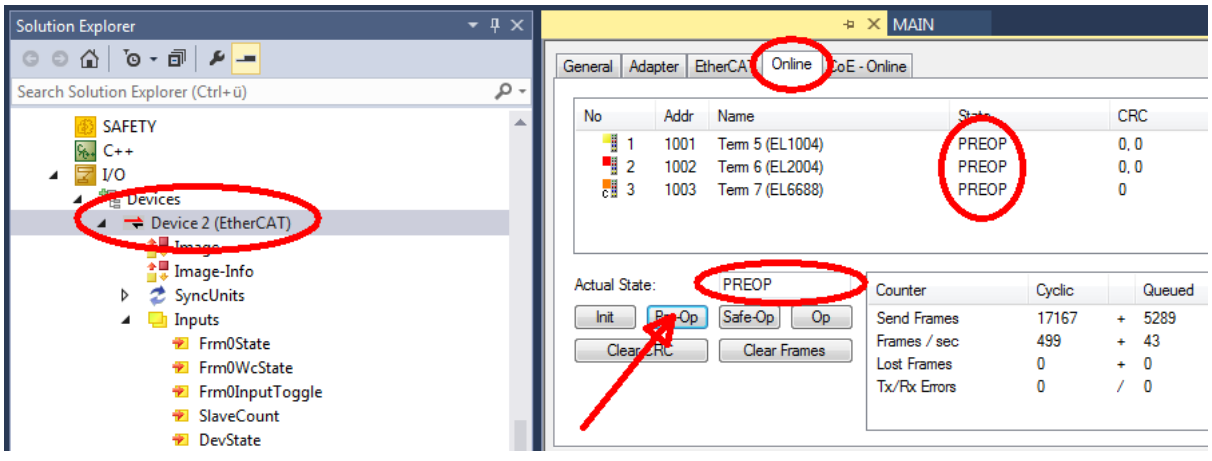
Proceed as follows, unless instructed otherwise by Beckhoff support. Valid for TwinCAT 2 and 3 as EtherCAT master.

- 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.



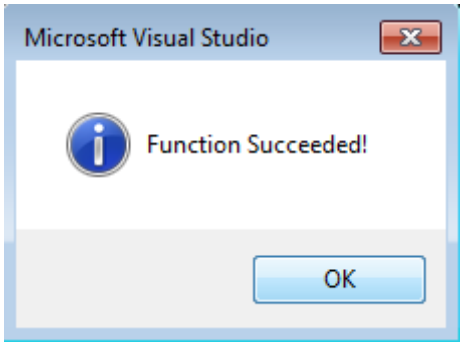


- Switch EtherCAT Master to PreOP



- 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.

9.2.4 FPGA firmware *.rbf

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

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

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

Determining the version via the System Manager

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

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

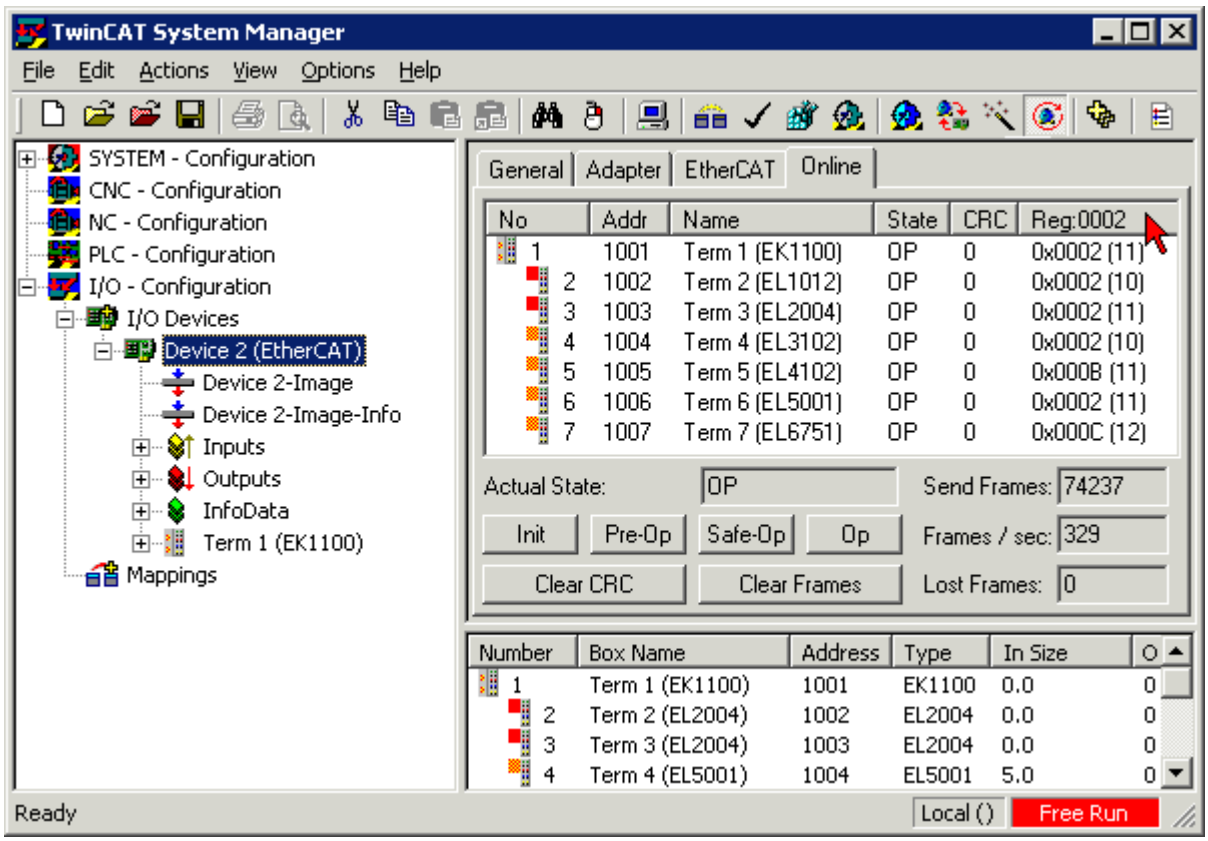


Fig. 121: FPGA firmware version definition

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

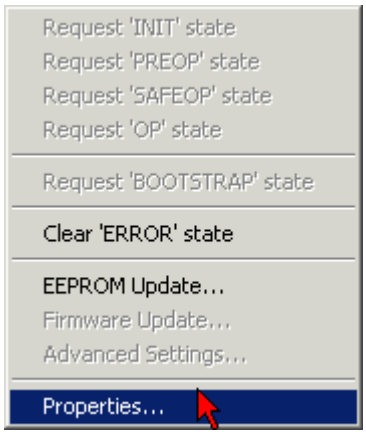


Fig. 122: Context menu *Properties*

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.

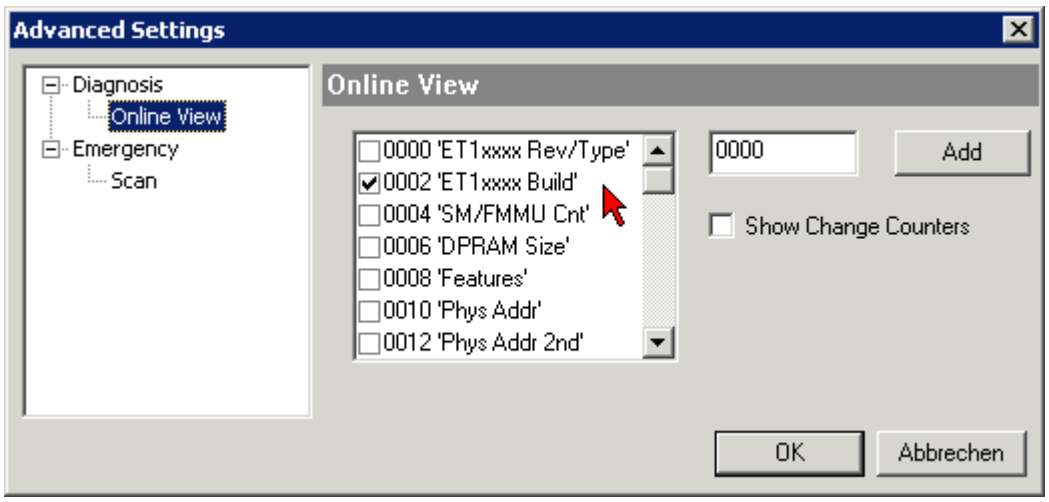


Fig. 123: Dialog *Advanced Settings*

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.

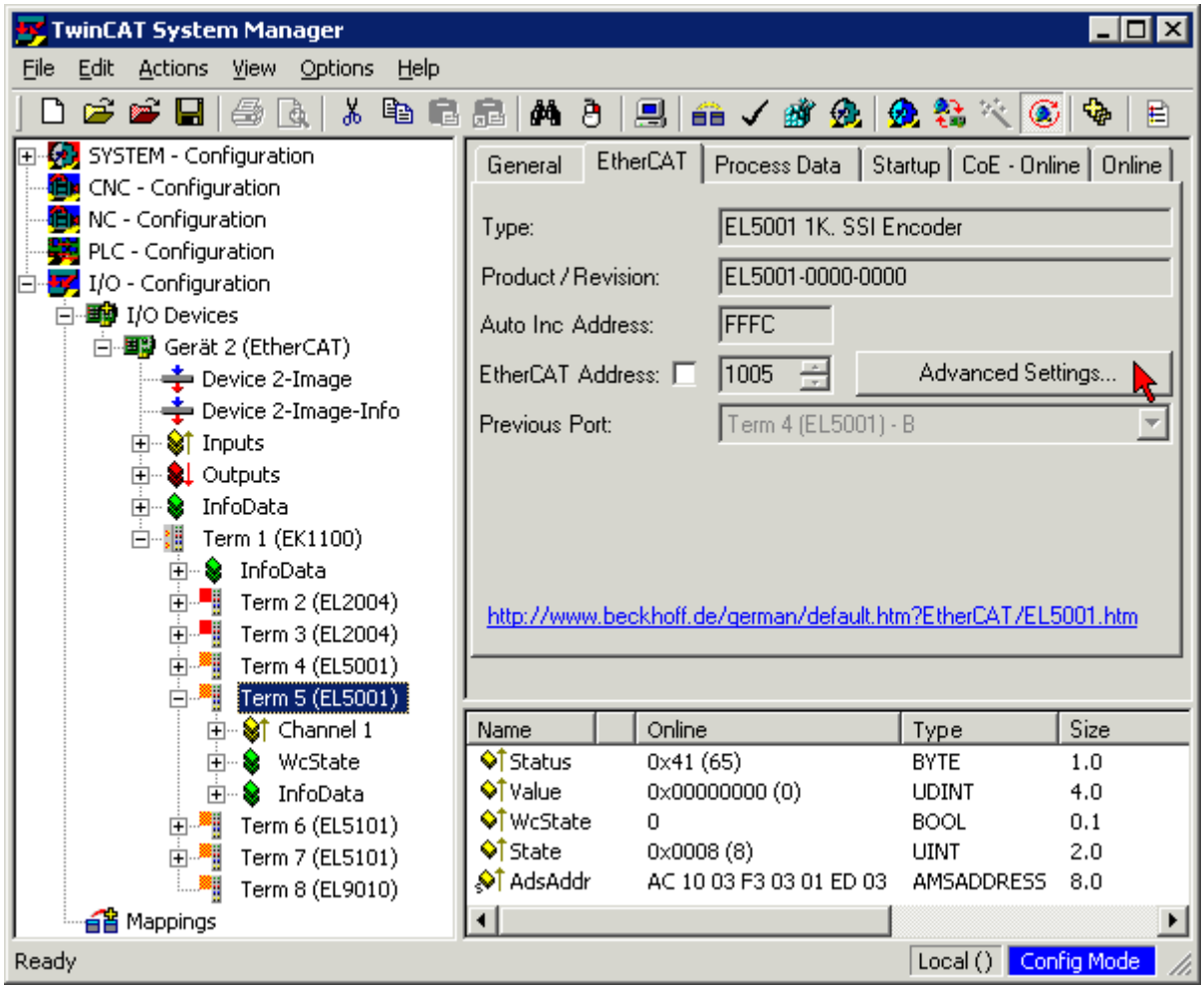
Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

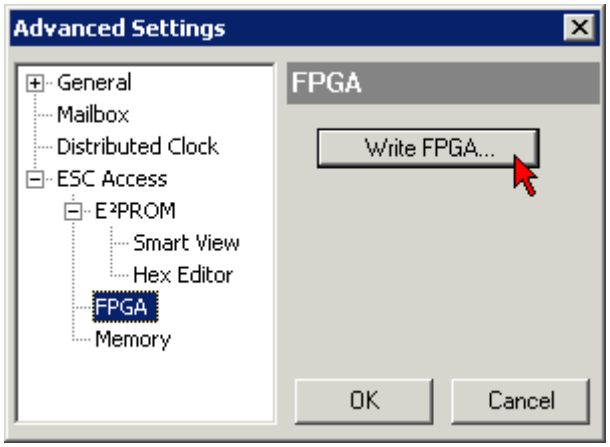
The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

- 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.

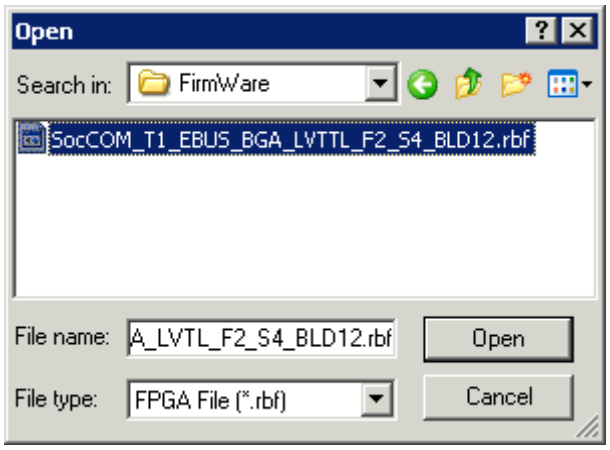
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and
  click the *Advanced Settings* button in the *EtherCAT* tab:



- 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

NOTE

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!

9.2.5 Simultaneous updating of several EtherCAT devices

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

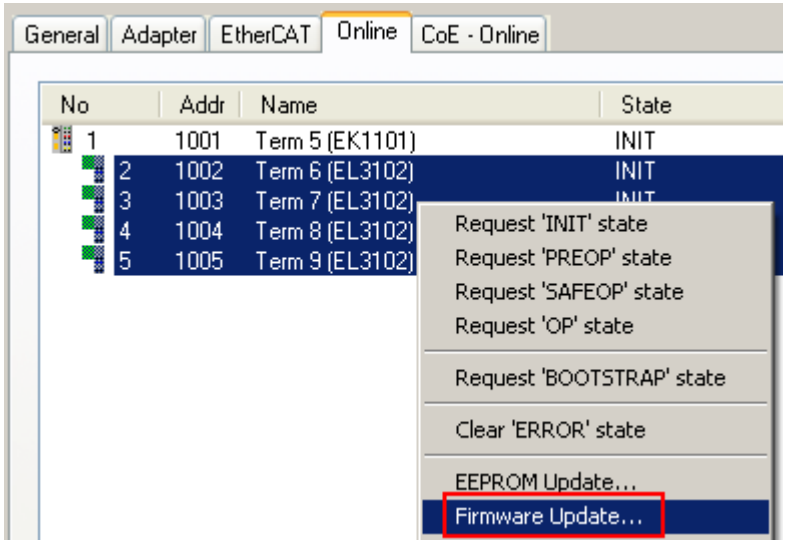


Fig. 124: Multiple selection and firmware update

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

9.3 Restoring the delivery state

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

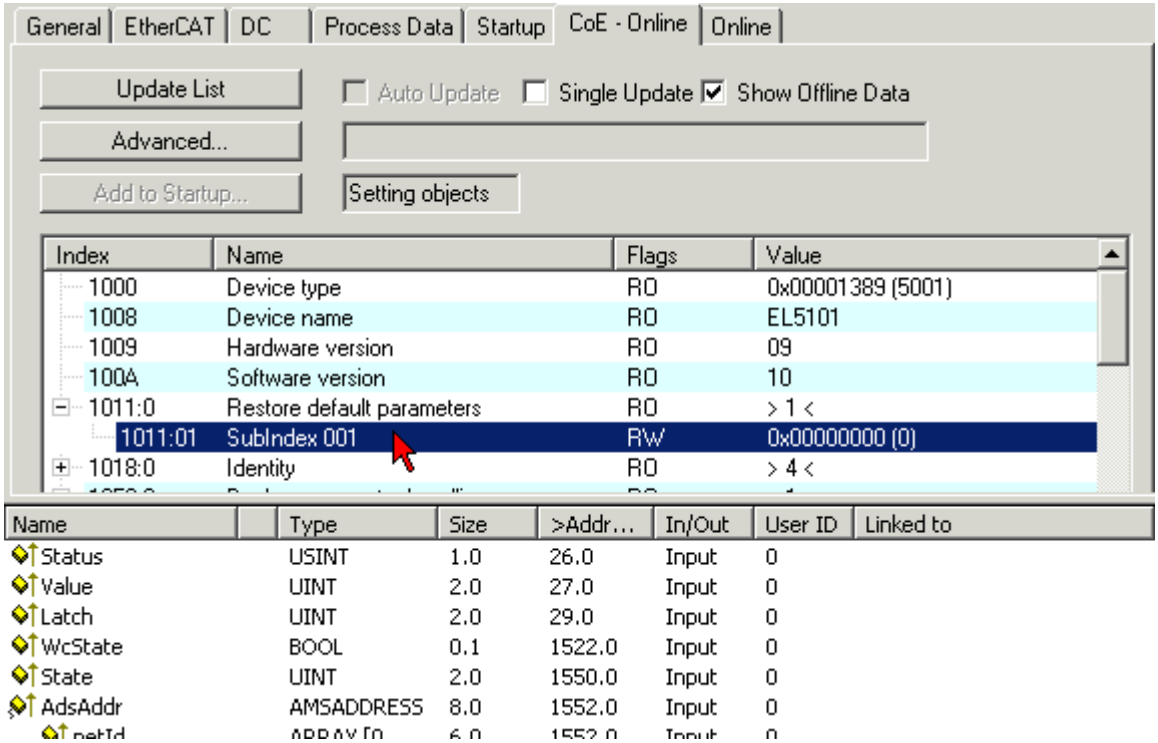


Fig. 125: Selecting the *Restore default parameters* PDO

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

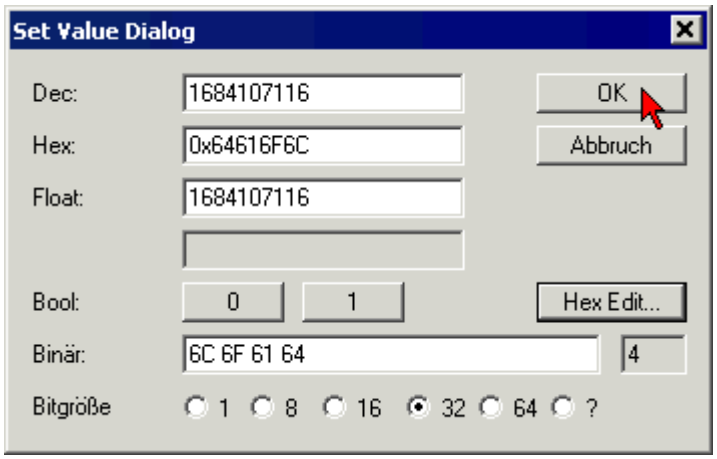


Fig. 126: Entering a restore value in the Set Value dialog

Alternative restore value

In some older terminals the backup objects can be switched with an alternative restore value: Decimal value: 1819238756, Hexadecimal value: 0x6C6F6164An incorrect entry for the restore value has no effect.

9.4 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: https://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

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

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

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

Beckhoff Service

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

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

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

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963 0
Fax:	+49 5246 963 198
e-mail:	info@beckhoff.com
web:	https://www.beckhoff.com

More Information:
beckhoff.de/EL5072/

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