BECKHOFF New Automation Technology

Operating manual | EN

ELX5151

One channel incremental encoder interface, NAMUR, 32 bit, Ex i

2021-10-13 | Version: 1.6.0

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
1.6.0	• Technical data extended (cFMus added) • FM notice regarding ANSI/ISA EX added • Chapter *Marking of ELX terminals* updated • Chapter *Basic function principles* added • Chapter *Parameterization and programming* added • Chapter *Disposal* added • Design of the safety instructions adapted to IEC 82079-1 • New title page
1.5.0	• Contact assignment extended with sensor illustration • Chapter *Arrangement of ELX terminals within a bus terminal block* updated • Chapter *Marking of ELX terminals* updated • Technical data updated
1.4.0	• Chapter *Arrangement of ELX terminals at the bus terminal block* updated
1.3.0	• Chapter *Installation notes for ELX terminals* updated
1.2.0	• Chapter *Marking of ELX terminals* updated • Technical data updated
1.1.0	• Chapter *Marking of ELX terminals* updated
1.0.0	• Technical data updated
0.2	• Technical data updated • LED display updated
0.1	• First preliminary version

1.4 Marking of ELX terminals

Name

An ELX terminal has a 15-digit technical designation, composed of

- family key
- type
- software variant
- revision

example	family	type	software variant	revision
ELX1052-0000-0001	ELX terminal	1052: two-channel digital input terminal for NAMUR sensors, Ex i	0000: basic type	0001
ELX9560-0000-0001	ELX terminal	9560: power supply terminal	0000: basic type	0001

Notes

- The elements mentioned above result in the **technical designation**. ELX1052-0000-0001 is used in the example below.
- Of these, ELX1052-0000 is the order identifier, commonly called just ELX1052 in the "-0000" revision. “-0001” is the EtherCAT revision.
- The **order identifier** is made up of
  - family key (ELX)
  - type (1052)
  - software version (-0000)
- The **Revision** -0001 shows the technical progress, such as the extension of features with regard to the EtherCAT communication, and is managed by Beckhoff.
  In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation.
  Associated and synonymous with each revision there is usually a description (ESI, EtherCAT Slave Information) in the form of an XML file, which is available for download from the Beckhoff website.
  The revision has been applied to the terminals on the outside, see *ELX1052 with date code 3218FMFM, BTN 10000100 and Ex marking.*
- The hyphen is omitted in the labeling on the side of the terminal. Example:
  Name: ELX1052-0000
  Label: ELX1052 $_{0000}$
- The type, software version and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

Identification numbers

ELX terminals have two different identification numbers:

- date code (batch number)
- **B**eckhoff **T**raceability **N**umber, or BTN for short (as a serial number it clearly identifies each terminal)

Date code

The date code is an eight-digit number given by Beckhoff and printed on the ELX terminal. The date code indicates the build version in the delivery state and thus identifies an entire production batch, but does not distinguish between the terminals in a batch.

Structure of the date code: **WW YY FF HH**	Example with date code: 02180100:
WW - week of production (calendar week)	02 - week of production 02
YY - year of production	18 - year of production 2018
FF - firmware version	01 - firmware version 01
HH - hardware version	00 - hardware version 00

Beckhoff Traceability Number (BTN)

In addition, each ELX terminal has a unique **B**eckhoff **T**raceability **N**umber (BTN).

Ex marking

The Ex marking can be found at the top left on the terminal:

II 3 (1) G Ex ec [ia Ga] IIC T4 Gc
II (1) D [Ex ia Da] IIIC
I (M1) [Ex ia Ma] I
IECEx BVS 18.0005X
BVS 18 ATEX E 005 X

Examples

Fig. 1: ELX2008-0000 with date code 2519HMHM, BTN 0001f6hd and Ex marking

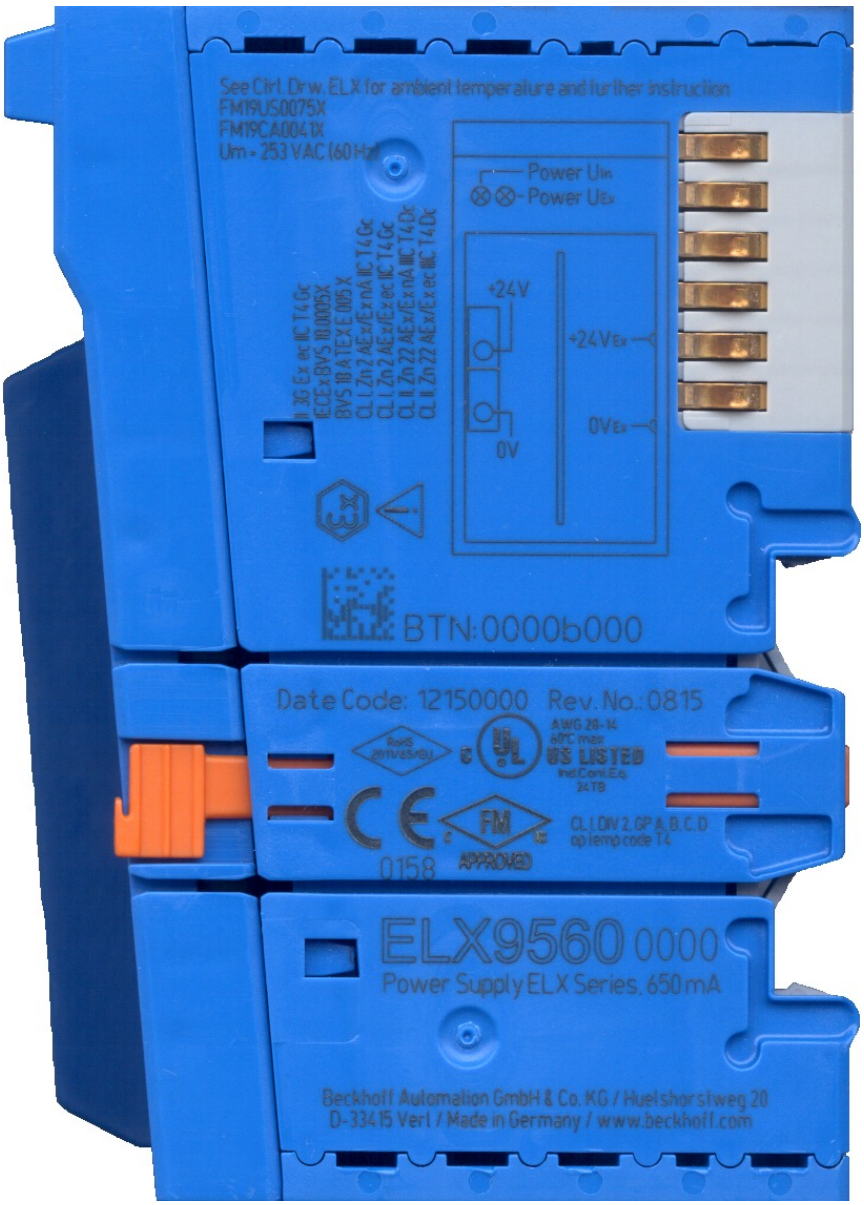


Fig. 2: ELX9560-0000 with date code 12150000, BTN 000b000 and Ex marking

Fig. 3: ELX9012 with date code 12174444, BTN 0000b0si and Ex marking

2 Product overview

2.1 ELX5151 - Introduction

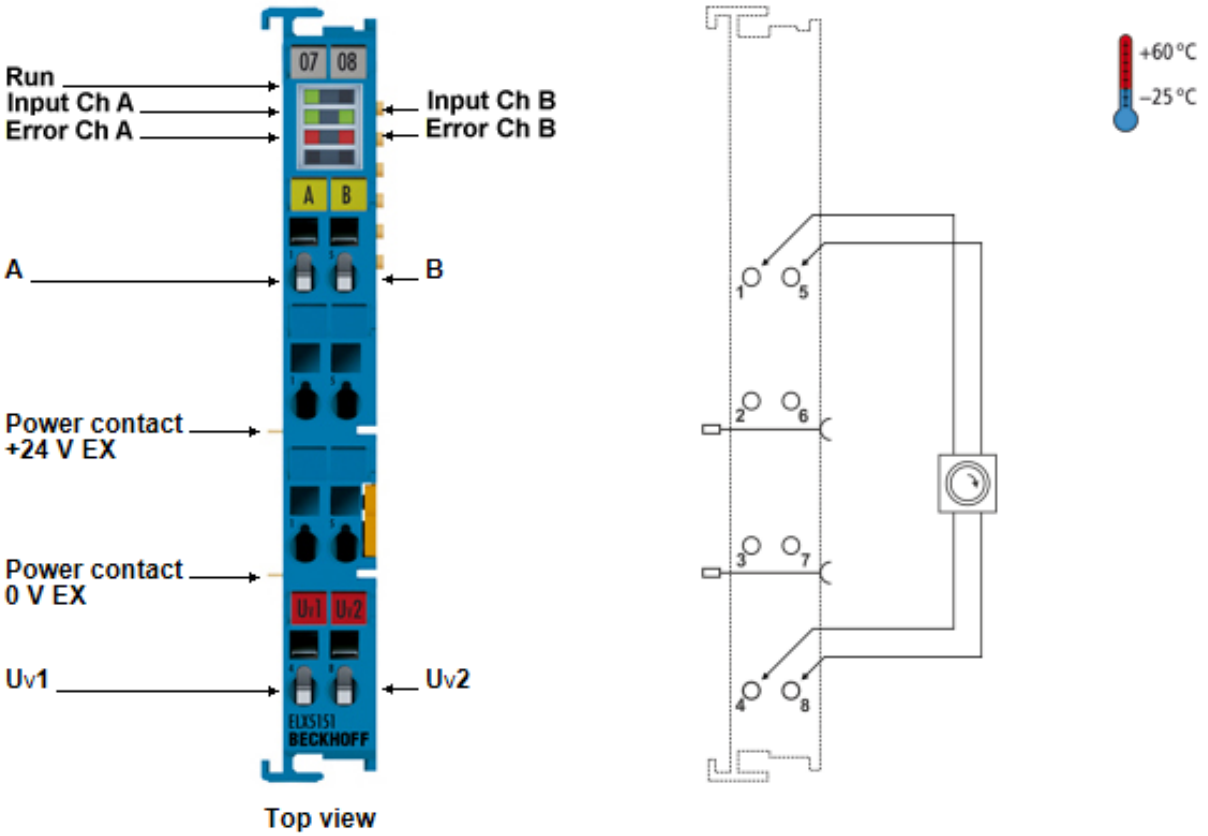


Fig. 4: ELX5151 - 1-channel incremental encoder interface, NAMUR, 32 bit, Ex i

The ELX5151 EtherCAT Terminal allows the direct connection of an intrinsically safe incremental encoder from Zone 0/20 and 1/21 hazardous areas. The terminal supplies the encoder with 8.2 V_{DC} and evaluates a diagnostic-capable NAMUR signal according to IEC 60947-5-6. In this way, a wire breakage or short circuit can be detected in addition to the switching state. The ELX5151 can also be used as an up/down counter and allows switching between 16 and 32 bit counting.

2.2 Technical data

Technical data	ELX5151-0000
Technology	NAMUR
Specification	NAMUR DC switching amplifier (IEC 60947-5-6)
Number of channels	1
Sensor inputs	1
Encoder connection	A, B
Encoder operating voltage	8.2 V_{DC}
Open circuit voltage	typical 8.2 V_{DC}
„0“ signal current	≤ 1.2 mA
„1“ signal current	≥ 2.1 mA
Switching hysteresis	0.2 mA
Short circuit current	typical 9 mA
Fault detection	I ≤ 200 µA (cable break), I ≥ 6,0 mA (short circuit)
Counter	1 x 16 or 32 bit, selectable
Input frequency	50 kHz
Supply voltage for electronics	via E-Bus (5 V_{DC}) and Power Contacts (24 V_{DC} Ex, feeding by ELX9560)
Current consumption from the E-Bus	typical 80 mA
Current consumption from Power Contacts	typical 15 mA + load (ELX9560 power supply)
Distributed clocks	yes
Special features	up/down counters
Configuration	no address or configuration settings required
Bit width at the process image	compact PDO: 10 byte (default) standard PDO: 16 byte
Electrical isolation	1500 V (E-Bus / field voltage)
Weight	ca. 50 g
Permissible ambient temperature range during operation	-25°C ... + 60°C
Permissible ambient temperature range during storage	-40°C ... + 85°C
Permissible relative humidity	95%, no condensation
Permissible air pressure (operation, storage, transport)	800 hPa to 1100 hPa (this corresponds to a height of approx. -690 m to 2000 m over sea level assuming an international standard atmosphere)
Dimensions (W x H x D)	app. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)
Mounting [▶ 22]	on 35 mm mounting rail conforms to EN 60715
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protect. class	IP20
Permissible installation position	See chapter Installation position and minimum distances [▶ 21]
Approvals / markings	CE, UL, ATEX, IECEx, cFMus

Technical data for explosion protection		ELX5151-0000	
Ex marking	ATEX	II 3 (1) G Ex ec [ia Ga] IIC T4 Gc II (1) D [Ex ia Da] IIIC I (M1) [Ex ia Ma] I	
	IECEx	Ex ec [ia Ga] IIC T4 Gc [Ex ia Da] IIIC [Ex ia Ma] I	
	cFMus	AIS Class I, II, III, Division 1, Groups A thru G Class I, Division 2, Groups A, B, C, D Class I, Zone 2, AEx/Ex ec [ia Ga] IIC T4 Gc [AEx/Ex ia Da] IIIC T4	
Certificate numbers		IECEx BVS 18.0005X BVS 18 ATEX E 005 X FM19US0075X, FM19CA0041X	
Power supply		Invariable in connection with ELX9560	
Field interfaces		U_O = 10.72 V I_O = 12.4 mA P_O = 33 mW Characteristic curve: linear	
Reactance (without consideration of the simultaneousness)		**L_0**	**C_0**
	Ex ia I	100 mH	58 µF
	Ex ia IIA	100 mH	66 µF
	Ex ia IIB	100 mH	15 µF
	Ex ia IIC	100 mH	2.14 µF
	Ex ia IIIC	100 mH	15 µF

2.3 Intended use

⚠ WARNING

Endangering the safety of persons and equipment!

The ELX components may only be used for the purposes described below!

⚠ CAUTION

Observe ATEX and IECEx!

The ELX components may only be used in accordance with the ATEX directive and the IECEx scheme!

The ELX terminals extend the field of application of the Beckhoff bus terminal system with functions for integrating intrinsically safe field devices from hazardous areas. The intended field of application is data acquisition and control tasks in discrete and process engineering automation, taking into account explosion protection requirements.

The ELX terminals are protected by the type of protection "Increased safety" (Ex e) according to IEC 60079-7 and must only be operated in hazardous areas of Zone 2 or in non-hazardous areas.

The field interfaces of the ELX terminals achieve explosion protection through the type of protection "intrinsic safety" (Ex i) according to IEC 60079-11. For this reason, only appropriately certified, intrinsically safe devices may be connected to the ELX terminals. Observe the maximum permissible connection values for voltages, currents and reactances. Any infringement can damage the ELX terminals and thus eliminate the explosion protection.

The ELX terminals are open, electrical equipment for installation in lockable cabinets, enclosures or operating rooms. Make sure that access to the equipment is only possible for authorized personnel.

⚠ CAUTION

Ensure traceability!

The buyer has to ensure the traceability of the device via the Beckhoff Traceability Number (BTN).

3 Mounting and wiring

3.1 Special conditions of use for ELX terminals

⚠ WARNING

Observe the special conditions of use for the intended use of Beckhoff ELX terminals in potentially explosive areas (ATEX directive 2014/34/EU)!

- The certified components are to be installed in a suitable housing that guarantees an ingress protection of at least IP54 in accordance with EN 60079-0 and EN 60529! The prescribed environmental conditions during installation, operation and maintenance are thereby to be taken into account! Inside the housing, pollution degree 1 and 2 are permissible.
- If the temperatures during rated operation are higher than 70°C at the feed-in points of cables, lines or pipes, or higher than 80°C at the wire branching points, then cables must be selected whose temperature data correspond to the actual measured temperature values!
- Observe the permissible ambient temperature range of -25 to +60°C of Beckhoff ELX terminals!
- Measures must be taken to protect against the rated operating voltage being exceeded by more than 40% due to short-term interference voltages! The power supply of the ELX9560 power supply terminal must correspond to overvoltage category II according to EN 60664-1
- The individual terminals may only be unplugged or removed from the bus terminal system if all supply voltages have been switched off or if a non-explosive atmosphere is ensured!
- The connections of the ELX9560 power supply terminal may only be connected or disconnected if all supply voltages have been switched off or if a non-explosive atmosphere is ensured!
- Address selectors and switches may only be adjusted if all supply voltages have been switched off or if a non-explosive atmosphere is ensured!

3.2 Installation notes for ELX terminals

NOTE

Storage, transport and mounting

- Transport and storage are permitted only in the original packaging!
- Store in a dry place, free from vibrations.
- A brand new ELX terminal with a certified build version is delivered only in a sealed carton. Therefore, check that the carton and all seals are intact before unpacking.
- Do not use the ELX terminal if
  - its packaging is damaged
  - the terminal is visibly damaged or
  - you cannot be sure of the origin of the terminal.
- ELX terminals with a damaged packaging seal are regarded as used.

⚠ WARNING

Observe the accident prevention regulations

During mounting, commissioning, operation and maintenance, adhere to the safety regulations, accident prevention regulations and general technical rules applicable to your devices, machines and plants.

⚠ CAUTION

Observe the erection regulations

Observe the applicable erection regulations.

NOTE

Protect the terminals against electrostatic discharge (ESD)

Electronic components can be destroyed by electrostatic discharge. Therefore, take the safety measures to protect against electrostatic discharge as described in DIN EN 61340-5-1 among others. In conjunction with this, ensure that the personnel and surroundings are suitably earthed.

NOTE

Do not place terminals on E-bus contacts

Do not place the ELX terminals on the E-bus contacts located on the right-hand side. The function of the E-bus contacts can be negatively affected by damage caused by this, e.g. scratches.

NOTE

Protect the terminals against dirt

To ensure the functionality of the ELX terminals they must be protected against dirt, especially on the contact points. For this reason use only clean tools and materials.

NOTE

Handling

- It is forbidden to insert conductive or non-conductive objects of any kind into the interior of the housing (e.g. through the ventilation slots in the housing).
- Use only the openings provided in the housing front and appropriate tools to actuate the spring-loaded terminal contacts on the front side for attaching connection cables to the terminal; see chapter Wiring [▶ 25].
- The opening of the housing, the removal of parts and any mechanical deformation or machining of an ELX terminal are not permitted!

If an ELX terminal is defective or damaged it must be replaced by an equivalent terminal. Do not carry out any repairs to the devices. For safety reasons repairs may only be carried out by the manufacturer.

NOTE

Contact marking and pin assignment

The colored inscription labels above the front connection contacts shown in the illustrations in the introduction chapter are only examples and are not part of the scope of delivery!
A clear assignment of channel and terminal designation according to the chapter contact assignment to the actual terminal point can be made via the lasered channel numbers 1 to 8 on the left above the respective terminal point as well as via the laser image.
Observe any possible polarity dependency of connected intrinsically safe circuits!

3.3 Arrangement of ELX terminals within a bus terminal block

⚠ WARNING

Observe the following instructions for the arrangement of ELX terminals!

- ELX signal terminals must always be installed behind an ELX9560 power supply terminal, without exception!
- Only signal terminals of the ELX series may be installed behind an ELX9560 power supply terminal!
- Multiple ELX9560 power supply terminals may be set in one terminal block as long as one ELX9410 is placed before each additional ELX9560!
- An ELX9410 power supply terminal must not be mounted to the right of an ELX9560 nor to the left of any ELX signal terminal!
- The last terminal of each ELX segment is to be covered by an ELX9012 bus end cover, unless two ELX9410 power supply terminals are installed in direct succession for continuing the same terminal segment with standard Beckhoff EtherCAT terminals (e.g. EL/ES/EK)!

Examples for the arrangement of ELX terminals

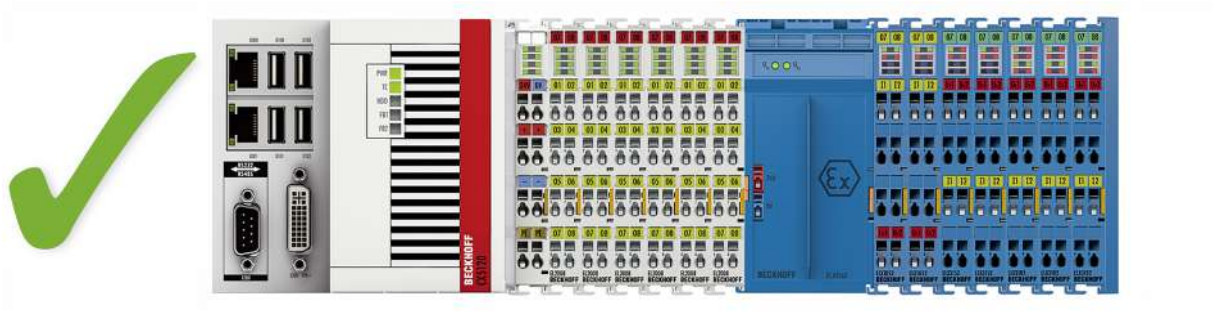


Fig. 5: Valid arrangement of the ELX terminals (right terminal block).

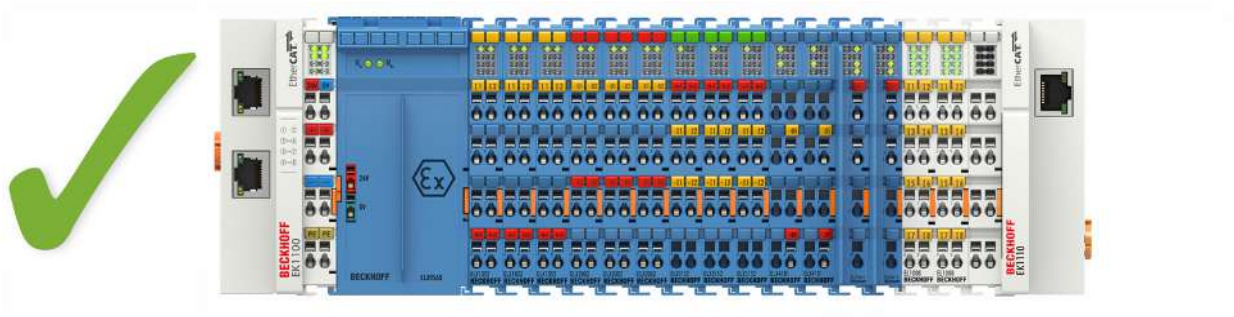


Fig. 6: Valid arrangement - terminals that do not belong to the ELX series are set before and after the ELX terminal segment. The separation is realized by the ELX9560 at the beginning of the ELX terminal segment and two ELX9410 at the end of the ELX terminal segment.

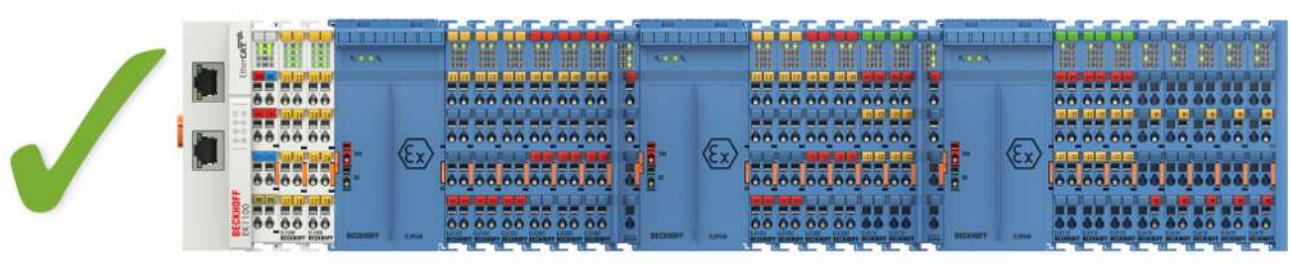


Fig. 7: Valid arrangement - multiple power supplies by ELX9560, each with an upstream ELX9410.

Fig. 8: Valid arrangement - ELX9410 in front of an ELX9560 power supply terminal.

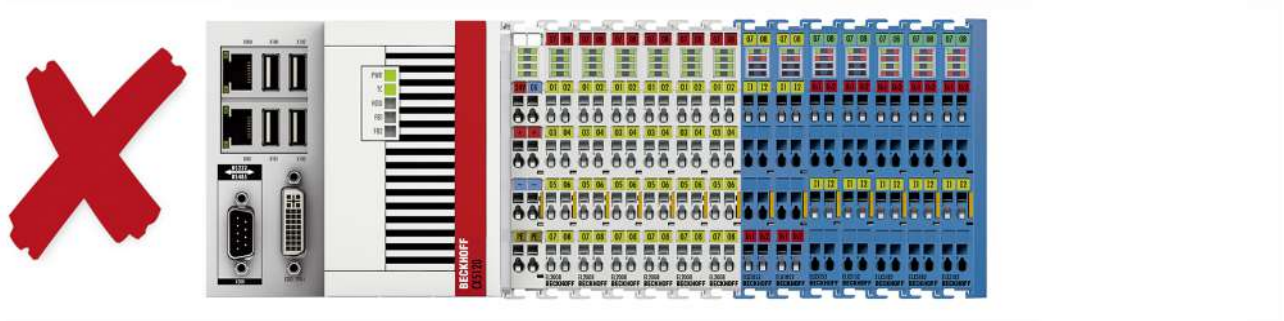


Fig. 9: Invalid arrangement - missing ELX9560 power supply terminal.

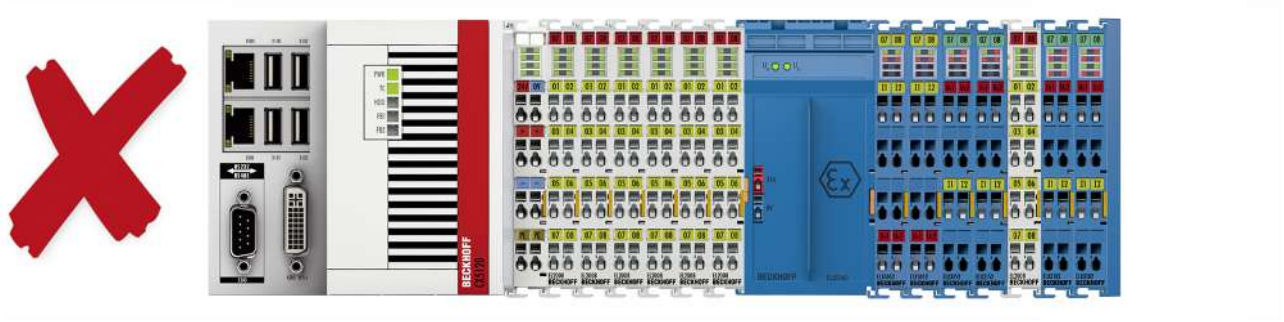


Fig. 10: Invalid arrangement - terminal that does not belong to the ELX series within the ELX terminal segment.

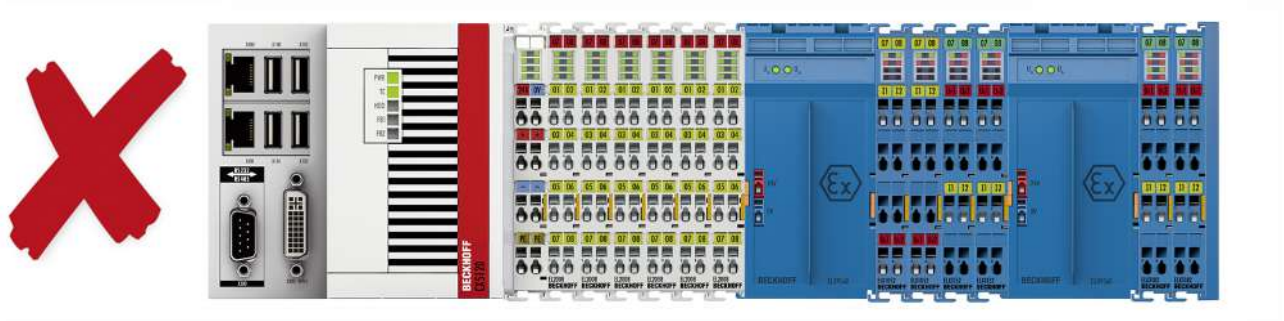


Fig. 11: Invalid arrangement - second ELX9560 power supply terminal within the ELX terminal segment without an upstream ELX9410.

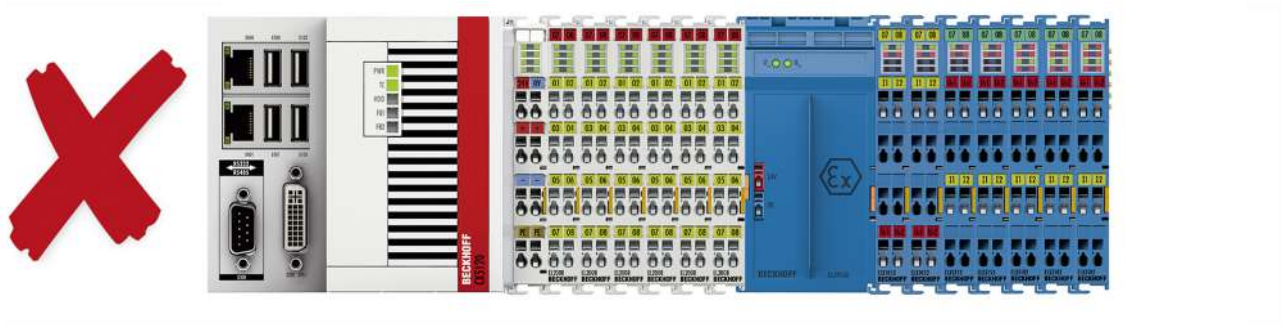


Fig. 12: Invalid arrangement - missing ELX9012 bus end cover.

NOTE

Observe the maximum output current of the ELX9560

When configuring the ELX terminal segment, please note the maximum available output current of the ELX9560 power supply terminal in accordance with the specified technical data.
If required, an additional power supply terminal ELX9560 with an upstream ELX9410 connected (see mounting examples) must be installed or a completely new terminal block must be assembled.

3.4 Installation position and minimum distances

Installation position

For the prescribed installation position the mounting rail is installed horizontally and the mating surfaces of the ELX terminals point toward the front (see illustration below). The terminals are ventilated from below, which enables optimum cooling of the electronics through convection. The direction indication “down” corresponds to the direction of positive acceleration due to gravity.

Minimum distances

Observe the following minimum distances to ensure optimum convection cooling:

- above and below the ELX terminals: 35 mm (required!)
- besides the bus terminal block: 20 mm (recommended)

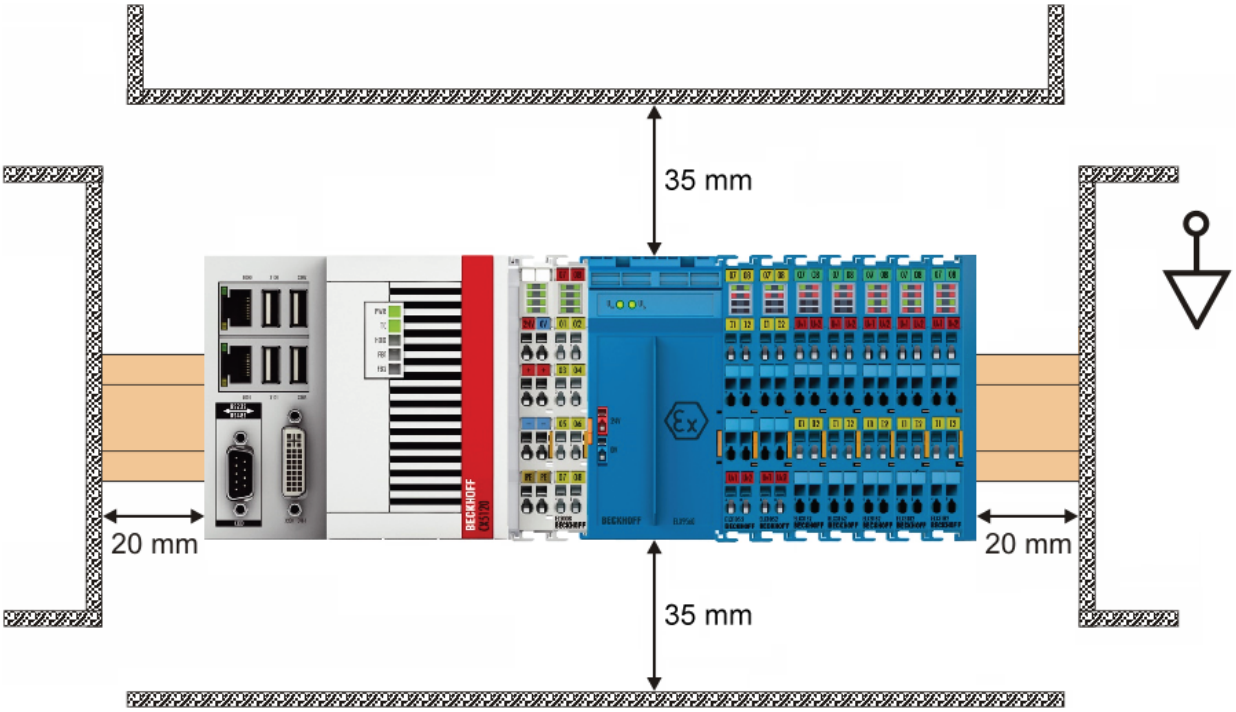


Fig. 13: Installation position and minimum distances

⚠ WARNING

Observe the minimum separation distances according to IEC 60079-14!

Observe the prescribed minimum separation distances between intrinsically safe and non-intrinsically safe circuits according to IEC 60079-14.

3.5 Installation of ELX terminals 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!

⚠ CAUTION

Danger of injury due to power contacts!

For your own protection, pay attention to careful and careful handling of the ELX terminals. In particular, the left side mounted, sharp-edged blade contacts pose a potential risk of injury.

Assembly

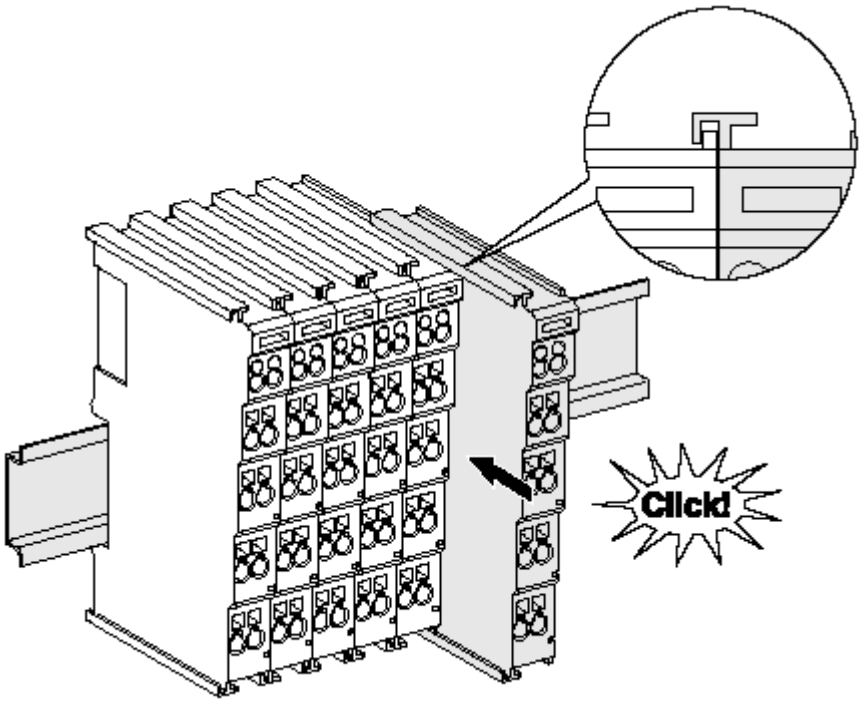


Fig. 14: Attaching on mounting rail

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

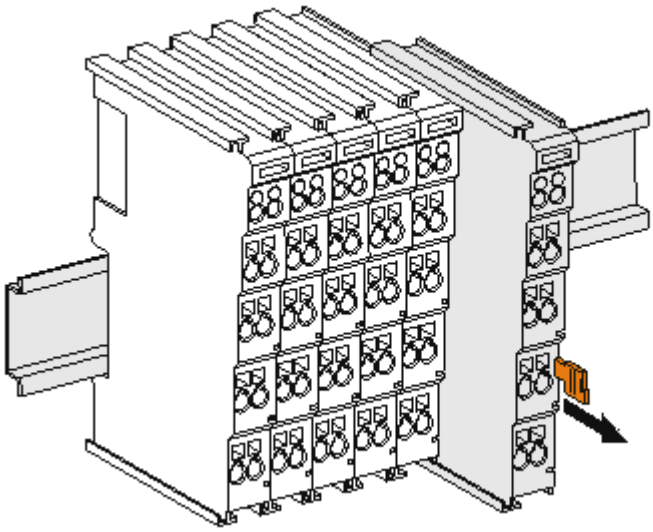


Fig. 15: Disassembling of terminal

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 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 of the ELX terminals are supplied by the ELX9560 power terminal. This interrupts the power contacts and thus represents the beginning of a new supply rail.

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.

3.6 Connection

3.6.1 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!

The terminals of ELXxxxx series include electronics and connection level in a single enclosure.

Standard wiring

Fig. 16: Standard wiring

The terminals of ELXxxxx series feature integrated screwless spring force technology for fast and simple assembly.

High Density Terminals (HD Terminals)

Fig. 17: *High Density Terminals*

The Bus Terminals from these series with 16 connection 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.

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 below!

3.6.2 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

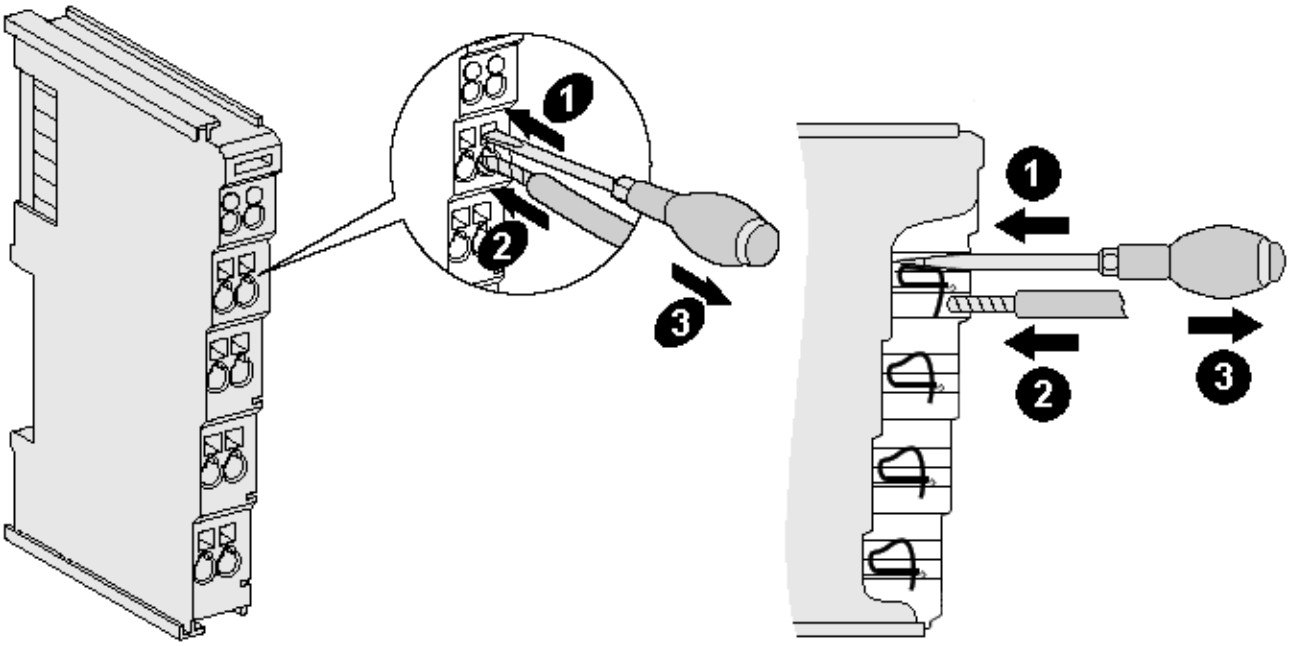


Fig. 18: Connecting a cable on a terminal point

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.

Observe the requirements for connecting cables and cross sections according to IEC 60079-7 and IEC 60079-11. See the following tables for the suitable wire size width.

Terminal housing	Standard wiring	ELX9560
Wire size width (single core wires)	0.08 ... 2.5 mm²	0.14 ... 1.5 mm²
Wire size width (fine-wire conductors)	0.08 ... 2.5 mm²	0.14 ... 1.5 mm²
Wire size width (conductors with a wire end sleeve)	0.14 ... 1.5 mm²	0.14 ... 1.0 mm²
Wire stripping length	8 ... 9 mm	8 ... 9 mm

NOTE

Maximum screwdriver width for ELX9560

Use a screwdriver with a maximum width of 2 mm to wire the ELX9560 power supply terminal. Wider screwdrivers can damage the terminal points.

High Density Terminals (HD Terminals) 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 mm^2
Wire size width (fine-wire conductors)	0.25 ... 1.5 mm^2
Wire size width (conductors with a wire end sleeve)	0.14 ... 0.75 mm^2
Wire size width (ultrasonically “bonded" conductors)	only 1.5 mm^2
Wire stripping length	8 ... 9 mm

3.6.3 Proper line connection

Always connect only one wire per terminal point.

When using fine-wire conductors it is recommended to connect them with wire end sleeves in order to establish a safe, conductive connection.

In addition, make sure that the pin assignment is correct to prevent damage to the ELX terminals and the connected devices.

3.6.4 Shielding and potential separation

Shielding

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

⚠ CAUTION

Observe installation requirements in areas of potentially explosive atmospheres!

During installation, observe the requirements for cables, shielding and earth potential equalization in areas of potentially explosive atmospheres according to IEC 60079-11, IEC 60079-14 and IEC 60079-25.

⚠ WARNING

Ensure potential separation of the 24 V Ex busbar!

In any case, make sure that the galvanic isolation made by the ELX9560 between the 24 V Ex busbar (power contacts +24 V Ex and 0 V Ex) and other system potentials (if applicable also functional or protective earths) is not removed.

3.6.5 Contact assignment

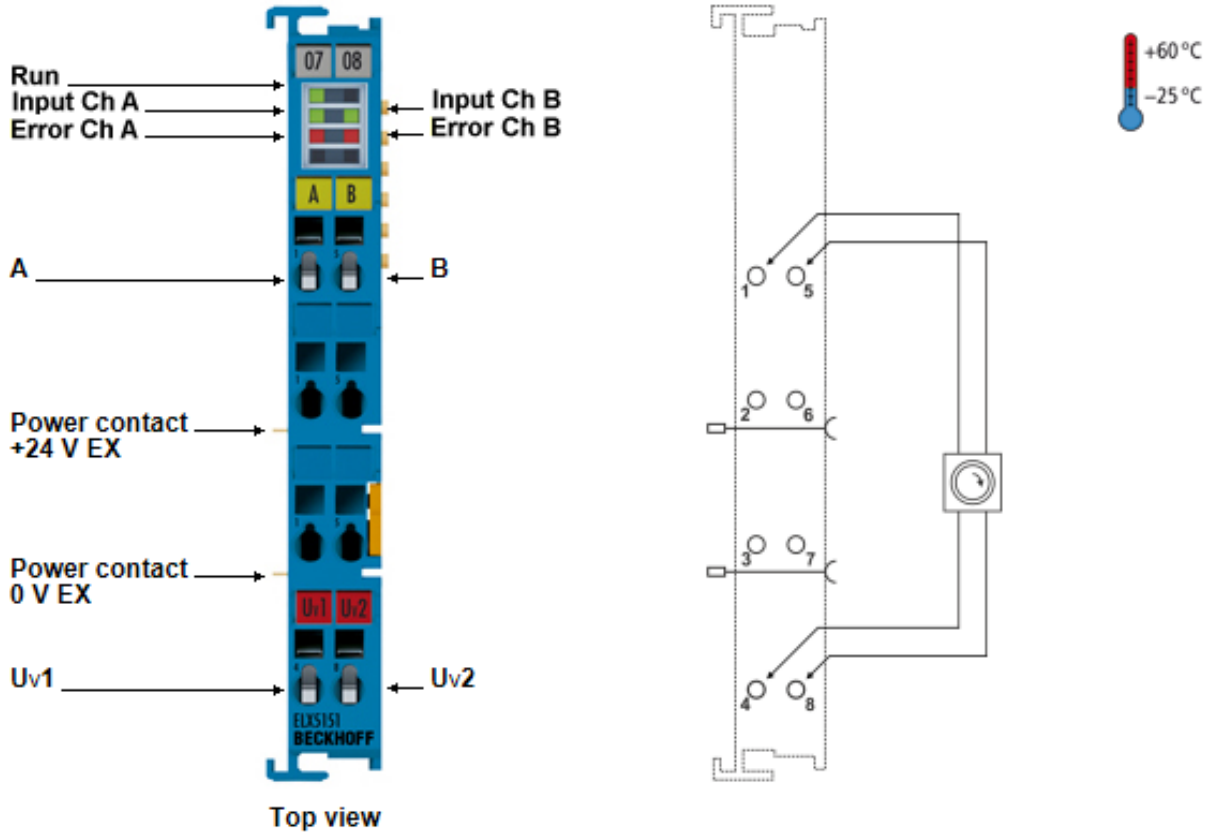


Fig. 19: ELX5151 - Contact assignment

Terminal point		Description
Name	No.	
A	1	Encoder input channel A
	2	not implemented
	3	not implemented
Uv1	4	Encoder supply channel A
B	5	Encoder input channel B
	6	not implemented
	7	not implemented
Uv2	8	Encoder supply channel B

LED display

LED	Color	Meaning	
RUN	green	This LED indicates the terminal's operating state:	
		off	State of the EtherCAT State Machine: **INIT** = initialization of the terminal or **BOOTSTRAP** = function for firmware updates of the terminal
		flashing	State of the EtherCAT State Machine: **PREOP** = function for mailbox communication and different standard-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
Input Ch A	green	flashing when pulses are present	
Error Ch A	red	shines at short circuit or wire breakage	
Input Ch B	green	flashing when pulses are present	
Error Ch B	red	shines at short circuit or wire breakage	

3.7 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.

4 Basic function principles

4.1 EtherCAT basics

Please refer to the EtherCAT System Documentation for the EtherCAT fieldbus basics, also available as PDF file from the download area of your ELX terminal on https://www.beckhoff.com/ELXxxxx.

4.2 Basic function principle of incremental encoders

The terminal acquires the 90° phase-shifted digital output signal of an incremental encoder on channels A and B. These signals are converted into a position value with quadruple evaluation with the aid of the quadrature encoder and the 32-bit counter. The latch and reset functions enable the exact referencing and saving of the counter value, irrespective of the speed.

Incremental encoders divide a 360° rotation of the encoder axis into individual steps (increments).
The phase angle between the signals on channels A and B sets the counting direction.

- Up: signal on channel A leads signal on channel B by 90°
- Down: signal on channel A lags signal on channel B by 90°

In case of single evaluation, the positive edges on channel A are counted.
In case of quadruple evaluation, the positive and negative edges on channel A and channel B are counted.

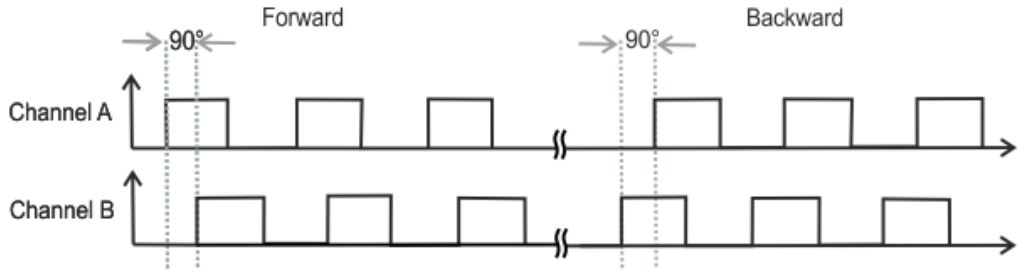


Fig. 20: *Quadrature decoder*

Whereas absolute value encoders deliver an absolute and unambiguous position value over the complete travel path directly after switching on, it is necessary with incremental encoders to perform a reference run (homing) after switching on in order to be able to determine an unambiguous position.

4.3 Process data

General

The ELX5151 incremental encoder interface terminal enables connection of incremental encoders with the Bus Coupler or the PLC.

The ELX5151 can also be used as a single-channel 32/16 bit counter on channel A, in which case the signal level on channel B defines the count direction. The changeover to this mode takes place via the CoE directory.

The *Counter Value* input value represents a 32-bit “position counter”. At the period input the period between two positive edges of channel A is measured with a resolution of 100 ns (default setting, decimal value x 100 ns). Depending on the setting (index 0x8000:14 [▶ 112], index 0x8000:16 [▶ 112]), the period length may be up to 1.6 s or 3.2 s.

4.3.1 Process data selection

Online display of the process data and structural contents

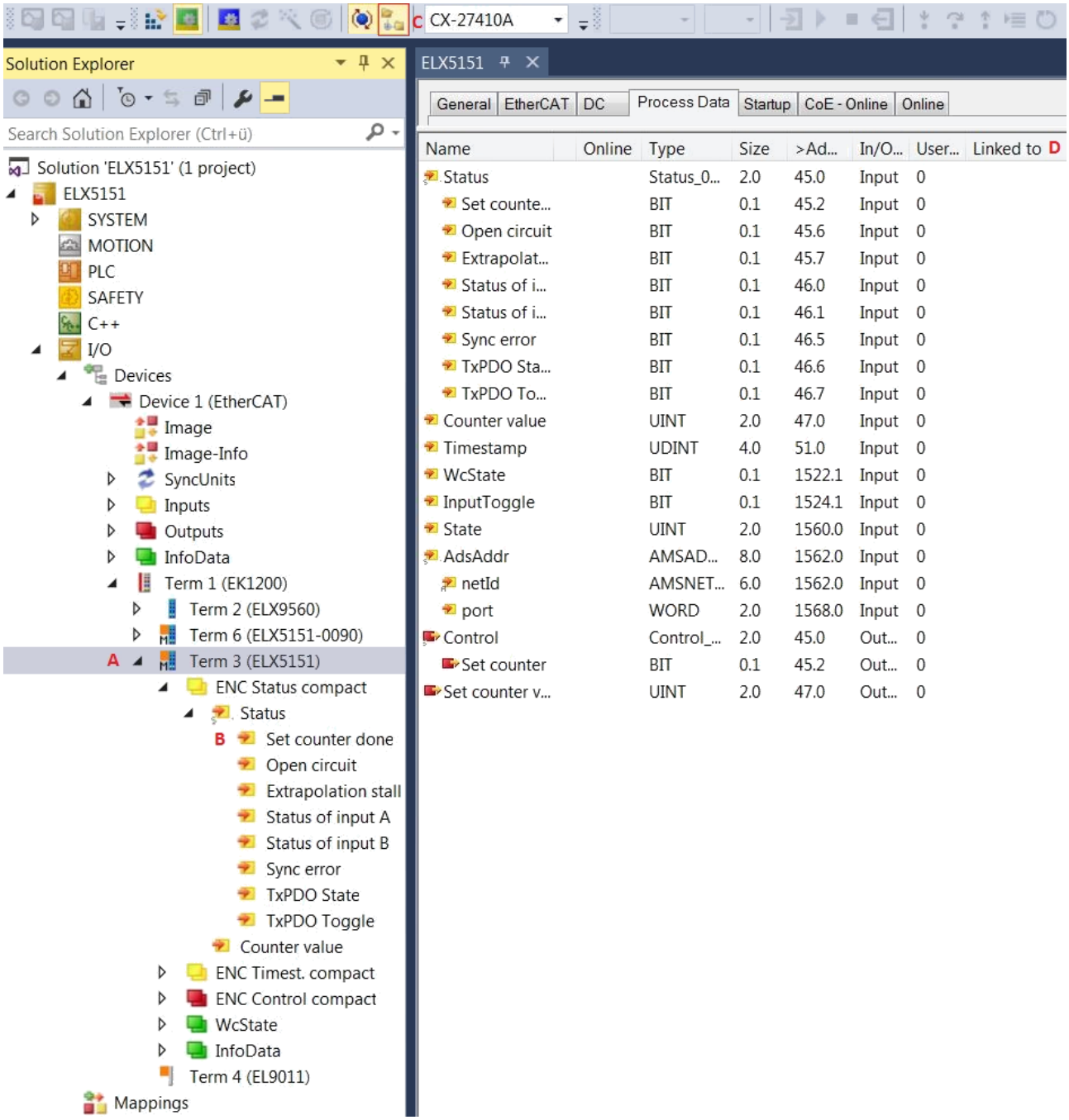


Fig. 21: ELX5151 - Online display of the process data and structural contents

The plain text display of the bit meanings is particularly helpful not only in commissioning, but also for linking to the PLC program.
By right-clicking on the Status variable in the configuration tree (A), the structure can be opened for linking (B).
Activation of the *Show Sub Variables* button (C) displays all subvariables and links to the PLC (D) in the online view.

Selection of the operating mode - DC (Distributed Clocks)

The operating mode is selected via the *DC* tab in the *Operating Mode* dialog box. The supported operating modes are displayed in the selection dialog.

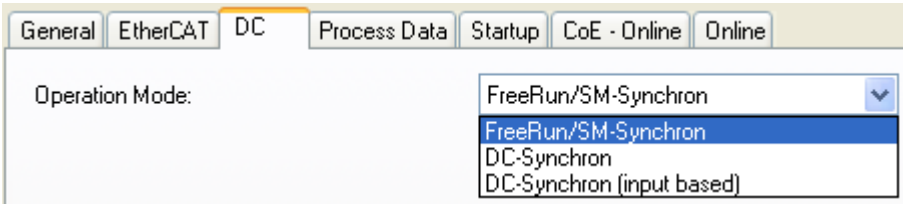
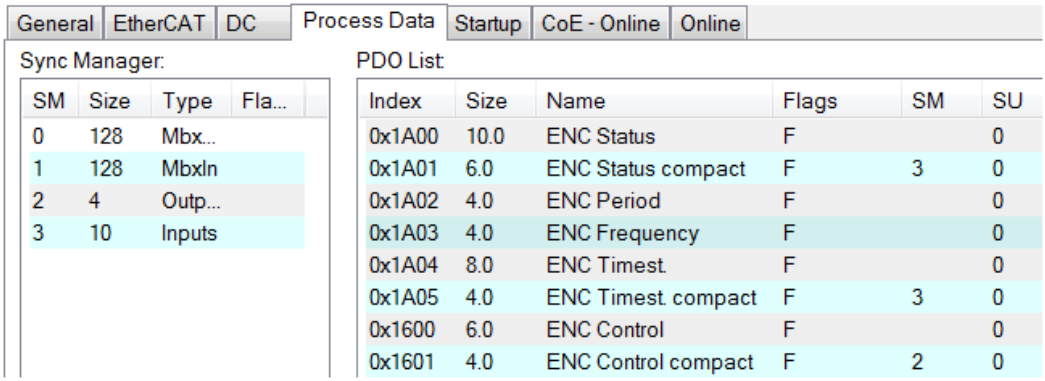


Fig. 22: DC tab

- **FreeRun**: the terminal operates frame-triggered. Cyclic operation is started via the SyncManagers of the EtherCAT frame processing.
- **DC-Synchron**: cyclic operation in the terminal is started by the local distributed clock at exact intervals. The start time is chosen such that it coincides with other output slaves in the EtherCAT system.
- **DC-Synchron (input based)**: as DC-Synchron mode, with the cyclic start time chosen such that it coincides with other input slaves in the EtherCAT system.

ELX5151 - Parameterization

An ELX5151 is parameterized via two dialog windows in the TwinCAT System Manager. Dialog box (A) shows the process data that can be parameterized based on the CoE directory (B).



SM	Size	Type	Fla...
0	128	Mbx...	
1	128	MbxIn	
2	4	Outp...	
3	10	Inputs	

Index	Size	Name	Flags	SM	SU
0x1A00	10.0	ENC Status	F		0
0x1A01	6.0	ENC Status compact	F	3	0
0x1A02	4.0	ENC Period	F		0
0x1A03	4.0	ENC Frequency	F		0
0x1A04	8.0	ENC Timest.	F		0
0x1A05	4.0	ENC Timest. compact	F	3	0
0x1600	6.0	ENC Control	F		0
0x1601	4.0	ENC Control compact	F	2	0

Fig. 23: ELX5151 - Parameterization

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

Main PDO

The EL515x offers two main PDOs per channel for the transmission of the basic process data.

The main PDO for the inputs contains the status data and a counter value. The counter value is represented and transmitted as a 32-bit value or a 16-bit value, depending on the selection of the corresponding PDO.

The main PDOs for the outputs contain the control data and a value for setting the counter value. The value for setting the counter value is represented and transmitted as a 32-bit value or a 16-bit value, depending on the selection of the corresponding PDO.

Optional PDOs

A PDO with a 32-bit frequency value or a 32-bit period value can optionally be added to the main PDO per channel.

A 32-bit or 64-bit time stamp is available as an additional PDO. The time stamp indicates the time of the last registered increment edge, based on the Distributed Clock system.

An overview of the available PDOs and their contents can be found in the System Manager (see fig. Selection dialog "Predefined PDO Assignment", example of EL5151 [▶ 32] D and E) or in the Section Operation modes [▶ 33].

Please observe the permissible combination possibilities (operation modes) when selecting the PDOs.

Predefined PDO Assignment selection dialog (from TwinCAT 2.11 build 1544 onwards)

The process data to be transmitted (PDO, ProcessDataObjects) can be selected by the user

- for all TwinCAT versions via the selection dialog *Predefined PDO Assignment* (see fig. *ELX5151 - Selection dialog Predefined PDO Assignment; A*) or
- selectively for individual PDOs (see fig. *ELX5151 - Selection dialog Predefined PDO Assignment; B*).

These changes become effective after activation and an EtherCAT restart or a reload.
Please observe the permissible combinations shown in the section Operation modes.

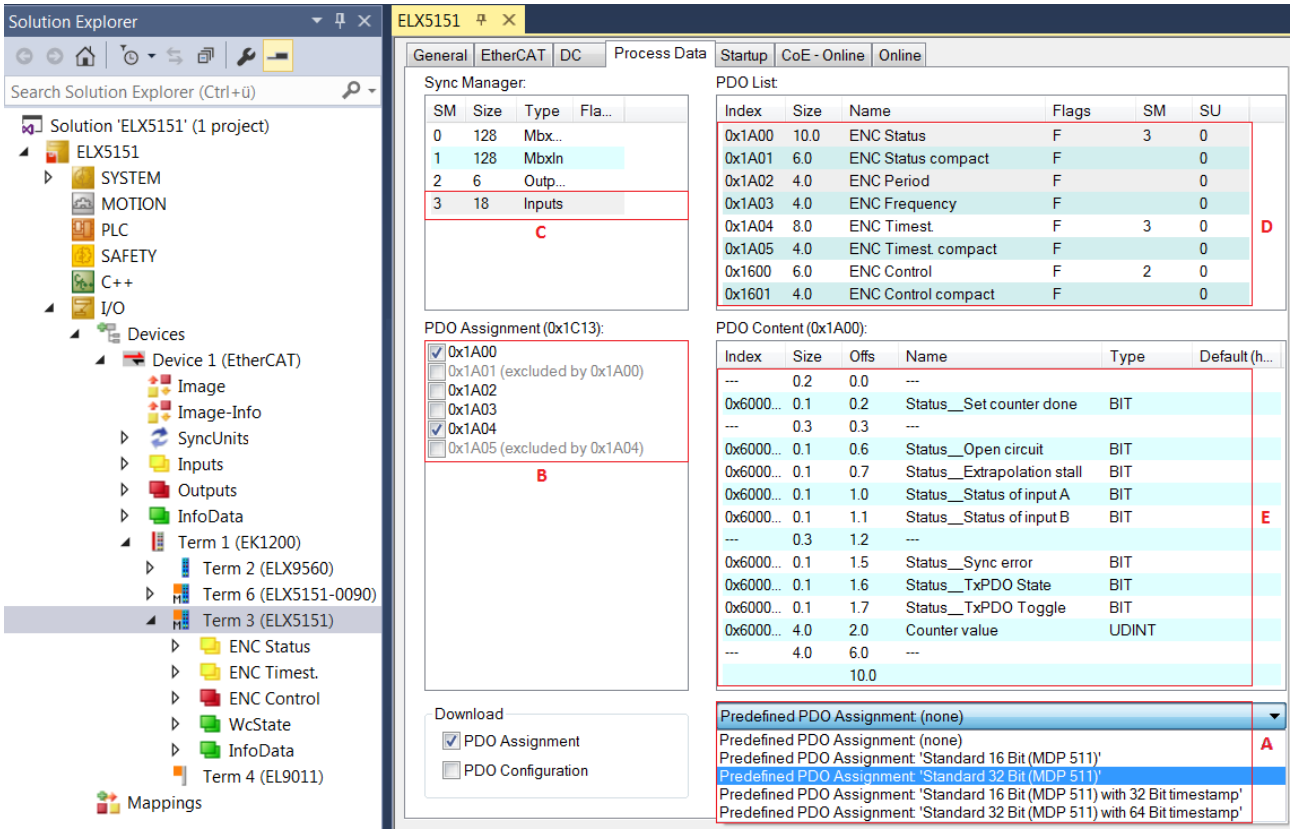


Fig. 24: ELX5151 - Selection dialog Predefined PDO Assignment

A Selection dialog *Predefined PDO Assignment*
B Display of (optional) PDOs (process data objects)
C Selection of the required Sync Manager
D Display of the PDOs available for selection
E Display of the contents of the PDO selected in D

The following predefined PDOs are available with the EL515x

- Standard 16-bit (MDP 511)
- Standard 32-bit (MDP 511)
- Standard 16-bit (MDP 511) with 32-bit timestamp, operation mode DC
- Standard 32-bit (MDP 511) with 64-bit timestamp, operation mode DC

The contents of the *Predefined PDO Assignment* can be found in the section Operation modes [▶ 33] or in the System Manager (see fig. *Selection dialog Predefined PDO Assignment; B, C*).

Loss of links when changing the PDO

In case of PDO changes, links already created in the changed objects are lost. When changing from Predefined PDO Assignment *Standard 16-bit (MDP 511)* to *Standard 32-bit (MDP 511)*, for example, the already created links of the objects *Counter value* (index 0x6000:11 [▶ 113]), *Latch value* (index 0x6000:12 [▶ 113]) and *Set Counter value* (index 0x7000:11 [▶ 113]) are deleted.

4.3.2 Operating modes

Permissible operating modes for the ELX5151

The following modes are available for the ELX5151. They apply both for the encoder analysis and counter terminal mode.

The combinations of DC, PDO and CoE settings listed below are permissible per mode. Other settings can lead to irregular process data and error messages in the TwinCAT System Manager Logger window.

Mode	DC	Main PDO	Optional PDO 1	Optional PDO 2	Features CoE
1	FreeRun	Predefined PDO Assignment: *Standard 32-bit (MDP 511)*: 0x1A00 + 0x1600 + 0x1A02			CoE combinations 0x8000:nn
		0x1A00 [▶ 127] Inputs: 16 Bit Status, 32 Bit Counter Value + 0x1600 [▶ 126] Outputs: 16 Bit Control, 32 Bit Set Counter Value	0x1A02 [▶ 128] 32 Bit Period or 0x1A03 [▶ 128] 32 Bit Frequency	-	
2	FreeRun	Predefined PDO Assignment: *Standard 16-bit (MDP 511)*: 0x1A01 + 0x1601 + 0x1A02			CoE combinations 0x8000:nn
		0x1A01 [▶ 128] Inputs: 16 Bit Status, 16 Bit Counter Value + 0x1601 [▶ 126] Outputs: 16 Bit Control 16 Bit Set Counter Value	0x1A02 [▶ 128] 32 Bit Period or 0x1A03 [▶ 128] 32 Bit Frequency	-	
3	DC/DCi	Predefined PDO Assignment: *Standard 32 Bit with 64 Bit Timestamp (MDP 511)*: 0x1A00 + 0x1600 + 0x1A02 + 0x1A04			CoE combinations 0x0x8000:nn
		0x1A00 [▶ 127] Inputs: 16 Bit Status, 32 Bit Counter Value + 0x1600 [▶ 126] Outputs: 16 Bit Control, 32 Bit Set Counter Value	0x1A02 [▶ 128] 32 Bit Period or 0x1A03 [▶ 128] 32 Bit Frequency	0x1A04 [▶ 128] 64 Bit Timestamp or 0x1A05 [▶ 128] 32 Bit Timestamp (compact)	
4	DC/DCi	Predefined PDO Assignment: *Standard 16 Bit with 32 Bit Timestamp (MDP 511)*: 0x1A01 + 0x1601 + 0x1A02 + 0x1A05			CoE combinations 0x8000:nn
		0x1A01 [▶ 128] Inputs: 16 Bit Status, 16 Bit Counter Value + 0x1601 [▶ 126] Outputs: 16 Bit Control, 16 Bit Set Counter Value	0x1A02 [▶ 128] 32 Bit Period or 0x1A03 [▶ 128] 32 Bit Frequency	0x1A04 [▶ 128] 64 Bit Timestamp or 0x1A05 [▶ 128] 32 Bit Timestamp (compact)	

Parameterization of the ELX5151

- Use a CoE reset in order to deactivate any previous settings
- In order to activate the new operating mode, reload the EtherCAT slaves

4.3.3 Settings via the CoE directory

Depending on the main PDO/optional PDOs further settings can be selected in the CoE list (CAN over EtherCAT).

Parameterization via the CoE list (CAN over EtherCAT)

Please note the following general CoE information 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
- “CoE-Reload” for resetting the changes

The following CoE settings are possible from object 0x8000:0 and are shown below in their default settings:

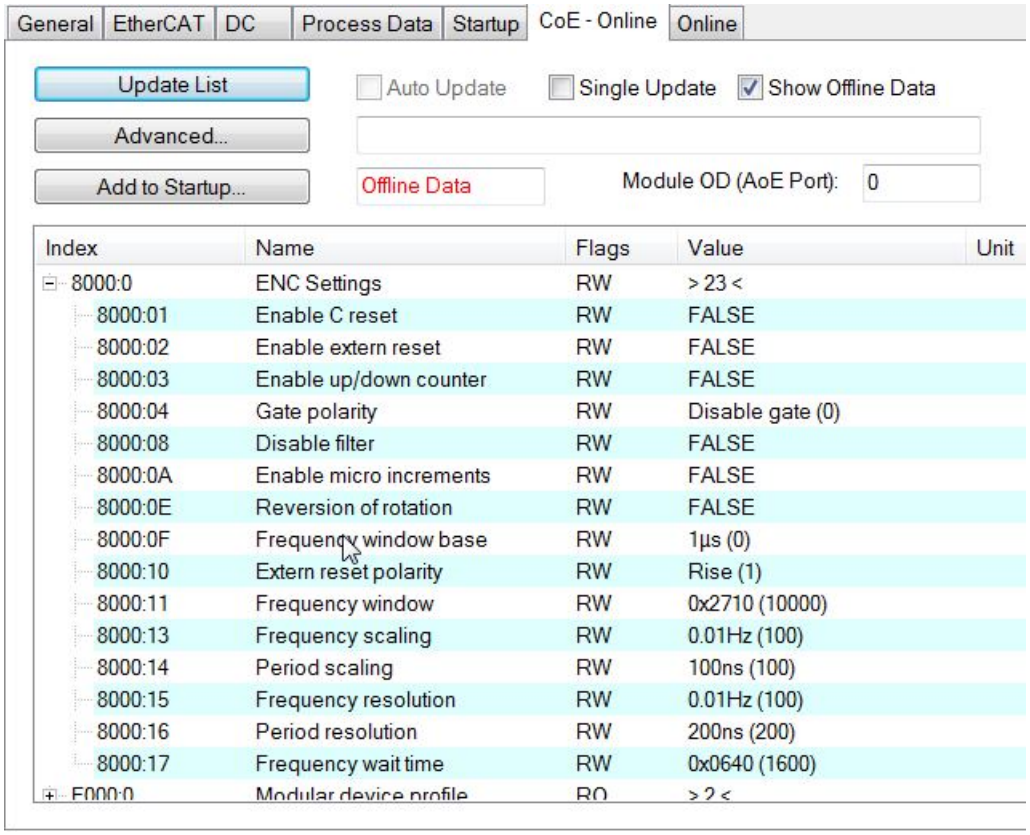


Fig. 25: EL5151 - CoE Online tab,

The parameters are described on the chapter Object description and parameterization.

4.3.4 Explanatory notes for parameters and modes

Frequency

- The timeframe for the frequency calculation as well as the resolution can be parameterized in the CoE objects *Frequency window* (index: 0x8000:11 [▶ 112]), *Frequency scaling* (index: 0x8000:13 [▶ 112]), *Frequency resolution* (index: 0x8000:15 [▶ 112]) and *Frequency wait time* (index: 0x8000:17 [▶ 112]).
- The positive edges of track A are counted in the specified timeframe (see Frequency modes [▶ 35]) and the next subsequent edge including the time until it arrives is counted. The waiting time for the edge can be set in the CoE object *Frequency Wait Time* (index: 0x8000:17 [▶ 112]) (unit: ms) and is set as standard to 1.6 seconds. This is also the maximum value.
- The frequency is always specified as a positive number, irrespective of the sense of rotation.
- The size of the timeframe is 10 ms (default), but at the least the basic unit *Frequency window base* (index: 0x8000:0F [▶ 112]).
- This calculation is carried out in the terminal in free-running mode without reference to the distributed clocks system. It is therefore independent of the DC mode.
- The object *Frequency window base* (index: 0x8000:0F [▶ 112]) is used for switching the basic unit for the *Frequency window* between 1 µs and 1 ms, in order to adjust the time window for the measurement. The following maximum measuring windows are therefore possible:

Basic unit	Max. timeframe
1 µs	65.5 ms
1 ms	65 s

- on expiry of the measuring window *Frequency window* (index: 0x8000:11 [▶ 112]), the subsequent positive edge on track A is awaited, but at the longest for 1.6 s or the time from *Frequency wait time* (index: 0x8000:17 [▶ 112]).
- The frequency is measured with different accuracies depending on the selected basic unit *Frequency window base* (index 0x8000:0F [▶ 112]) and the window size.

Frequency mode A

The measurement is automatically performed in frequency mode A if the window size is smaller than or equal to 600 ms.

- Basic unit 1 µs: all window sizes
- Basic unit 1 ms: up to 600 ms window size

Measurement sequence

- The measurement starts with a positive edge at track A. The current counter value and time (resolution: 100 ns) are stored.
- On expiry of the measuring window *Frequency window* (index: 0x8000:11 [▶ 112]), the subsequent rising edge on track A is awaited, but at the longest for 1.6 s or the time from *Frequency wait time* (index: 0x8000:17 [▶ 112]).
- The frequency is calculated from the edge difference and the actual elapsed time.

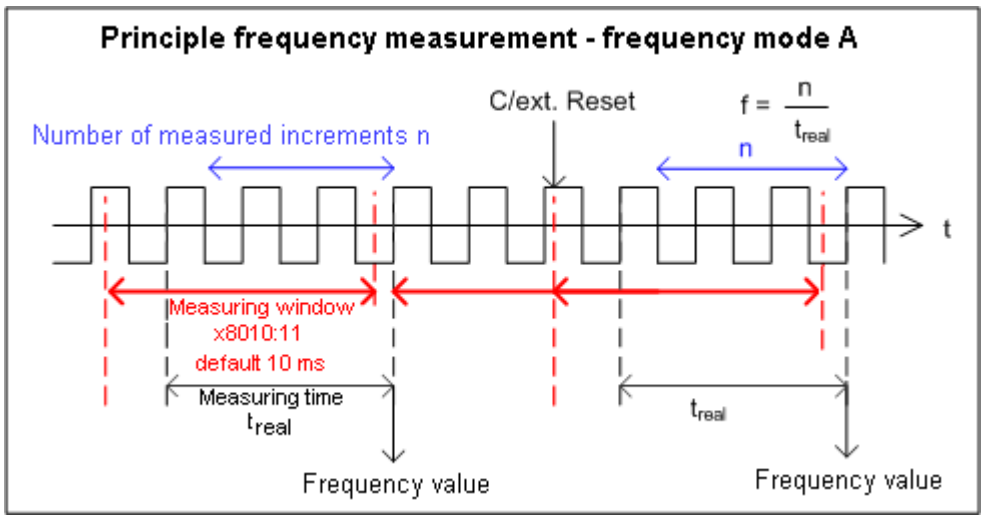


Fig. 26: Frequency measurement principle - frequency mode A

Frequency mode B

The measurement is automatically performed in frequency mode B if the window size is greater than 600 ms.

- Basic unit 1 ms: from 601 ms window size

Measurement sequence

- At the start of the measurement the time and the current position are stored with a resolution of 100 ns, irrespective of the current signal position.
- After the measurement the current position is stored irrespective of the current signal position.
- The frequency is calculated from the number of increments and the actual elapsed time.
- The frequency measurement therefore takes place with reduced accuracy.
- The larger the measuring window in relation to the basic unit, the more precise the frequency calculation.

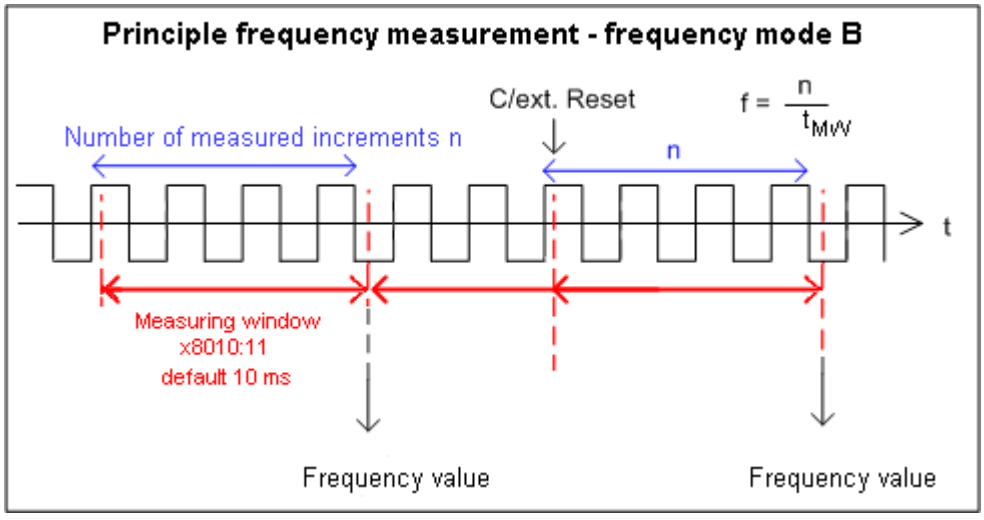


Fig. 27: Frequency measurement principle - frequency mode *B*

Period calculation

- This calculation is carried out free-running in the slave without reference to the distributed clocks system. It is therefore independent of the DC mode.
- During each cycle the interval between two positive edges of input A is counted.
- Depending on the setting, periods of up to 1.6 s or 3.2 s in length are measurable.
- If no edge change occurs for approx. 1.6 s, any period specification is cancelled.

Up/down counter

- The operating mode (encoder or up/down counter) is selected via the CoE object *Enable up/down counter* (index: 0x8000:03 [▶ 112]).
  On the *CoE - Online* tab, click on the row of the index to be parameterized, enter the corresponding value in the *SetValue* dialog and confirm with *OK*.
  - 0: the up/down counter is not active
  - 1: the up/down counter is active
- The counting direction (up/down) is specified via the signal level at channel B.
- An additional option for reversing the direction of rotation is to set the *Reversion of rotation* bit (index: 0x8000:0E [▶ 112]).
- Connection:

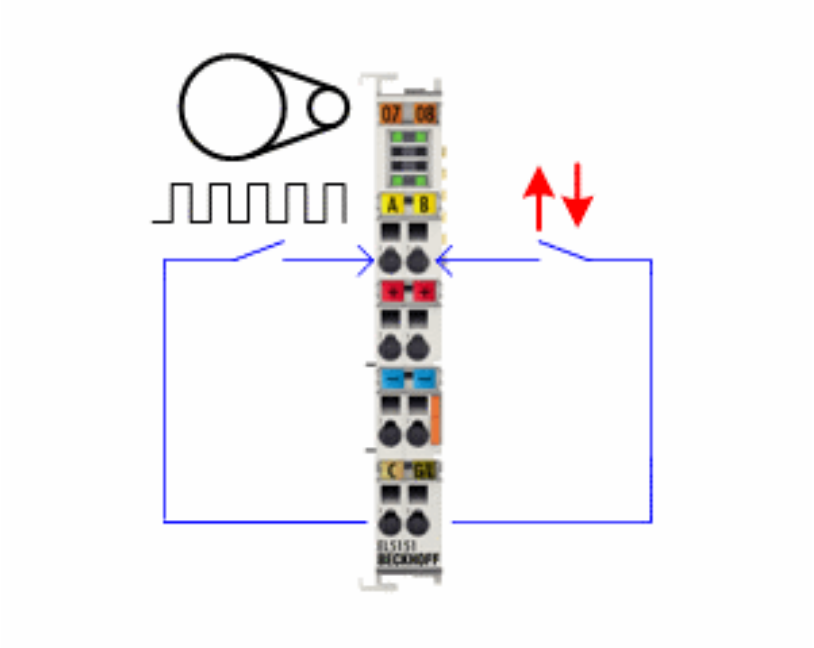


Fig. 28: ELX5151 - Counter connection principle

Micro-increments

- Works with and without distributed clocks, but in the ELX5151 this is only meaningful in conjunction with one of the DC modes
- By setting the counter value only the integer component can be modified.
- The principle:

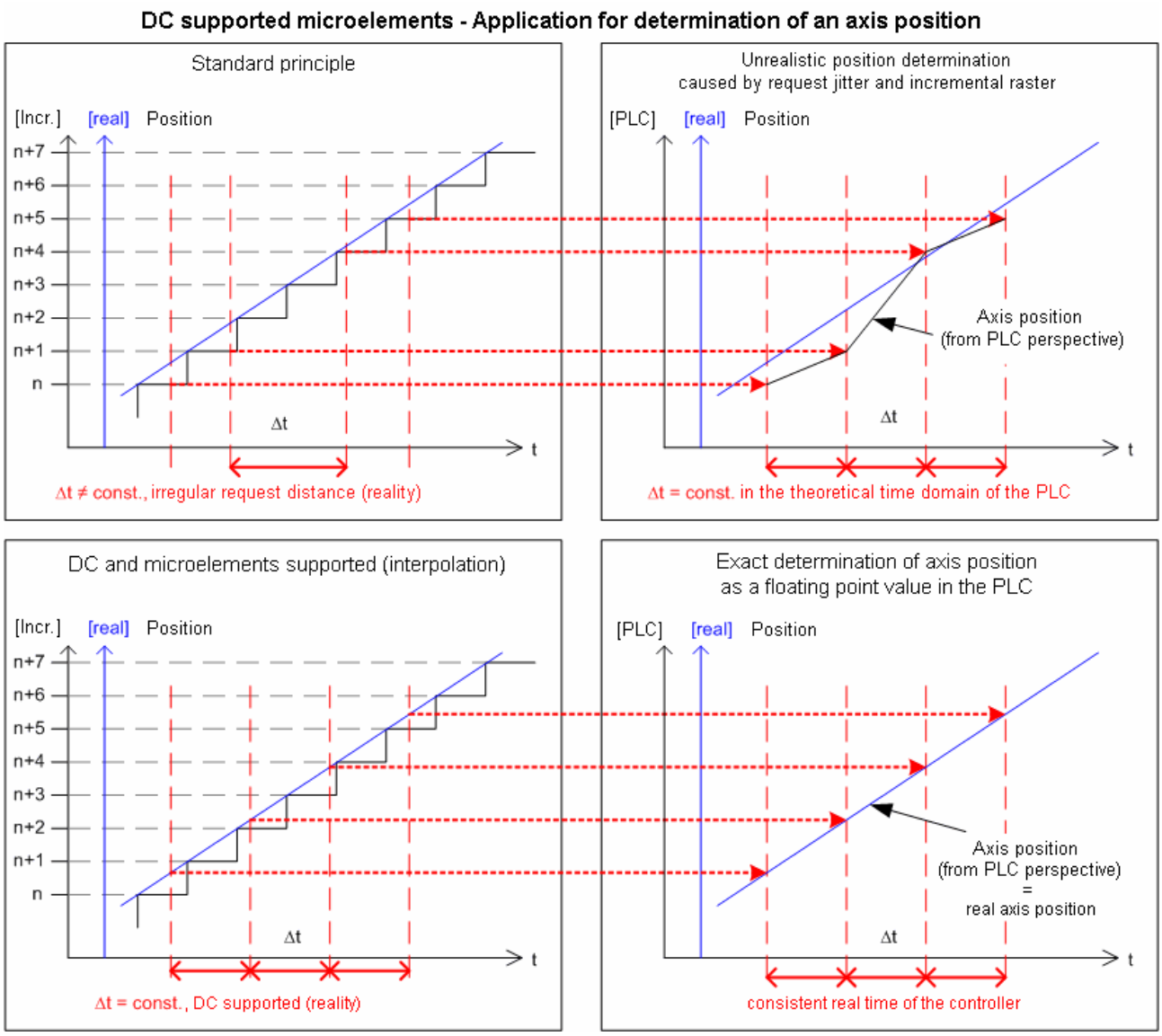


Fig. 29: Principle of frequency measurement

The highly constant query cycles (accuracy: 100 ns) of the distributed clocks system permit the ELX5151 to interpolate interpolated axis positions between the counted encoder increments above a certain speed. The interpolation resolution is 8 bit, corresponding to 256 values. A standard encoder with 1,024 bars with 4-way evaluation and micro-increments thus becomes a high-resolution axis encoder with
4096 * 256 = 1,048,567 bars.
If the speed falls below the minimum speed, this is displayed by the object *Extrapolation stall* (index: 0x6000:08 [▶ 113]) in the process data.

Set the counter value - referencing (Index 7000:03)

Since incremental encoders do not deliver an unambiguous position value after switching on, a homing [▶ 29] must be carried out.

The EL5151-0021 offers the option to set the reference point manually via *Set counter* (index 0x7000:03 [▶ 113]) on reaching the zero pulse.

- **Set counter (index** 0x7000:03**)**
  - The value to be set as reference value (default: 0) is written in *Set counter value* (index 0x7000:11 [▶ 113]).
  - The function is activated by setting the bit in *Set counter* (index 0x7000:03 [▶ 113]) to TRUE.
  - The value from *Set counter value* (index 0x7000:11 [▶ 113]) is written in C*ounter value* (index 0x6000:11 [▶ 113]).
  - The value of the bit in *Set counter done* (index 0x6000:03 [▶ 113]) is set to TRUE.
  - After re-activation of *Set counter* (index 0x7000:03 [▶ 113]), the next reference value is written into *Counter value* (index 0x6000:11 [▶ 113]) only if the value of the *Set counter done* bit (index 0x6000:03 [▶ 113]) is FALSE.
    The *Set counter done* bit (index 0x6000:03 [▶ 113]) is reset if *Set counter* (index 0x7000:03 [▶ 113]) has been reset.

4.3.5 TwinSAFE SC

4.3.5.1 TwinSAFE SC - operating principle

The TwinSAFE SC (Single Channel) technology enables the use of standard signals for safety tasks in any networks of fieldbuses. To do this, EtherCAT Terminals from the areas of analog input, angle/displacement measurement or communication (4…20 mA, incremental encoder, IO-Link, etc.) are extended by the TwinSAFE SC function. The typical signal characteristics and standard functionalities of the I/O components are retained. TwinSAFE SC I/Os have a yellow strip at the front of the housing to distinguish them from standard I/Os.

The TwinSAFE SC technology enables communication via a TwinSAFE protocol. These connections can be distinguished from the usual safe communication via Safety over EtherCAT.

The data of the TwinSAFE SC components are transferred via a TwinSAFE protocol to the TwinSAFE logic, where they can be used in the context of safety-relevant applications. Detailed examples for the correct application of the TwinSAFE SC components and the respective normative classification, which were confirmed/calculated by TÜV SÜD, can be found in the TwinSAFE application manual.

4.3.5.2 TwinSAFE SC - configuration

The TwinSAFE SC technology enables communication with standard EtherCAT terminals via the Safety over EtherCAT protocol. These connections use another checksum, in order to be able to distinguish between TwinSAFE SC and TwinSAFE. Eight fixed CRCs can be selected, or a free CRC can be entered by the user.

By default the TwinSAFE SC communication channel of the respective TwinSAFE SC component is not enabled. In order to be able to use the data transfer, the corresponding TwinSAFE SC module must first be added under the Slots tab. Only then is it possible to link to a corresponding alias device.

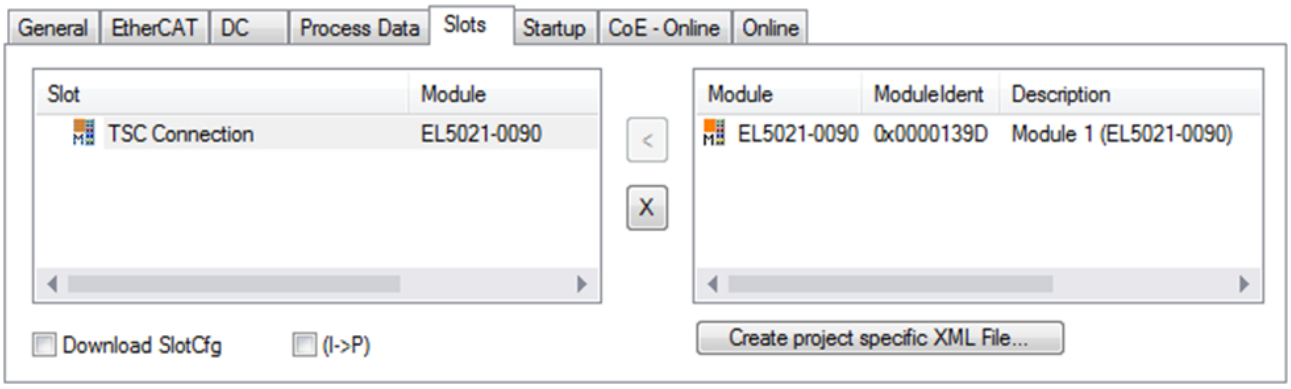


Fig. 30: Adding the TwinSAFE SC process data under the component, e.g. EL5021-0090

Additional process data with the ID TSC Inputs, TSC Outputs are generated (TSC - TwinSAFE Single Channel).

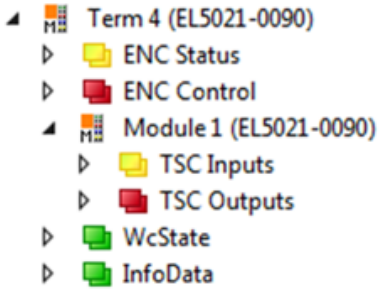


Fig. 31: TwinSAFE SC component process data, example EL5021-0090

A TwinSAFE SC connection is added by adding an alias devices in the safety project and selecting TSC (*TwinSAFE Single Channel*)

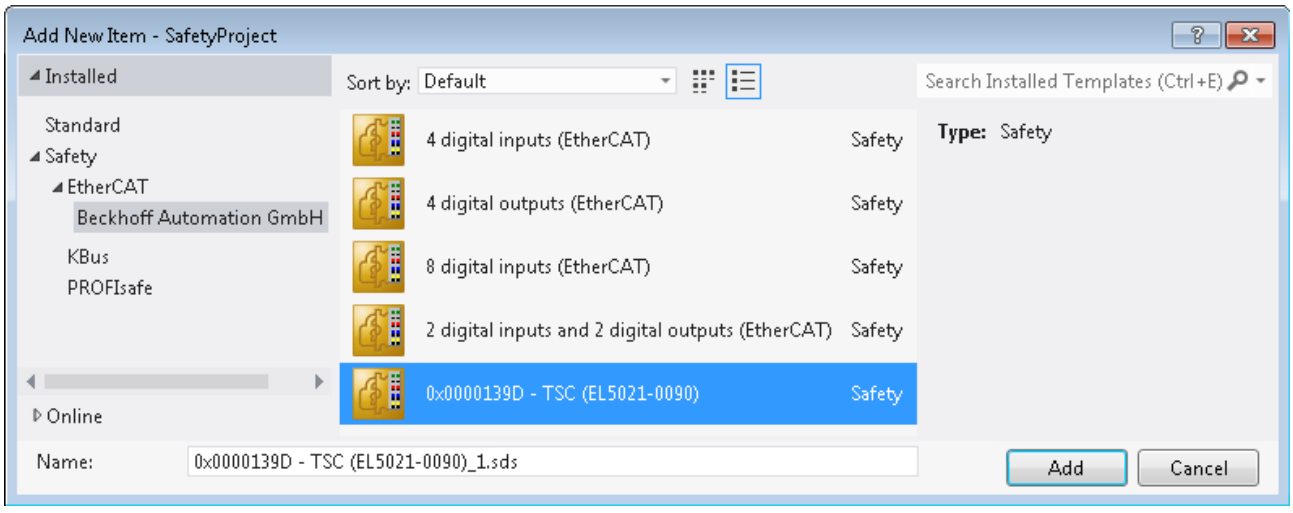


Fig. 32: Adding a TwinSAFE SC connection

After opening the alias device by double-clicking, select the Link button  next to *Physical Device,* in order to create the link to a TwinSAFE SC terminal. Only suitable TwinSAFE SC terminals are offered in the selection dialog.

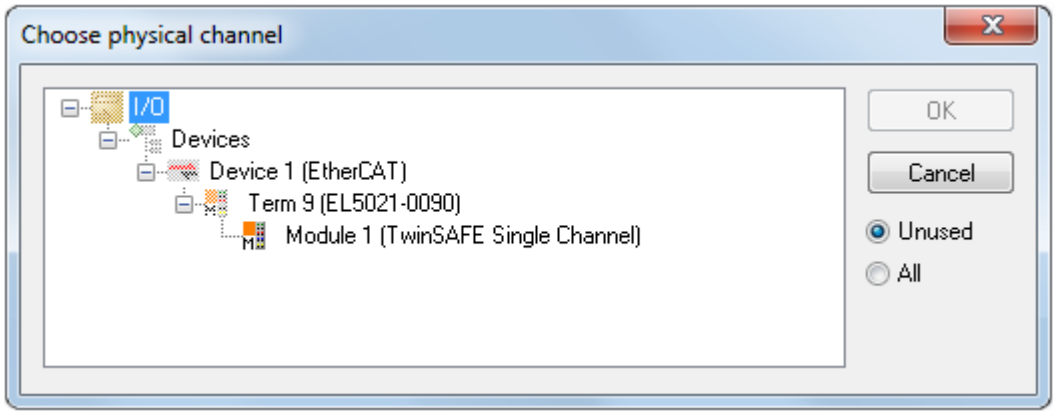


Fig. 33: Creating a link to TwinSAFE SC terminal

The CRC to be used can be selected or a free CRC can be entered under the Connection tab of the alias device.

Entry Mode	Used CRCs
TwinSAFE SC CRC 1 master	0x17B0F
TwinSAFE SC CRC 2 master	0x1571F
TwinSAFE SC CRC 3 master	0x11F95
TwinSAFE SC CRC 4 master	0x153F1
TwinSAFE SC CRC 5 master	0x1F1D5
TwinSAFE SC CRC 6 master	0x1663B
TwinSAFE SC CRC 7 master	0x1B8CD
TwinSAFE SC CRC 8 master	0x1E1BD

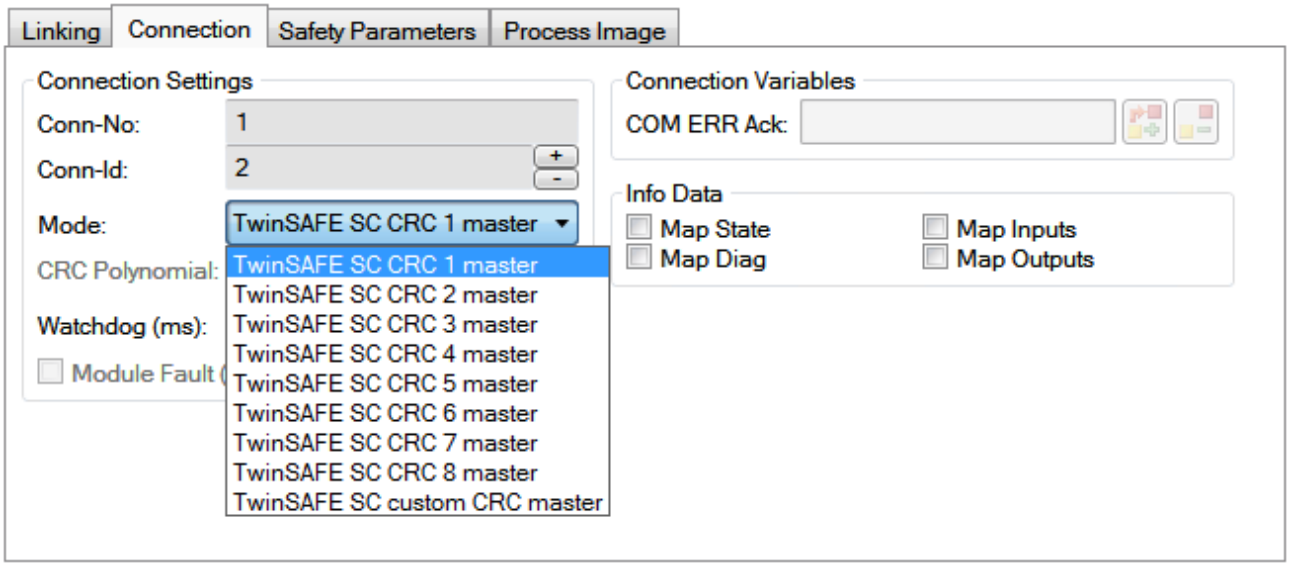


Fig. 34: Selecting a free CRC

These settings must match the settings in the CoE objects of the TwinSAFE SC component.
The TwinSAFE SC component initially makes all available process data available. The *Safety Parameters* tab typically contains no parameters. The process data size and the process data themselves can be selected under the *Process Image* tab.

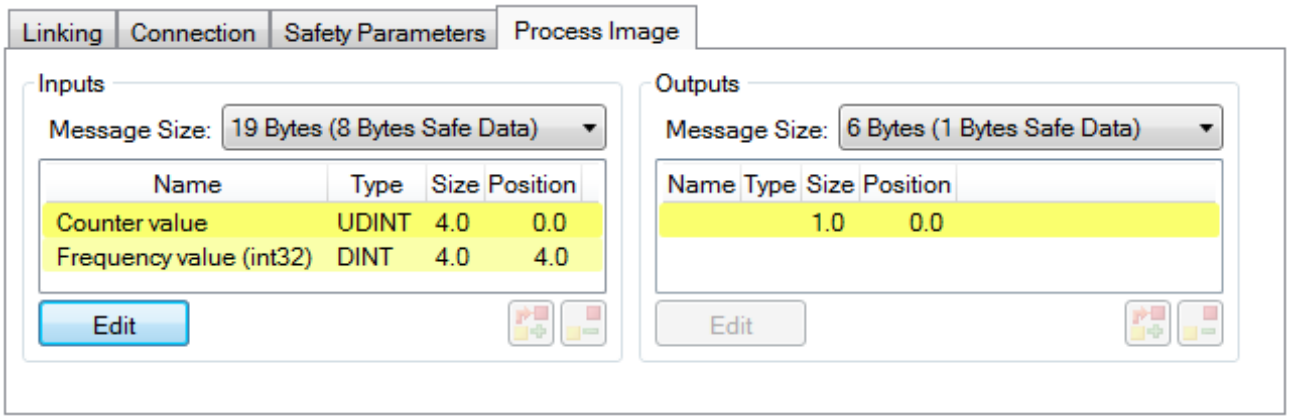


Fig. 35: Selecting the process data size and the process data

The process data (defined in the ESI file) can be adjusted to user requirements by selecting the *Edit* button in the dialog *Configure I/O element(s)*.

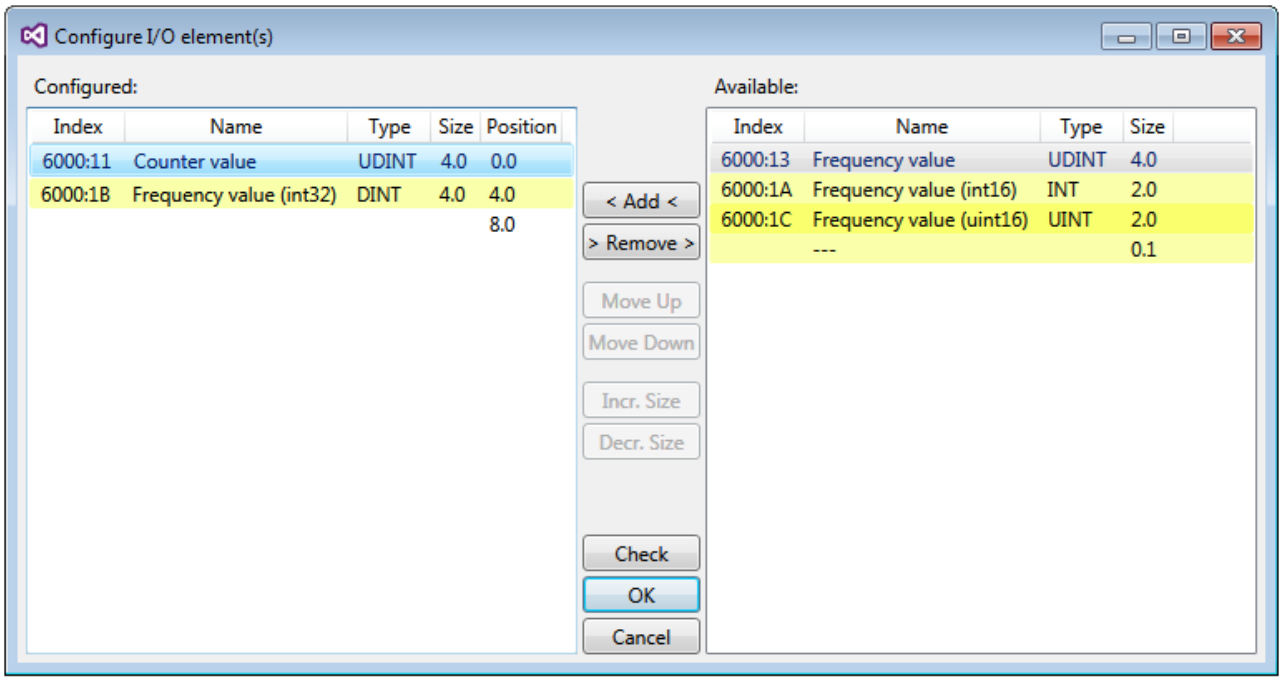


Fig. 36: Selection of the process data

The safety address together with the CRC must be entered on the TwinSAFE SC slave side. This is done via the CoE objects under *TSC settings* of the corresponding TwinSAFE SC component (here, for example, EL5021-0090, 0x8010: 01 and 0x8010: 02). The address set here must also be set in the *alias device* as *FSoE* address under the *Linking* tab.

Under the object 0x80n0:02 Connection Mode the CRC to be used is selected or a free CRC is entered. A total of 8 CRCs are available. A free CRC must start with 0x00ff in the high word.

8010:0	TSC Settings	RW	> 2 <
8010:01	Address	RW	0x0000 (0)
8010:02	Connection Mode	RW	TwinSAFE SC CRC1 master (97039)

Fig. 37: CoE objects 0x8010:01 and 0x8010:02

Object *TSC Settings*

Depending on the terminal, the index designation of the configuration object *TSC Settings* can vary. Example:
- EL3214-0090 and EL3314-0090, TSC Settings, Index 8040
- EL5021-0090, TSC Settings, Index 8010
- EL6224-0090, TSC Settings, Index 800F

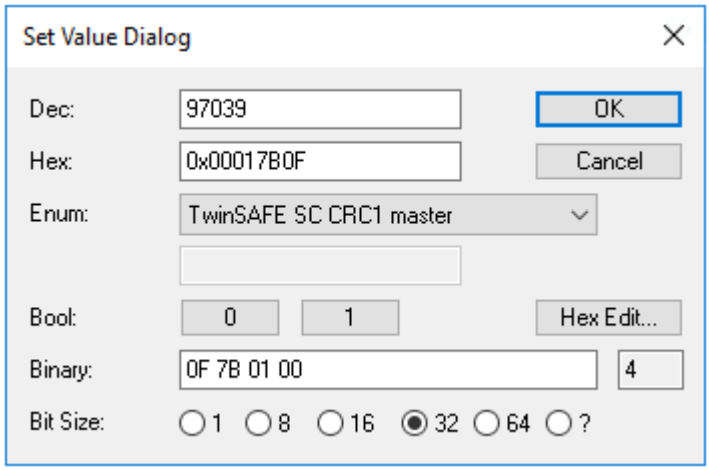


Fig. 38: Entering the safety address and the CRC

TwinSAFE SC connections

If several TwinSAFE SC connections are used within a configuration, a different CRC must be selected for each TwinSAFE SC connection.

4.3.5.3 TwinSAFE SC process data ELX5151-0090

The ELX5151-0090 transmits the following process data to the TwinSAFE logic:

Index (hex)	Name	Type	Size
6000:1D	Counter value (uint16)	UINT	2.0
6000:11	Counter value	UDINT	4.0
6000:13	Frequency value	UDINT	4.0
6000:14	Period value	UDINT	4.0
6000:1C	Frequency value (uint16)	UINT	2.0
6000:1E	Period value (uint16)	UINT	2.0

The Counter Value (uint16) (0x6000:1D) is transferred as default value. Via the “Process Image” tab, other data types can be selected or completely deselected in the Safety Editor.

Depending on the TwinCAT 3.1 version, process data can be renamed automatically when linking to the Safety Editor.

TwinSAFE SC Objects

The TwinSAFE SC objects of the ELX5151-0090 are listed in chapter Objects TwinSAFE Single Channel (ELX5151-0090) [▶ 132].

5 Parameterization and programming

5.1 TwinCAT Quick Start

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

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

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

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

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

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

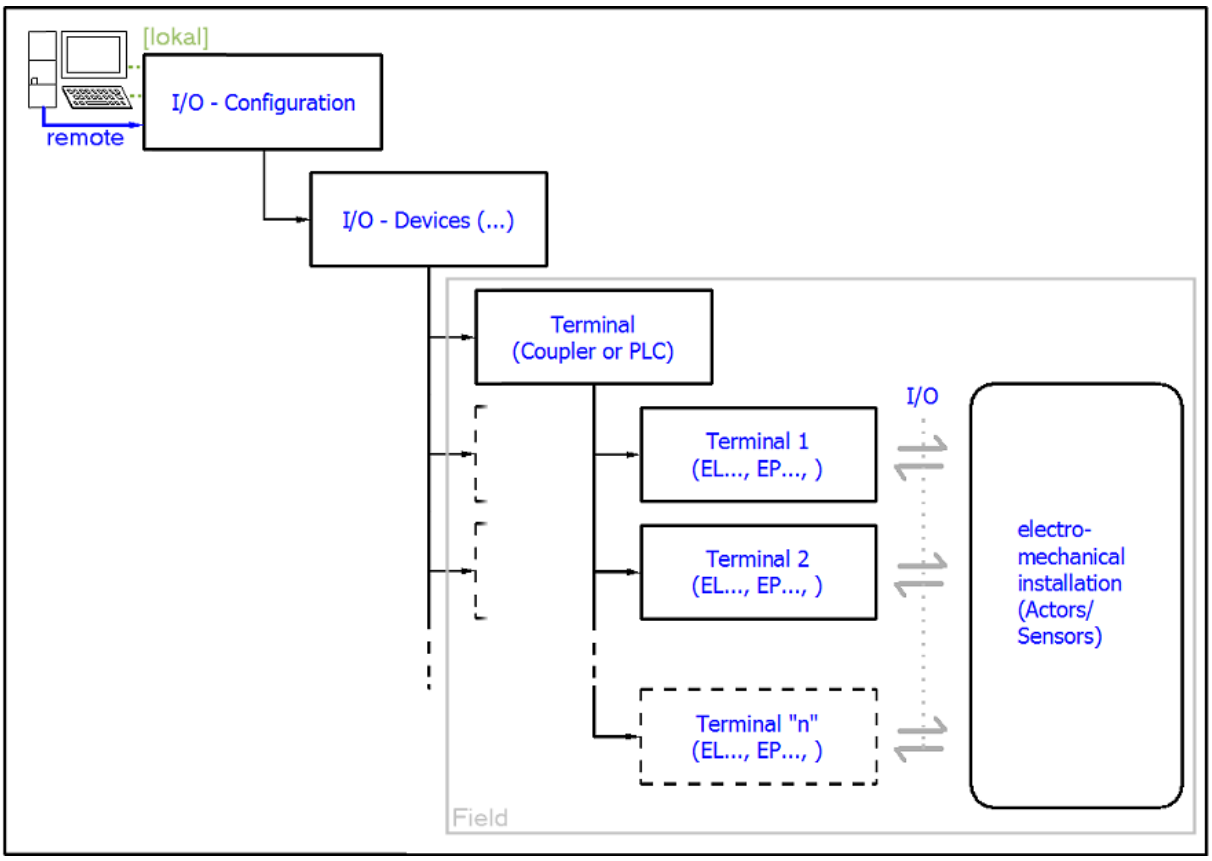


Fig. 39: Relationship between user side (commissioning) and installation

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

Sample configuration (actual configuration)

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

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

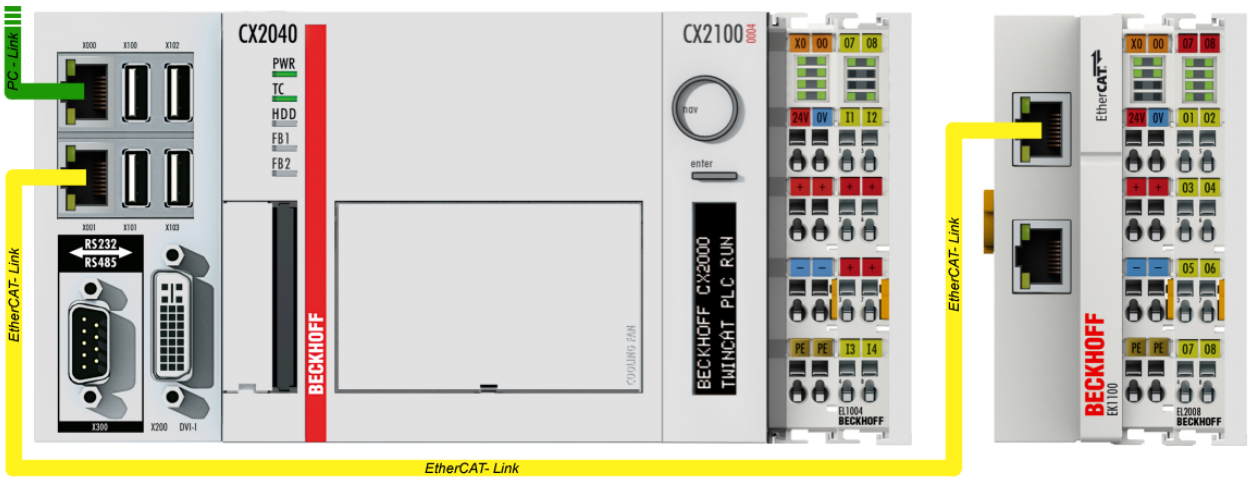


Fig. 40: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

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

5.1.1 TwinCAT 2

Startup

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

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

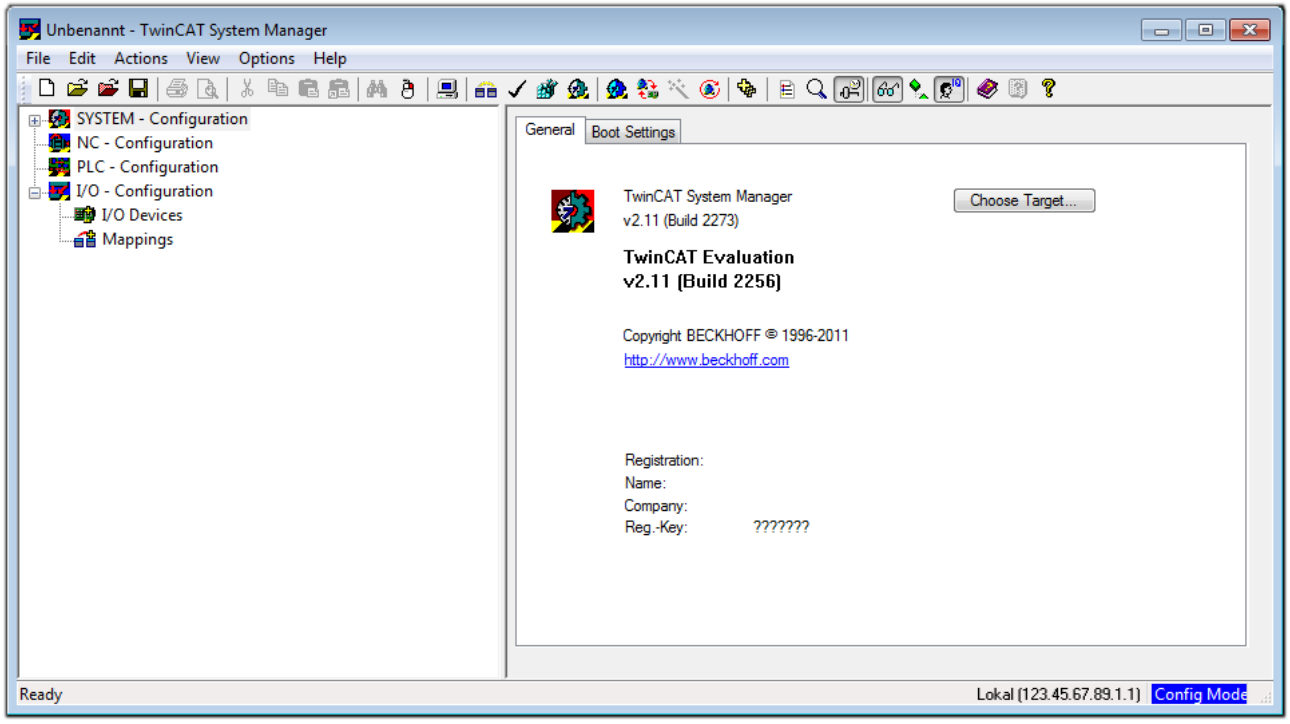


Fig. 41: Initial TwinCAT 2 user interface

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC, TwinCAT can be used in local mode and thereby the next step is “Insert Device [▶ 49]”.

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

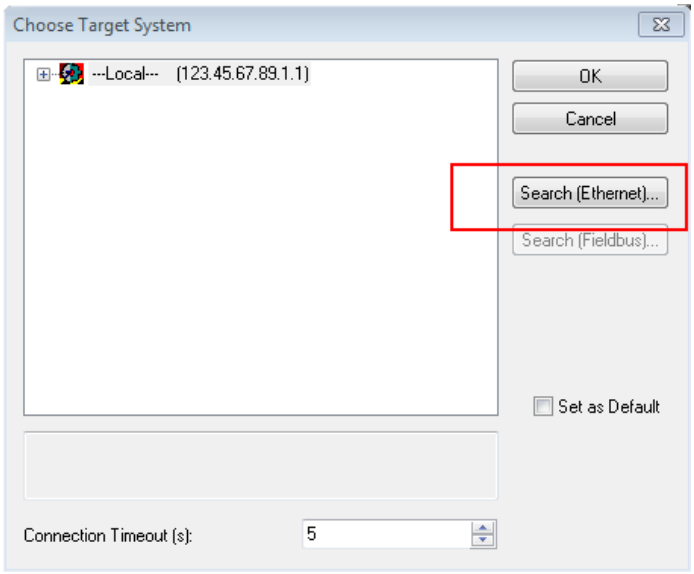


Fig. 42: Selection of the target system

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

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

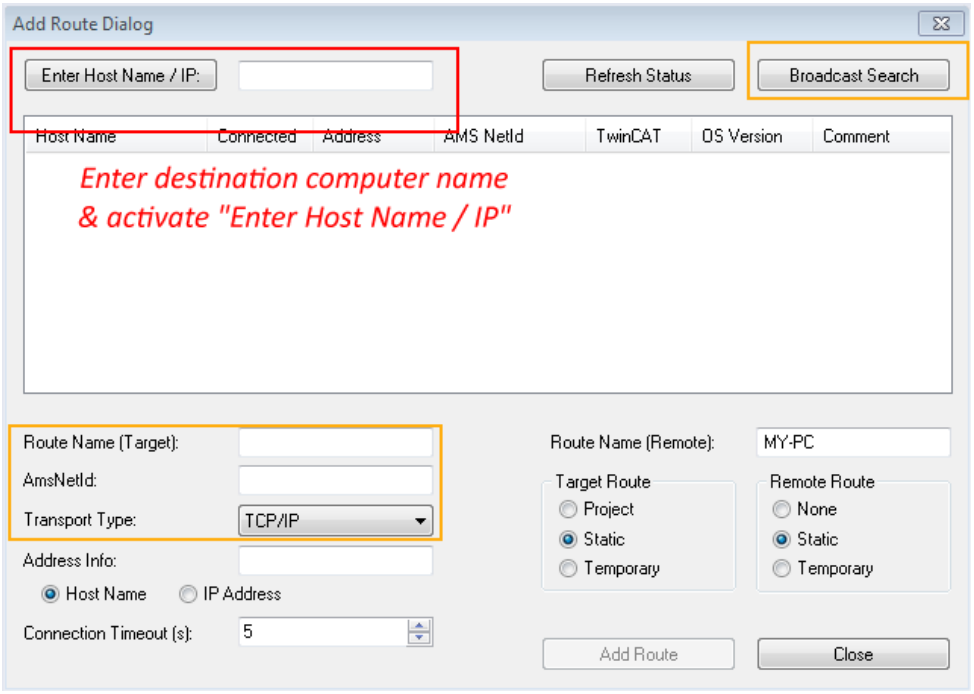
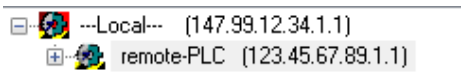


Fig. 43: Specify the PLC for access by the TwinCAT System Manager: selection of the target system

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



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

Adding devices

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

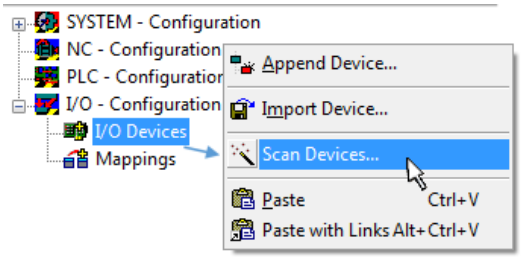


Fig. 44: Select “Scan Devices...”

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

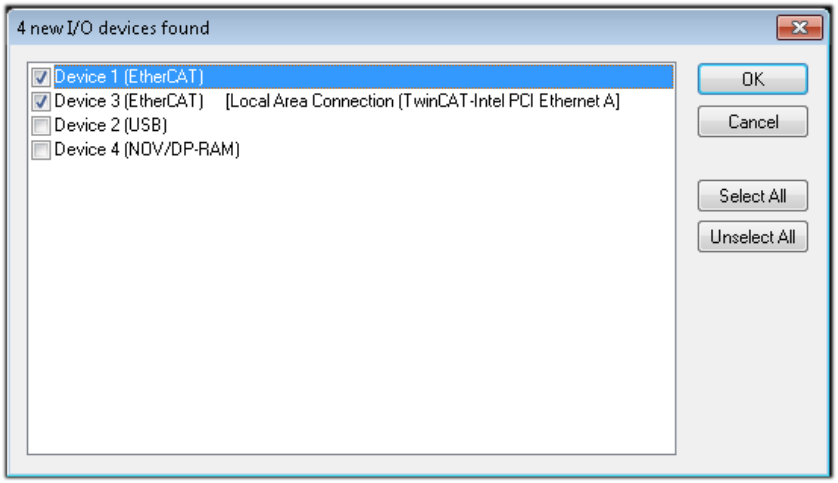


Fig. 45: Automatic detection of I/O devices: selection the devices to be integrated

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

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

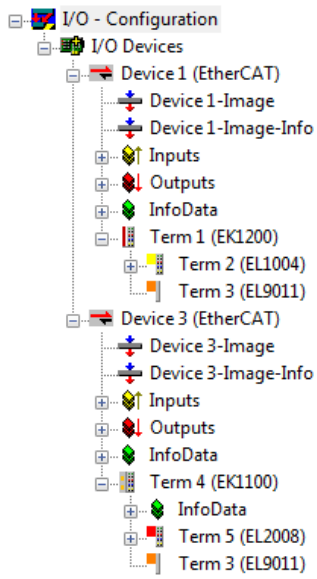


Fig. 46: Mapping of the configuration in the TwinCAT 2 System Manager

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

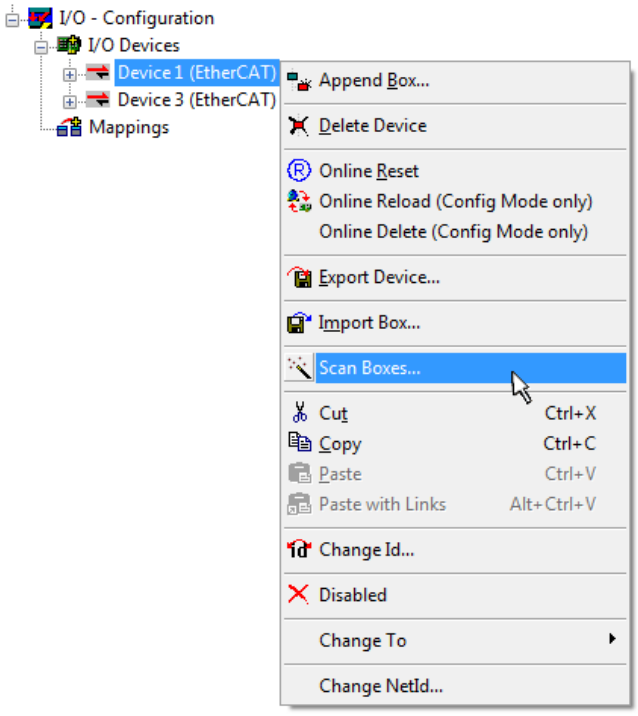


Fig. 47: Reading of individual terminals connected to a device

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

Programming and integrating the PLC

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

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

- Structured Text (ST)

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

The following section refers to Structured Text (ST).

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

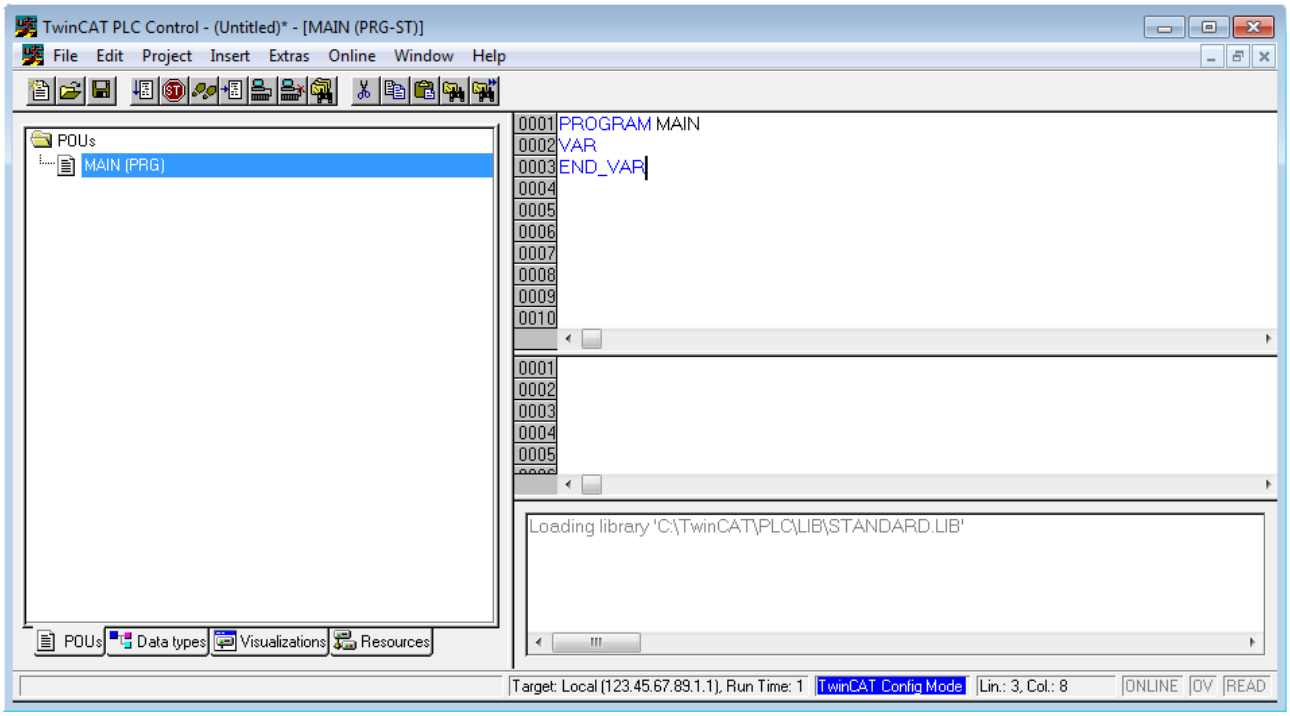


Fig. 48: TwinCAT PLC Control after startup

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

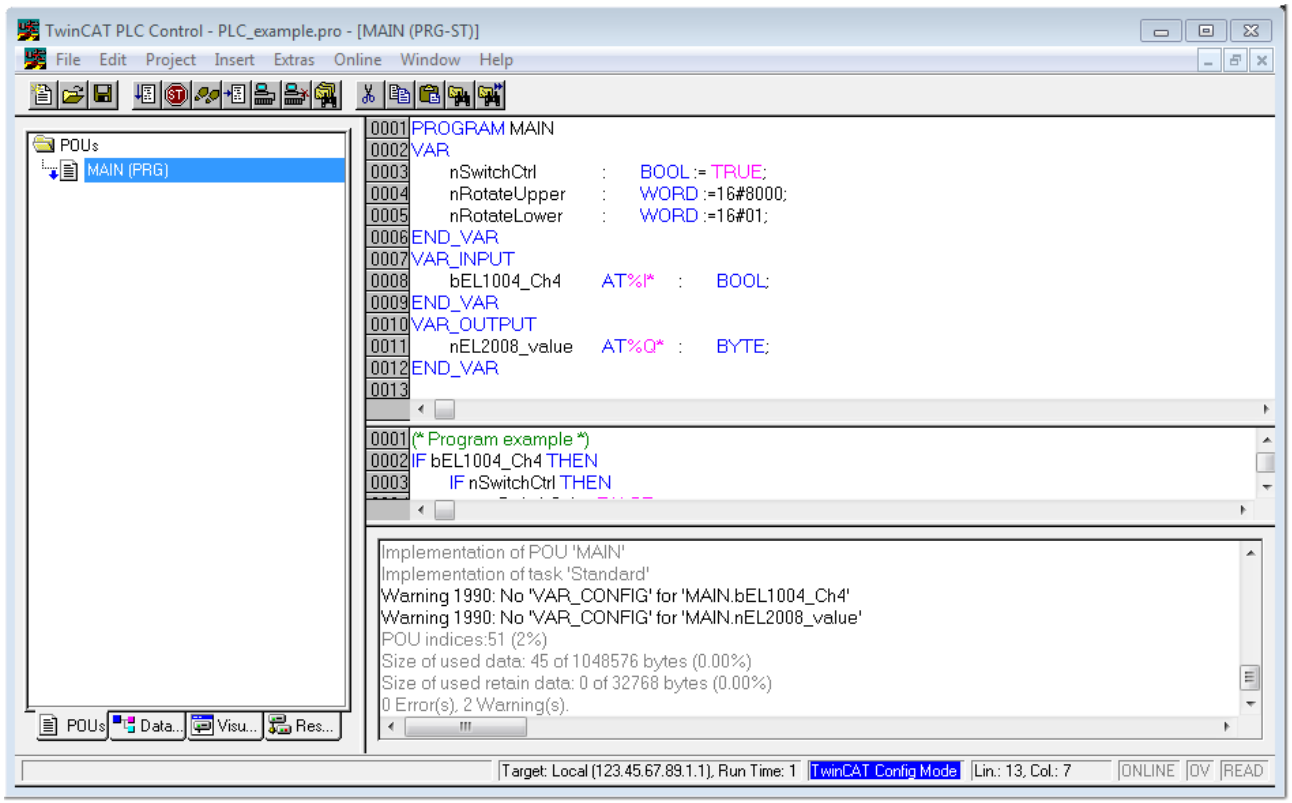


Fig. 49: Sample program with variables after a compile process (without variable integration)

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

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

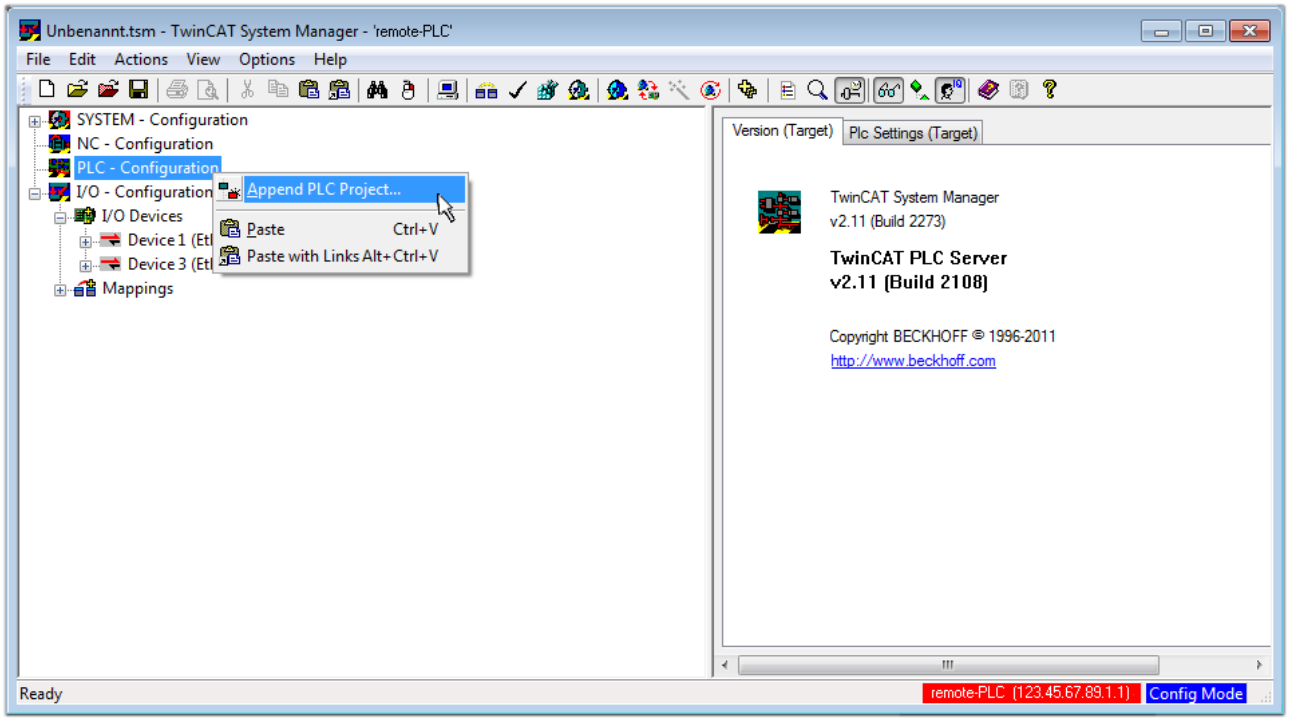


Fig. 50: Appending the TwinCAT PLC Control project

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

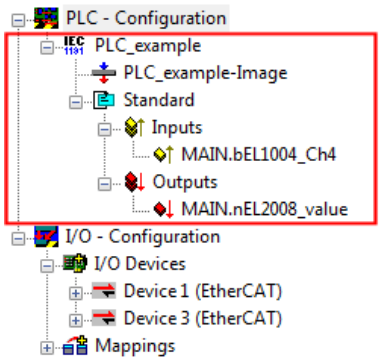


Fig. 51: PLC project integrated in the PLC configuration of the System Manager

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

Assigning variables

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

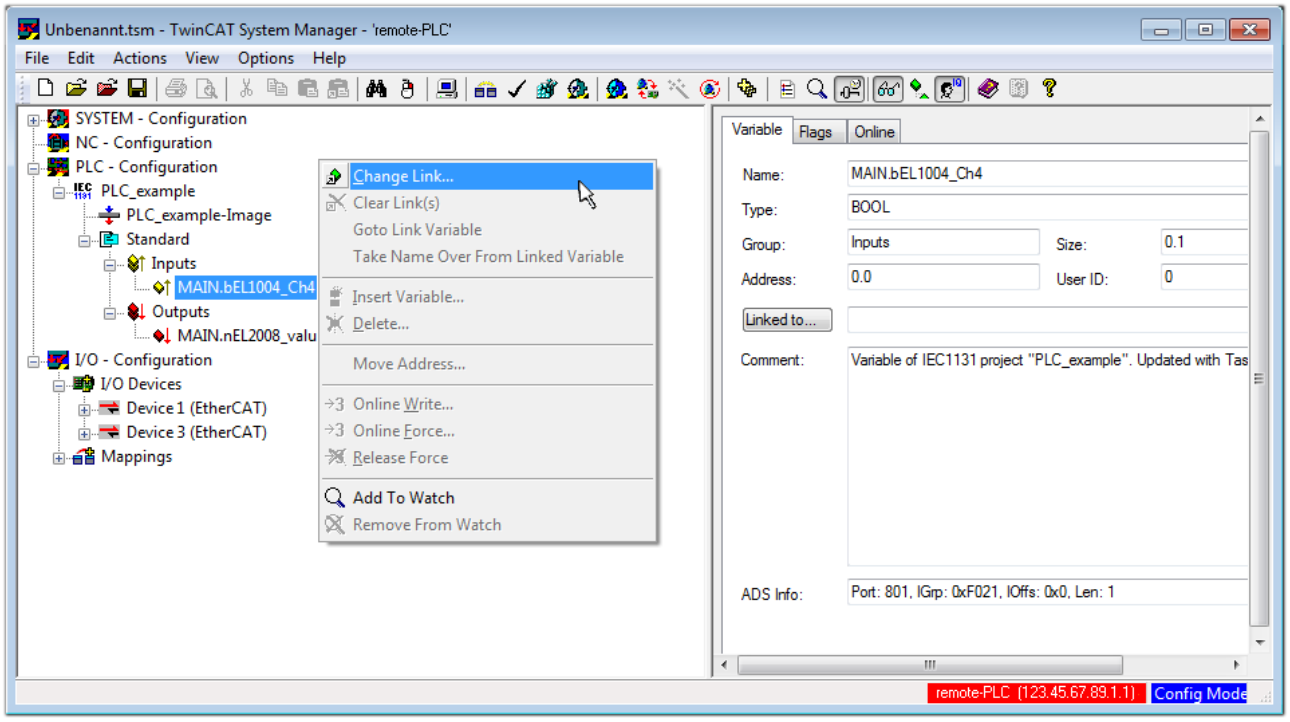


Fig. 52: Creating the links between PLC variables and process objects

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

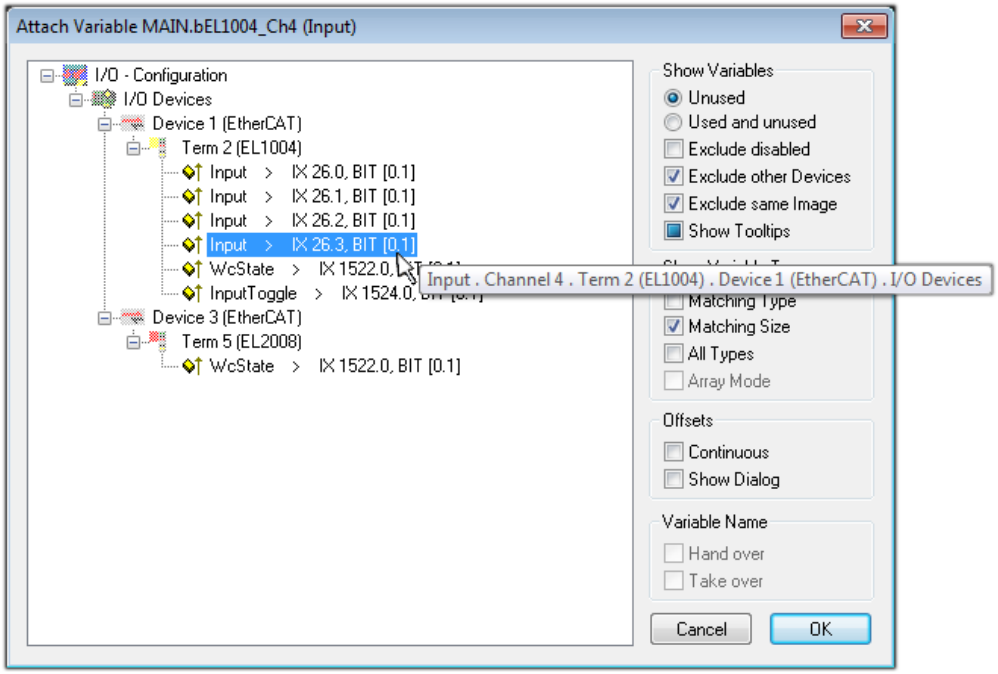


Fig. 53: Selecting PDO of type BOOL

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

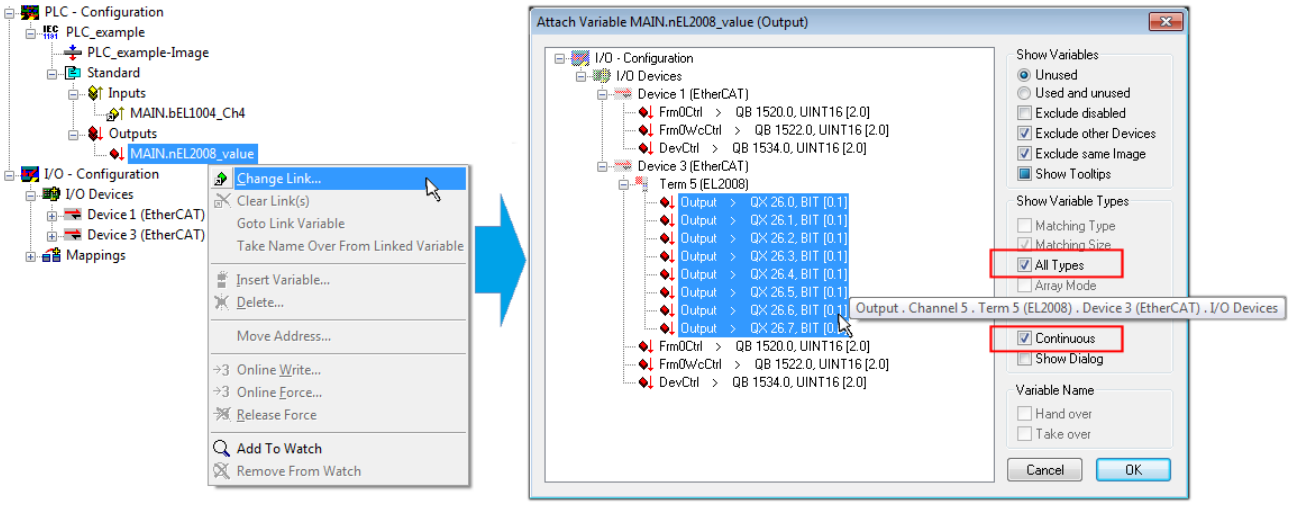


Fig. 54: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

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

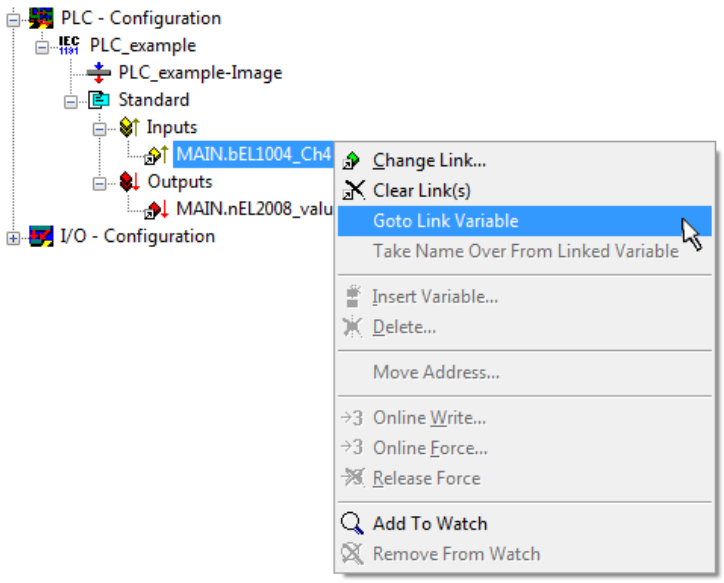
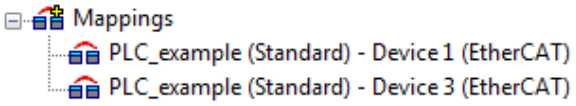


Fig. 55: Application of a “Goto Link” variable, using “MAIN.bEL1004_Ch4” as a sample

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

This can be visualized in the configuration:



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

Activation of the configuration

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

A few seconds later the real-time status 

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

Starting the controller

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

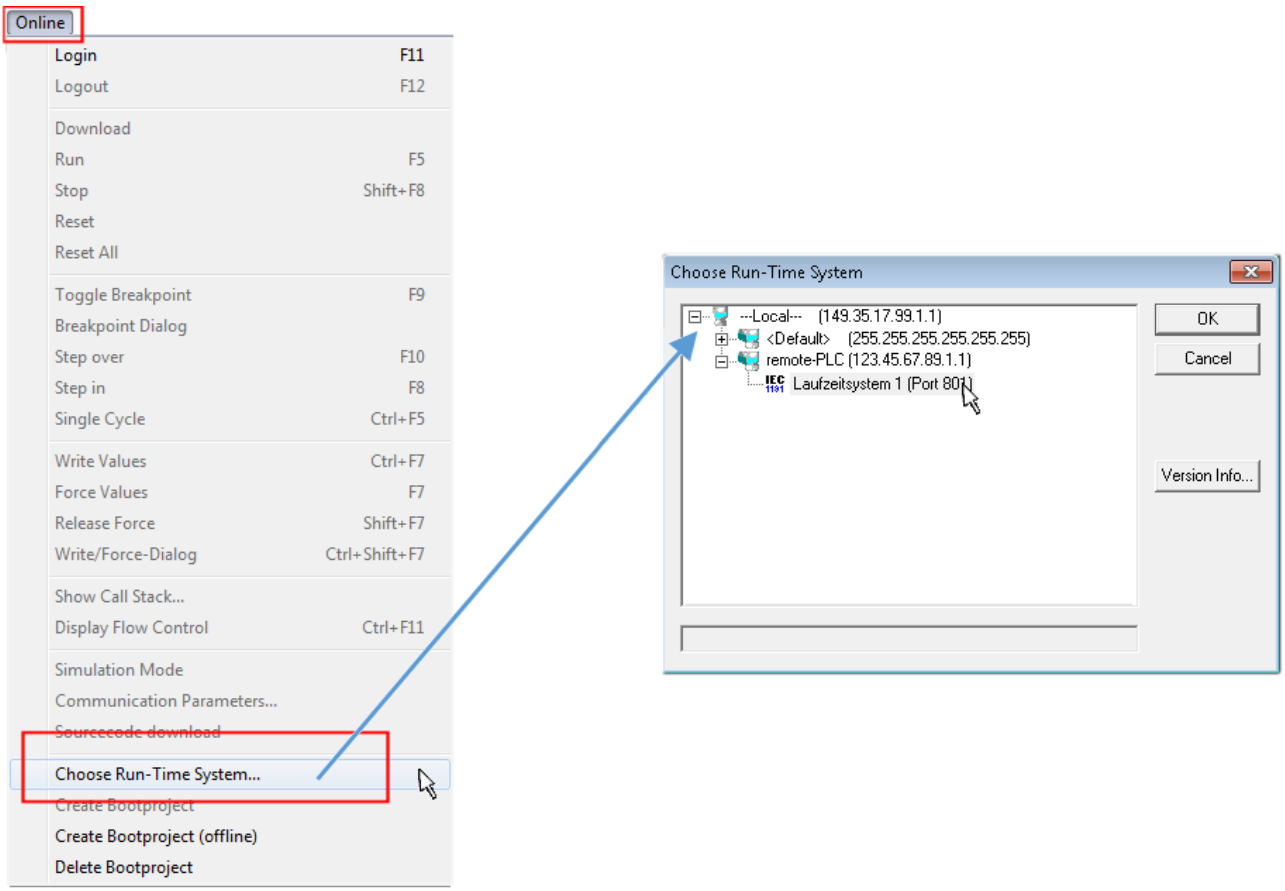


Fig. 56: Choose target system (remote)

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

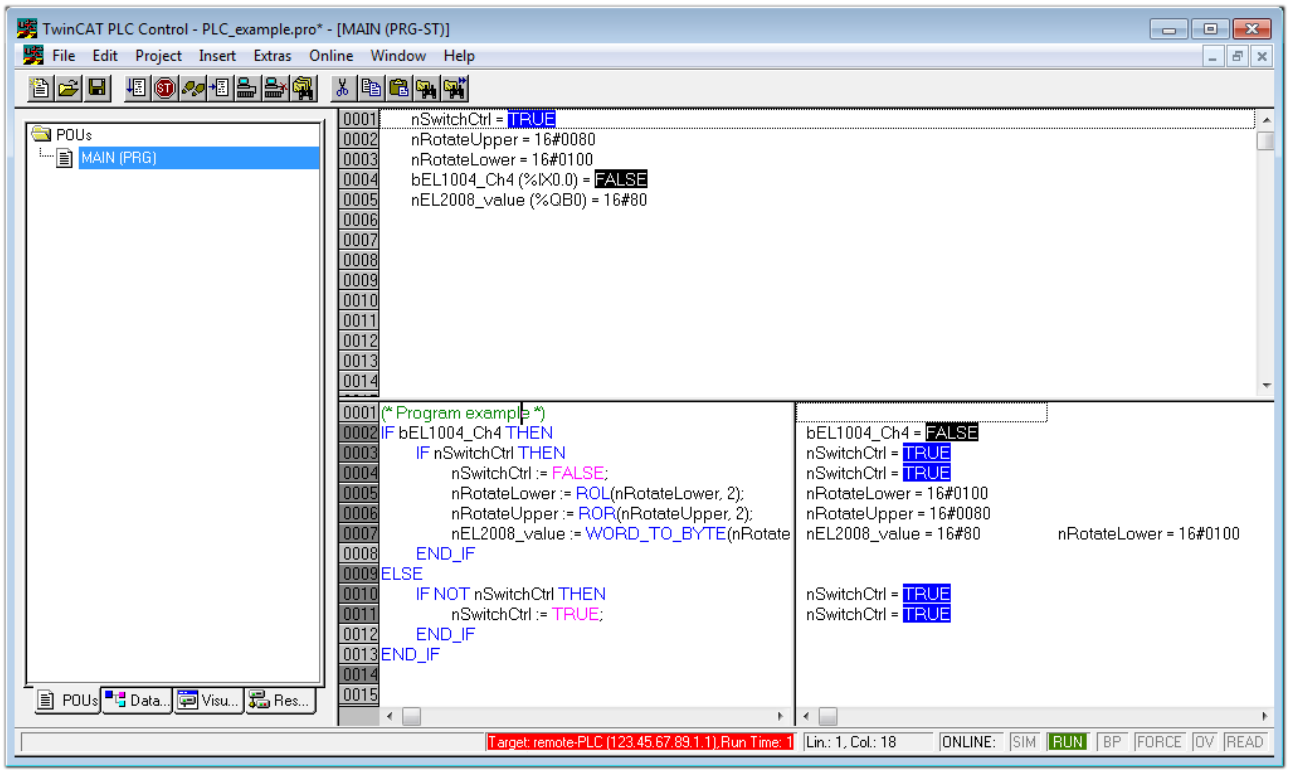


Fig. 57: PLC Control logged in, ready for program startup

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

5.1.2 TwinCAT 3

Startup

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

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

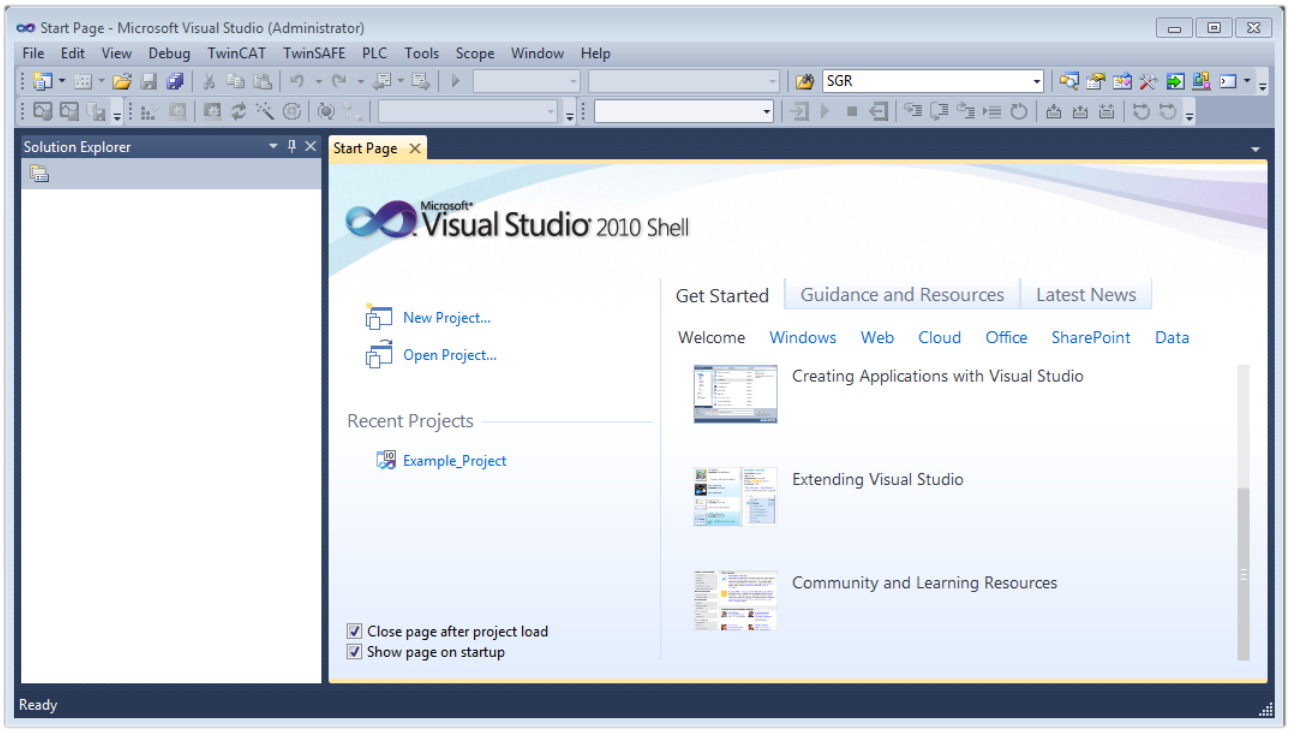


Fig. 58: Initial TwinCAT 3 user interface

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

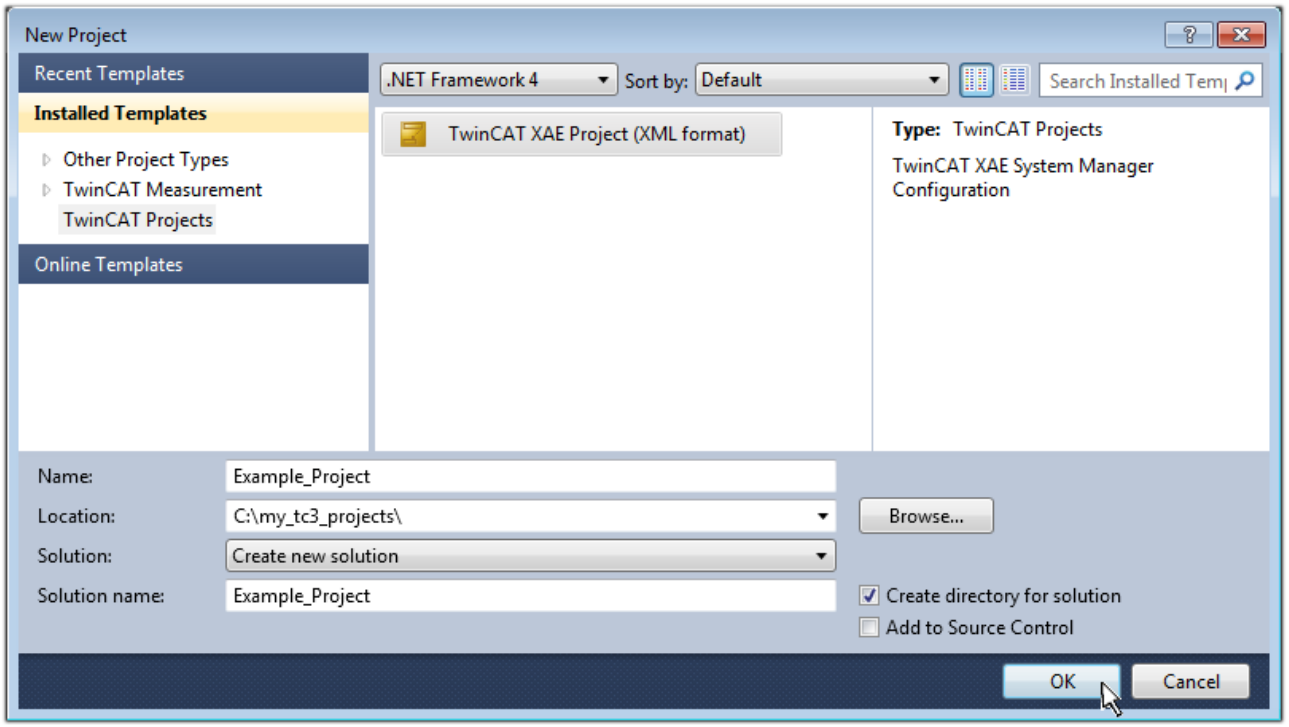


Fig. 59: Create new TwinCAT project

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

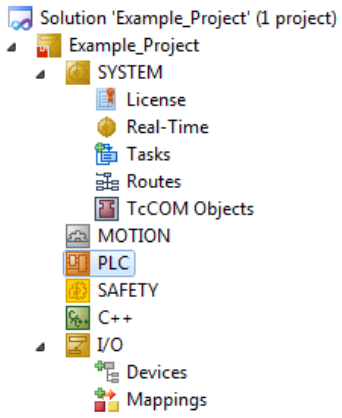
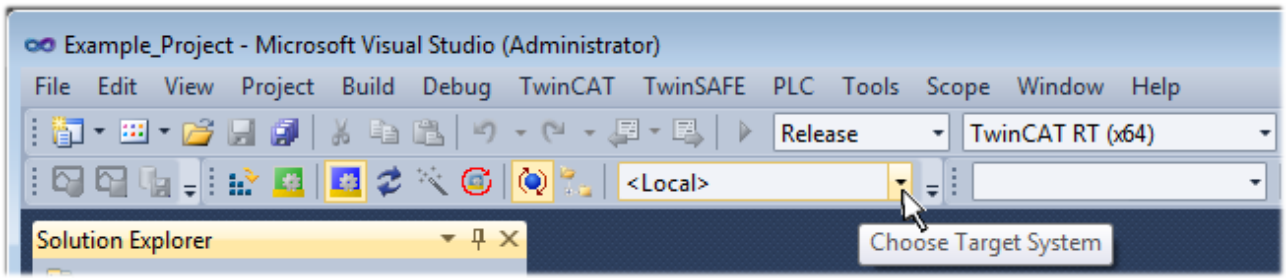


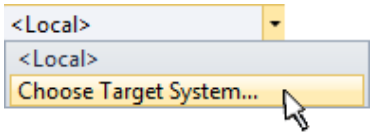
Fig. 60: New TwinCAT3 project in the project folder explorer

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC, TwinCAT can be used in local mode and thereby the next step is "Insert Device [▶ 60]".

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



expand the pull-down menu:



and open the following window:

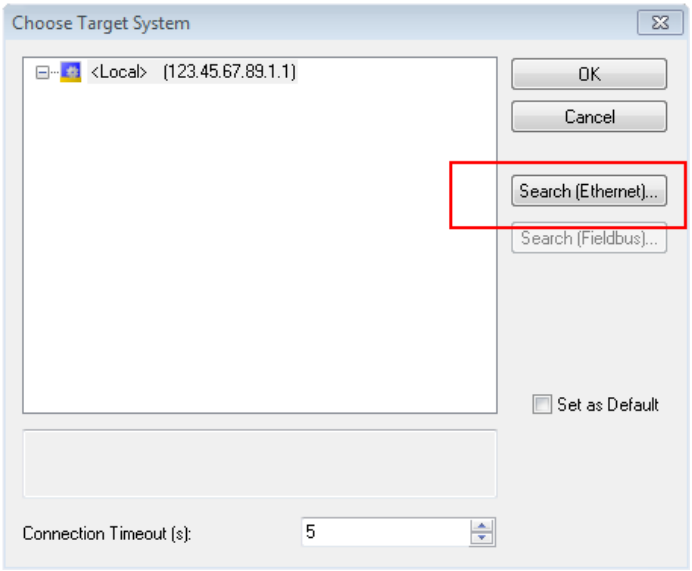


Fig. 61: Selection dialog: Choose the target system

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

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

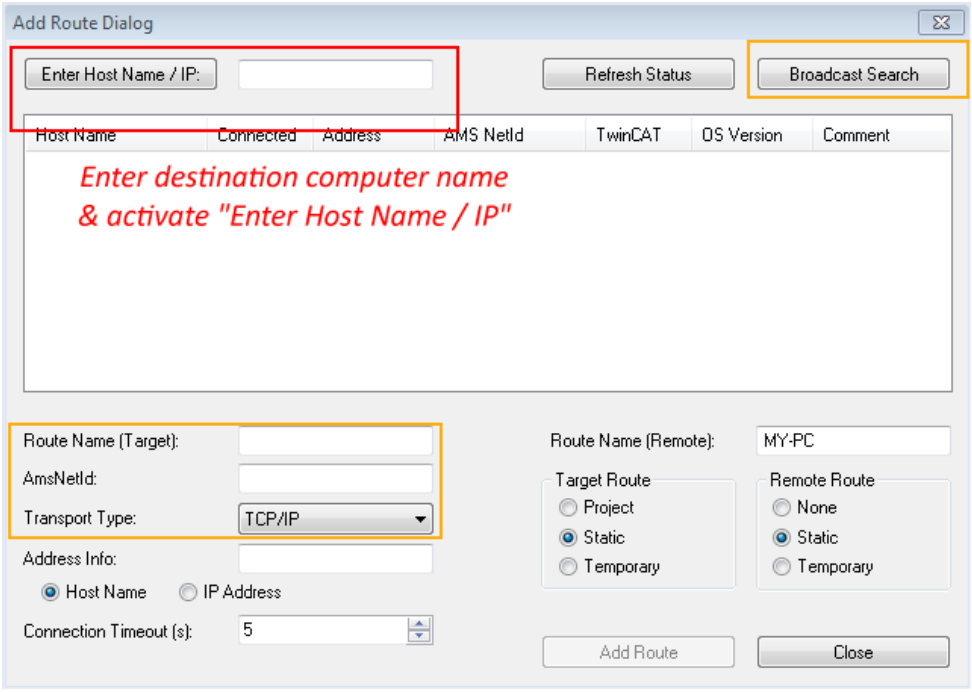
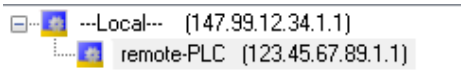


Fig. 62: Specify the PLC for access by the TwinCAT System Manager: selection of the target system

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



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

Adding devices

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

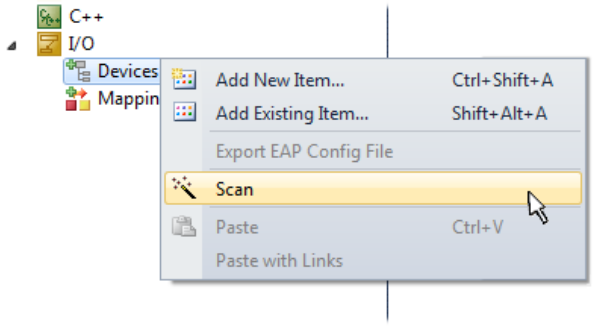


Fig. 63: Select “Scan”

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

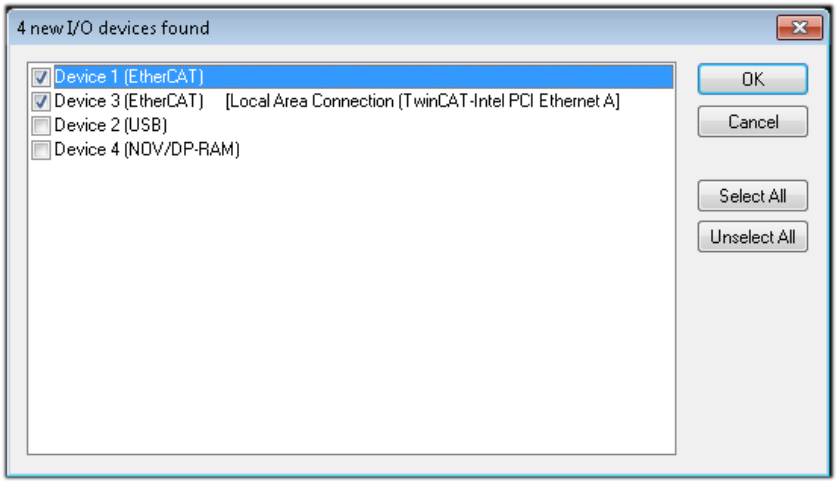


Fig. 64: Automatic detection of I/O devices: selection the devices to be integrated

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

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

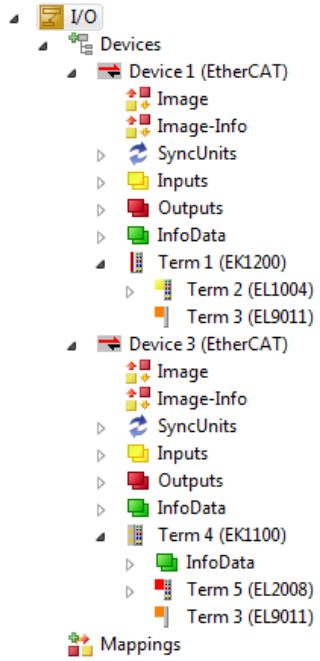


Fig. 65: Mapping of the configuration in VS shell of the TwinCAT3 environment

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

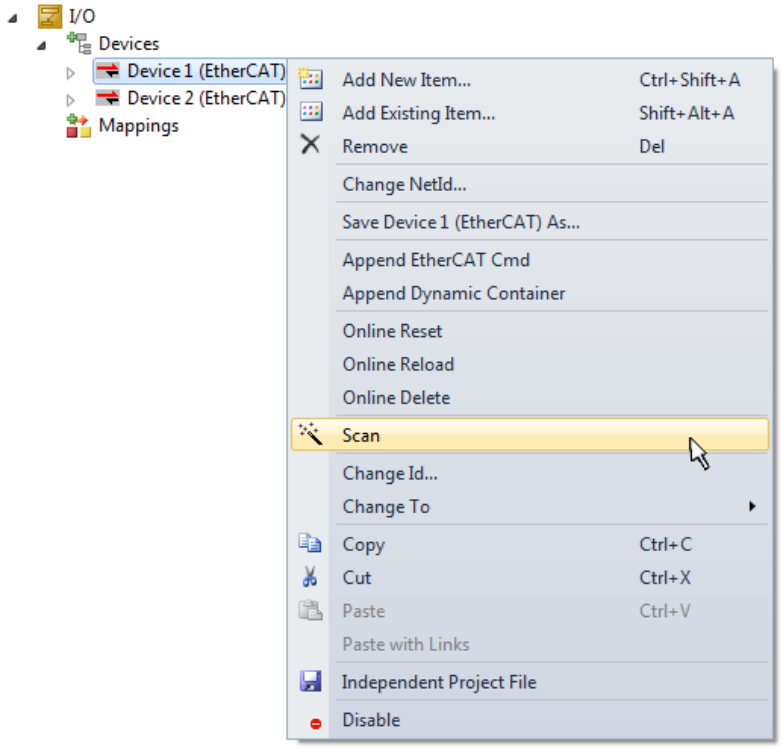


Fig. 66: Reading of individual terminals connected to a device

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

Programming the PLC

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

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

The following section refers to Structured Text (ST).

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

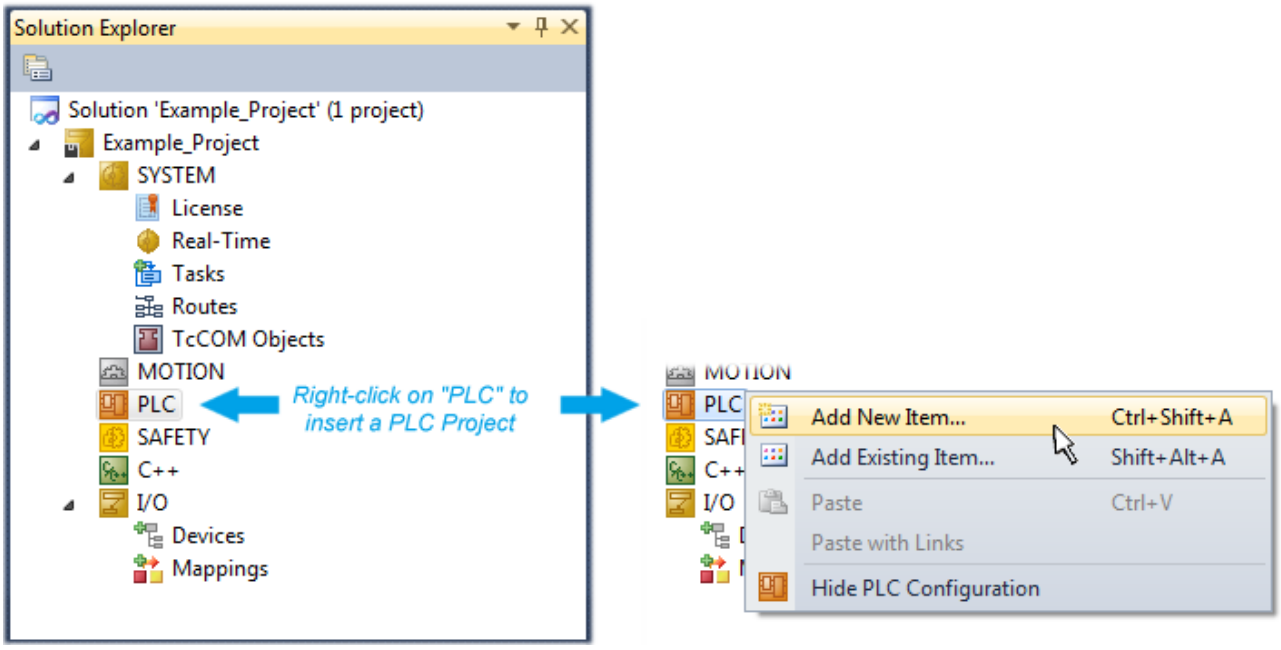


Fig. 67: Adding the programming environment in “PLC”

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

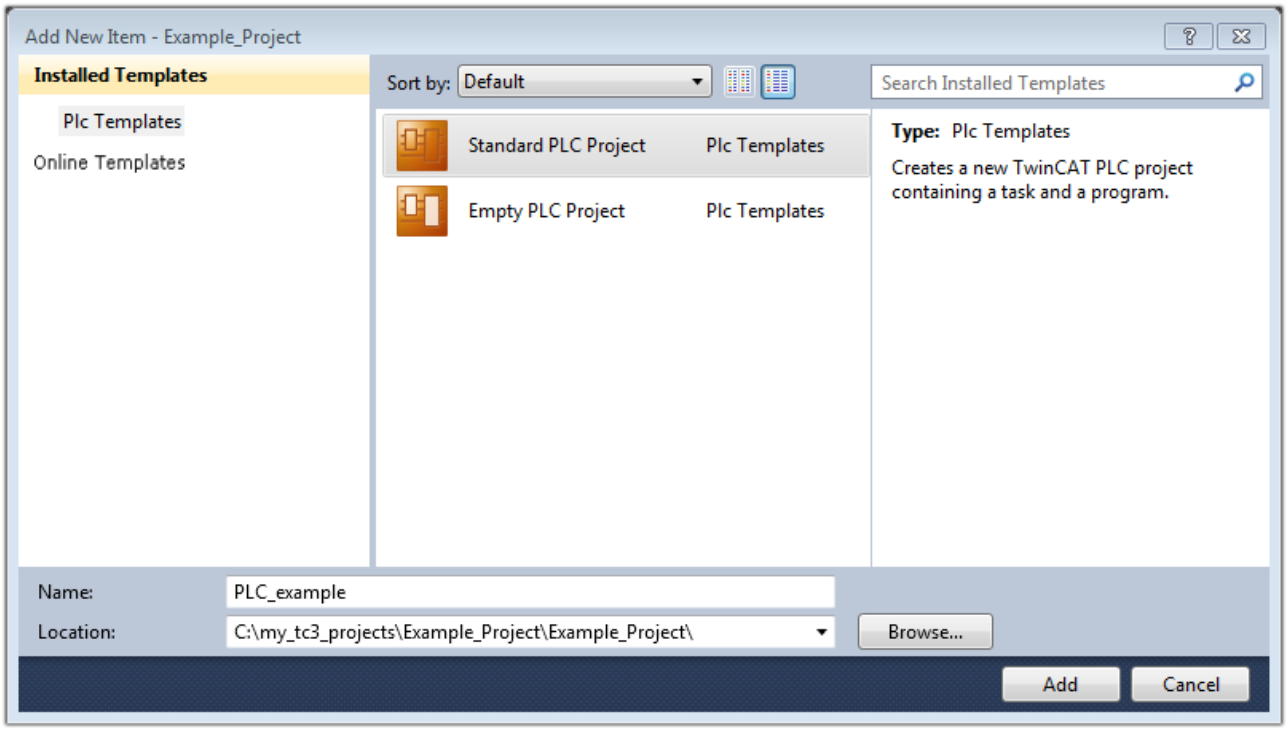


Fig. 68: Specifying the name and directory for the PLC programming environment

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

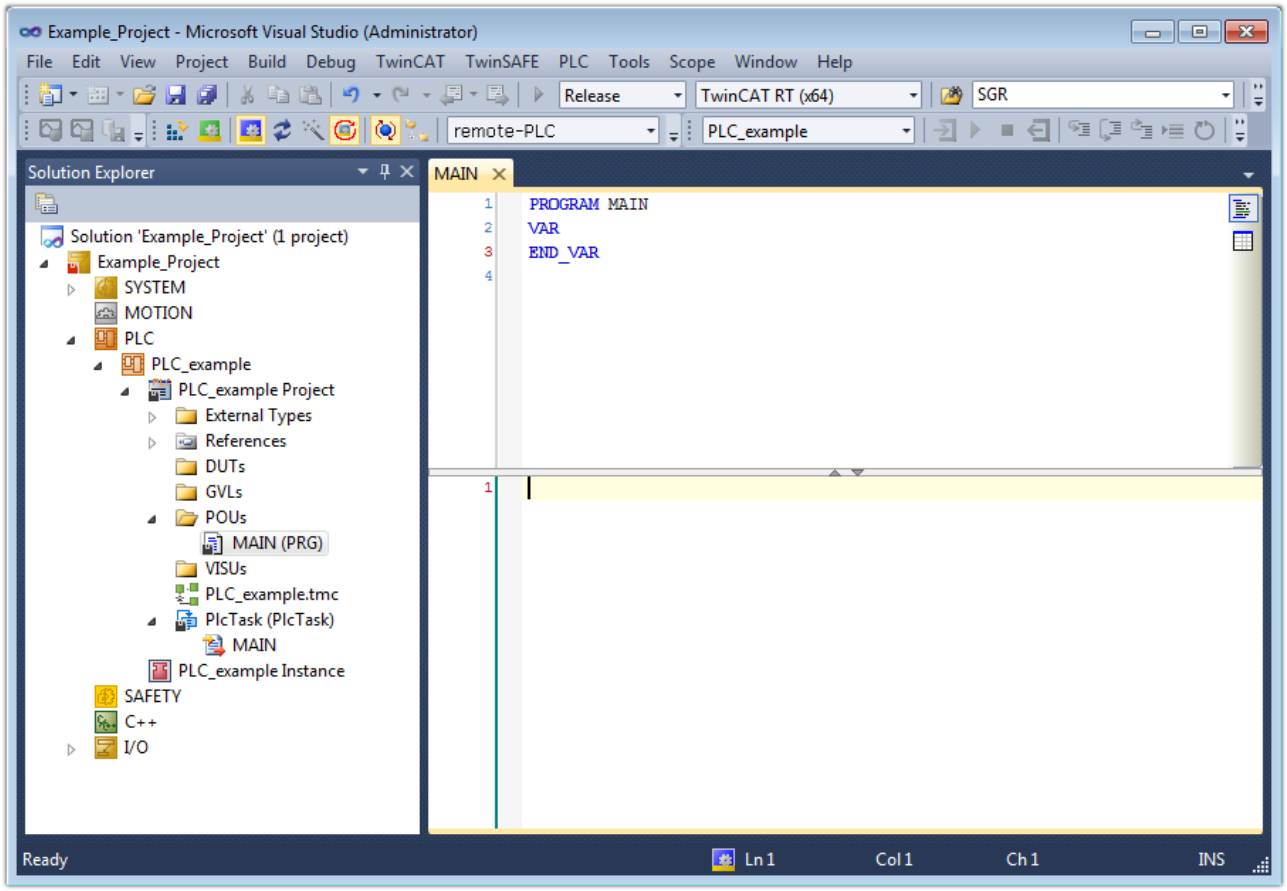


Fig. 69: Initial “Main” program of the standard PLC project

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

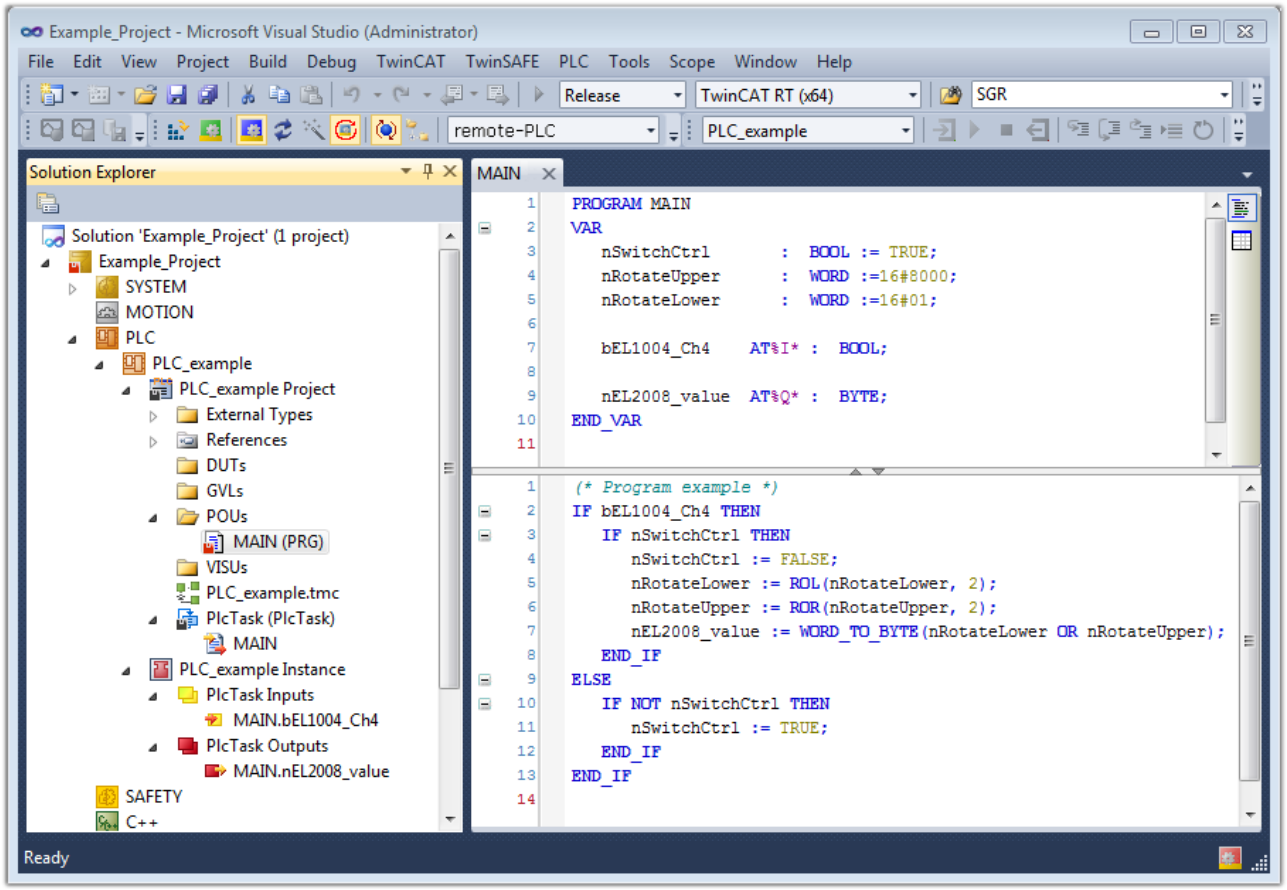


Fig. 70: Sample program with variables after a compile process (without variable integration)

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

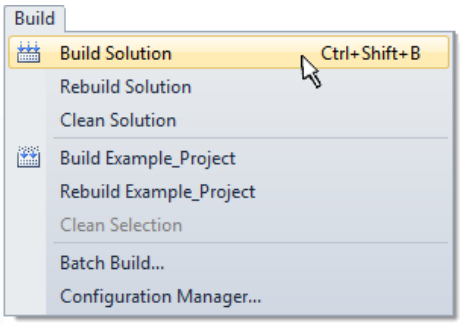
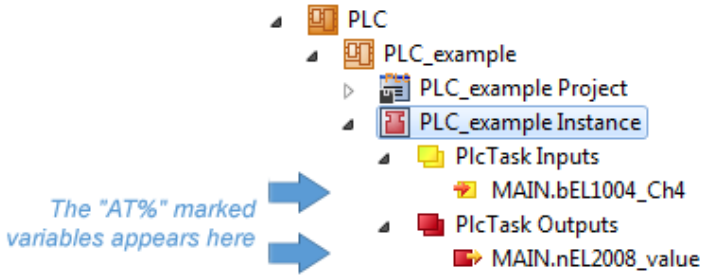


Fig. 71: Start program compilation

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



Assigning variables

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

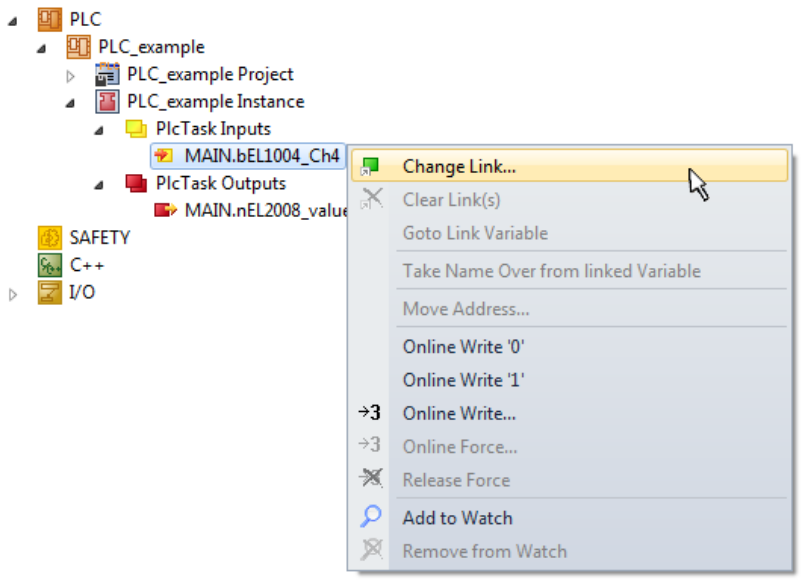


Fig. 72: Creating the links between PLC variables and process objects

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

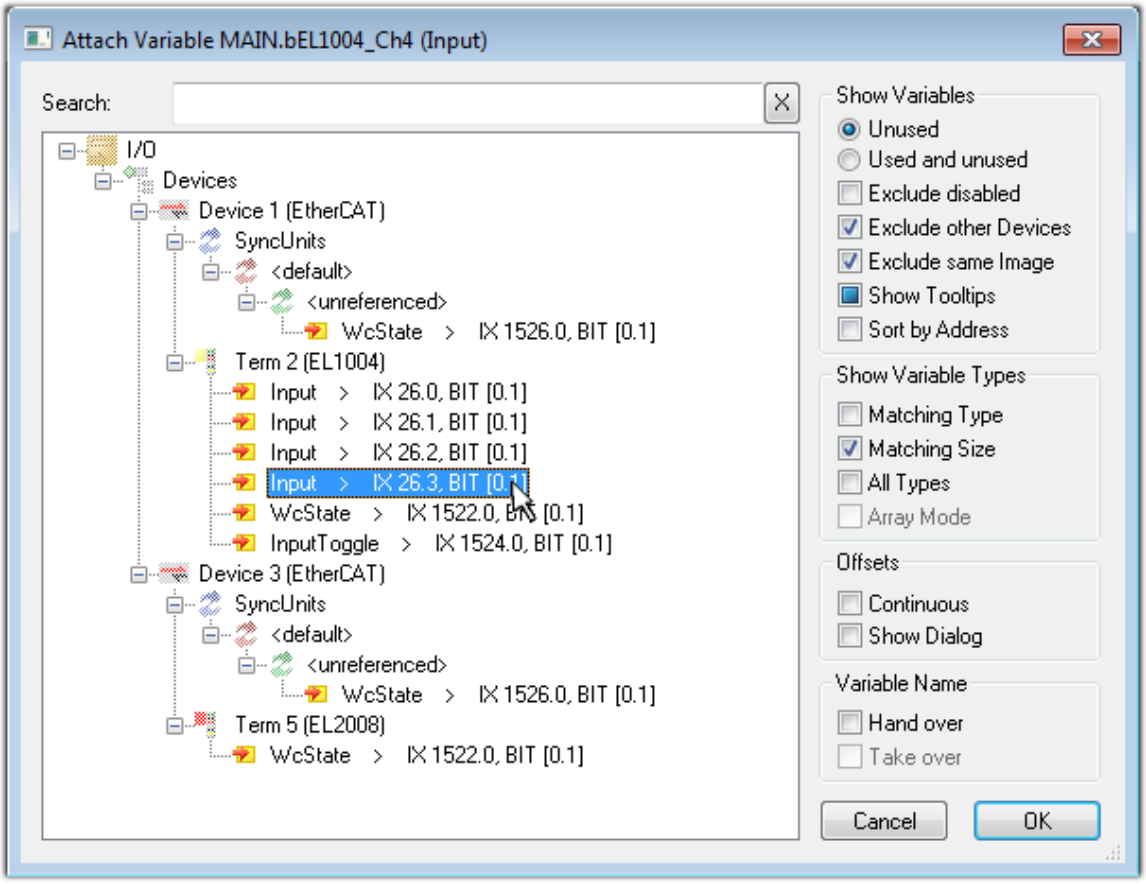


Fig. 73: Selecting PDO of type BOOL

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

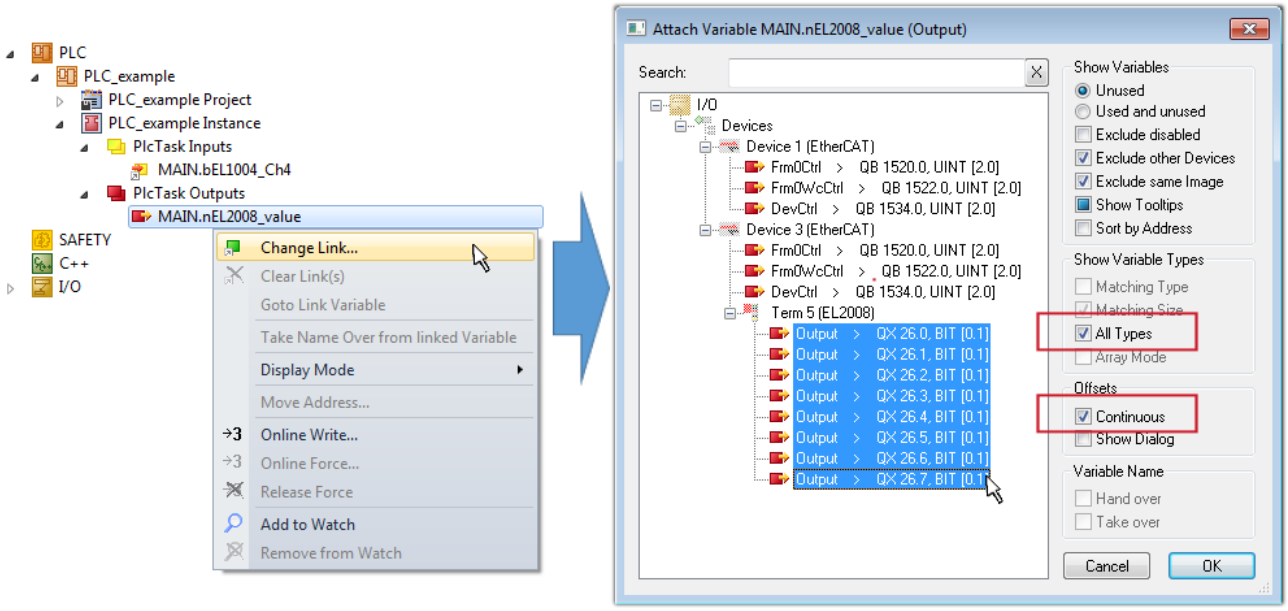


Fig. 74: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

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

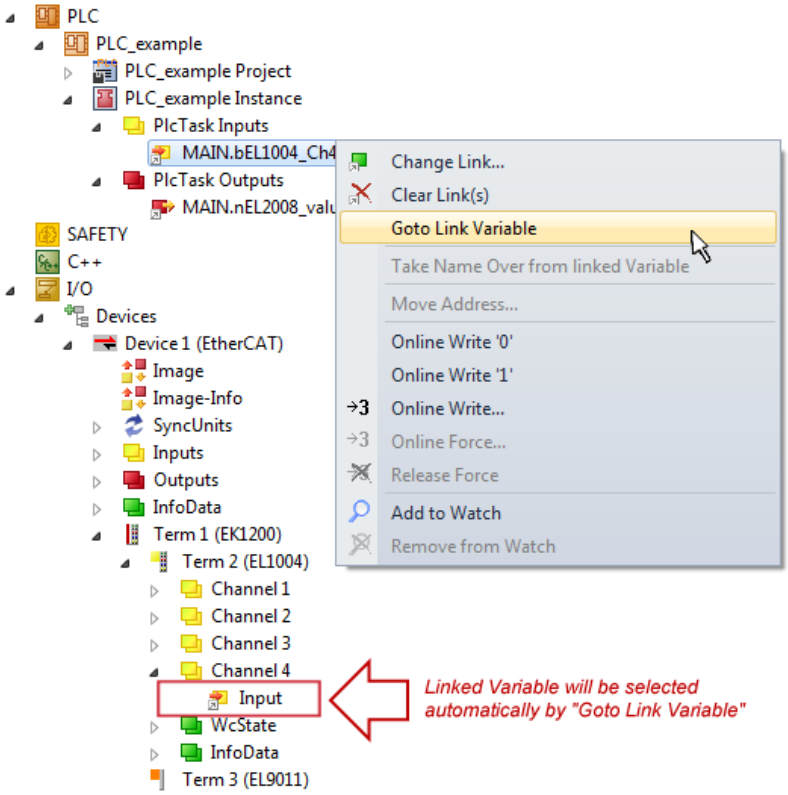


Fig. 75: Application of a “Goto Link” variable, using “MAIN.bEL1004_Ch4” as a sample

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

similar PDO, it is possible to allocate this a set of bit-standardized variables (type “BOOL”). Here, too, a “Goto Link Variable” from the context menu of a PDO can be executed in the other direction, so that the respective PLC instance can then be selected.

Note on the type of variable assignment

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

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

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

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

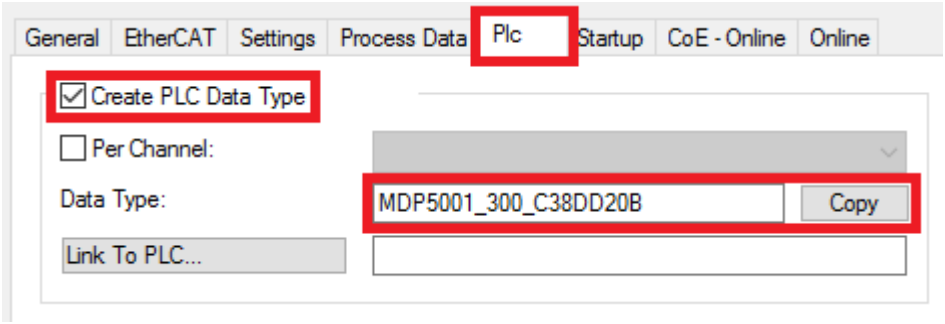


Fig. 76: Creating a PLC data type

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

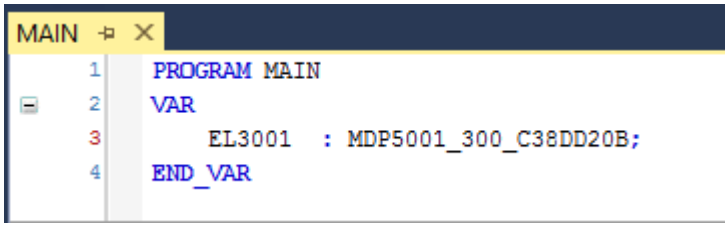


Fig. 77: Instance_of_struct

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

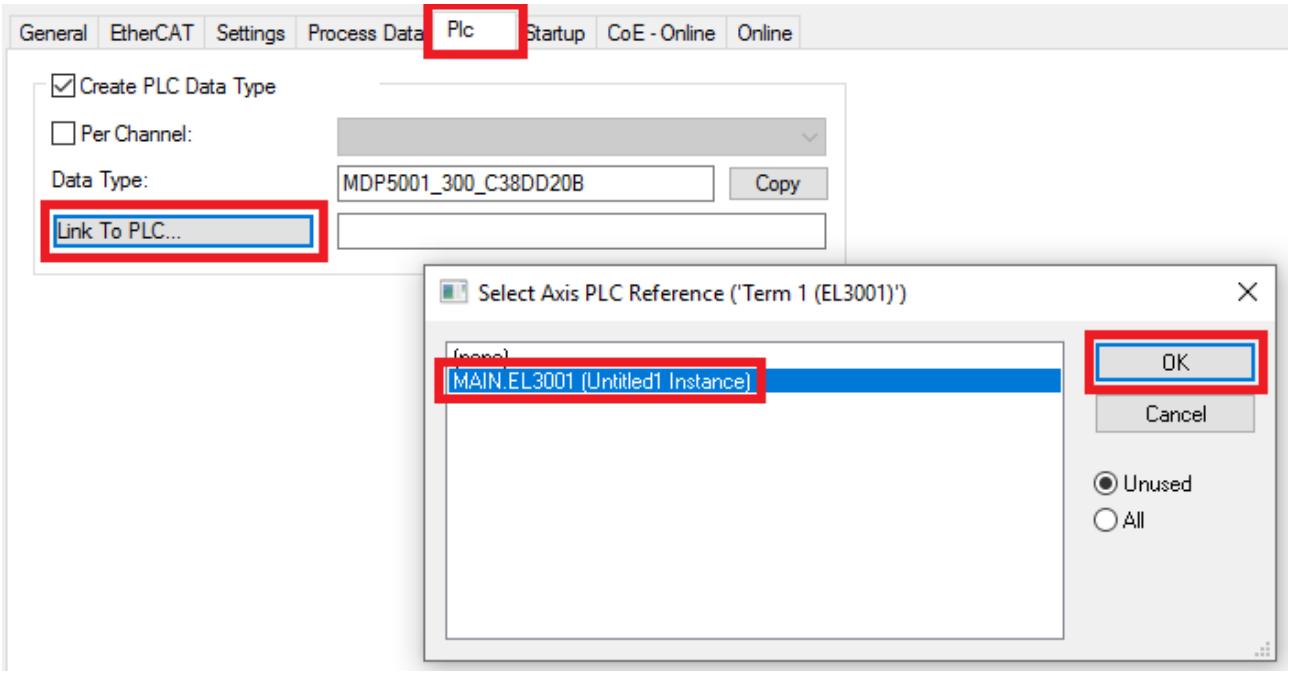


Fig. 78: Linking the structure

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

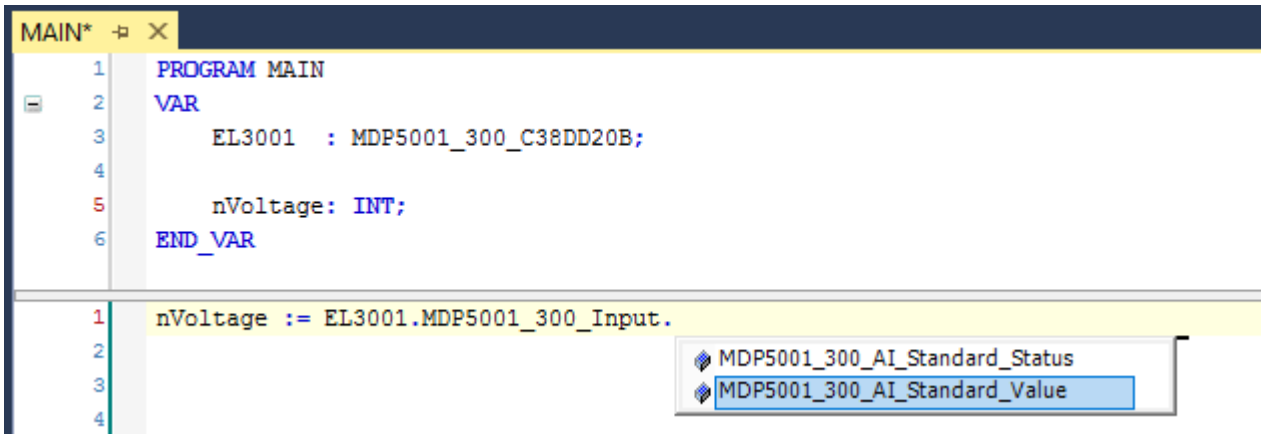
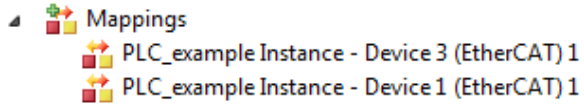


Fig. 79: Reading a variable from the structure of the process data

Activation of the configuration

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



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

Starting the controller

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

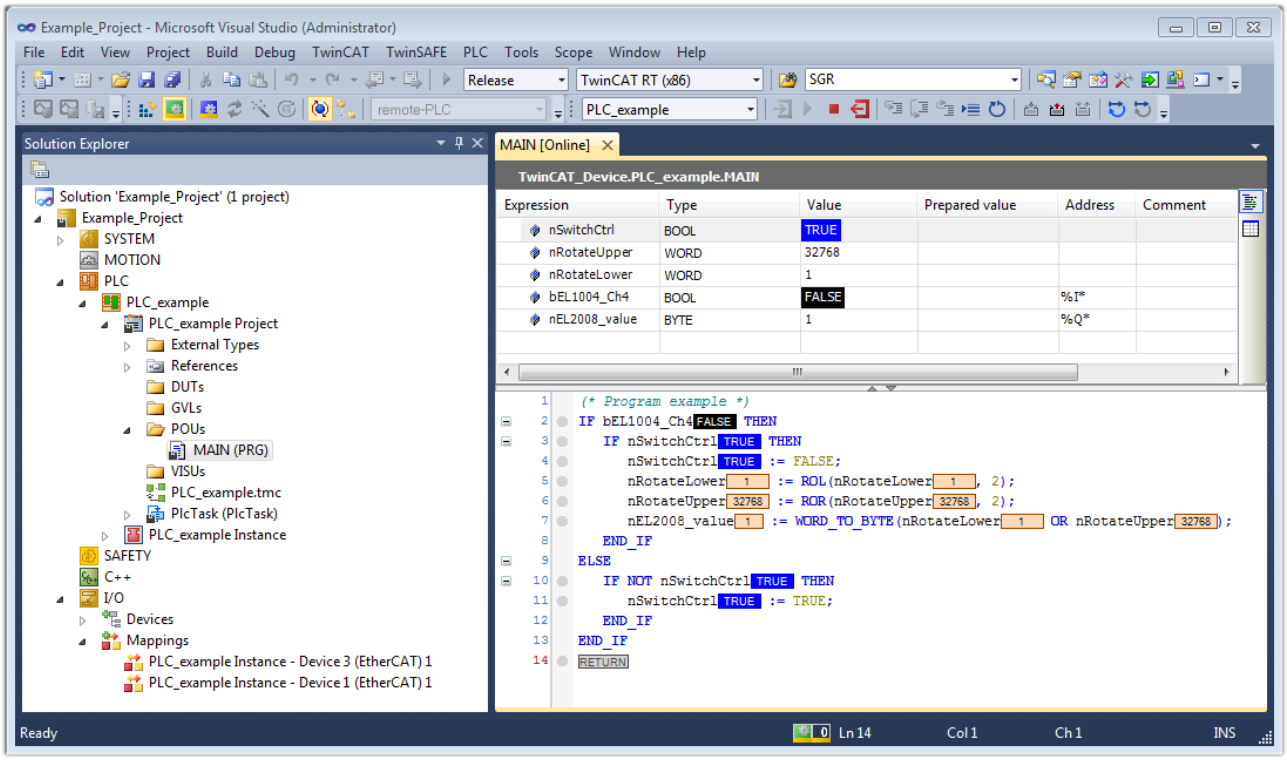


Fig. 80: TwinCAT development environment (VS shell): logged-in, after program startup

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

5.2 TwinCAT Development Environment

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

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

Details:

- **TwinCAT 2:**
  - Connects I/O devices to tasks in a variable-oriented manner
  - Connects tasks to tasks in a variable-oriented manner
  - Supports units at the bit level
  - Supports synchronous or asynchronous relationships
  - Exchange of consistent data areas and process images
  - Datalink on NT - Programs by open Microsoft Standards (OLE, OCX, ActiveX, DCOM+, etc.)

- Integration of IEC 61131-3-Software-SPS, Software- NC and Software-CNC within Windows NT/2000/XP/Vista, Windows 7, NT/XP Embedded, CE
- Interconnection to all common fieldbusses
- More...

Additional features:

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

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

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

5.2.1 Installation of the TwinCAT real-time driver

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

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

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

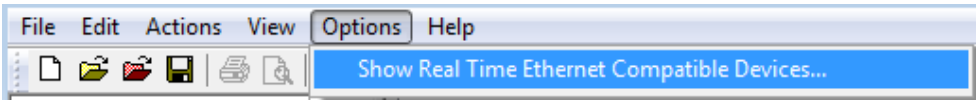


Fig. 81: System Manager “Options” (TwinCAT 2)

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

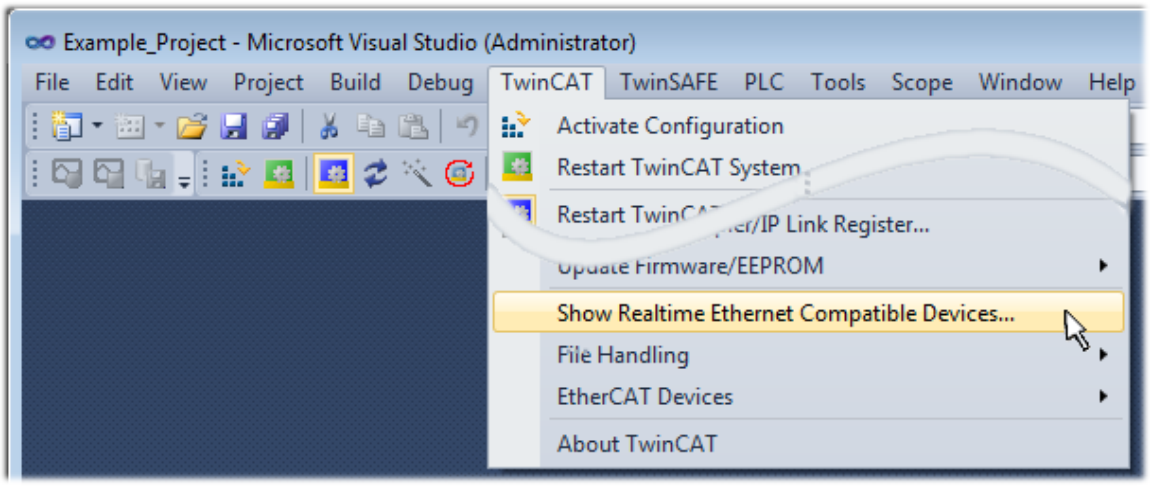


Fig. 82: Call up under VS Shell (TwinCAT 3)

The following dialog appears:

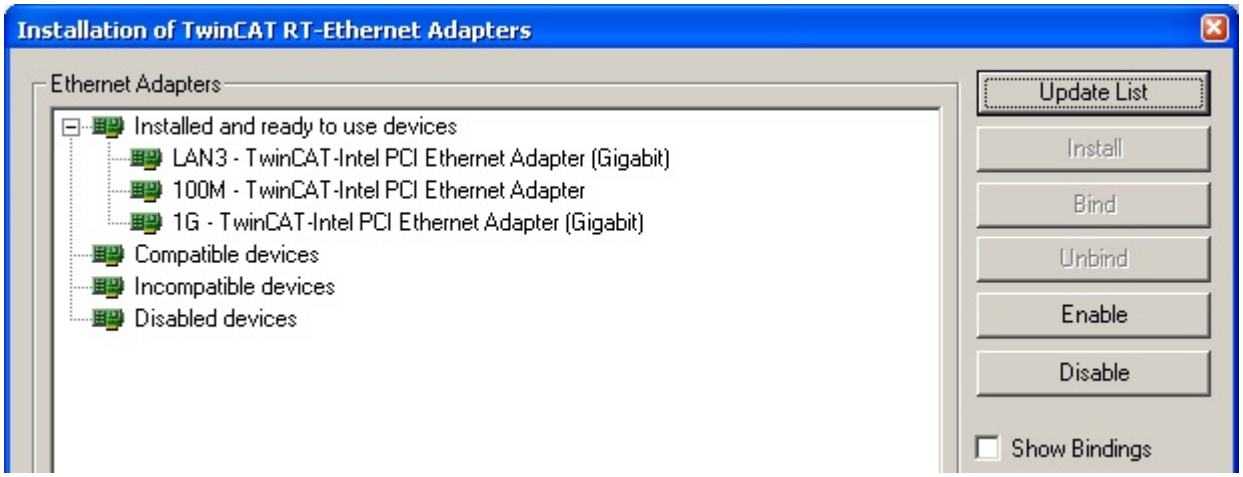


Fig. 83: Overview of network interfaces

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

Fig. 84: EtherCAT device properties(TwinCAT 2): click on “Compatible Devices…” of tab “Adapte””

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)

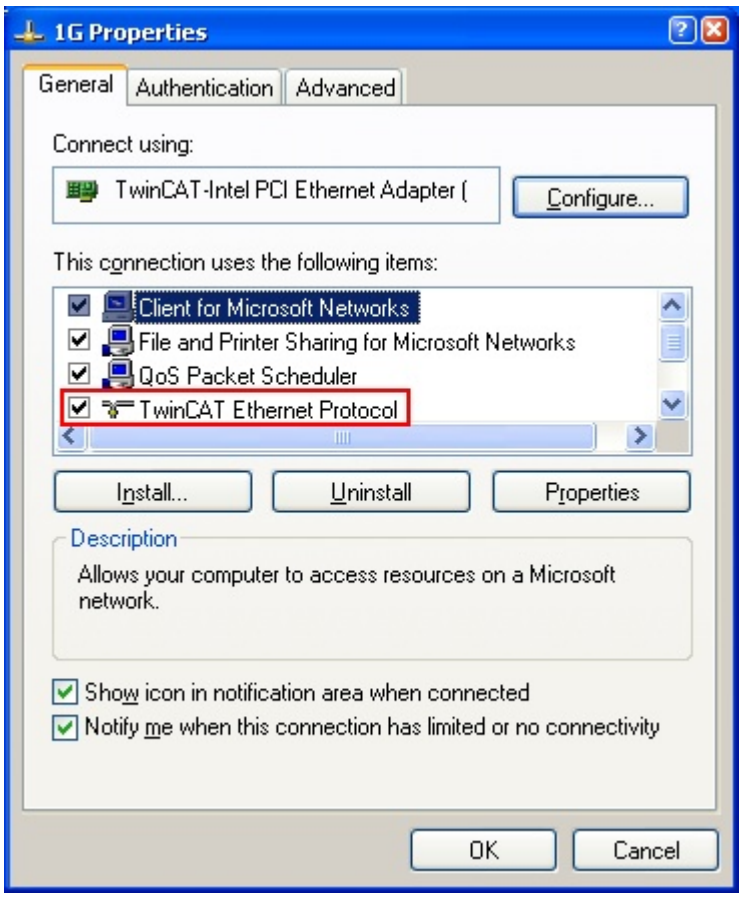


Fig. 85: Windows properties of the network interface

A correct setting of the driver could be:

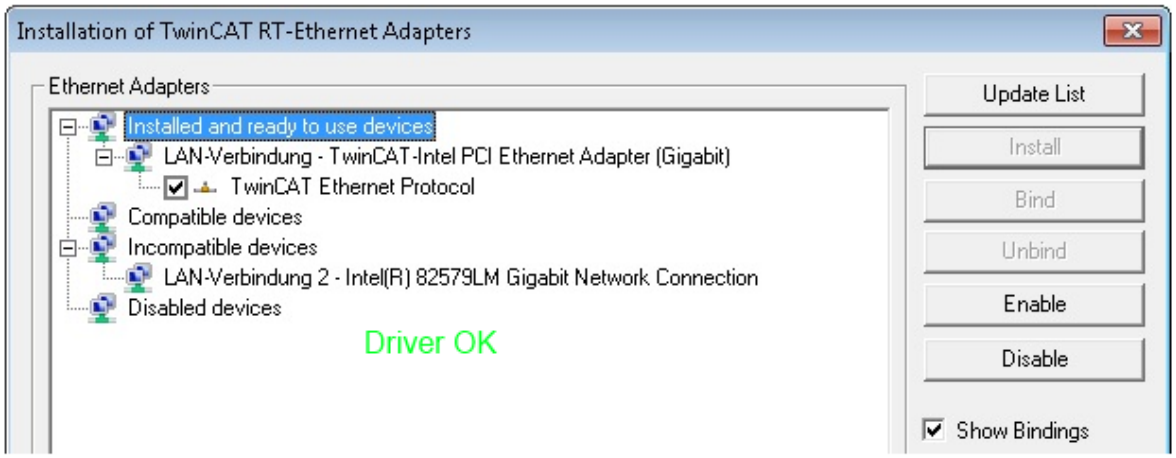
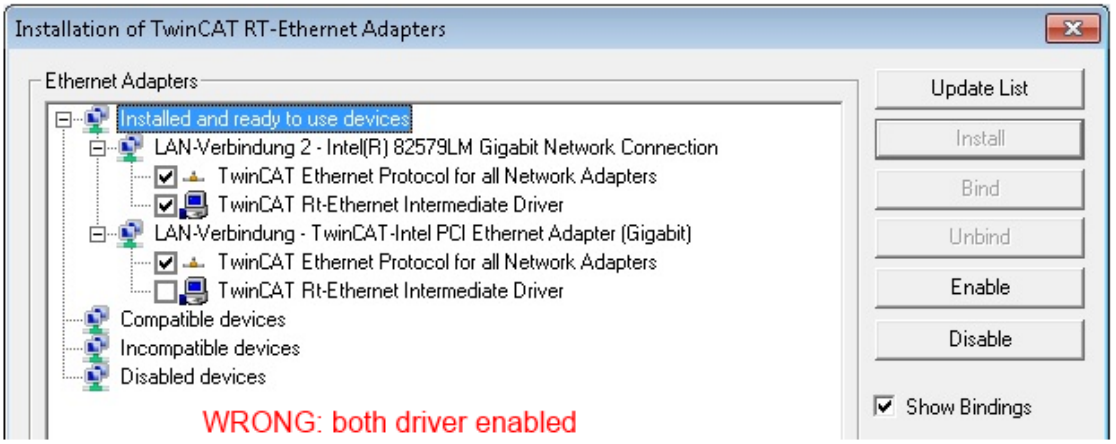
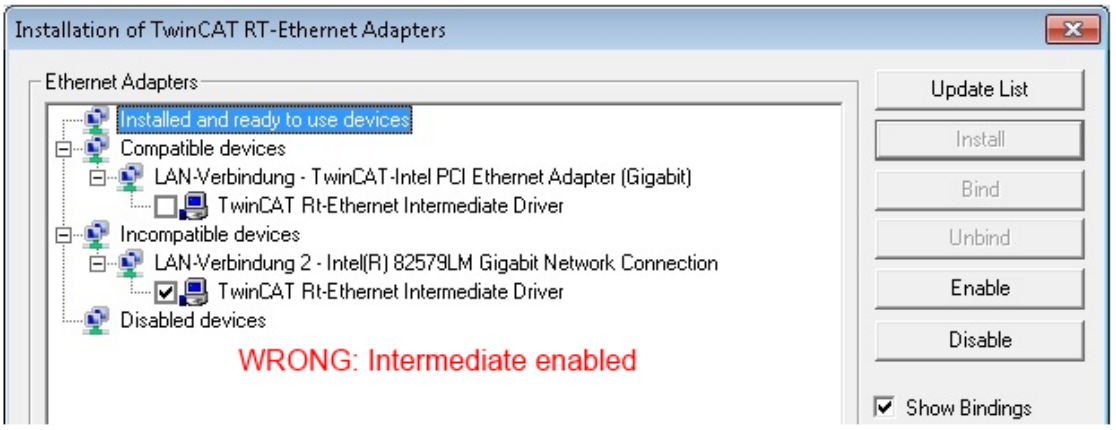
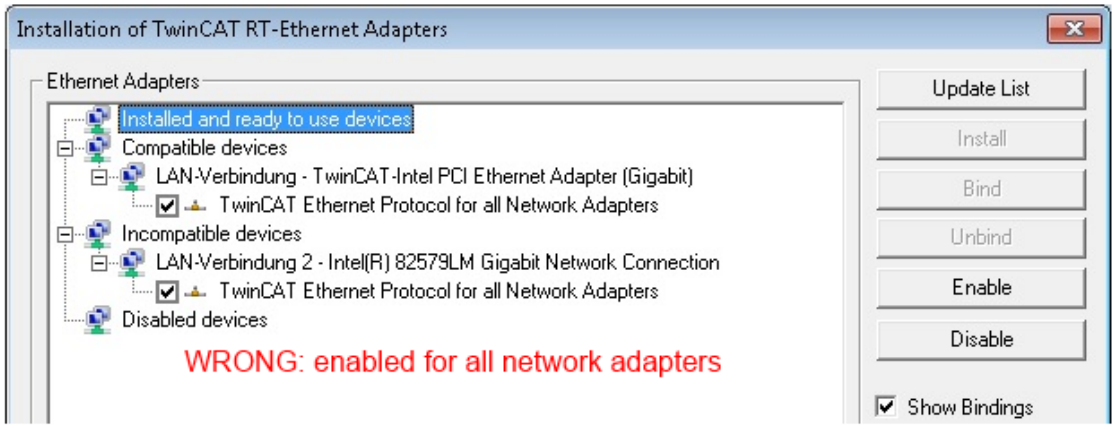


Fig. 86: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:







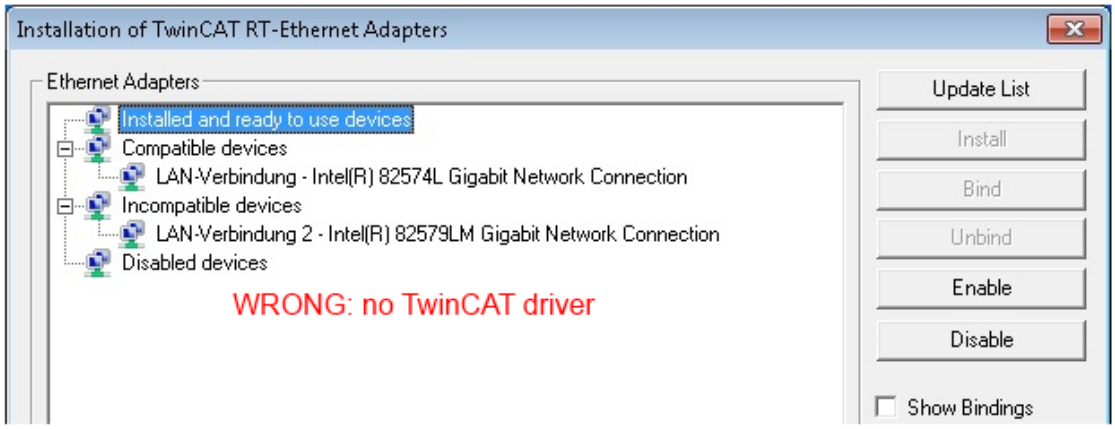


Fig. 87: Incorrect driver settings for the Ethernet port

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.

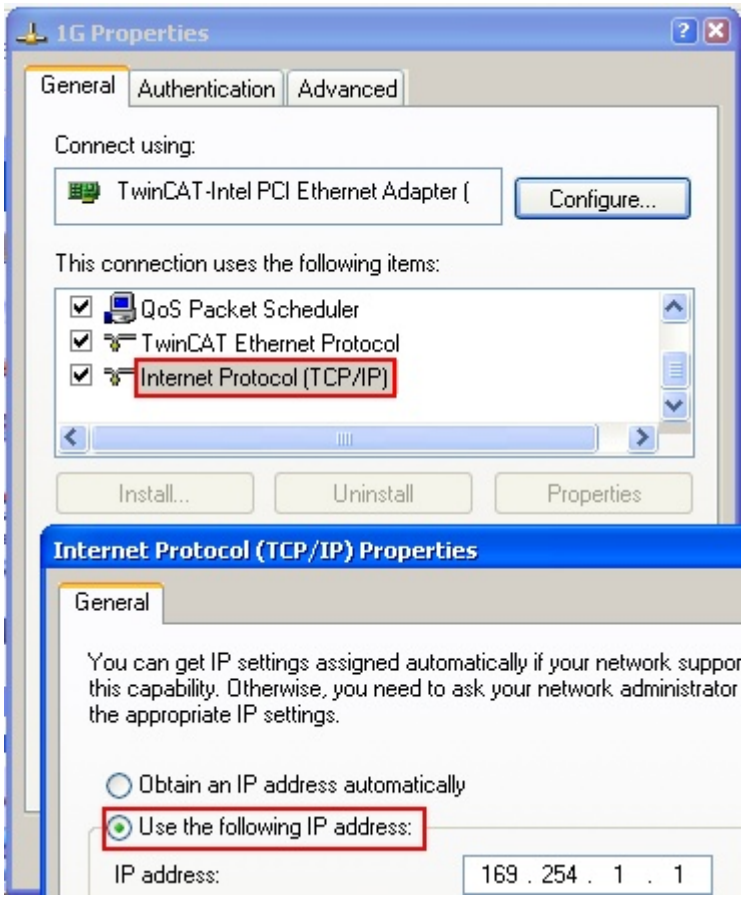


Fig. 88: TCP/IP setting for the Ethernet port

5.2.2 Notes regarding ESI device description

Installation of the latest ESI device description

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

The ESI files for Beckhoff EtherCAT devices are available on the Beckhoff website.

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

Default settings:

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

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

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

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

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

The TwinCAT ESI Updater 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”

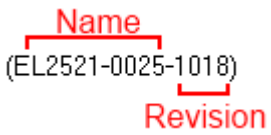


Fig. 89: Identifier structure

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.

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.

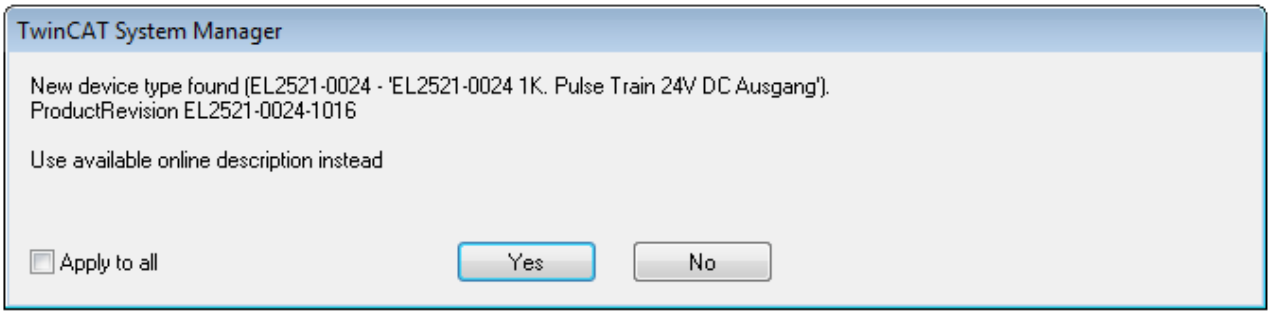


Fig. 90: OnlineDescription information window (TwinCAT 2)

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

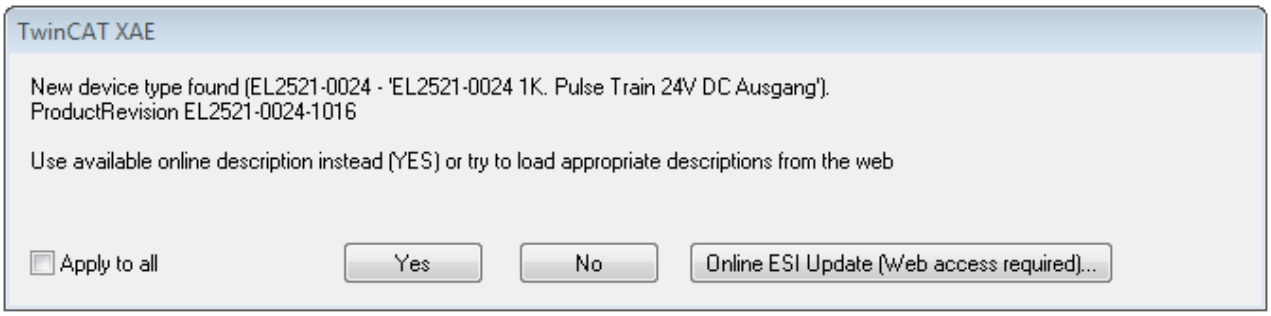


Fig. 91: Information window OnlineDescription (TwinCAT 3)

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".

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. 92: 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)*.

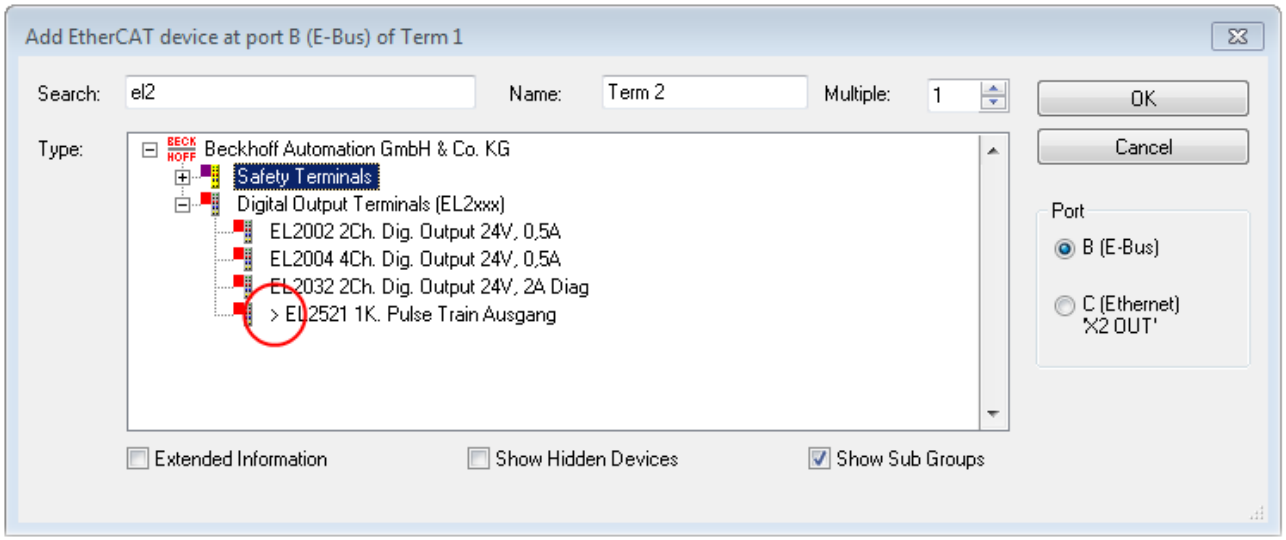


Fig. 93: Indication of an online recorded ESI of EL2521 as an example

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.

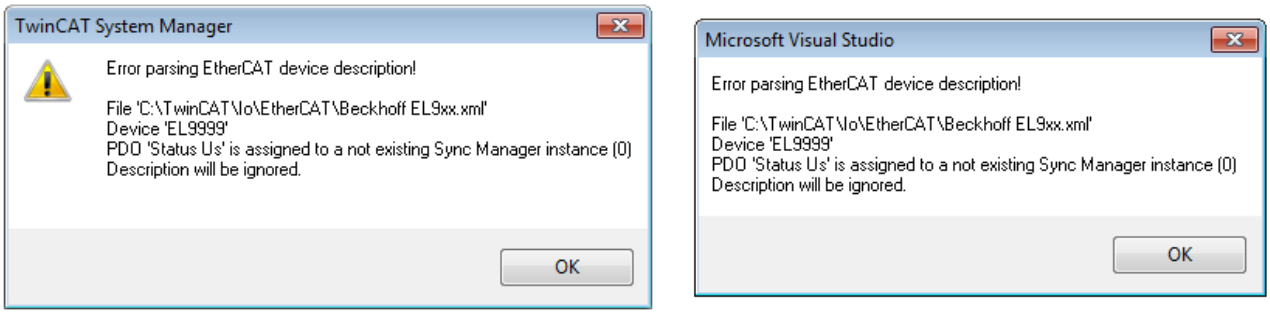


Fig. 94: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

Reasons may include:

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

5.2.3 TwinCAT ESI Updater

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

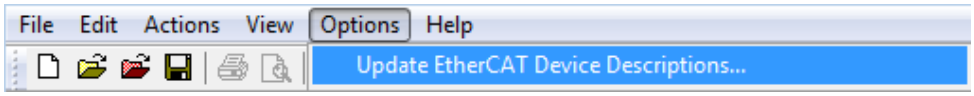


Fig. 95: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:

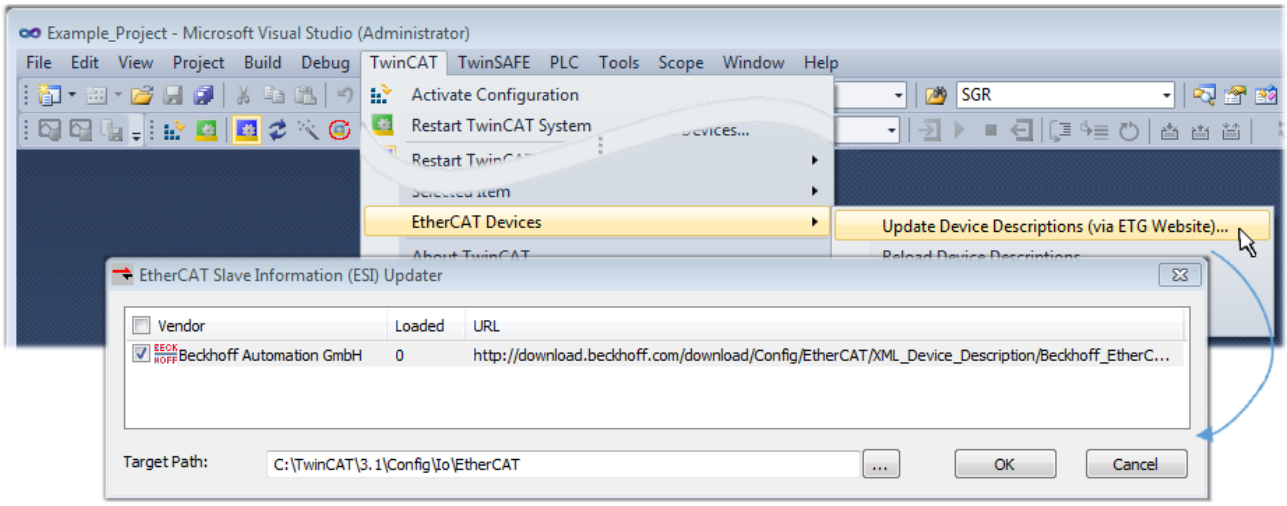


Fig. 96: Using the ESI Updater (TwinCAT 3)

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

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

5.2.4 Distinction between Online and Offline

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

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

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

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 [▶ 86] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 87]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 90]

The scan with existing configuration [▶ 91] can also be carried out for comparison.

5.2.5 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

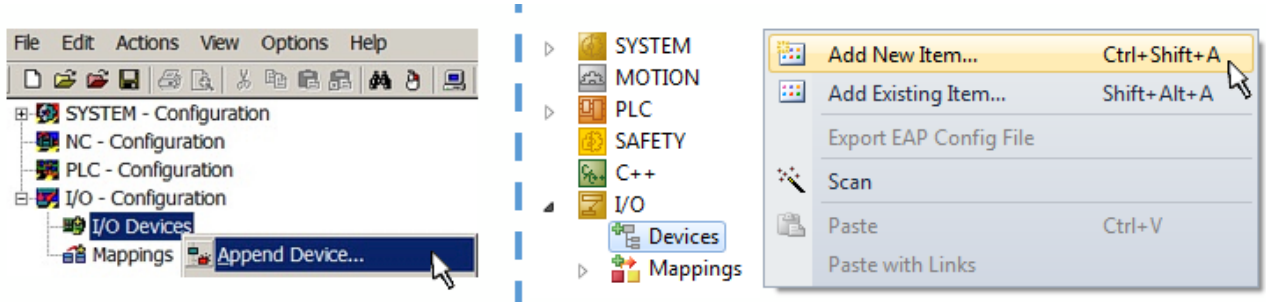


Fig. 97: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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”.

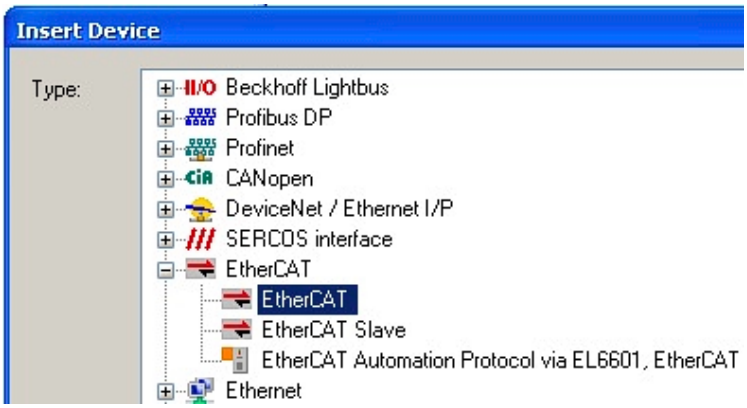


Fig. 98: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

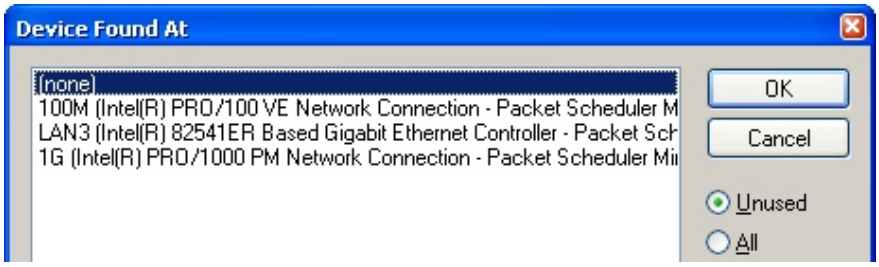


Fig. 99: Selecting the Ethernet port

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)”.

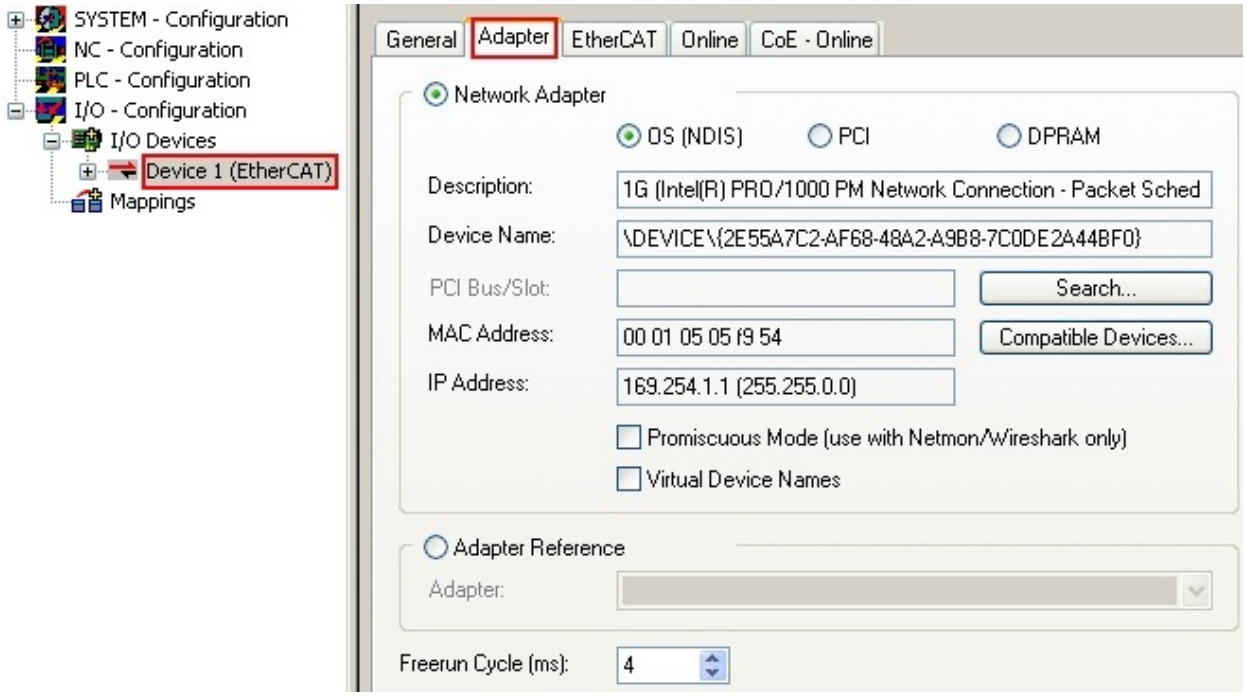
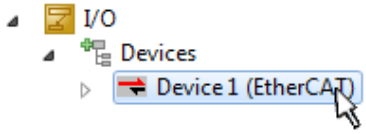


Fig. 100: EtherCAT device properties (TwinCAT 2)

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.

Defining EtherCAT slaves

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

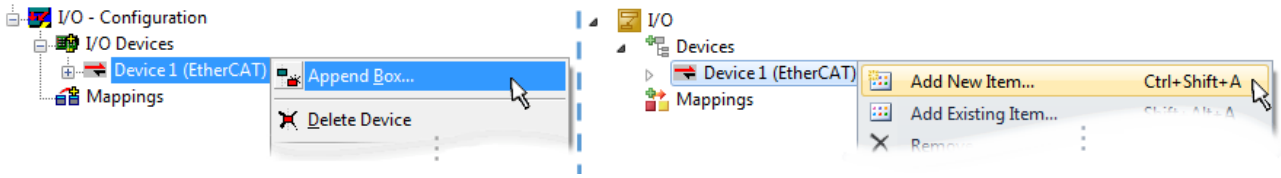


Fig. 101: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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

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

Overview of physical layer

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

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

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

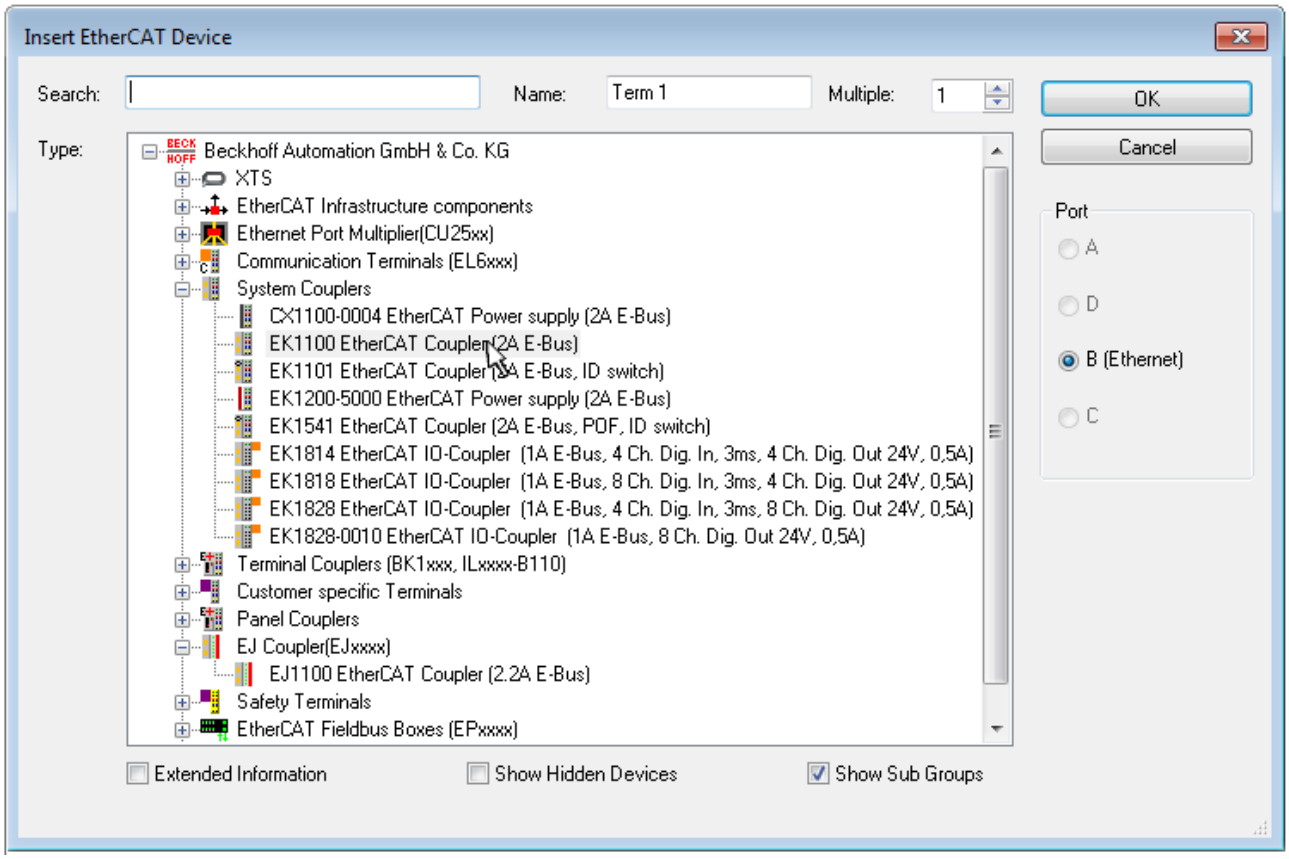


Fig. 102: Selection dialog for new EtherCAT device

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”.

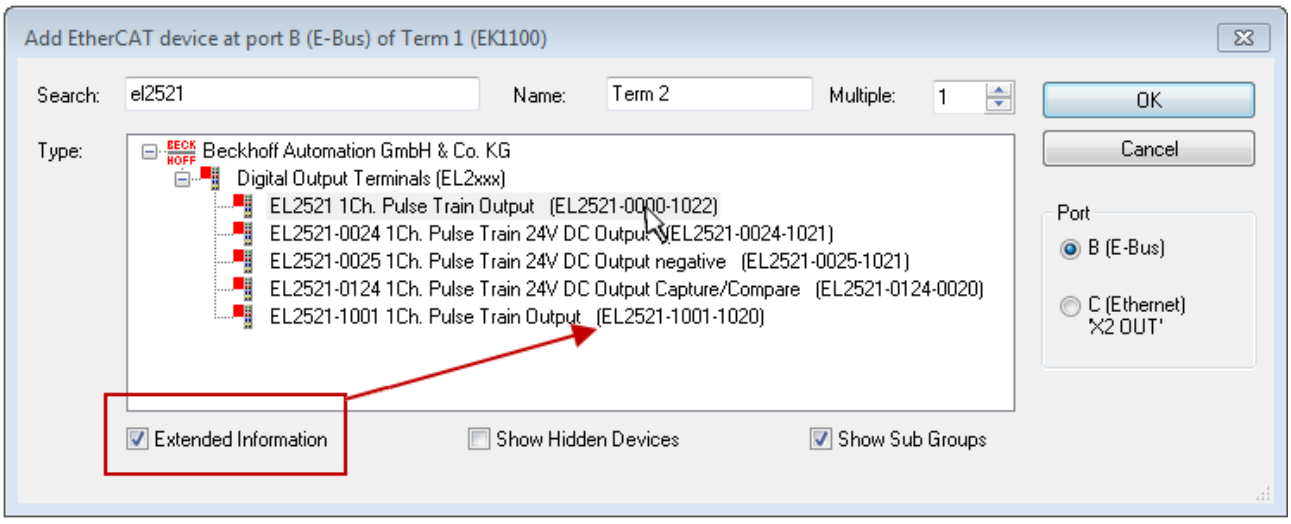


Fig. 103: Display of device revision

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”.

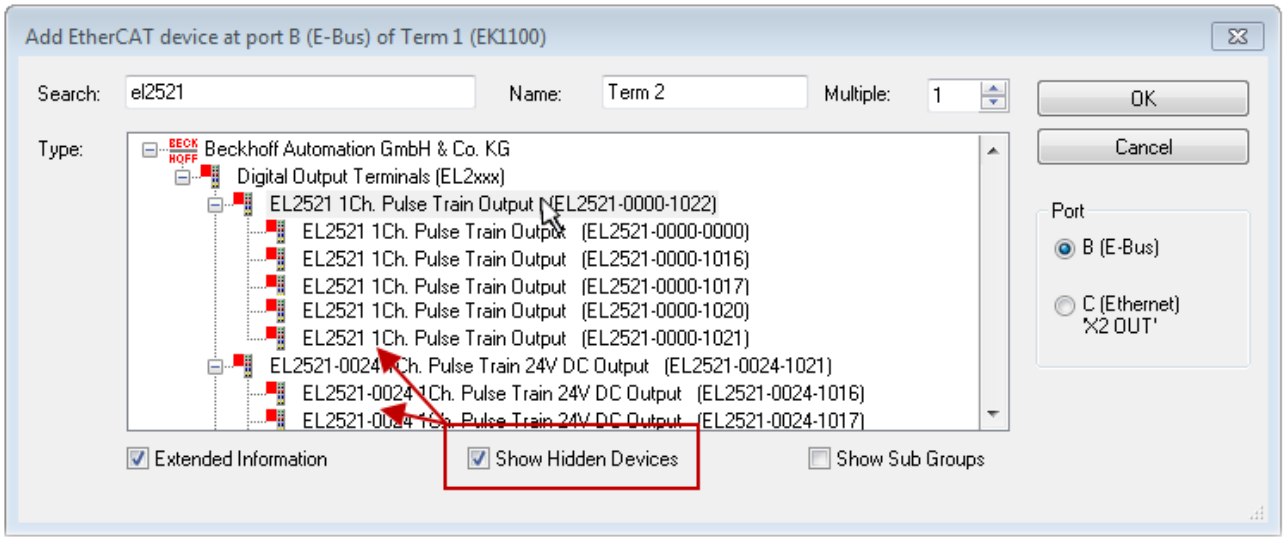


Fig. 104: Display of previous revisions

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.

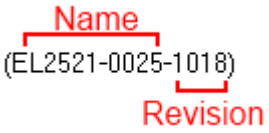


Fig. 105: Name/revision of the terminal

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, ...

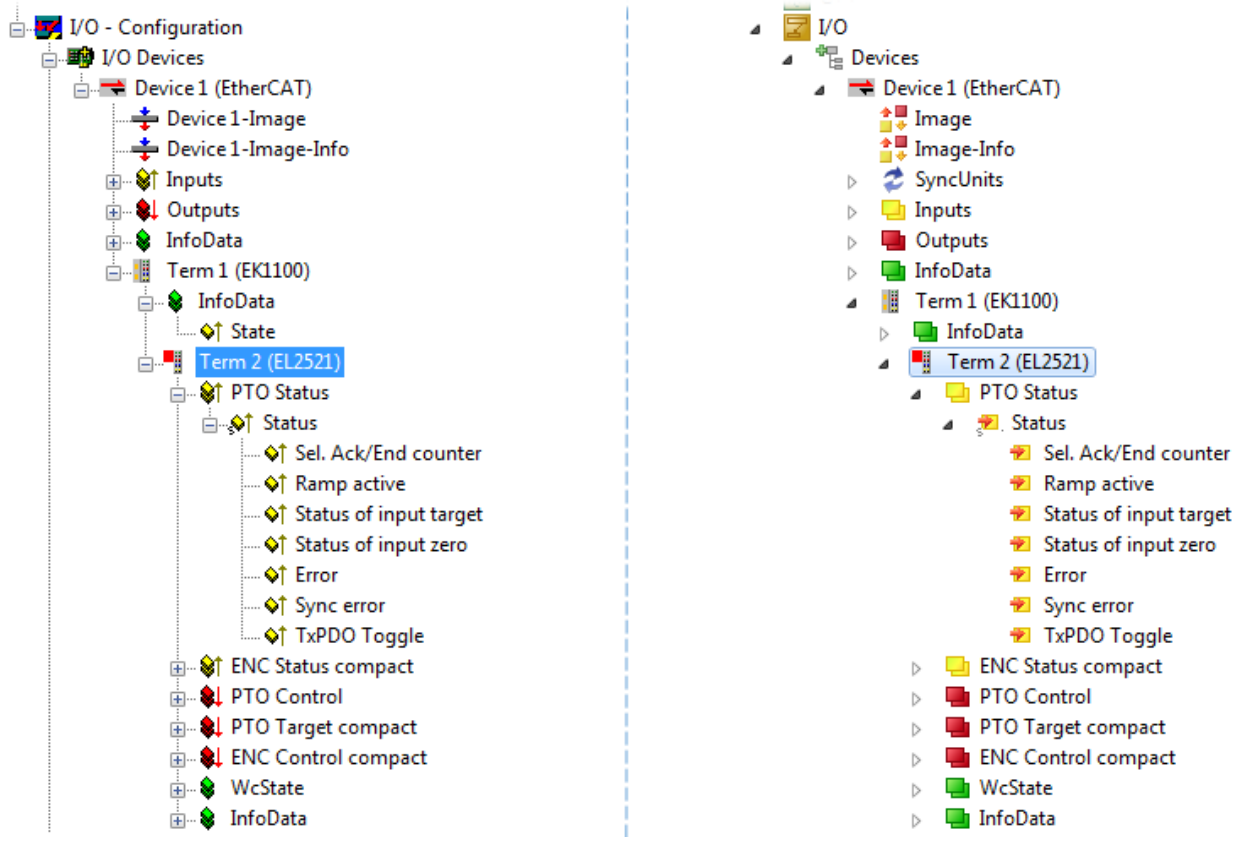


Fig. 106: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)

5.2.6 ONLINE configuration creation

Detecting/scanning of the EtherCAT device

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

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

TwinCAT can be set into this mode:

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

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.

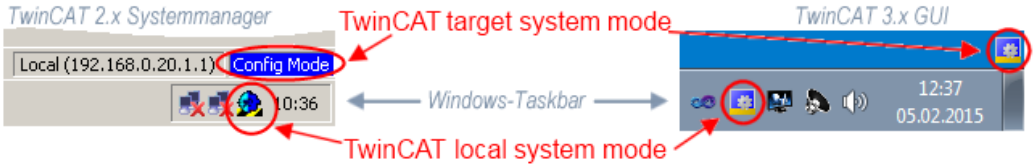


Fig. 107: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

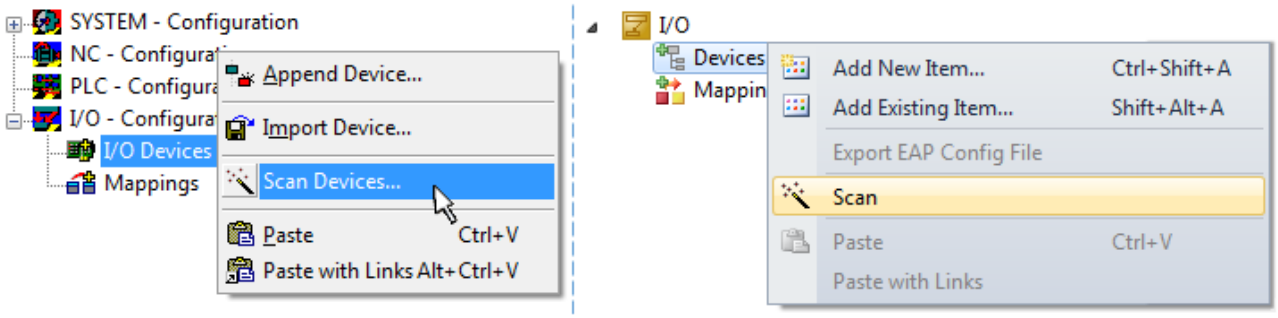


Fig. 108: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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.

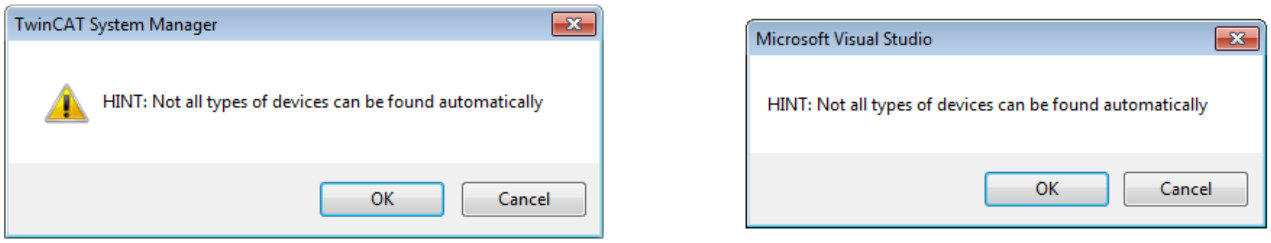


Fig. 109: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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” .

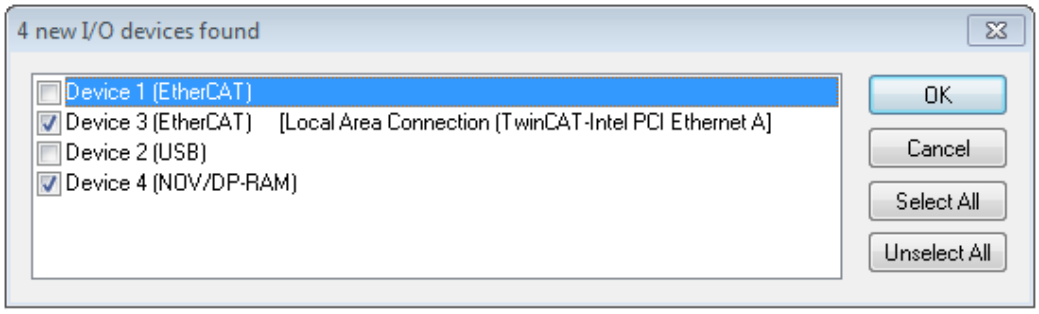


Fig. 110: Detected Ethernet devices

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.

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.

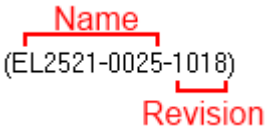


Fig. 111: Example default state

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 [▶ 91] 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:

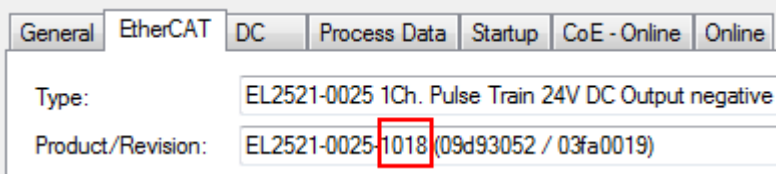


Fig. 112: Installing EthetCAT terminal with revision -1018

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 [▶ 91] 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:

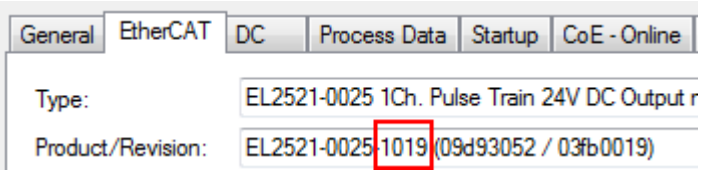


Fig. 113: Detection of EtherCAT terminal with revision -1019

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

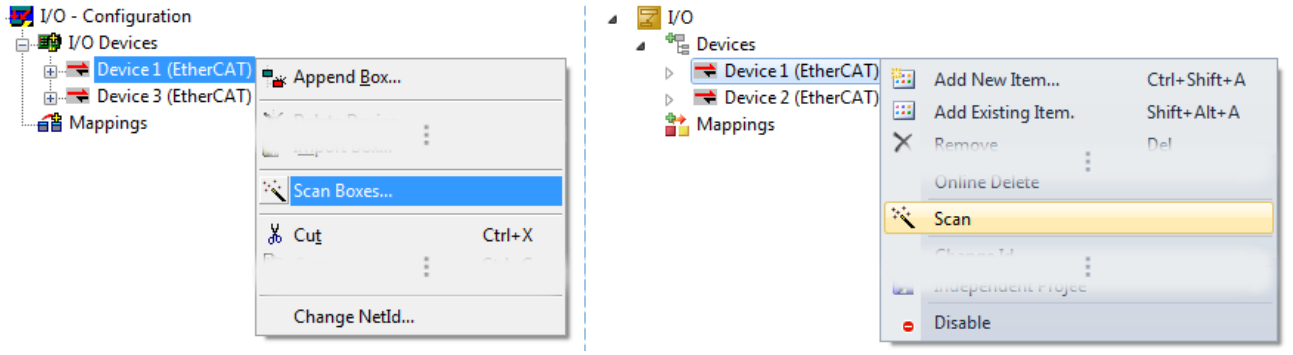


Fig. 115: Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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.

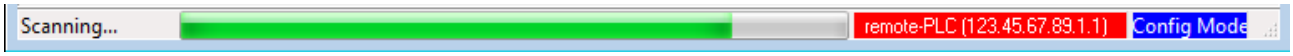


Fig. 116: *Scan progressexemplary by TwinCAT 2*

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

Fig. 117: 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).

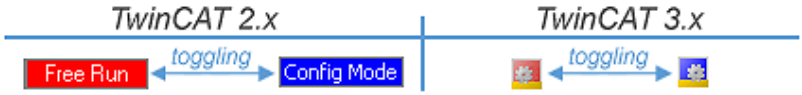


Fig. 118: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar

Fig. 119: 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*.

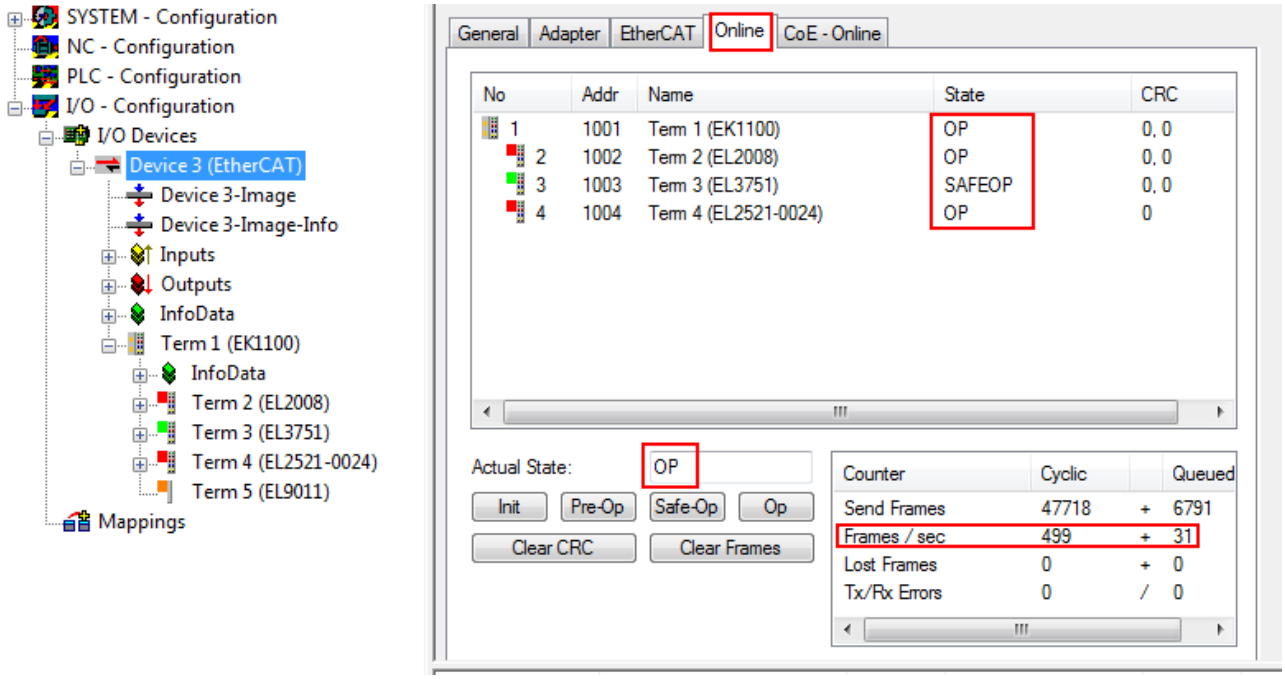


Fig. 120: Online display example

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.

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.

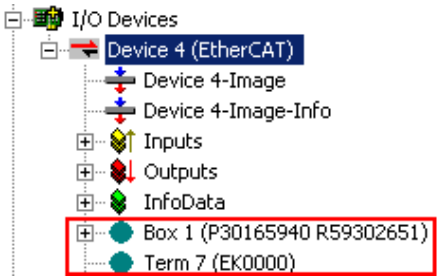


Fig. 121: Faulty identification

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. 122: 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.

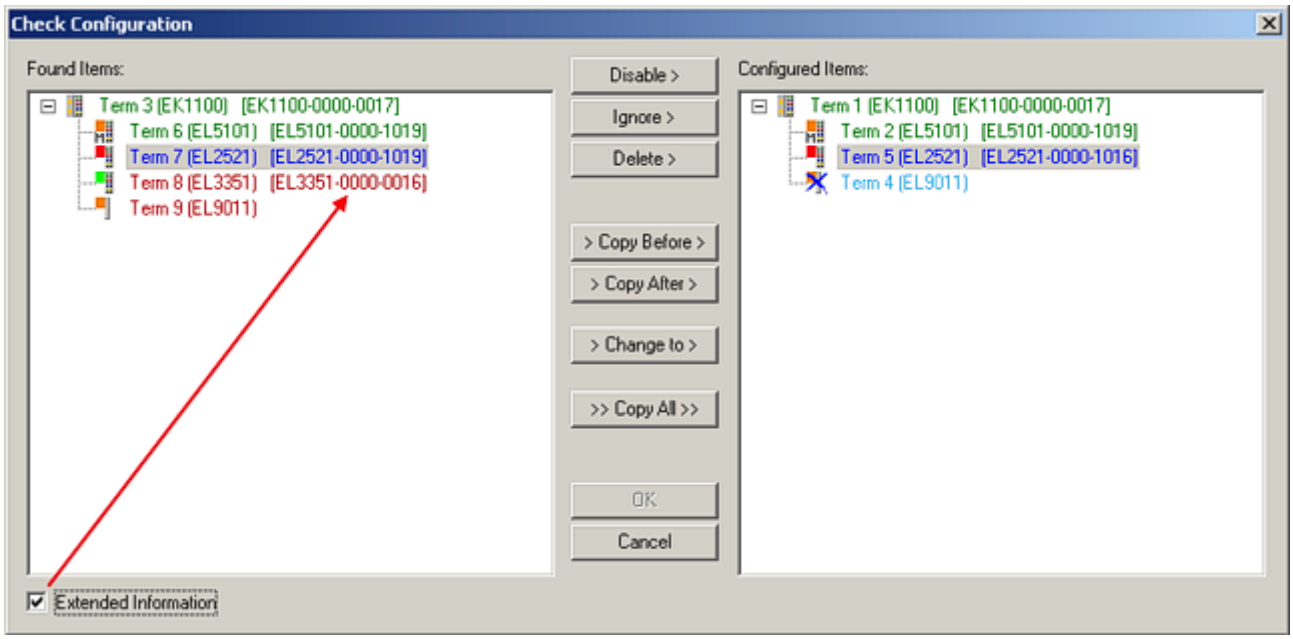


Fig. 123: Correction dialog

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.

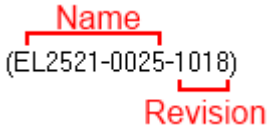


Fig. 124: Name/revision of the terminal

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, ...

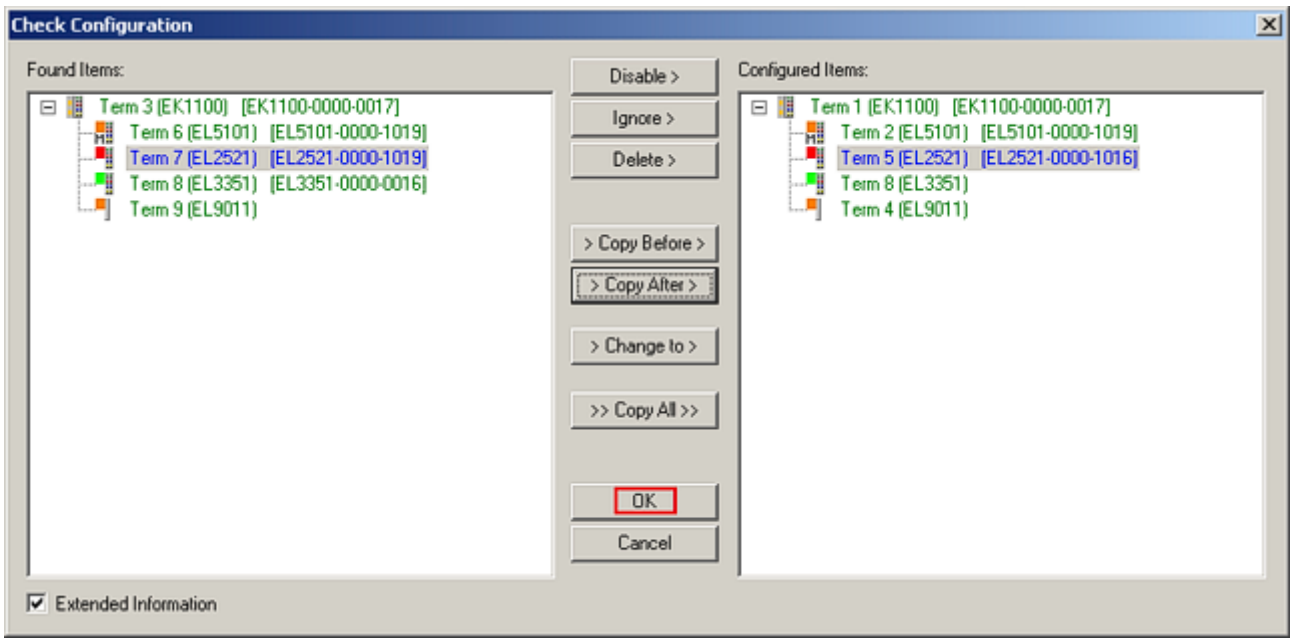


Fig. 125: Correction dialog with modifications

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.

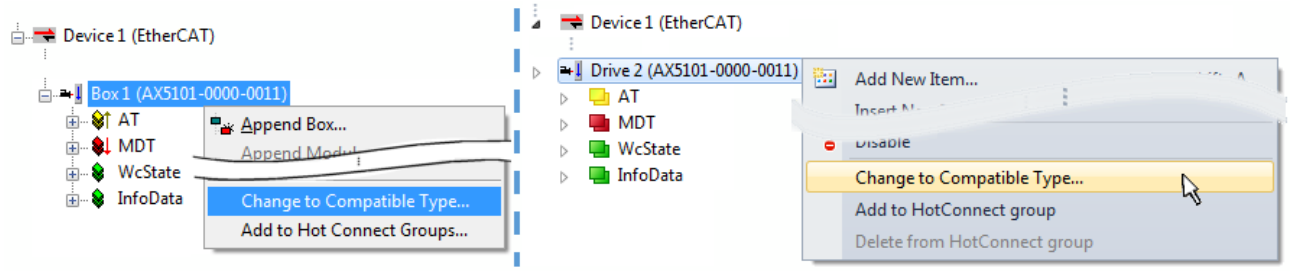


Fig. 126: Dialog “Change to Compatible Type…” (left: TwinCAT 2; right: TwinCAT 3)

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

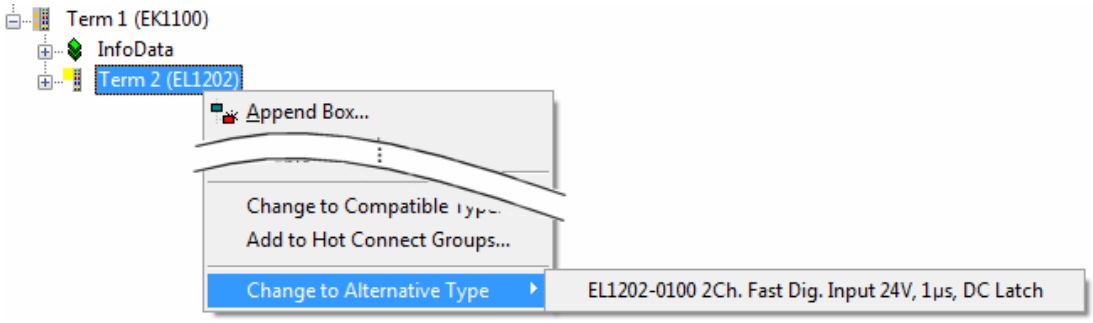


Fig. 127: TwinCAT 2 Dialog Change to Alternative Type

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

5.2.7 EtherCAT subscriber configuration

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

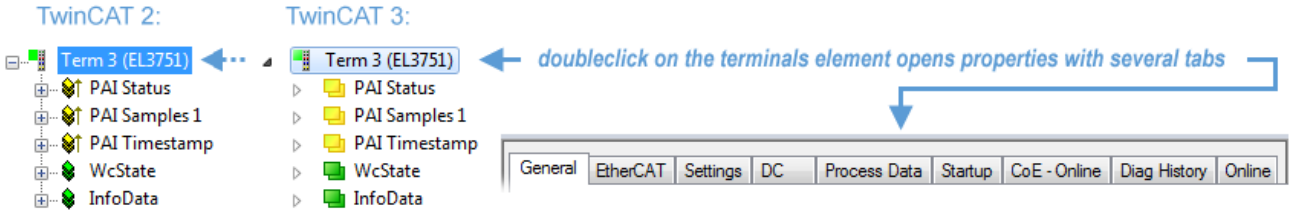


Fig. 128: Branch element as terminal EL3751

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

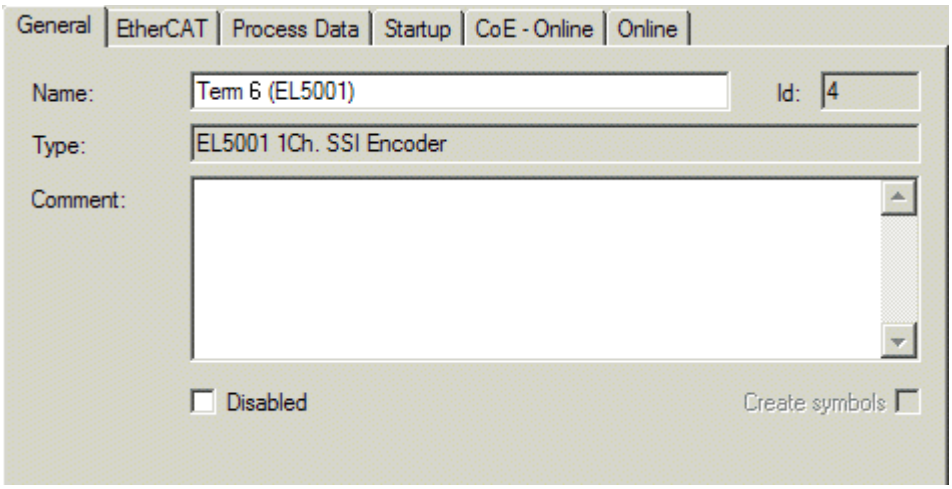


Fig. 129: “General” tab

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

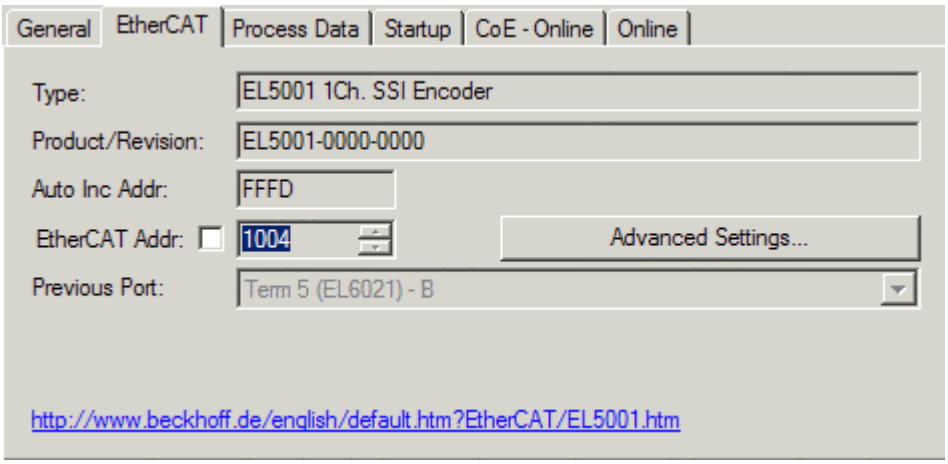


Fig. 130: “EtherCAT” tab

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.

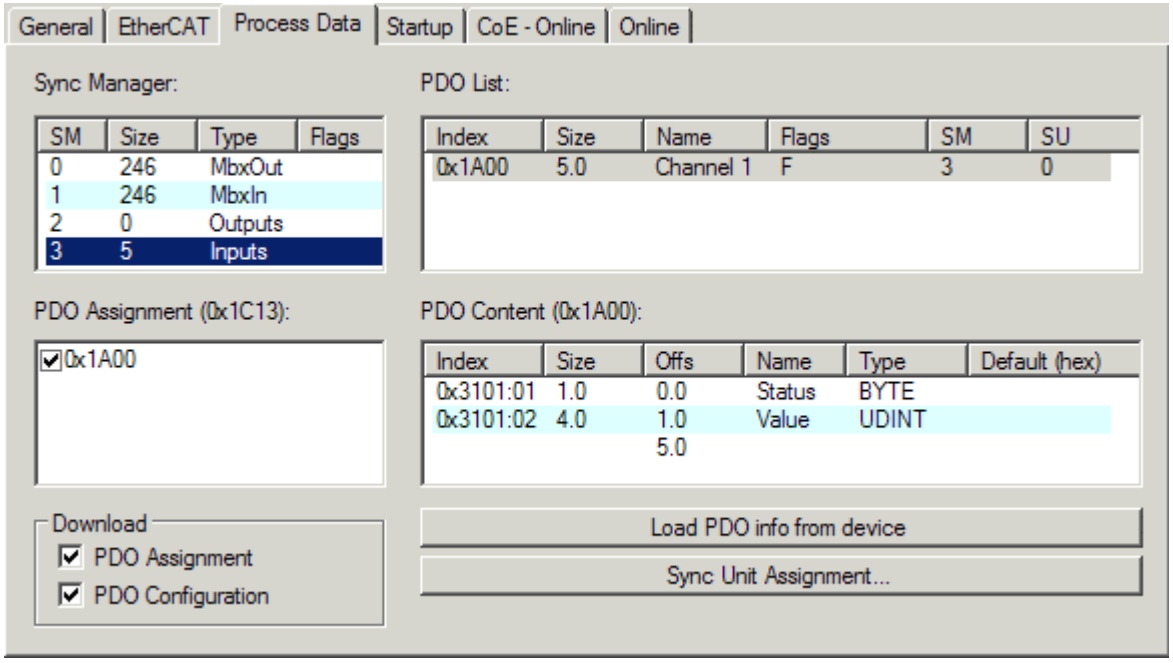


Fig. 131: “Process Data” tab

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”).

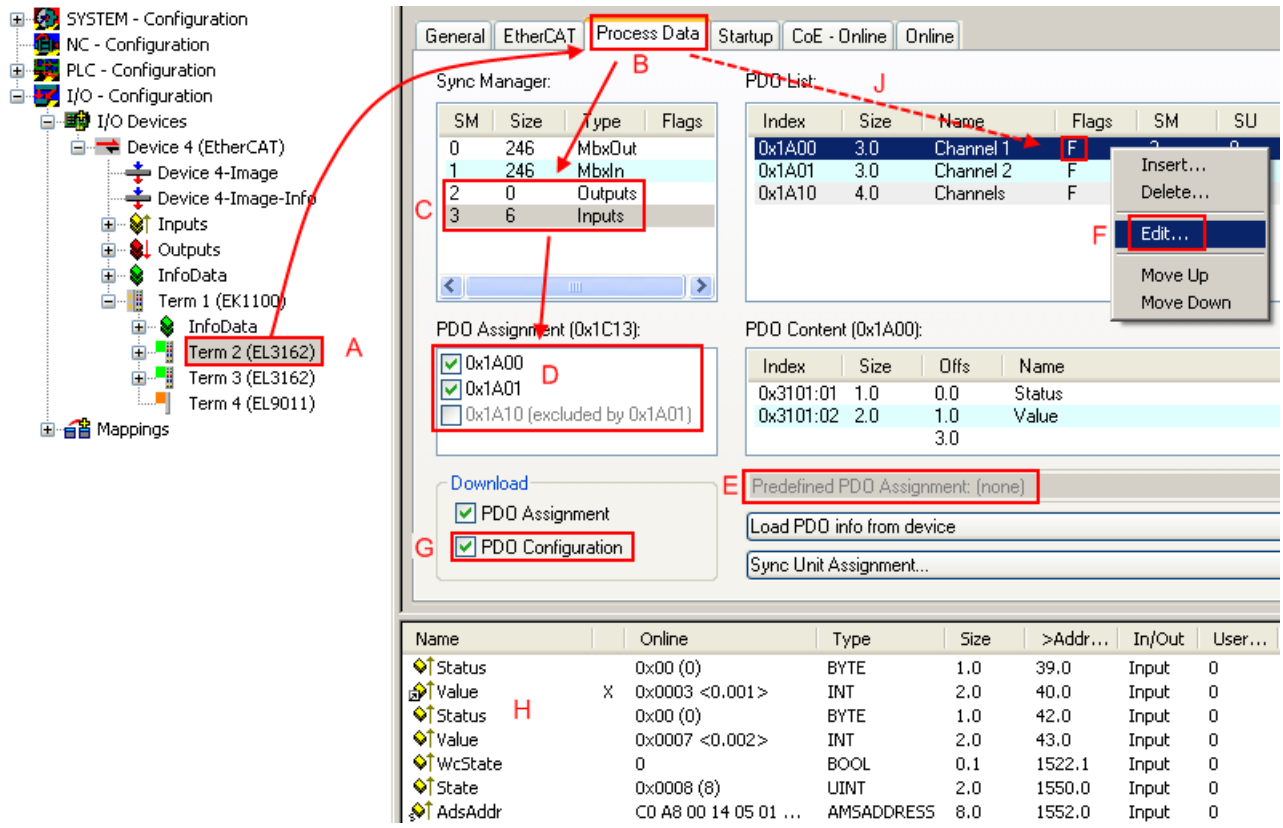


Fig. 132: Configuring the process data

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 [▶ 102] 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.

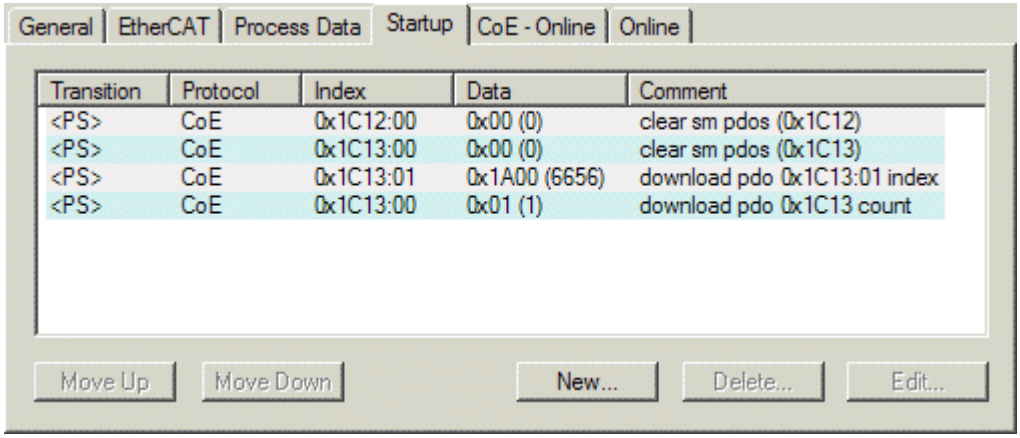


Fig. 133: “Startup” tab

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.

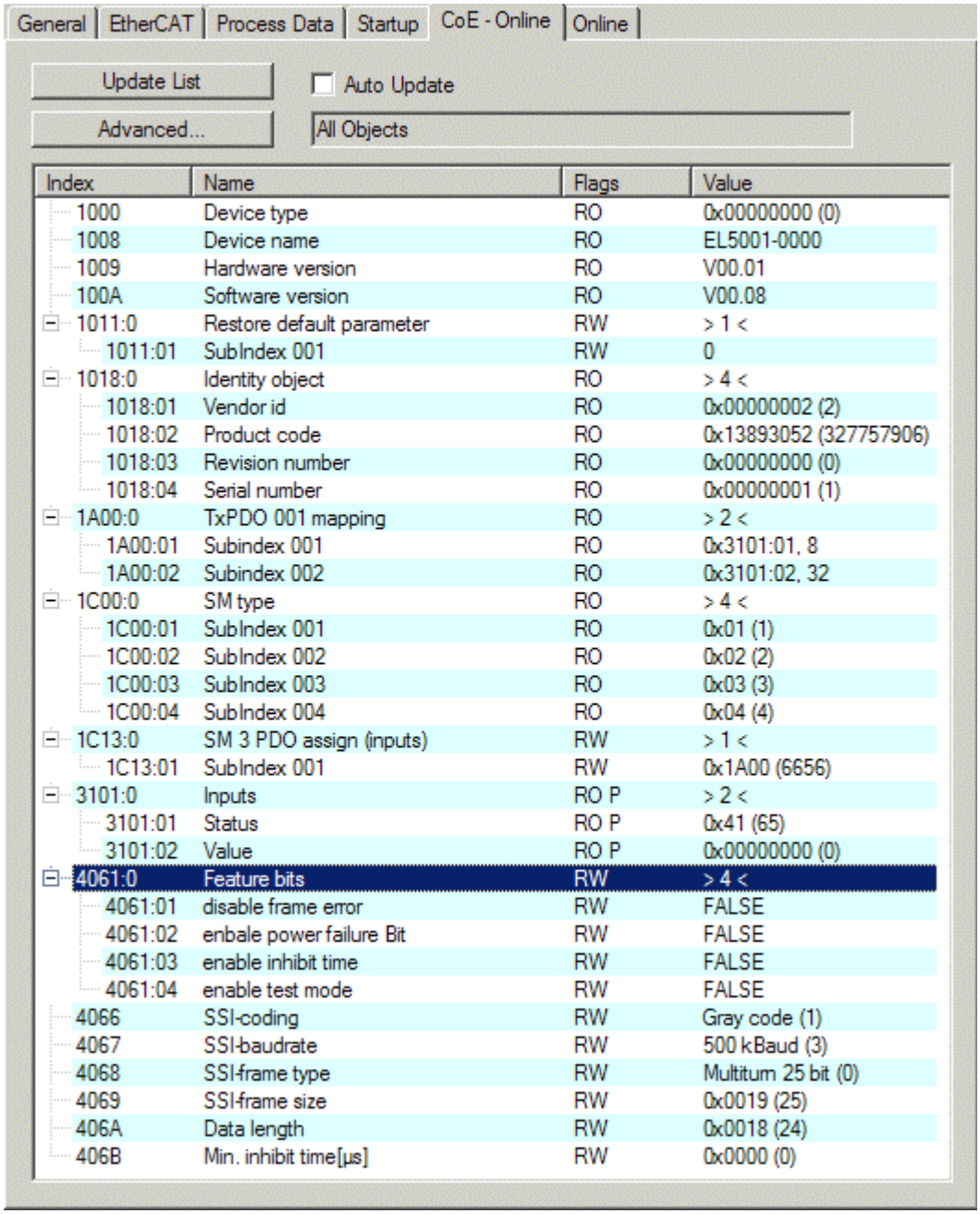


Fig. 134: “CoE - Online” tab

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.

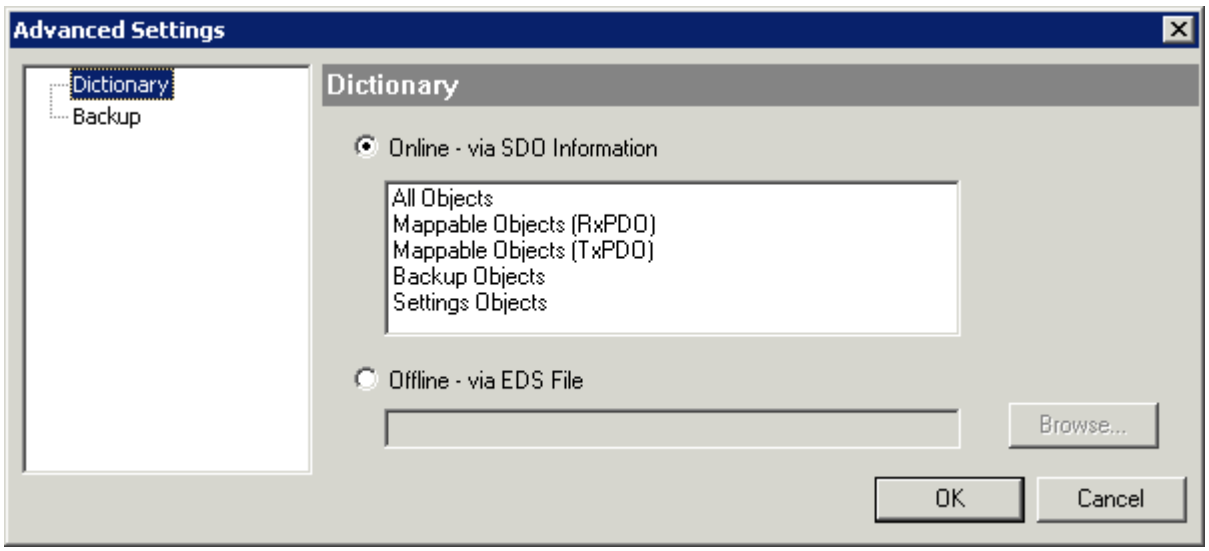


Fig. 135: Dialog “Advanced settings”

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

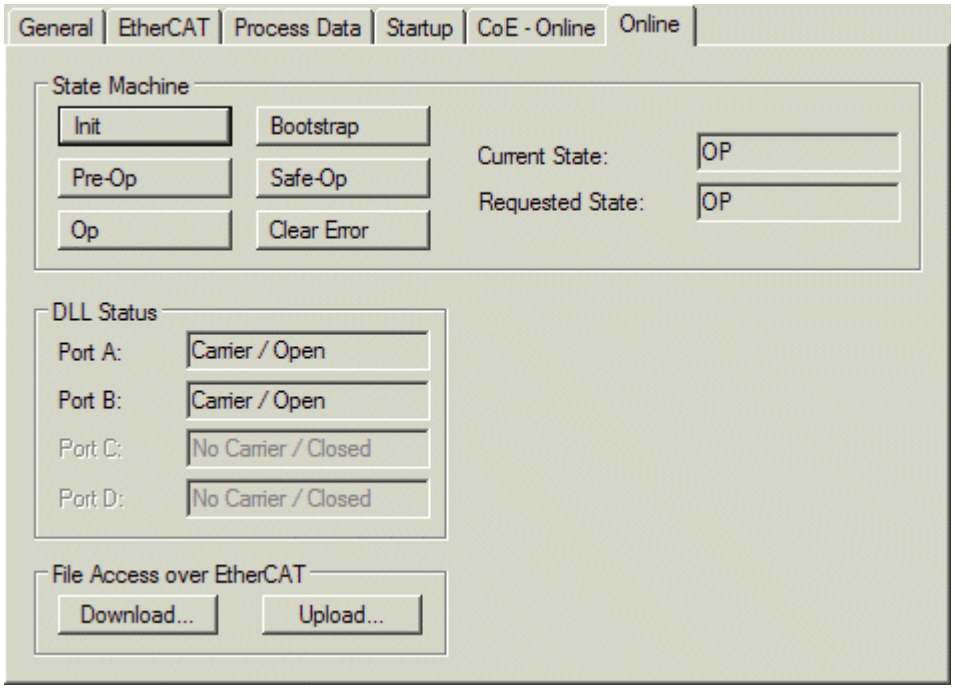


Fig. 136: “Online” tab

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)

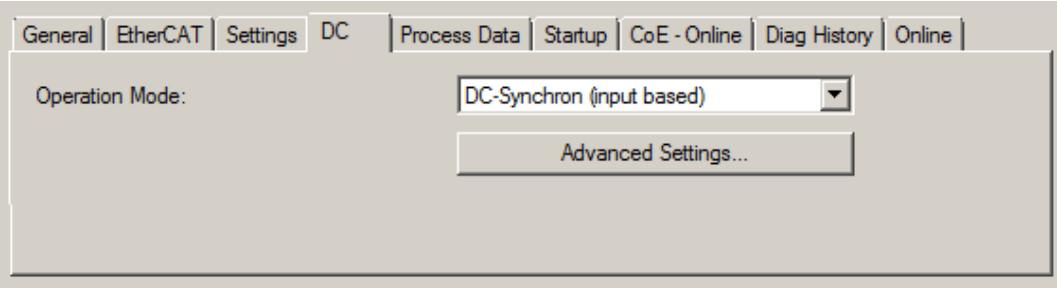


Fig. 137: “DC” tab (Distributed Clocks)

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

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

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

5.2.7.1 Detailed description of Process Data tab

Sync Manager

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

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

PDO Assignment

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

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

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

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 [▶ 100]),

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 [▶ 97] tab.

PDO Configuration

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

5.3 General Notes - EtherCAT Slave Application

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

Diagnosis in real time: WorkingCounter, EtherCAT State and Status

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

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

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

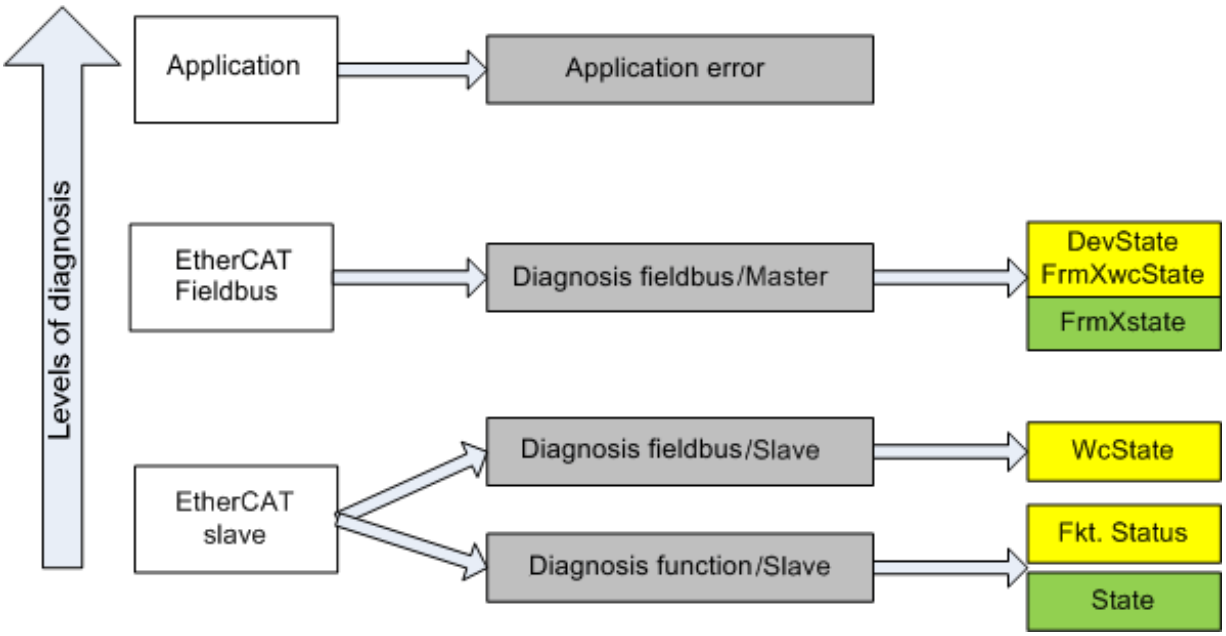


Fig. 138: Selection of the diagnostic information of an EtherCAT Slave

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*.

Color	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.

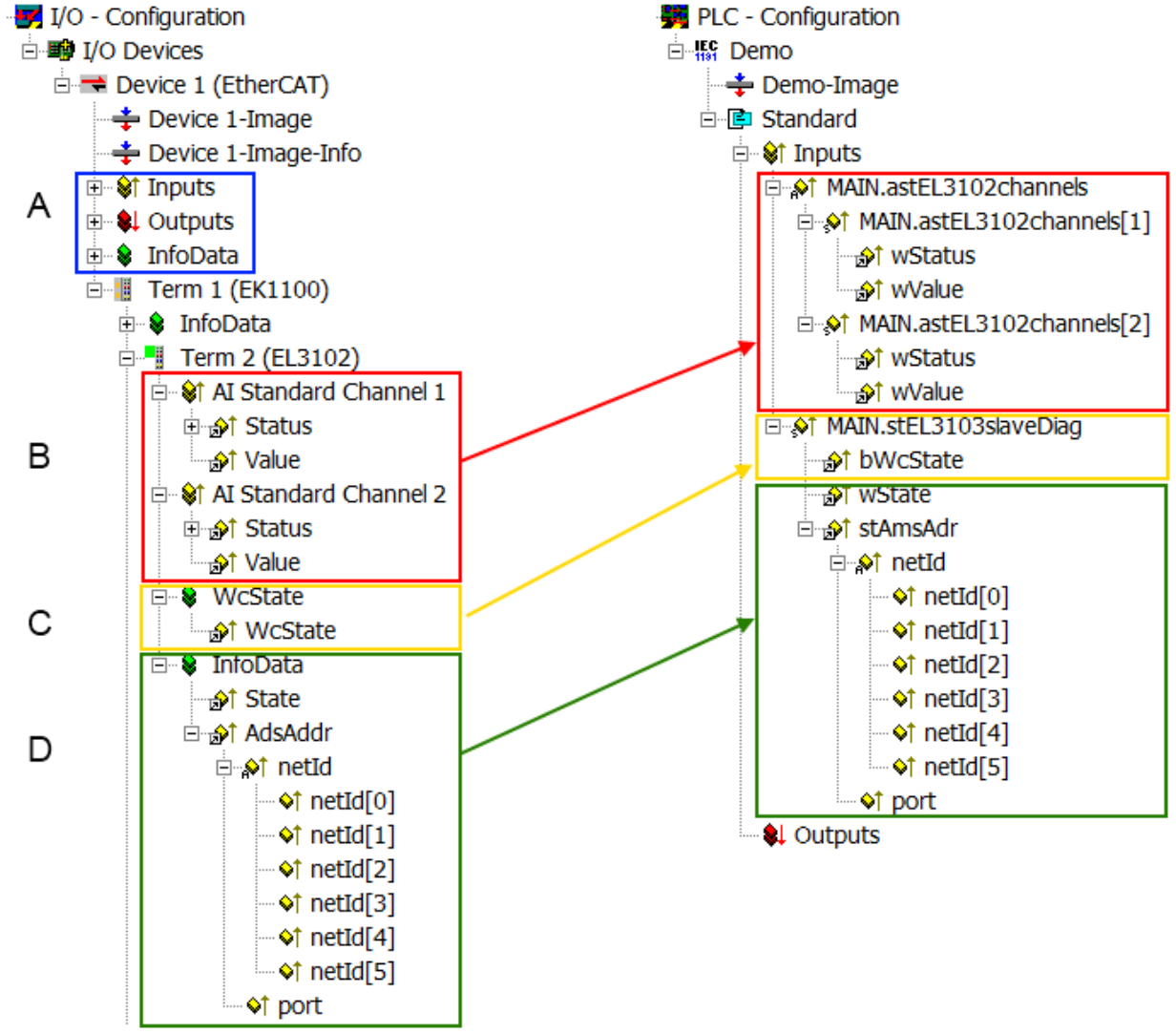


Fig. 139: Basic EtherCAT Slave Diagnosis in the PLC

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*:

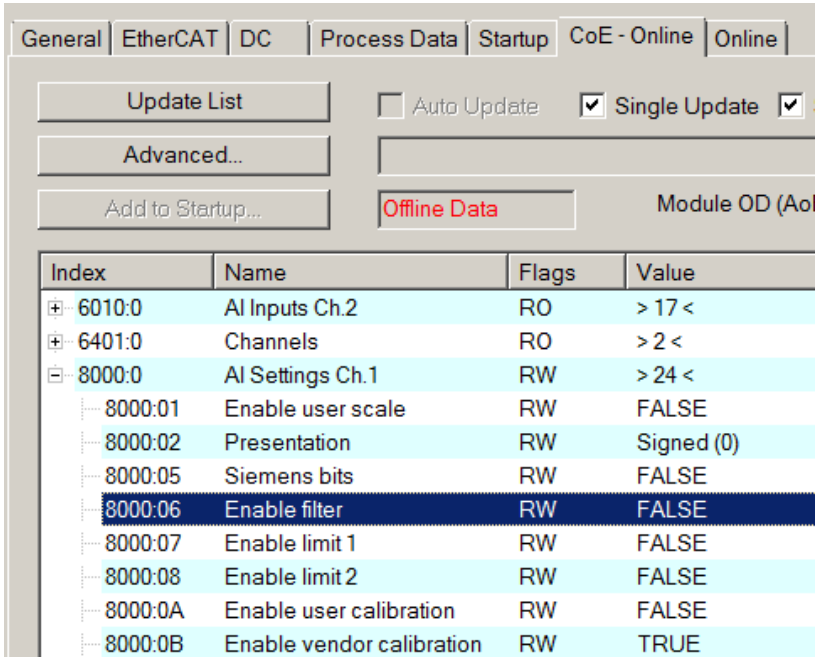


Fig. 140: EL3102, CoE directory

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.

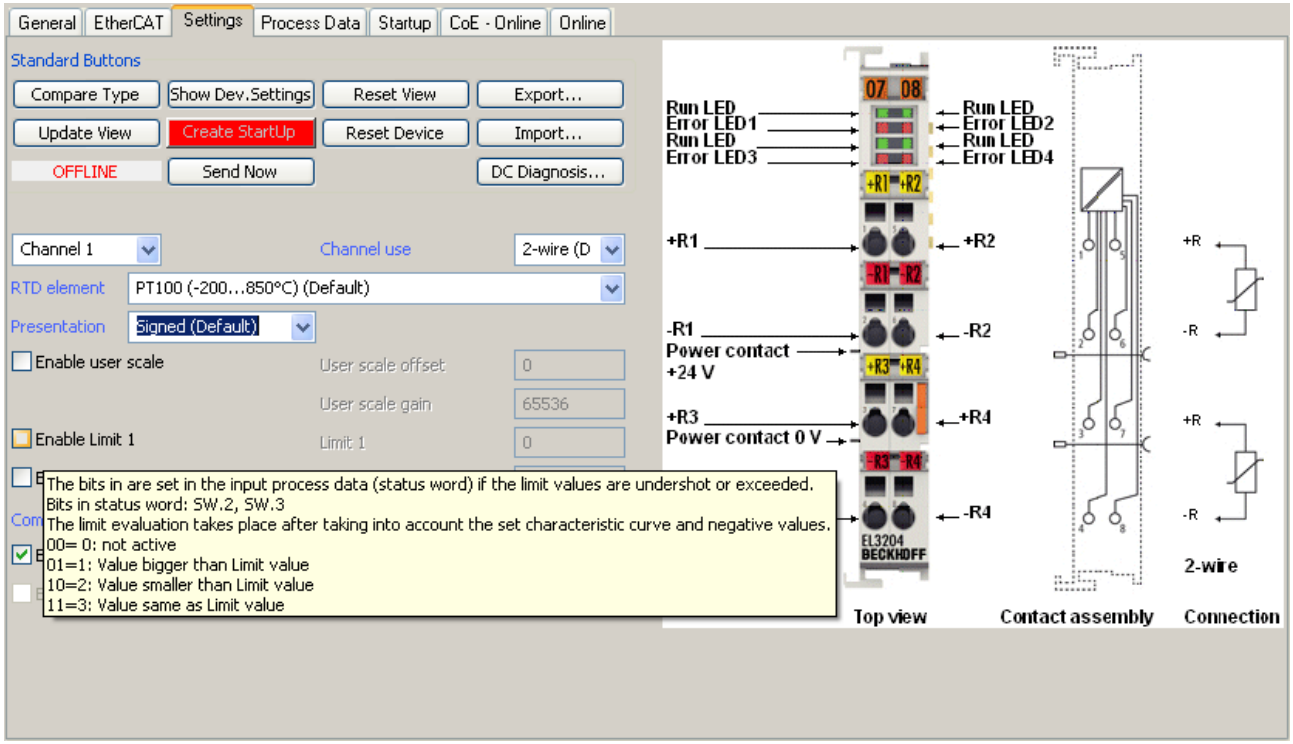


Fig. 141: Example of commissioning aid for a EL3204

This commissioning process simultaneously manages

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

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

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

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

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

- INIT
- PREOP
- SAFEOP
- OP

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

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

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

Standard setting

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

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

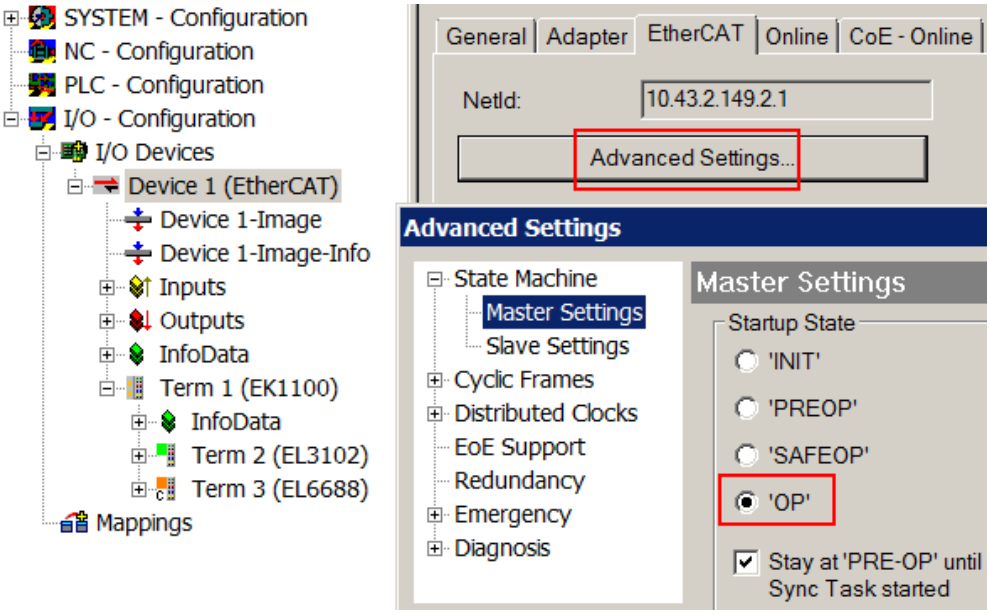


Fig. 142: Default behaviour of the System Manager

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

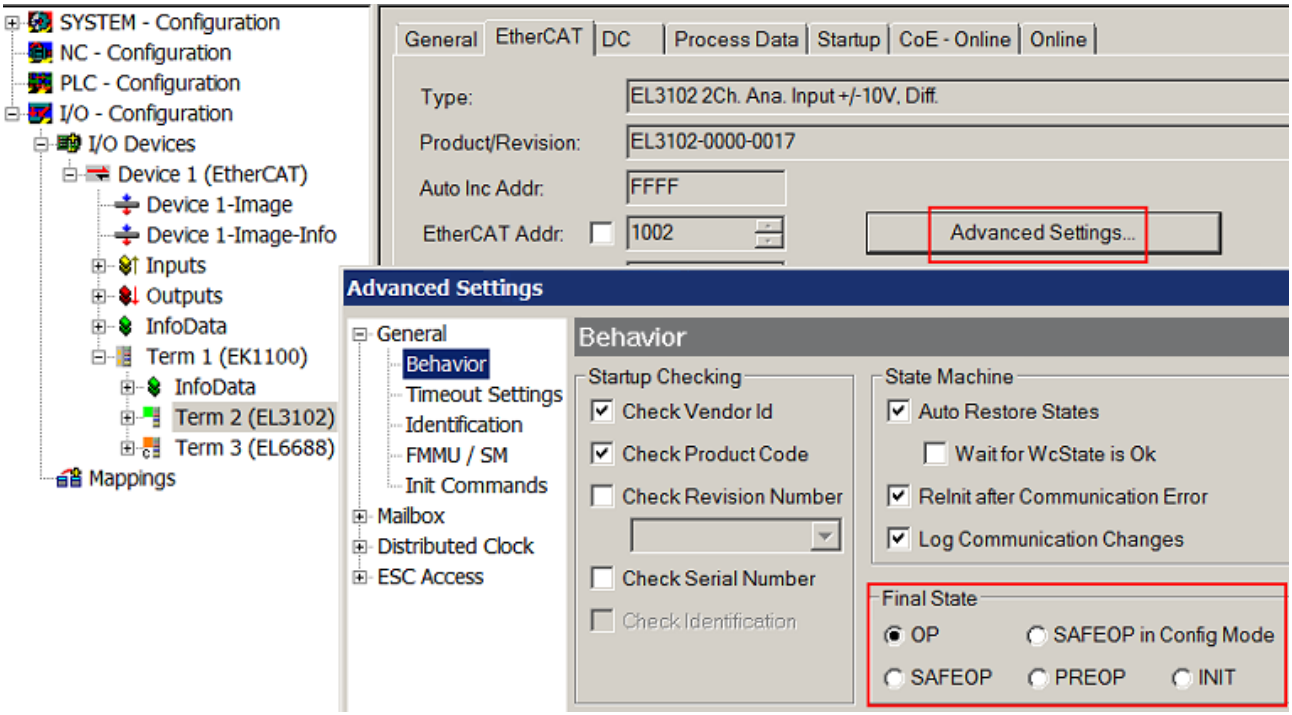


Fig. 143: Default target state in the Slave

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.

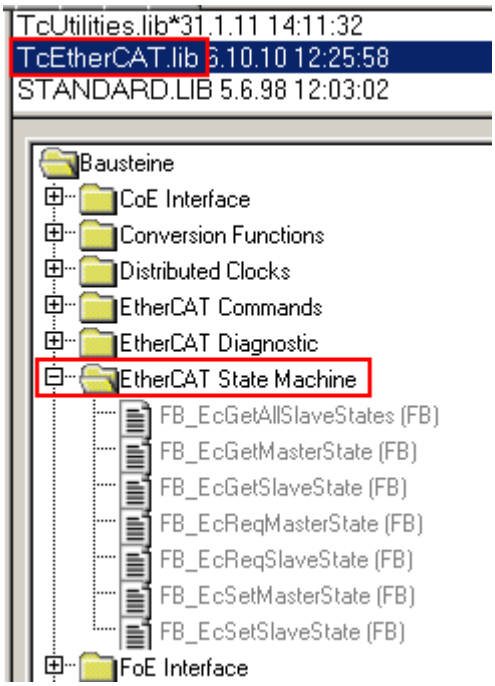
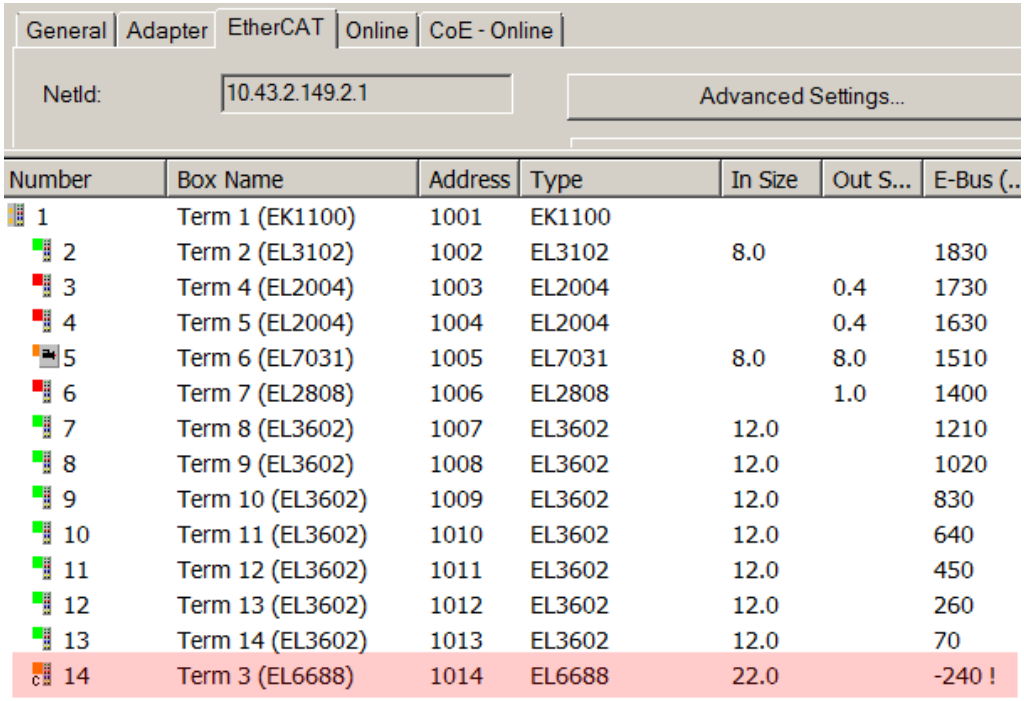


Fig. 144: PLC function blocks

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. 145: 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:

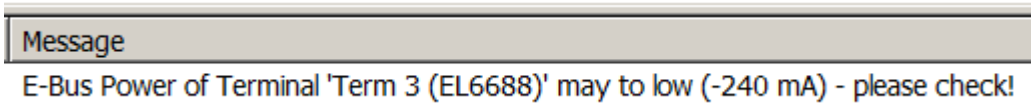


Fig. 146: Warning message for exceeding E-Bus current

NOTE

Caution! Malfunction possible!

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

5.4 ELX5151-0000 - CoE object description

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 [▶ 98] (double-click on the respective object) or via the Process Data tab [▶ 95](allocation of PDOs). Please note the following general CoE notes when using/manipulating the CoE parameters:

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

Introduction

The CoE overview contains objects for different intended applications:

5.4.1 Restore object

Index 1011 Restore default parameters

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

5.4.2 Configuration data

Index 8000 ENC Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8000:0	ENC Settings	Maximum subindex	UINT8	RW	0x17 (23_{dec})
8000:03	Enable up/ down counter [▶ 37]	Enablement of the up/down counter in place of the encoder with the bit set.	BOOLEAN	RW	0x00 (0_{dec})
8000:0A	Enable micro increments [▶ 38]	The counter value is extrapolated by 8 bit.	BOOLEAN	RW	0x00 (0_{dec})
8000:0B	Open Circuit detection A		BOOLEAN	RW	0x00 (0_{dec})
8000:0C	Open Circuit detection B		BOOLEAN	RW	0x00 (0_{dec})
8000:0E	Reversion of rotation [▶ 37]	Activates reversion of rotation	BOOLEAN	RW	0x00 (0_{dec})
8000:0F	Frequency window base [▶ 35]	Basic unit of *Frequency window* (index 0x8000:11 [▶ 112]) 0: µs 1: ms	BIT1	RW	0x00 (0_{dec})
8000:11	Frequency window [▶ 35]	This is the minimum time over which the frequency is determined [1 µs], default: 10 ms. Measuring window < 600 ms: Measurement takes place in frequency mode A. [▶ 35] Measuring window > 600 ms: Measurement takes place in frequency mode B. [▶ 36] The frequency determined in this way is output in index 0x6000:13 [▶ 113] .	UINT16	RW	0x2710 (10000_{dec})
8000:13	Frequency scaling [▶ 35]	Scaling of the frequency measurement (must be divided by this value to obtain the unit in Hz): 100: "0.01 Hz	UINT16	RW	0x0064 (100_{dec})
8000:14	Period scaling [▶ 36]	Scaling of the period in the process data: (must be divided by this value to obtain the unit in ns): 100: “100 ns” period value is a multiple of 100 ns Only the setting “100” is currently possible here.	UINT16	RW	0x0064 (100_{dec})
8000:15	Frequency resolution [▶ 35]	Resolution of the frequency measurement: 100: "0.01 Hz”	UINT16	RW	0x0064 (100_{dec})
8000:16	Period resolution [▶ 36]	Internal resolution of the period measurement: 100: “100 ns” 200: “200 ns” The period is calculated internally with a resolution of 100 ns. The max. measurable period is 1.6 s. Only 100 ns and 200 ns can be set.	UINT16	RW	0x00C8 (200_{dec})
8000:17	Frequency Wait Time [▶ 35]	Waiting time [ms] for frequency measurement Default: 1.6 s (maximum possible value) If the time from Frequency window [▶ 112] has elapsed, the next positive edge from track A is awaited for this time (frequency mode A only). This enables the update speed for the Frequency process data to be optimized, depending on the expected frequencies. At least double the period of the minimum frequency to be measured should be entered here. t >= 2* (1 / f_{min}).	UINT16	RW	0x0640 (1600_{dec})

5.4.3 Input data

Index 6000 ENC Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default value
6000:0	ENC Inputs	Maximum subindex	UINT8	RO	0x1E (22_{dec})
6000:03	Set counter done	The counter was set.	BOOLEAN	ROP	0x00 (0_{dec})
6000:07	Open Circuit		BOOLEAN	ROP	0x00 (0_{dec})
6000:08	Extrapolation stall [▶ 38]	The extrapolated part of the counter is invalid. The speed has fallen below the minimum speed required to use the micro-increments [▶ 38].	BOOLEAN	ROP	0x00 (0_{dec})
6000:09	Status of input A	Status of input A	BOOLEAN	ROP	0x00 (0_{dec})
6000:0A	Status of input B	Status of input B	BOOLEAN	ROP	0x00 (0_{dec})
6000:0E	Sync Error	The *Sync error* bit is only required for DC mode. It indicates whether a synchronization error has occurred during the previous cycle.	BOOLEAN	ROP	0x00 (0_{dec})
6000:0F	TxPDO State		BOOLEAN	ROP	0x00 (0_{dec})
6000:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated.	BOOLEAN	ROP	0x00 (0_{dec})
6000:11	Counter value	Counter value	UINT32	ROP	0x00000000 (0_{dec})
6000:13	Frequency value	The frequency (setting of the scaling in index 0x8000:13 and the resolution in index 0x8000:15)	UINT32	ROP	0x00000000 (0_{dec})
6000:14	Period value	The period (setting of the scaling in index 0x8000:14 and the resolution in index 0x8000:16)	UINT32	ROP	0x00000000 (0_{dec})
6000:16	Timestamp [▶ 31]	Time stamp of the last counter change.	UINT64	ROP	0x0000000000000000 (0_{dec})
6000:1C	Frequency Value (uint16)		UINT16	ROP	0x0000 (0_{dec})
6000:1D	Counter Value (uint16)		UINT16	ROP	0x0000 (0_{dec})
6000:1E	Period value (uint16)		UINT16	ROP	0x0000 (0_{dec})

5.4.4 Output data

Index 7000 ENC Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default value
7000:0	ENC Outputs	Maximum subindex	UINT8	RO	0x11 (17_{dec})
7000:03	Set counter	Set counter	BOOLEAN	RO P	0x00 (0_{dec})
7000:11	Set counter value	This is the counter value to be set via *Set counter* (index 0x7000:03 [▶ 113]).	UINT32	RO P	0x00000000 (0_{dec})

5.4.5 Standard objects (0x1000-0x1FFF)

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default value
1000:0	Device type	Device type of the EtherCAT slave: the Lo-Word contains the CoE profile used (5001). The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	0x01FF1389 (33493897_{dec})

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default value
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	ELX5151

Index 1009 Hardware version

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

Index 100A Software version

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

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default value
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	0x970C3FF9 (2534162425_{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 value
10F0:0	Backup parameter handling	Information for standardized loading and saving of backup entries	UINT8	RO	0x01 (1_{dec})
10F0:01	Checksum	Checksum across all backup entries of the EtherCAT slave	UINT32	RO	0x00000000 (0_{dec})

Index 1400 ENC RxPDO-Par Control

Index (hex)	Name	Meaning	Data type	Flags	Default value
1400:0	ENC RxPDO-Par Control	PDO Parameter RxPDO 1	UINT8	RO	0x06 (6_{dec})
1400:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 1	OCTET-STRING[2]	RO	01 16

Index 1401 ENC RxPDO-Par Control compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1401:0	ENC RxPDO-Par Control compact	PDO Parameter RxPDO 2	UINT8	RO	0x06 (6_{dec})
1401:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 2	OCTET-STRING[2]	RO	00 16

Index 1600 ENC RxPDO-Map Control

Index (hex)	Name	Meaning	Data type	Flags	Default value
1600:0	ENC RxPDO-Map Control	PDO Mapping RxPDO 1	UINT8	RO	0x05 (5_{dec})
1600:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (ENC Outputs), entry 0x01 (Enable latch C))	UINT32	RO	0x0000:00, 2
1600:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (ENC Outputs), entry 0x02 (Enable latch extern on positive edge))	UINT32	RO	0x7000:03, 1
1600:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (ENC Outputs), entry 0x03 (Set counter))	UINT32	RO	0x0000:00, 5
1600:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (ENC Outputs), entry 0x04 (Enable latch extern on negative edge))	UINT32	RO	0x0000:00, 8
1600:05	SubIndex 005	5. PDO Mapping entry (4 bits align)	UINT32	RO	0x7000:11, 32

Index 1601 ENC RxPDO-Map Control compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1601:0	ENC RxPDO-Map Control compact	PDO Mapping RxPDO 2	UINT8	RO	0x05 (5_{dec})
1601:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (ENC Outputs), entry 0x01 (Enable latch C))	UINT32	RO	0x0000:00, 2
1601:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (ENC Outputs), entry 0x02 (Enable latch extern on positive edge))	UINT32	RO	0x7000:03, 1
1601:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (ENC Outputs), entry 0x03 (Set counter))	UINT32	RO	0x0000:00, 5
1601:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (ENC Outputs), entry 0x04 (Enable latch extern on negative edge))	UINT32	RO	0x0000:00, 8
1601:05	SubIndex 005	5. PDO Mapping entry (4 bits align)	UINT32	RO	0x7000:11, 16

Index 1800 ENC TxPDO-Par Status

Index (hex)	Name	Meaning	Data type	Flags	Default value
1800:0	ENC TxPDO-Par Status	PDO parameter TxPDO 1	UINT8	RO	0x06 (6_{dec})
1800:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 1.	OCTET-STRING[2]	RO	01 1A

Index 1801 ENC TxPDO-Par Status compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1801:0	ENC TxPDO-Par Status compact	PDO parameter TxPDO 2	UINT8	RO	0x06 (6_{dec})
1801:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 2.	OCTET-STRING[2]	RO	00 1A

Index 1802 ENC TxPDO-Par Period

Index (hex)	Name	Meaning	Data type	Flags	Default value
1802:0	ENC TxPDO-Par Period	PDO parameter TxPDO 3	UINT8	RO	0x06 (6_{dec})
1802:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 3.	OCTET-STRING[2]	RO	03 1A

Index 1803 ENC TxPDO-Par Frequency

Index (hex)	Name	Meaning	Data type	Flags	Default value
1803:0	ENC TxPDO-Par Frequency	PDO parameter TxPDO 4	UINT8	RO	0x06 (6_{dec})
1803:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 4.	OCTET-STRING[2]	RO	02 1A

Index 1804 ENC TxPDO-Par Timest.

Index (hex)	Name	Meaning	Data type	Flags	Default value
1804:0	ENC TxPDO-Par Timest.	PDO parameter TxPDO 5	UINT8	RO	0x06 (6_{dec})
1804:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 5.	OCTET-STRING[2]	RO	05 1A

Index 1805 ENC TxPDO-Par Timest. compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1805:0	ENC TxPDO-Par Timest. compact	PDO parameter TxPDO 6	UINT8	RO	0x06 (6_{dec})
1805:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 6.	OCTET-STRING[2]	RO	04 1A

Index 1A00 ENC TxPDO-Map Status

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A00:0	ENC TxPDO-Map Status	PDO Mapping TxPDO 1	UINT8	RO	0x0D (13_{dec})
1A00:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x01 (Latch C valid))	UINT32	RO	0x0000:00, 2
1A00:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x02 (Latch extern valid))	UINT32	RO	0x6000:03, 1
1A00:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x03 (Set counter done))	UINT32	RO	0x0000:00, 3
1A00:04	SubIndex 004	4. PDO Mapping entry (4 bits align)	UINT32	RO	0x6000:07, 1
1A00:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x08 (Extrapolation stall))	UINT32	RO	0x6000:08, 1
1A00:06	SubIndex 006	6. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x09 (Status of input A))	UINT32	RO	0x6000:09, 1
1A00:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x0A (Status of input B))	UINT32	RO	0x6000:0A, 1
1A00:08	SubIndex 008	8. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x0B (Status of input C))	UINT32	RO	0x0000:00, 3
1A00:09	SubIndex 009	9. PDO Mapping entry (1 bits align)	UINT32	RO	0x6000:0E, 1
1A00:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x0D (Status of extern latch))	UINT32	RO	0x6000:0F, 1
1A00:0B	SubIndex 011	11. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x0E (Sync error))	UINT32	RO	0x6000:10, 1
1A00:0C	SubIndex 012	12. PDO Mapping entry (1 bits align)	UINT32	RO	0x6000:11, 32
1A00:0D	SubIndex 013	13. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x10 (TxPDO-Toggle))	UINT32	RO	0x0000:00, 32

Index 1A01 ENC TxPDO-Map Status compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A01:0	ENC TxPDO-Map Status compact	PDO Mapping TxPDO 2	UINT8	RO	0x0D (13_{dec})
1A01:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x01 (Latch C valid))	UINT32	RO	0x0000:00, 2
1A01:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x02 (Latch extern valid))	UINT32	RO	0x6000:03, 1
1A01:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x03 (Set counter done))	UINT32	RO	0x0000:00, 3
1A01:04	SubIndex 004	4. PDO Mapping entry (4 bits align)	UINT32	RO	0x6000:07, 1
1A01:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x08 (Extrapolation stall))	UINT32	RO	0x6000:08, 1
1A01:06	SubIndex 006	6. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x09 (Status of input A))	UINT32	RO	0x6000:09, 1
1A01:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x0A (Status of input B))	UINT32	RO	0x6000:0A, 1
1A01:08	SubIndex 008	8. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x0B (Status of input C))	UINT32	RO	0x0000:00, 3
1A01:09	SubIndex 009	9. PDO Mapping entry (1 bits align)	UINT32	RO	0x6000:0E, 1
1A01:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x0D (Status of extern latch))	UINT32	RO	0x6000:0F, 1
1A01:0B	SubIndex 011	11. PDO Mapping entry (object 0x6000 (ENC Inputs entry 0x0E (Sync error))	UINT32	RO	0x6000:10, 1
1A01:0C	SubIndex 012	12. PDO Mapping entry (1 bits align)	UINT32	RO	0x6000:11, 16
1A01:0D	SubIndex 013	13. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x10 (TxPDO-Toggle))	UINT32	RO	0x0000:00, 16

Index 1A02 ENC TxPDO-Map Period

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A02:0	ENC TxPDO-Map Period	PDO Mapping TxPDO 3	UINT8	RO	0x01 (1_{dec})
1A02:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x14 (Period value))	UINT32	RO	0x6000:14, 32

Index 1A03 ENC TxPDO-Map Frequency

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A03:0	ENC TxPDO-Map Frequency	PDO Mapping TxPDO 4	UINT8	RO	0x01 (1_{dec})
1A03:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x13 (Frequency value))	UINT32	RO	0x6000:13, 32

Index 1A04 ENC TxPDO-Map Timest.

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A04:0	ENC TxPDO-Map Timest.	PDO Mapping TxPDO 5	UINT8	RO	0x01 (1_{dec})
1A04:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x16 (Timestamp))	UINT32	RO	0x6000:16, 64

Index 1A05 ENC TxPDO-Map Timest. compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A05:0	ENC TxPDO-Map Timest. compact	PDO Mapping TxPDO 6	UINT8	RO	0x01 (1_{dec})
1A05:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x16 (Timestamp))	UINT32	RO	0x6000:16, 32

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default value
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x01 (1_{dec})
1C32:01	SubIndex 001	1. 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 value
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x01 (1_{dec})
1C13:01	SubIndex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A01 (6657_{dec})
1C13:02	SubIndex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:03	SubIndex 003	3. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})

Index 1C32 SM output parameter

Index (hex)	Name	Meaning	Data type	Flags	Default value
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: Synchronous with SM 2 event • 2: DC-Mode - Synchronous with SYNC0 Event • 3: DC-Mode - Synchronous with SYNC1 event	UINT16	RW	0x0001 (1_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchronous with SM 2 event: Master cycle time • DC-Mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	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: Synchronous with SM 2 event is supported • Bit 2-3 = 01: DC mode is supported • Bit 14 = 1: dynamic times (measurement by writing 0x1C32:08) (for revision no.: 17 – 25)	UINT16	RO	0xC807 (51207_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns) Default: 59.1ms	UINT32	RO	0x0001E848 (125000_{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		UINT32	RO	0x00000000 (0_{dec})
1C32:08	Command	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 0x1C32:03, 0x1C32:05, 0x1C32:06, 0x1C32:09, 0x1C33:03 [▶ 120], 0x1C33:06 and 0x1C33:09 [▶ 120] 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})
1C32:20	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 value
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32_{dec})
1C33:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchronous with SM 3 event (no outputs available) • 2: DC - Synchronous with SYNC0 Event • 3: DC - Synchronous with SYNC1 Event • 34: Synchronous with SM 2 event (outputs available)	UINT16	RW	0x0022 (34_{dec})
1C33:02	Cycle time	as 0x1C32:02 [▶ 119]	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 = 1: free run is supported • Bit 1 = 1: Synchronous with SM 2 event is supported • Bit 2-3 = 01: DC mode is supported • Bit 14 = 1: dynamic times (measurement by writing 0x1C32:08 [▶ 119]) (for revision no.: 17 – 25)	UINT16	RO	0xC807 (51207_{dec})
1C33:05	Minimum cycle time	as 1C32:05 [▶ 119]	UINT32	RO	0x0001E848 (125000_{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	0x0001E848 (125000_{dec})
1C33:07	Minimum delay time		UINT32	RO	0x00000000 (0_{dec})
1C33:08	Command	as 0x1C32:08 [▶ 119]	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	as 0x1C32:11 [▶ 119]	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	as 0x1C32:12 [▶ 119]	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	as 0x1C32:13 [▶ 119]	UINT16	RO	0x0000 (0_{dec})
1C33:20	Sync error	as 0x1C32:32 [▶ 119]	BOOLEAN	RO	0x00 (0_{dec})

Index A000 ENC Diag data

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
A000:0	MDP Profile Compatibility	Maximaler Subindex	UINT8	RO	0x02 (2_{dez})
A000:01	Open circuit A		BOOLEAN	RO	0x00 (0_{dez})
A000:02	Open circuit B		BOOLEAN	RO	0x00 (0_{dez})

Index F000 Modular device profile

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

Index F008 Code word

Index (hex)	Name	Meaning	Data type	Flags	Default value
F008:0	Code word	NoCoeStorage function: The input code of the code word 0x12345678 activates the NoCoeStorage function: Changes to the CoE directory are not saved if the function is active. The function is deactivated by: 1.) changing the code word or 2.) restarting the terminal.	UINT32	RW	0x00000000 (0_{dec})

Index F010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default value
F010:0	Module list	Maximum subindex	UINT8	RW	0x01 (1_{dec})
F010:01	SubIndex 001	reserved	UINT32	RW	0x000001FF (511_{dec})

Index F082 MDP Profile Compatibility

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
F082:0	MDP Profile Compatibility	Maximaler Subindex	UINT8	RO	0x01 (1_{dez})
F082:01	Compatible input cycle counter	reserviert	BOOLEAN	RW	0x00 (0_{dez})

5.5 ELX5151-0090 - CoE object description

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 [▶ 98] (double-click on the respective object) or via the Process Data tab [▶ 95](allocation of PDOs). Please note the following general CoE notes when using/manipulating the CoE parameters:

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

Introduction

The CoE overview contains objects for different intended applications:

5.5.1 Restore object

Index 1011 Restore default parameters

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

5.5.2 Configuration data

Index 8000 ENC Settings

Index (hex)	Name	Meaning	Data type	Flags	Default value
8000:0	ENC Settings	Maximum subindex	UINT8	RO	0x17 (23_{dec})
8000:03	Enable up/down counter	Enablement of the up/down counter in place of the encoder with the bit set.	BOOLEAN	RW	0x00 (0_{dec})
8000:0A	Enable micro increments	The counter value is extrapolated by 8 bit.	BOOLEAN	RW	0x00 (0_{dec})
8000:0B	Open Circuit detection A		BOOLEAN	RW	0x00 (0_{dec})
8000:0C	Open Circuit detection B		BOOLEAN	RW	0x00 (0_{dec})
8000:0E	Reversion of rotation	Activates reversion of rotation	BOOLEAN	RW	0x00 (0_{dec})
8000:0F	Frequency window base	Basic unit of *Frequency window* (index 0x8000:11) 0: µs 1: ms	BIT1	RW	0x00 (0_{dec})
8000:11	Frequency window	This is the minimum time over which the frequency is determined [1 µs], default: 10 ms. Measuring window < 600 ms: Measurement takes place in frequency mode A. Measuring window > 600 ms: Measurement takes place in frequency mode B. The frequency determined in this way is output in index 0x6000:13 [▶ 124] .	UINT16	RW	0x2710 (10000_{dec})
8000:13	Frequency scaling	Scaling of the frequency measurement (must be divided by this value to obtain the unit in Hz): 100: “0.01” Hz	UINT16	RW	0x0064 (100_{dec})
8000:14	Period scaling	Scaling of the period in the process data: (must be divided by this value to obtain the unit in ns): 100: “100 ns” period value is a multiple of 100 ns Only the setting “100” is currently possible here.	UINT16	RW	0x0064 (100_{dec})
8000:15	Frequency resolution	Resolution of the frequency measurement: 100: “0.01 Hz”	UINT16	RW	0x0064 (100_{dec})
8000:16	Period resolution	Internal resolution of the period measurement: 100: “100 ns” 200: “200 ns” The period is calculated internally with a resolution of 100 ns. The max. measurable period is 1.6 s. Only 100 ns and 200 ns can be set.	UINT16	RW	0x00C8 (200_{dec})
8000:17	Frequency Wait Time	Waiting time [ms] for frequency measurement Default: 1.6 s (maximum possible value) If the time from Frequency window has elapsed, the next positive edge from track A is awaited for this time (frequency mode A only). This enables the update speed for the Frequency process data to be optimized, depending on the expected frequencies. At least double the period of the minimum frequency to be measured should be entered here. t >= 2* (1 / f_{min}).	UINT16	RW	0x0640 (1600_{dec})

Also see about this

- Configuration data [▶ 123]

5.5.3 Input data

Index 6000 ENC Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default value
6000:0	ENC Inputs	Maximum subindex	UINT8	RO	0x1E (30_{dec})
6000:03	Set counter done	The counter was set.	BOOLEAN	ROP	0x00 (0_{dec})
6000:07	Open Circuit		BOOLEAN	ROP	0x00 (0_{dec})
6000:08	Extrapolation stall	The extrapolated part of the counter is invalid. The speed has fallen below the minimum speed required to use the micro-increments.	BOOLEAN	ROP	0x00 (0_{dec})
6000:09	Status of input A	Status of input A	BOOLEAN	ROP	0x00 (0_{dec})
6000:0A	Status of input B	Status of input B	BOOLEAN	ROP	0x00 (0_{dec})
6000:0E	Sync Error	The *Sync error* bit is only required for DC mode. It indicates whether a synchronization error has occurred during the previous cycle.	BOOLEAN	ROP	0x00 (0_{dec})
6000:0F	TxPDO State		BOOLEAN	ROP	0x00 (0_{dec})
6000:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated.	BOOLEAN	ROP	0x00 (0_{dec})
6000:11	Counter value	Counter value	UINT32	ROP	0x00000000 (0_{dec})
6000:13	Frequency value	The frequency (setting of the scaling in index 0x8000:13 and the resolution in index 0x8000:15)	UINT32	ROP	0x00000000 (0_{dec})
6000:14	Period value	The period (setting of the scaling in index 0x8000:14 and the resolution in index 0x8000:16)	UINT32	ROP	0x00000000 (0_{dec})
6000:16	Timestamp	Time stamp of the last counter change.	UINT64	ROP	0x0000000000000000 (0_{dec})
6000:1C	Frequency value (uint 16)	The frequency (16 bit Value)	UINT16	ROP	0x0000 (0_{dec})
6000:1D	Counter value (uint 16)	Counter value (16 bit Value)	UINT16	ROP	0x0000 (0_{dec})
6000:1E	Period value (uint 16)	The period (16 bit Value)	UINT16	ROP	0x0000 (0_{dec})

Also see about this

- Input data [▶ 124]
- Output data [▶ 124]

5.5.4 Output data

Index 7000 ENC Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default value
7000:0	ENC Outputs	Maximum subindex	UINT8	RO	0x11 (17_{dec})
7000:03	Set counter	Set counter	BOOLEAN	RO P	0x00 (0_{dec})
7000:11	Set counter value	This is the counter value to be set via *Set counter* (index 0x7000:03 [▶ 124]).	UINT32	RO P	0x00000000 (0_{dec})

5.5.5 Standard objects (0x1000-0x1FFF)

Index 1000 Device type

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

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default value
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	ELX5151-0090

Index 1009 Hardware version

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

Index 100A Software version

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

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default value
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	0x970C3FF9 (2534162425_{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 value
10F0:0	Backup parameter handling	Information for standardized loading and saving of backup entries	UINT8	RO	0x01 (1_{dec})
10F0:01	Checksum	Checksum across all backup entries of the EtherCAT slave	UINT32	RO	0x00000000 (0_{dec})

Index 1400 ENC RxPDO-Par Control

Index (hex)	Name	Meaning	Data type	Flags	Default value
1400:0	ENC RxPDO-Par Control	PDO Parameter RxPDO 1	UINT8	RO	0x06 (6_{dec})
1400:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 1	OCTET-STRING[2]	RO	01 16

Index 1401 ENC RxPDO-Par Control compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1401:0	ENC RxPDO-Par Control compact	PDO Parameter RxPDO 2	UINT8	RO	0x06 (6_{dec})
1401:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 2	OCTET-STRING[2]	RO	00 16

Index 1600 ENC RxPDO-Map Control

Index (hex)	Name	Meaning	Data type	Flags	Default value
1600:0	ENC RxPDO-Map Control	PDO Mapping RxPDO 1	UINT8	RO	0x05 (5_{dec})
1600:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (ENC Outputs), entry 0x01 (Enable latch C))	UINT32	RO	0x0000:00, 2
1600:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (ENC Outputs), entry 0x02 (Enable latch extern on positive edge))	UINT32	RO	0x7000:03, 1
1600:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (ENC Outputs), entry 0x03 (Set counter))	UINT32	RO	0x0000:00, 5
1600:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (ENC Outputs), entry 0x04 (Enable latch extern on negative edge))	UINT32	RO	0x0000:00, 8
1600:05	SubIndex 005	5. PDO Mapping entry (4 bits align)	UINT32	RO	0x7000:11, 32

Index 1601 ENC RxPDO-Map Control compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1601:0	ENC RxPDO-Map Control compact	PDO Mapping RxPDO 2	UINT8	RO	0x05 (5_{dec})
1601:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (ENC Outputs), entry 0x01 (Enable latch C))	UINT32	RO	0x0000:00, 2
1601:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (ENC Outputs), entry 0x02 (Enable latch extern on positive edge))	UINT32	RO	0x7000:03, 1
1601:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (ENC Outputs), entry 0x03 (Set counter))	UINT32	RO	0x0000:00, 5
1601:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (ENC Outputs), entry 0x04 (Enable latch extern on negative edge))	UINT32	RO	0x0000:00, 8
1601:05	SubIndex 005	5. PDO Mapping entry (4 bits align)	UINT32	RO	0x7000:11, 16

Index 1800 ENC TxPDO-Par Status

Index (hex)	Name	Meaning	Data type	Flags	Default value
1800:0	ENC TxPDO-Par Status	PDO parameter TxPDO 1	UINT8	RO	0x06 (6_{dec})
1800:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 1.	OCTET-STRING[2]	RO	01 1A

Index 1801 ENC TxPDO-Par Status compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1801:0	ENC TxPDO-Par Status compact	PDO parameter TxPDO 2	UINT8	RO	0x06 (6_{dec})
1801:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 2.	OCTET-STRING[2]	RO	00 1A

Index 1802 ENC TxPDO-Par Period

Index (hex)	Name	Meaning	Data type	Flags	Default value
1802:0	ENC TxPDO-Par Period	PDO parameter TxPDO 3	UINT8	RO	0x06 (6_{dec})
1802:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 3.	OCTET-STRING[2]	RO	03 1A

Index 1803 ENC TxPDO-Par Frequency

Index (hex)	Name	Meaning	Data type	Flags	Default value
1803:0	ENC TxPDO-Par Frequency	PDO parameter TxPDO 4	UINT8	RO	0x06 (6_{dec})
1803:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 4.	OCTET-STRING[2]	RO	02 1A

Index 1804 ENC TxPDO-Par Timest.

Index (hex)	Name	Meaning	Data type	Flags	Default value
1804:0	ENC TxPDO-Par Timest.	PDO parameter TxPDO 5	UINT8	RO	0x06 (6_{dec})
1804:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 5.	OCTET-STRING[2]	RO	05 1A

Index 1805 ENC TxPDO-Par Timest. compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1805:0	ENC TxPDO-Par Timest. compact	PDO parameter TxPDO 6	UINT8	RO	0x06 (6_{dec})
1805:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 6.	OCTET-STRING[2]	RO	04 1A

Index 1A00 ENC TxPDO-Map Status

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A00:0	ENC TxPDO-Map Status	PDO Mapping TxPDO 1	UINT8	RO	0x0D (13_{dec})
1A00:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x01 (Latch C valid))	UINT32	RO	0x0000:00, 2
1A00:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x02 (Latch extern valid))	UINT32	RO	0x6000:03, 1
1A00:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x03 (Set counter done))	UINT32	RO	0x0000:00, 3
1A00:04	SubIndex 004	4. PDO Mapping entry (4 bits align)	UINT32	RO	0x6000:07, 1
1A00:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x08 (Extrapolation stall))	UINT32	RO	0x6000:08, 1
1A00:06	SubIndex 006	6. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x09 (Status of input A))	UINT32	RO	0x6000:09, 1
1A00:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x0A (Status of input B))	UINT32	RO	0x6000:0A, 1
1A00:08	SubIndex 008	8. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x0B (Status of input C))	UINT32	RO	0x0000:00, 3
1A00:09	SubIndex 009	9. PDO Mapping entry (1 bits align)	UINT32	RO	0x6000:0E, 1
1A00:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x0D (Status of extern latch))	UINT32	RO	0x6000:0F, 1
1A00:0B	SubIndex 011	11. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x0E (Sync error))	UINT32	RO	0x6000:10, 1
1A00:0C	SubIndex 012	12. PDO Mapping entry (1 bits align)	UINT32	RO	0x6000:11, 32
1A00:0D	SubIndex 013	13. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x10 (TxPDO-Toggle))	UINT32	RO	0x0000:00, 32

Index 1A01 ENC TxPDO-Map Status compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A01:0	ENC TxPDO-Map Status compact	PDO Mapping TxPDO 2	UINT8	RO	0x0D (13_{dec})
1A01:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x01 (Latch C valid))	UINT32	RO	0x0000:00, 2
1A01:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x02 (Latch extern valid))	UINT32	RO	0x6000:03, 1
1A01:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x03 (Set counter done))	UINT32	RO	0x0000:00, 3
1A01:04	SubIndex 004	4. PDO Mapping entry (4 bits align)	UINT32	RO	0x6000:07, 1
1A01:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x08 (Extrapolation stall))	UINT32	RO	0x6000:08, 1
1A01:06	SubIndex 006	6. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x09 (Status of input A))	UINT32	RO	0x6000:09, 1
1A01:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x0A (Status of input B))	UINT32	RO	0x6000:0A, 1
1A01:08	SubIndex 008	8. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x0B (Status of input C))	UINT32	RO	0x0000:00, 3
1A01:09	SubIndex 009	9. PDO Mapping entry (1 bits align)	UINT32	RO	0x6000:0E, 1
1A01:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x0D (Status of extern latch))	UINT32	RO	0x6000:0F, 1
1A01:0B	SubIndex 011	11. PDO Mapping entry (object 0x6000 (ENC Inputs entry 0x0E (Sync error))	UINT32	RO	0x6000:10, 1
1A01:0C	SubIndex 012	12. PDO Mapping entry (1 bits align)	UINT32	RO	0x6000:11, 16
1A01:0D	SubIndex 013	13. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x10 (TxPDO-Toggle))	UINT32	RO	0x0000:00, 16

Index 1A02 ENC TxPDO-Map Period

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A02:0	ENC TxPDO-Map Period	PDO Mapping TxPDO 3	UINT8	RO	0x01 (1_{dec})
1A02:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x14 (Period value))	UINT32	RO	0x6000:14, 32

Index 1A03 ENC TxPDO-Map Frequency

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A03:0	ENC TxPDO-Map Frequency	PDO Mapping TxPDO 4	UINT8	RO	0x01 (1_{dec})
1A03:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x13 (Frequency value))	UINT32	RO	0x6000:13, 32

Index 1A04 ENC TxPDO-Map Timest.

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A04:0	ENC TxPDO-Map Timest.	PDO Mapping TxPDO 5	UINT8	RO	0x01 (1_{dec})
1A04:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x16 (Timestamp))	UINT32	RO	0x6000:16, 64

Index 1A05 ENC TxPDO-Map Timest. compact

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A05:0	ENC TxPDO-Map Timest. compact	PDO Mapping TxPDO 6	UINT8	RO	0x01 (1_{dec})
1A05:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x16 (Timestamp))	UINT32	RO	0x6000:16, 32

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default value
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x02 (2_{dec})
1C32:01	SubIndex 001	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1601 (5633_{dec})
1C32:02	SubIndex 002	2. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1602 (5634_{dec})

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default value
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	0x1A01 (6657_{dec})
1C13:02	SubIndex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A06 (6662_{dec})
1C13:03	SubIndex 003	3. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:04	SubIndex 004	4. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})

Index 1C32 SM output parameter

Index (hex)	Name	Meaning	Data type	Flags	Default value
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: Synchronous with SM 2 event • 2: DC-Mode - Synchronous with SYNC0 Event • 3: DC-Mode - Synchronous with SYNC1 event	UINT16	RW	0x0001 (1_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchronous with SM 2 event: Master cycle time • DC-Mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	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: Synchronous with SM 2 event is supported • Bit 2-3 = 01: DC mode is supported • Bit 14 = 1: dynamic times (measurement by writing 0x1C32:08) (for revision no.: 17 – 25)	UINT16	RO	0xC807 (51207_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns) Default: 100 ms	UINT32	RO	0x0001E848 (125000_{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	Minimum time between Sync-1 Event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C32:08	Command	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 0x1C32:03, 0x1C32:05, 0x1C32:06, 0x1C32:09, 0x1C33:03 [▶ 131], 0x1C33:06 and 0x1C33:09 [▶ 131] 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})
1C32:20	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 value
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32_{dec})
1C33:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchronous with SM 3 event (no outputs available) • 2: DC - Synchronous with SYNC0 Event • 3: DC - Synchronous with SYNC1 Event • 34: Synchronous with SM 2 event (outputs available)	UINT16	RW	0x0022 (34_{dec})
1C33:02	Cycle time	as 0x1C32:02	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 = 1: free run is supported • Bit 1 = 1: Synchronous with SM 2 event is supported • Bit 2-3 = 01: DC mode is supported • Bit 14 = 1: dynamic times (measurement by writing 0x1C32:08) (for revision no.: 17 – 25)	UINT16	RO	0xC807 (51207_{dec})
1C33:05	Minimum cycle time	as 1C32:05	UINT32	RO	0x0001E848 (125000_{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	0x0001E848 (125000_{dec})
1C33:07	Minimum delay time	Minimum time between Sync-1 Event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:08	Command	as 0x1C32:08	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	as 0x1C32:11	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	as 0x1C32:12	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	as 0x1C32:13	UINT16	RO	0x0000 (0_{dec})
1C33:20	Sync error	as 0x1C32:32	BOOLEAN	RO	0x00 (0_{dec})

Index A000 ENC Diag data

Index (hex)	Name	Bedeutung	Datentyp	Flags	Default
A000:0	MDP Profile Compatibility	Maximaler Subindex	UINT8	RO	0x02 (2_{dez})
A000:01	Open circuit A		BOOLEAN	RO	0x00 (0_{dez})
A000:02	Open circuit B		BOOLEAN	RO	0x00 (0_{dez})

Index F000 Modular device profile

Index (hex)	Name	Meaning	Data type	Flags	Default value
F000:0	Modular device profile	General information for the modular device profile	UINT8	RO	0x02 (2_{dec})
F000:01	Module index distance	Index spacing 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 value
F008:0	Code word	NoCoeStorage function: The input code of the code word 0x12345678 activates the NoCoeStorage function: Changes to the CoE directory are not saved if the function is active. The function is deactivated by: 1.) changing the code word or 2.) restarting the terminal.	UINT32	RW	0x00000000 (0_{dec})

Index F010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default value
F010:0	Module list	Maximum subindex	UINT8	RW	0x02 (2_{dec})
F010:01	SubIndex 001	reserved	UINT32	RW	0x000001FF (511_{dec})
F010:02	SubIndex 002	reserved	UINT32	RW	0x000003B6 (950_{dec})

Index F082 MDP Profile Compatibility

Index (hex)	Name	Meaning	Data type	Flags	Default
F082:0	MDP profile compatibility	Maximum subindex	UINT8	RO	0x01 (1_{dec})
F082:01	Compatible input cycle counter	reserved	BOOLEAN	RW	0x00 (0_{dec})

5.5.6 Objects TwinSAFE Single Channel (ELX5151-0090)

Index 1602 TSC RxPDO-Map Master Message

Index (hex)	Name	Meaning	Data type	Flags	Default
1602:0	TSC RxPDO-Map Master Message	PDO Mapping RxPDO 17	UINT8	RO	0x04 (4_{dec})
1602:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (TSC Master Frame Elements), entry 0x01 (TSC___Master Cmd))	UINT32	RO	0x7010:01, 8
1602:02	SubIndex 002	2. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1602:03	SubIndex 003	3. PDO Mapping entry (object 0x7010 (TSC Master Frame Elements), entry 0x03 (TSC__Master CRC_0))	UINT32	RO	0x7010:03, 16
1602:04	SubIndex 004	4. PDO Mapping entry (object 0x7010 (TSC Master Frame Elements), entry 0x02 (TSC__Master ConnID))	UINT32	RO	0x7010:02, 16

Index 1A06 TSC TxPDO-Map Slave Message

Index (hex)	Name	Meaning	Data type	Flags	Default
1A06:0	TSC TxPDO-Map Slave Message	PDO Mapping TxPDO	UINT8	RW	0x4 (4_{dec})
1A06:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (TSC Slave Frame Elements), entry 0x01 (TSC___Slave Cmd))	UINT32	RW	0x6010:01, 8
1A06:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x1D (Counter value (uint 16)))	UINT32	RW	0x6000:1D, 16
1A06:03	SubIndex 003	3. PDO Mapping entry (object 0x6010 (TSC Slave Frame Elements), entry 0x03 (TSC__Slave CRC_0))	UINT32	RW	0x6010:03, 16
1A06:04	SubIndex 004	4. PDO Mapping entry (object 0x6010 (TSC Slave Frame Elements), entry 0x02 (TSC___Slave ConnID))	UINT32	RW	0x6010:02, 16
1A06:05	SubIndex 005	reserviert	UINT32	RW	0x0000:00
1A06:06	SubIndex 006	reserviert	UINT32	RW	0x0000:00
1A06:07	SubIndex 007	reserviert	UINT32	RW	0x0000:00
1A06:08	SubIndex 008	reserviert	UINT32	RW	0x0000:00
1A06:09	SubIndex 009	reserviert	UINT32	RW	0x0000:00
1A06:0A	SubIndex 010	reserviert	UINT32	RW	0x0000:00
1A06:0B	SubIndex 011	reserviert	UINT32	RW	0x0000:00
1A06:0C	SubIndex 012	reserviert	UINT32	RW	0x0000:00
1A06:0D	SubIndex 013	reserviert	UINT32	RW	0x0000:00
1A06:0E	SubIndex 014	reserviert	UINT32	RW	0x0000:00
1A06:0F	SubIndex 015	reserviert	UINT32	RW	0x0000:00
1A06:10	SubIndex 016	reserviert	UINT32	RW	0x0000:00
1A06:11	SubIndex 017	reserviert	UINT32	RW	0x0000:00
1A06:12	SubIndex 018	reserviert	UINT32	RW	0x0000:00
1A06:13	SubIndex 019	reserviert	UINT32	RW	0x0000:00
1A06:14	SubIndex 020	reserviert	UINT32	RW	0x0000:00

Index 6010 TSC Slave Frame Elements

Index (hex)	Name	Meaning	Data type	Flags	Default
6010:0	TSC Slave Frame Elements	Max. Subindex	UINT8	RO	0x0B (11_{dec})
6010:01	TSC__Slave Cmd	reserved	UINT8	RO P	0x00 (0_{dec})
6010:02	TSC__Slave ConnID	reserved	UINT16	RO P	0x0000 (0_{dec})
6010:03	TSC__Slave CRC_0	reserved	UINT16	RO P	0x0000 (0_{dec})
6010:04	TSC__Slave CRC_1	reserved	UINT16	RO P	0x0000 (0_{dec})
6010:05	TSC__Slave CRC_2	reserved	UINT16	RO P	0x0000 (0_{dec})
6010:06	TSC__Slave CRC_3	reserved	UINT16	RO P	0x0000 (0_{dec})
6010:07	TSC__Slave CRC_4	reserved	UINT16	RO P	0x0000 (0_{dec})
6010:08	TSC__Slave CRC_5	reserved	UINT16	RO P	0x0000 (0_{dec})
6010:09	TSC__Slave CRC_6	reserved	UINT16	RO P	0x0000 (0_{dec})
6010:0A	TSC__Slave CRC_7	reserved	UINT16	RO P	0x0000 (0_{dec})
6010:0B	TSC__Slave CRC_8	reserved	UINT16	RO P	0x0000 (0_{dec})

Index 7010 TSC Master Frame Elements

Index (hex)	Name	Meaning	Data type	Flags	Default
7010:0	TSC Master Frame Elements	Maximaler Subindex	UINT8	RO	0x03 (3_{dec})
7010:01	TSC__Master Cmd	reserved	UINT8	RO P	0x00 (0_{dec})
7010:02	TSC__Master ConnID	reserved	UINT16	RO P	0x0000 (0_{dec})
7010:03	TSC__Master CRC_0	reserved	UINT16	RO P	0x0000 (0_{dec})

Index 8010 TSC Settings

Index (hex)	Name	Meaning		Data type	Flags	Default
8010:0	TSC Settings	Max. Subindex		UINT8	RW	0x02 (2_{dec})
8010:01	Address	TwinSAFE SC Address		UINT16	RW	0x0000 (0_{dec})
8010:02	Connection Mode	Selection of the TwinSAFE SC CRC		UINT32	RW	0x00000000 (0_{dec})
		97039_{dec}	TwinSAFE SC CRC1 master			
		153375_{dec}	TwinSAFE SC CRC2 master			
		20469_{dec}	TwinSAFE SC CRC3 master			
		283633_{dec}	TwinSAFE SC CRC4 master			
		389589_{dec}	TwinSAFE SC CRC5 master			
		419387_{dec}	TwinSAFE SC CRC6 master			
		506061_{dec}	TwinSAFE SC CRC7 master			
		582077_{dec}	TwinSAFE SC CRC8 master			

5.6 NC - Configuration

Installation of the latest XML device description

Please ensure that you have installed the corresponding latest XML device description in TwinCAT. This can be downloaded from the Beckhoff Website and installed according to the installation instructions.

The configuration of the axes and linking in the TwinCAT System Manager (Config mode) are described below, taking the EL5151 as an example. Proceed as follows:

1. The terminal must already have been added manually under I/O devices or have been scanned in by the system (see section “Configuration set-up in TwinCAT”).
2. Click with the right mouse button on *NC Configuration -> Append task*.

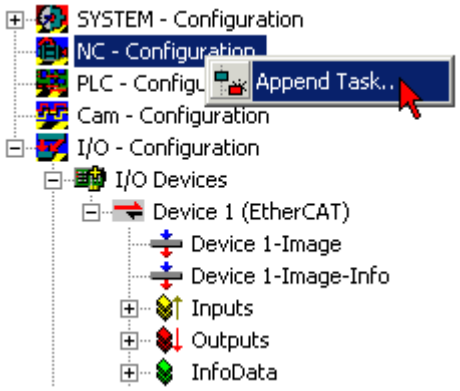


Fig. 147: NC - Configuration, Append Task

3. Select a name for the task and confirm with *OK*.

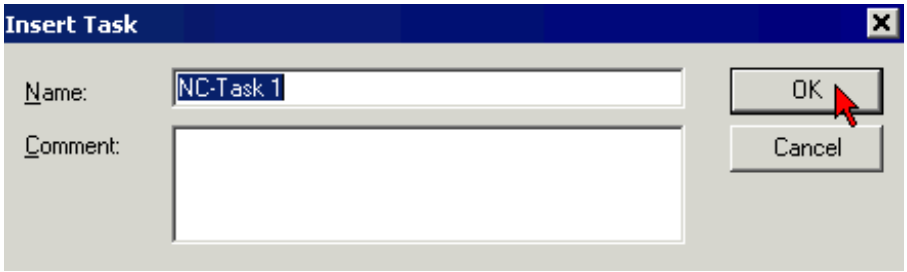


Fig. 148: Entering a name for the task and confirming

4. Click with the right mouse button on *Axes - >Append axis.*

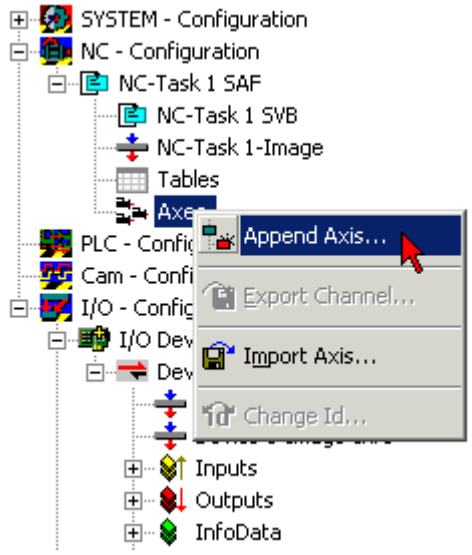


Fig. 149: Insert axis

5. Select a name and type for the axis and confirm with *OK*.

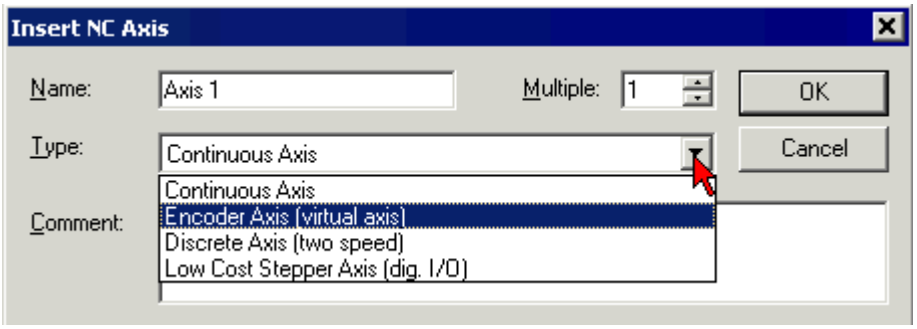


Fig. 150: Entering a name for the axis and selecting a type

6. After selecting the *NC-Encoder* tab, select the encoder *KL5101/KI5111/IP5109/EL5101* in the pull-down menu *Type*.

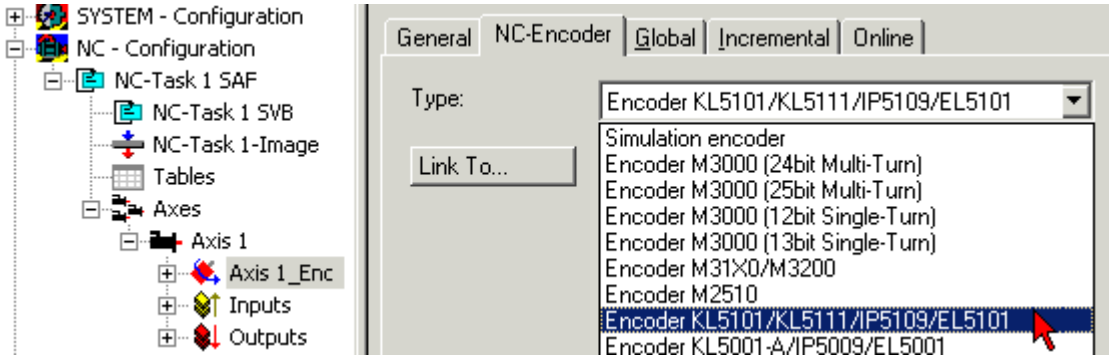


Fig. 151: Selecting the encoder

7. Click the *Linked with...* button, select the *EL5151* terminal and confirm with *OK*.

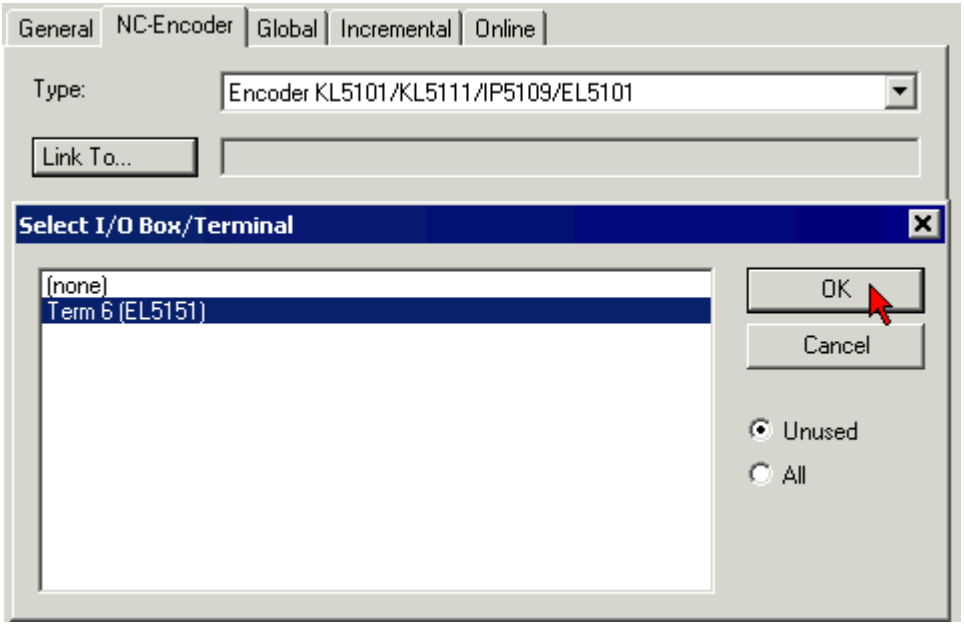


Fig. 152: Selecting and confirming an encoder terminal

8. The respective inputs of the EL5151 are now linked with the NC task.

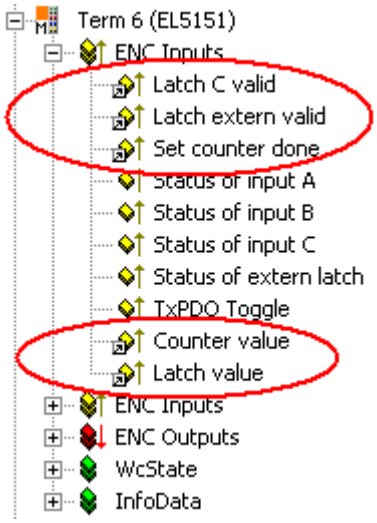


Fig. 153: EL5151 inputs linked with the NC task

5.7 Distributed Clocks - Settings

EtherCAT and Distributed Clocks

A basic introduction into EtherCAT and distributed clocks (DC) can is available for download from the Beckhoff website: EtherCAT-System documentation.

The incremental encoder terminals support the distributed clocks function. In order for the EL51xx to be able to make the current counter value available in the designated process data in time before the arrival of the querying EtherCAT datagram, a suitable signal must be generated cyclically within the terminal. This signal can be triggered in the EL51xx through two events:

1. the SyncManager (SM)
2. the Distributed Clock (DC)

Under operating mode selection the following options are available (see Fig. *“DC” tab*):

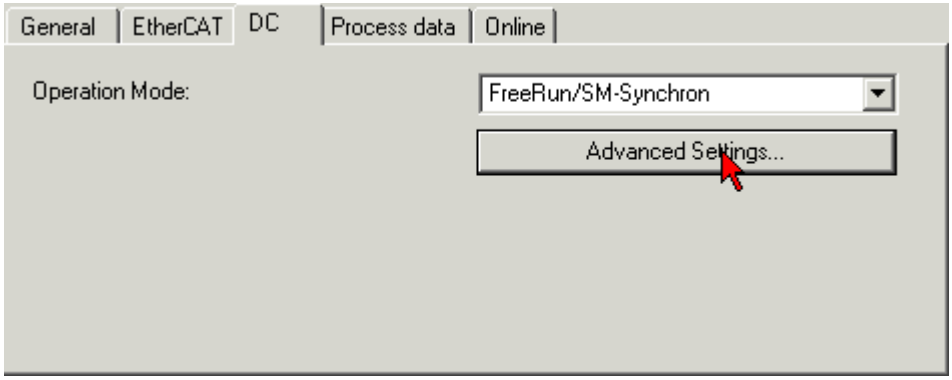


Fig. 154: *DC* tab (Distributed Clocks)

- **FreeRun/SM-Synchron**
  The SyncManager event occurs when an EtherCAT frame successfully exchanges process data with the EL51xx. Frame-triggered, the current counter value is thus cyclically determined, but with the low temporal jitter of the Ethernet frame. In this mode an Ethernet frame triggers the process data provision for the *next* retrieving frame. This generally only occurs after 1x cycle time.
- **DC Synchronous**
  In DC mode, a counter reading is triggered by the integrated DC unit with a constant cycle, usually in synchrony with the bus cycle, although with a constant shift (phase, shift time, offset). Sampling is significantly more uniform (synchronization accuracy: 100 ns), which means a higher-level control algorithm can be supplied with higher-quality position data, for example. In the ELX5151 the trigger is the SYNC0 signal, which is set like an output component in *DC-synchron* mode. See EtherCAT-System documentation -> Distributed clocks.
  The DC modes enable the start time of the process data provision to be offset by an offset value (shift value). This offset value can only by set on EtherCAT startup and can then no longer be changed during the uptime. Based on the general distributed clocks SYNC function model, the terminal-specific SYNC signal can either occur *before* or *after* the expect frame pass-through time:
- In the case of input terminals the SYNC signal is generated *before* the frame in order to make current input data available for onward transport.
- In the case of output terminals the SYNC signal is set to a time *after* the frame pass-through so that the output data just delivered can be output immediately.

Since only one of the two modes is possible, the user can set the optimum mode for his application. “DC Synchronous” corresponds to the output module configuration. The local SYNC event is triggered shortly after the EtherCAT frame has passed.

- **DC-Synchron (input based)**
  In the *DC-Synchronous (input based)* mode this ELX5151 is assigned to the group of input modules and the shift time (see Fig. *Advanced Distributed Clock (DC) settings, EL51xx terminal*) is calculated accordingly.

When *DC-Synchronous* operating mode is activated, TwinCAT selects settings that ensure reliable operation of the ELX5151 and the acquisition of current position data. This means that determination of the current counter value is started by the SYNC0 signal at highly constant intervals and in the operating mode *DC-Synchronous (input based)* in good time – i.e. with an adequate safety buffer – before the retrieving EtherCAT datagram.

Duration of the process data provision in the ELX5151

The ELX5151 requires approx. 80 µs after the SYNC event to determine the position data and provide them for retrieval. This value depends on the configuration and parameterization. Using the internal DC functions, the current actually required time can be read (see the CoE setting in 0x1C32:08) and the result written into 0x1C32:05.

If necessary, the SYNC0 signal can be shifted along the time axis to the right/later or left/earlier in associated dialogs by specifying a *User defined Shift Time*, see Fig. *ELX5151 - Advanced Settings, Distributed Clock (DC).*

- A right-shift (positive shift value) will delay the counter value query, which means the position value becomes more current from the PLC perspective. However, this increases the risk that the position determination may not be finished in time before the arrival of EtherCAT frame, so that no current position value is available in this cycle.
- A left-shift (negative shift value) means the counter value will be queried earlier, resulting in older position values, with an associated increase in the safety buffer before the arrival of the EtherCAT datagram. This setting may be useful in systems with high real-time jitter, if no Industrial PCs from Beckhoff are used for control purposes, for example.

NOTE

Caution! Risk of device damage!

The mentioned notes and information should be used advisedly. The EtherCAT master automatically allocates SYNC0 and SYNC1 settings that support reliable and timely process data acquisition. User intervention at this point may lead to undesired behavior. If these settings are changed in the System Manager, no plausibility checks are carried out on the software side. Correct function of the terminal with all conceivable setting options cannot be guaranteed.

Default setting

The cyclic read of the inputs is triggered by the SYNC0 pulse (interrupt) from the DC in the ELX5151. The EtherCAT master sets the Sync Unit Cycle time value to the PLC cycle time and therefore the EtherCAT cycle time as standard. See Fig. *ELX5151 - Advanced Settings, Distributed Clock (DC).*

4000 µs = 4 ms, as TwinCAT is in configuration mode.

ELX5151 - DC settings

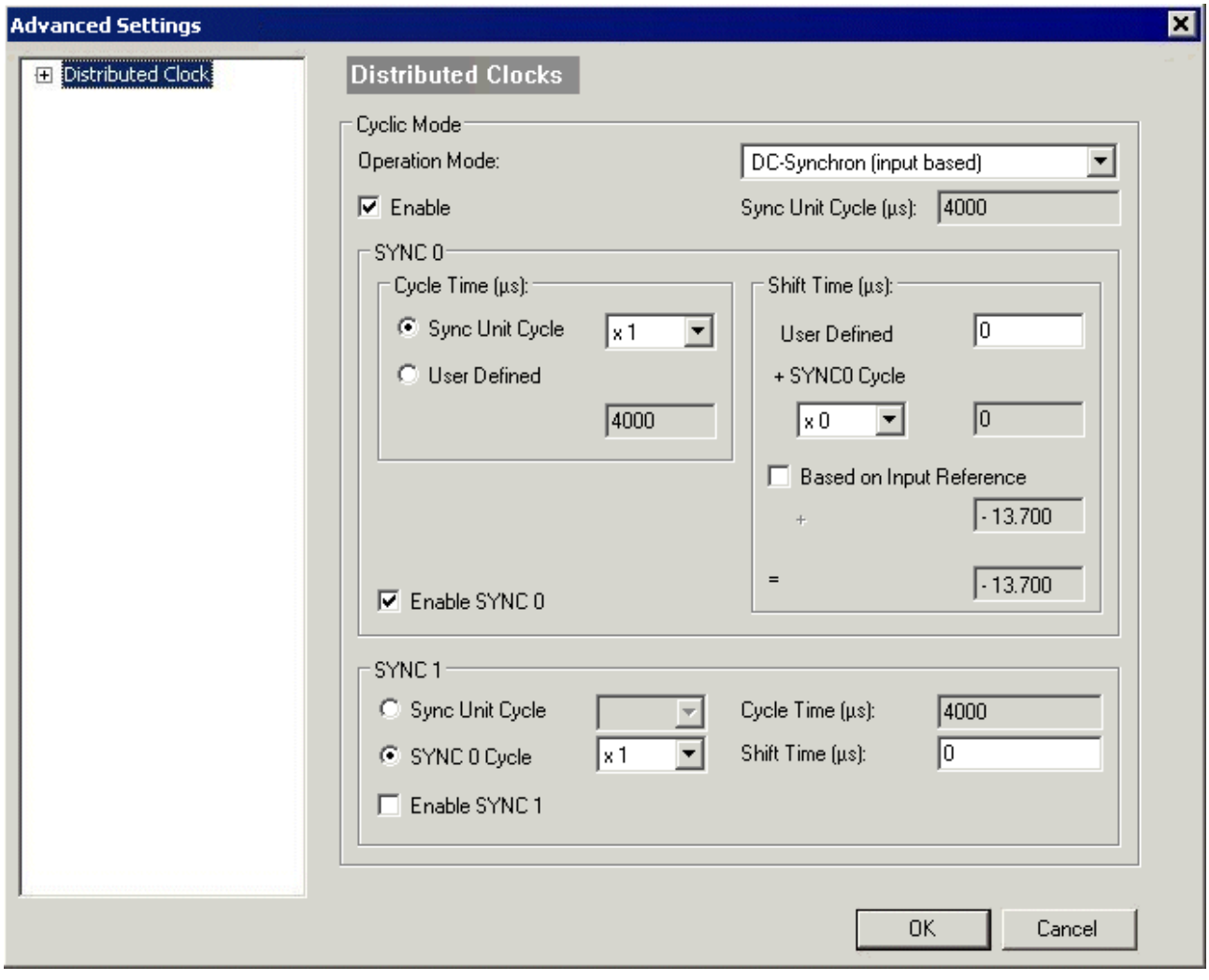
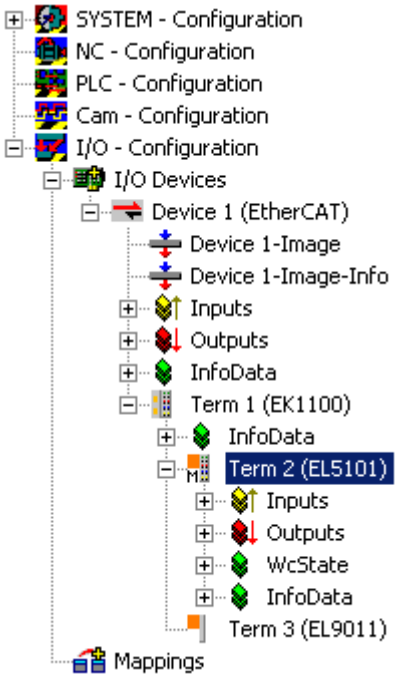


Fig. 155: ELX5151 - Advanced Settings, Distributed Clock (DC)

- **SYNC0**
  Sync unit cycle: a multiple of the bus cycle time. The counter value is periodically determined at this interval (in µs).
- **User-defined**
  Arbitrary number up to 2^{32} ns ≈ 4.3 secs. Decimal point values are possible.
- **Shift Time**
  The Shift Time can be used to shift the SYNC0 pulse for this ELX5151relative to other terminals and the global SYNC pulse in nanosecond steps. If the data of several ELX5151 terminals are to be read simultaneously, the same value must be entered here.
- **Based on input reference**
  If this option is activated an additional Input Shift is added to the configurable terminal-specific SYNC0 shift (user-defined). This value is calculated and made available by the EtherCAT master (SysMan/ EtherCAT device/EtherCAT tab/Advanced Settings/Distributed Clocks/Input Shift Time/, see Fig. *EtherCAT Master, EtherCAT tab, Advanced* + *EtherCAT Master, Advanced Settings, Distributed Clock*). In this way *all* input terminals in the system (EL1xxx, EL3xxx and appropriately set ELxxxx such as the EL51xx) read their inputs as close as possible to the time of the EtherCAT frame that will fetch them, thereby supplying the most recent possible input data to the controller. In input-based mode this value is taken into account automatically.
- **Enable SYNC0**
  Automatically activated in DC Synchronous operating mode.
- **SYNC1**
  Additional SYNC pulse, derived from SYNC0 or from the DC itself. Not required by the ELX5151.

DC settings for EtherCAT Master

Higher-level distributed clock parameters can be modified under advanced settings for the EtherCAT Master. Refer also to the basic introduction to the topic of EtherCAT and Distributed Clocks; download: the "EtherCAT-System documentation -> Distributed clocks".



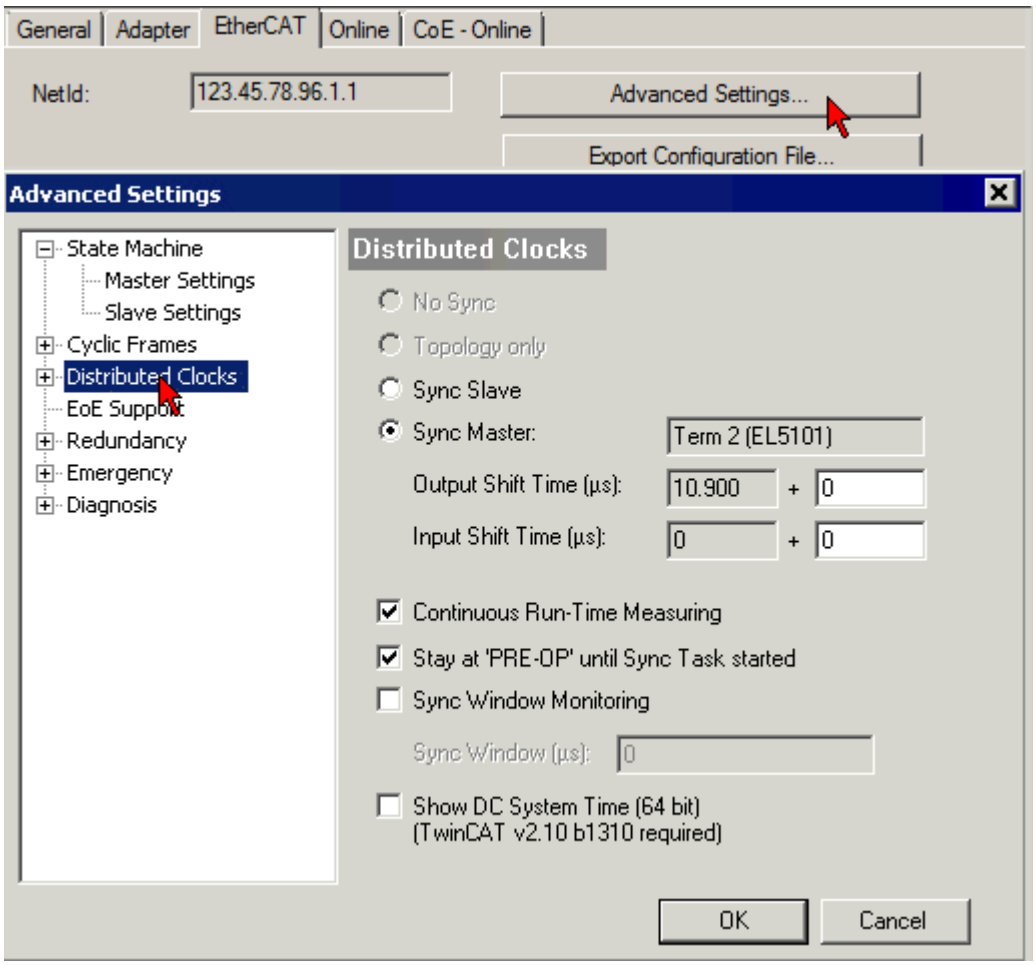


Fig. 156: EtherCAT Master, Advanced Settings, Distributed Clock

6 Appendix

6.1 EtherCAT AL Status Codes

For detailed information please refer to the EtherCAT system description.

6.2 UL notice

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

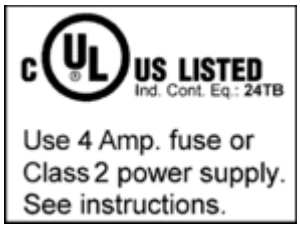
Basic principles

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

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

c UL US

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



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

Application

If terminals certified *with restrictions* are used, then the current consumption at 24 V_{DC} must be limited accordingly by means of supply

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

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

6.3 FM notice

Special notice regarding ANSI/ISA Ex

⚠ WARNING

Observe the permissible range of application!

The I/O modules of the ELX series may only be used in potentially explosive areas of Class I, Division 2, Group A, B, C, D or in non-explosive areas!

⚠ WARNING

Consider the *Control Drawing ELX* documentation!

When installing the I/O modules of the ELX series, be sure to read the *Control Drawing ELX* documentation, which is available in the download area of your ELX terminal on https://www.beckhoff.com/ELXxxxx!

6.4 Support and Service

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

Beckhoff's branch offices and representatives

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

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

Table of figures

More Information:
www.beckhoff.com/ELX5151

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