BECKHOFF New Automation Technology

Documentation | EN

EL1202, EL1252, EL1254

Digital Input Terminals (2 and 4 Channel, Distributed Clocks)

2020-08-06 | Version: 2.8

Table of contents

1 Foreword

1.1 Product overview: 2-/4-channel digital input terminals

EL1202 [▶ 15] 1-channel digital input terminal 24 V_{DC}, T_{ON}/T_{OFF} 1 µs
EL1202-0100 [▶ 15] 1-channel digital input terminal 24 V_{DC}, T_{ON}/T_{OFF} 1 µs, Distributed Clocks

EL1252 [▶ 17] 2-channel digital input terminal 24 V_{DC}, with time stamp
EL1252-0050 [▶ 17] 2-channel digital input terminal 5 V_{DC}, with time stamp

EL1254 [▶ 17] 4-channel digital input terminal 24 V_{DC}, with time

1.2 Notes on the documentation

Intended audience

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

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

Copyright

1.3 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

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.

1.4 Documentation issue status

Version	Comment
2.8	• Addenda chapter "Sensitivity of the input" • Update structure
2.7	• Correction chapter "EL1252, EL1254 - LEDs and pin assignment" • Update structure
2.6	• EL1254 added • Chapter "Beckhoff Identification Code (BIC) " added
2.5	• Update chapter "Commissioning"
2.4	• EL1202-0100 added • Update chapter "Technical data" • Update structure
2.3	• Update chapter "Commissioning"
2.2	• Update chapter "Technical data" • Update chapter "Connection system" ->"Connection" • Update structure
2.1	• Update chapter "Notes on the documentation" • Update chapter "Technical data" • Addenda chapter "Instructions for ESD protection" • Addenda chapter "Installation instructions for enhanced mechanical load capacity" • Addenda chapter "ATEX - Special conditions (extended temperature range)" • Addenda chapter "TwinCAT Quickstart" • Update chapter "TwinCAT 2.1x" -> "TwinCAT Development Environment" • Update chapter "EL1252 - Process data"
2.0	• First publication in PDF format
1.8	• Technical data update
1.7	• EL1250-0050 amended • Structural adaptation • Technical data update
1.6	• Firmware compatibility list inserted
1.5	• Notes on device description update amended; note on trademarks inserted
1.4	• Description of sample program amended
1.3	• Technical description amended
1.2	• Technical description amended
1.1	• Technical data and safety instructions amended
1.0	• Technical changes added, description of principle amended
0.1	• Provisional documentation for EL1202, EL1252

1.5 Version identification of EtherCAT devices

Designation

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

- family key
- type
- version
- revision

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

Notes

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

Identification number

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

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

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

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

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

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

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

Syntax: D ww yy x y z u

D - prefix designation
ww - calendar week
yy - year
x - firmware version of the bus PCB

y - hardware version of the bus PCB
z - firmware version of the I/O PCB
u - hardware version of the I/O PCB

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

Unique serial number/ID, ID number

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

See also the further documentation in the area

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

Examples of markings

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

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

Fig. 3: CU2016 switch with serial/ batch number

Fig. 4: EL3202-0020 with serial/ batch number 26131006 and unique ID-number 204418

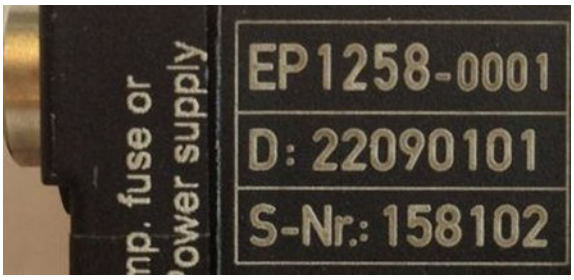


Fig. 5: EP1258-00001 IP67 EtherCAT Box with batch number/ date code 22090101 and unique serial number 158102

Fig. 6: EP1908-0002 IP67 EtherCAT Safety Box with batch number/ date code 071201FF and unique serial number 00346070

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

Fig. 8: ELM3604-0002 terminal with unique ID number (QR code) 100001051 and serial/ batch number 44160201

1.5.1 Beckhoff Identification Code (BIC)

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

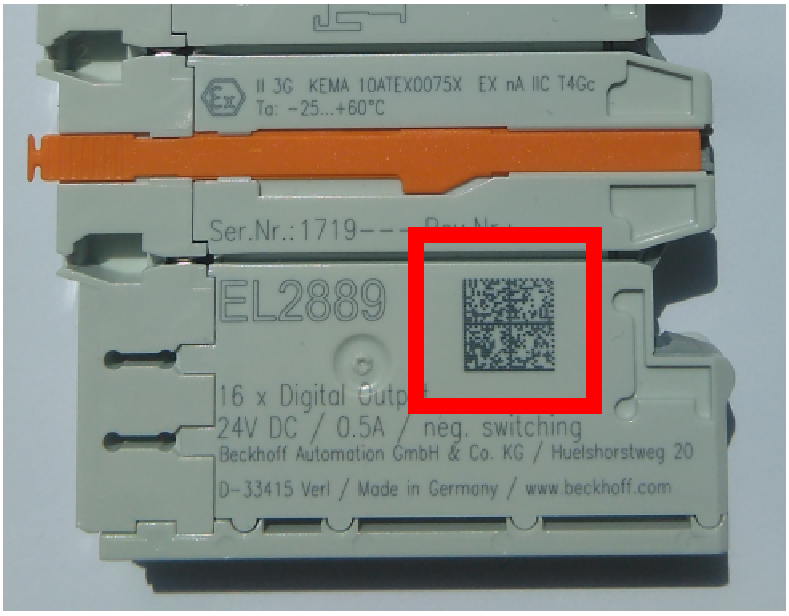


Fig. 9: BIC as data matrix code (DMC, code scheme ECC200)

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

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

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

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

Each piece of information can be uniquely identified using the so-called data identifier (ANSI MH10.8.2-2016). The data identifier is followed by a character string. Both together have a maximum length according to the table below. If the information is shorter, spaces are added to it. The data under positions 1 to 4 are always available.

The following information is contained:

Item no.	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	**Beckhoff order number**	1P	8	1P072222
2	Beckhoff Traceability Number (BTN)	**Unique serial number, see note below**	S	12	SBTNk4p562d7
3	Article description	**Beckhoff article description, e.g. EL1008**	1K	32	1KEL1809
4	Quantity	**Quantity in packaging unit, e.g. 1, 10, etc.**	Q	6	Q1
5	Batch number	Optional: Year and week of production	2P	14	2P401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	51S678294104
7	Variant number	Optional: Product variant number on the basis of standard products	30P	32	30PF971, 2*K183
...					

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

Structure of the BIC

Example of composite information from item 1 to 4 and 6. The data identifiers are marked in red for better display:

BTN

An important component of the BIC is the Beckhoff Traceability Number (BTN, item no. 2). The BTN is a unique serial number consisting of eight characters that will replace all other serial number systems at Beckhoff in the long term (e.g. batch designations on IO components, previous serial number range for safety products, etc.). The BTN will also be introduced step by step, so it may happen that the BTN is not yet coded in the BIC.

NOTE

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

2 Product overview

2.1 EL1202

2.1.1 EL1202 - Introduction

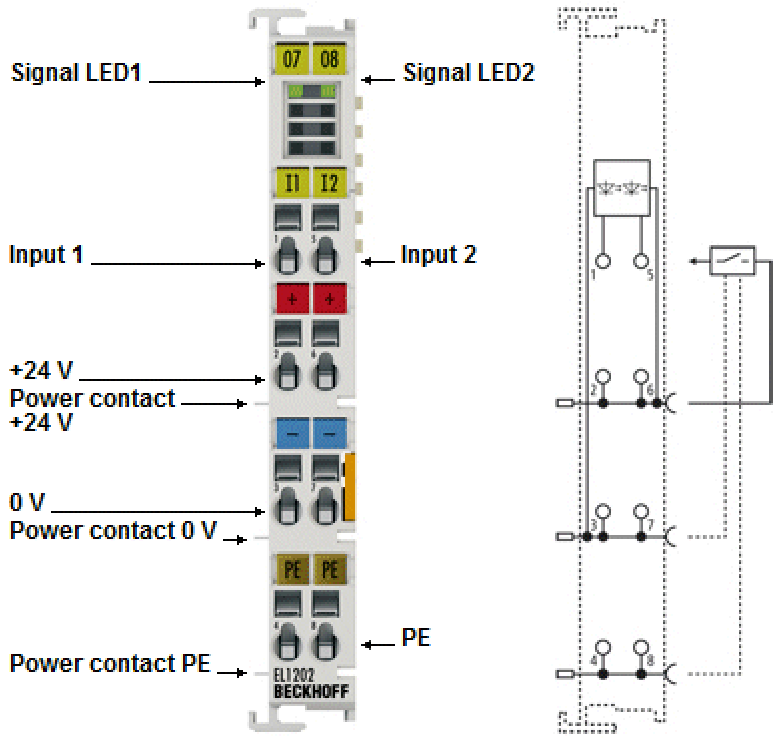


Fig. 10: EL1202

2-channel digital input terminal 24 V_{DC}, T_{ON}/T_{OFF} 1 µs

The EL1202 digital input terminal acquires the binary control signals from the process level and transmits them, in an electrically isolated form, to the higher-level automation unit. It is suitable for particularly fast signals due to its very low input delay. For the EL1202-0100 variant, Distributed Clocks are activated, i.e. the input data can be monitored synchronous with other data that are also linked to distributed clock terminals. Therefore, the accuracy across the system is <<1 µs. The EL1202 contains two channels, the signal state of which is indicated via LEDs.

Quick links

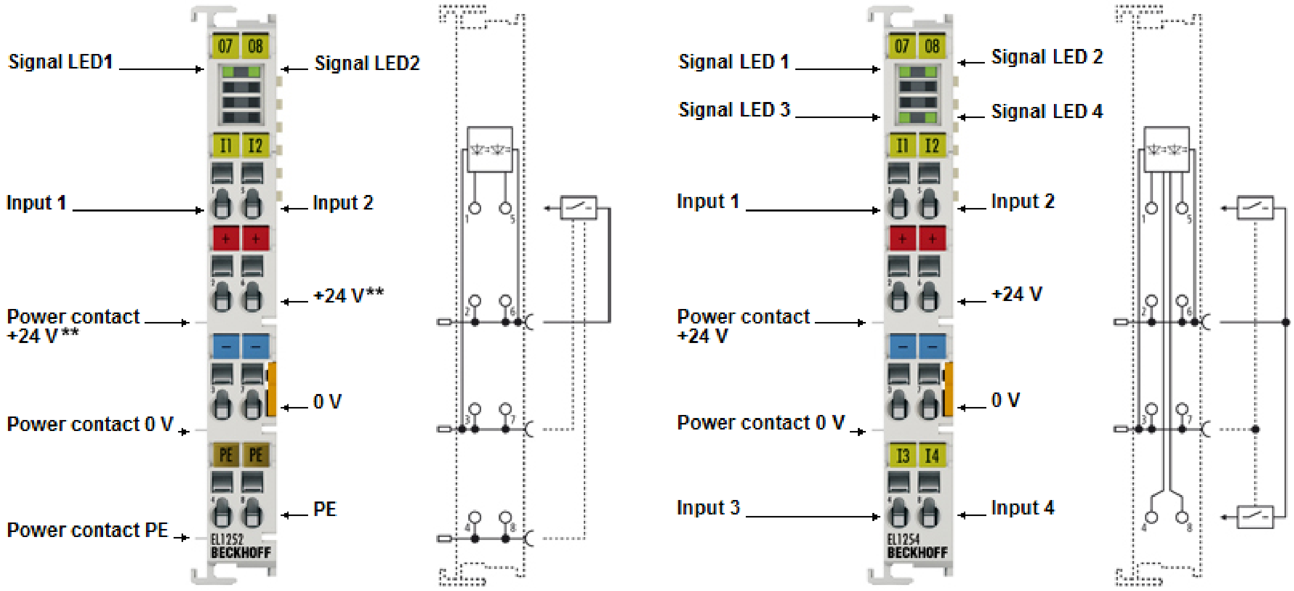
- EtherCAT function principles
- LEDs and pin assignment [▶ 40]
- Commissioning [▶ 42]

2.1.2 EL1202 - Technical data

Technical data	EL1202-0000	EL1202-0100
Digital inputs	2	
Nominal voltage of the inputs	24 V_{DC} (-15%/+20%)	
Signal voltage "0"	-3 V… +5 V (based on EN 61131-2, type 3)	
Signal voltage "1"	+11 V… +30 V (based on EN 61131-2, type 3)	
Input current	typ. 3 mA (based on EN 61131-2, type 3)	
Input delay T_{ON}/T_{OFF}	< 1 µs	
Distributed Clocks (DC)	no	yes
Power supply for the electronics	via the E-Bus	
Current consumption from the E-bus	typ. 110 mA	
Electrical isolation	500 V (E-bus/field voltage)	
Bit width in the process image	2 input bits	
Configuration	no address or configuration settings required	
Weight	approx. 55 g	
Permissible ambient temperature range during operation	0°C ... +55°C	
Permissible ambient temperature range during storage	-25°C ... +85°C	
Permissible relative humidity	95%, no condensation	
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)	
Mounting [▶ 26]	on 35 mm mounting rail conforms to EN 60715	
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27, see also installation instructions for enhanced mechanical load capacity [▶ 30]	
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4	
Protection class	IP20	
Installation position	variable	
Approval	CE ATEX [▶ 37] cULus [▶ 39]	

2.2 EL1252, EL1254

2.2.1 EL1252, EL1254 - Introduction



**) +5 V for EL1252-0050

Fig. 11: EL1252, EL1252-0050 and EL1254

2-/4-channel digital input terminal with time stamp

The EL1252 and EL1254 digital input terminals acquire the fast binary control signals from the process level and transmits them, in an electrically isolated form, to the controller. The signals are furnished with a time stamp that identifies the time of the last edge change with a resolution of 1 ns. This technology enables signals to be traced exactly over time and synchronized with the distributed clocks across the system. With this technology, machine-wide parallel hardware wiring of digital inputs or encoder signals for synchronization purposes is often no longer required. In conjunction with the EL2252 EtherCAT Terminal (digital output terminal with time stamp), the EL1252/EL1254 enables responses with equidistant time intervals, largely independent of the bus cycle time.

In the EL1252-0050 a variant is provided with a 5 V input voltage (TTL level).

Quick links

- EtherCAT function principles
- LEDs and pin assignment [▶ 41]
- Commissioning [▶ 42]

2.2.2 EL1252, EL1254 - Technical data

Technical data	EL1252-0000	EL1252-0050	EL1254-0000
digital inputs	2		4
Nominal voltage of the inputs	24 V_{DC} (-15%/+20%)	5 V_{DC} (-15%/+20%)	24 V_{DC} (-15%/+20%)
Signal voltage "0"	-3 V … +5 V (based on EN 61131-2, type 3)	< 0.8 V	-3 V … +5 V (based on EN 61131-2, type 3)
Signal voltage "1"	+11 V … +30 V (based on EN 61131-2, type 3)	>2.4 V	+11 V … +30 V (based on EN 61131-2, type 3)
Input current	typ. 3 mA (based on EN 61131-2, type 3)	typ. 50 µA	typ. 3 mA (based on EN 61131-2, type 3)
Input delay T_{ON}/T_{OFF}	< 1 µs		
Time stamp resolution	1 ns		
Time stamp accuracy in the terminal	10 ns (+ input delay)		
Distributed Clocks (DC) accuracy	<< 1 µs		
Power supply for the electronics	via the E-Bus		
Current consumption from the E-bus	typ. 110 mA	typ. 90 mA	typ. 110 mA
Electrical isolation	500 V (E-bus/field voltage)		
Bit width in the process image	2 input bits + 2 time stamps		4 input bits + 8 time stamps
Configuration	no address or configuration settings required		
Weight	approx. 55 g		
Permissible ambient temperature range during operation	-25°C ... +60°C (extended temperature range)	0°C ... +55°C	
Permissible ambient temperature range during storage	-40°C ... +85°C	-25°C ... +85°C	
Permissible relative humidity	95%, no condensation		
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)		
Mounting [▸ 26]	on 35 mm mounting rail conforms to EN 60715		
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27, see also installation instructions for enhanced mechanical load capacity [▸ 30]		
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4		
Protection class	IP20		
Installation position	variable		
Approval	CE ATEX [▸ 38] cULus [▸ 39]		CE

2.3 Start

For commissioning:

- install the EL12xx as described in the Mounting and wiring [▸ 26] section
- Configure the EL12xx in TwinCAT as described in section Commissioning [▸ 42].

3 Basics communication

3.1 EtherCAT basics

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

3.2 EtherCAT cabling – wire-bound

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

Cables and connectors

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

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

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

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

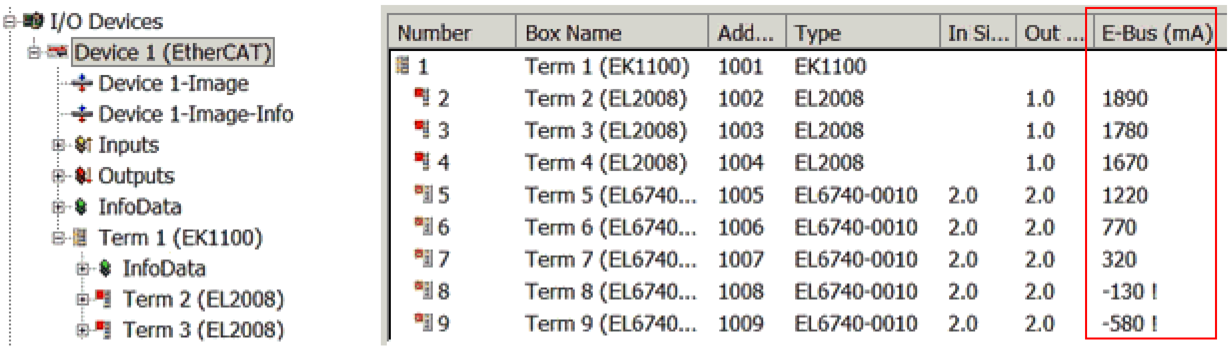
Recommended cables

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

E-Bus supply

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

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



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

Fig. 12: System manager current calculation

NOTE

Malfunction possible!

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

3.3 General notes for setting the watchdog

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

The EtherCAT slave controller (ESC) in the EL2xxx terminals features two watchdogs:

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

SM watchdog (SyncManager Watchdog)

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

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

PDI watchdog (Process Data Watchdog)

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

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

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

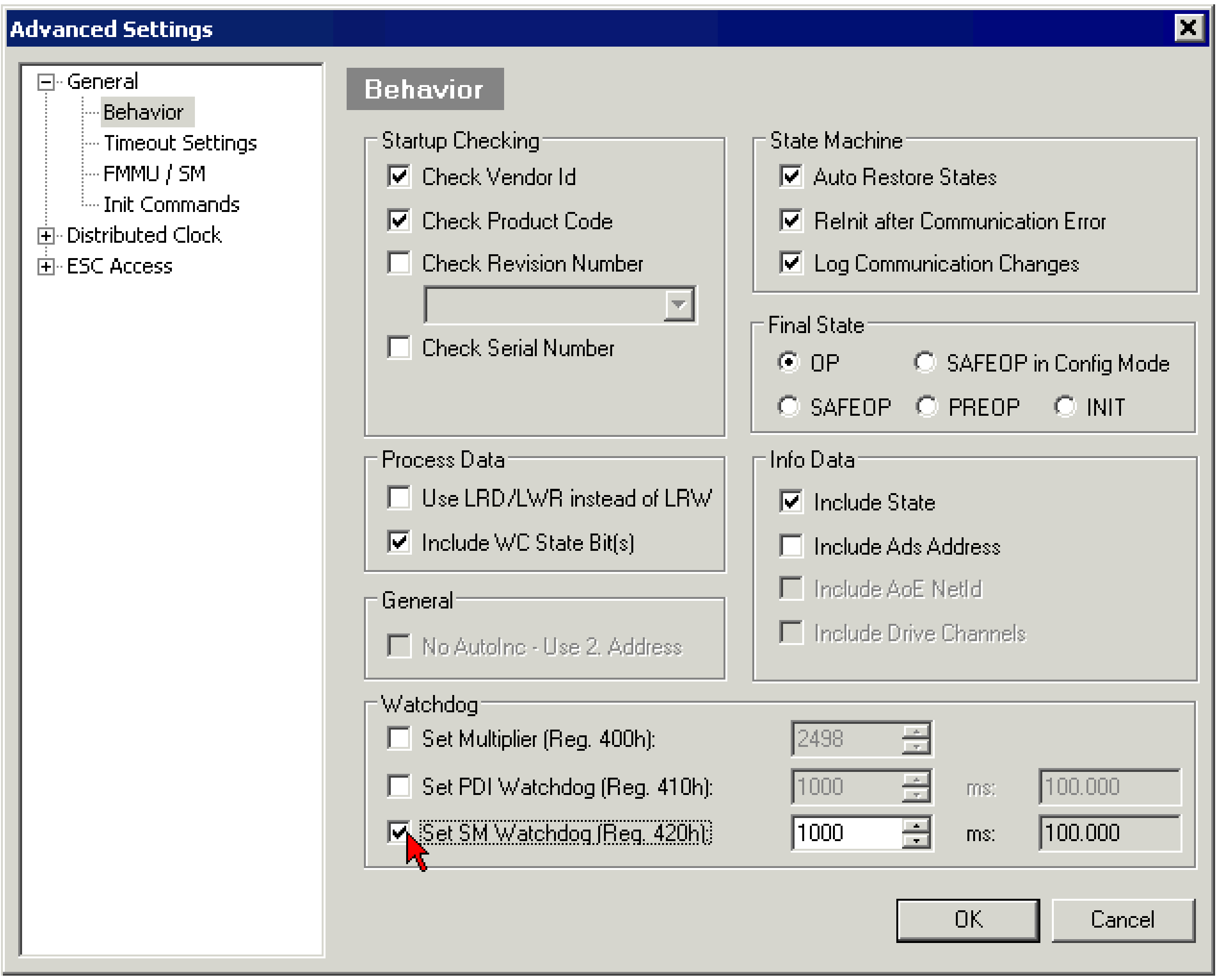


Fig. 13: EtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

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

Multiplier

Multiplier

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

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

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

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

Example “Set SM watchdog”

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

Calculation

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

⚠ CAUTION

Undefined state possible!

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

⚠ CAUTION

Damage of devices and undefined state possible!

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

3.4 EtherCAT State Machine

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

A distinction is made between the following states:

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

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

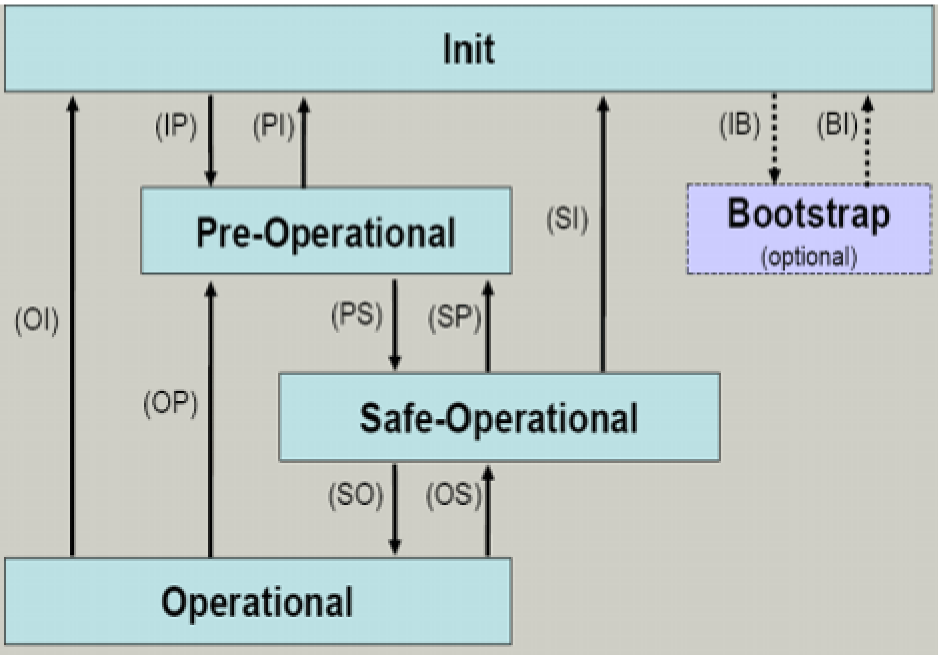


Fig. 14: States of the EtherCAT State Machine

Init

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

Pre-Operational (Pre-Op)

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

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

Safe-Operational (Safe-Op)

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

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

Outputs in SAFEOP state

The default set watchdog [▶ 20] monitoring sets the outputs of the module in a safe state - depending on the settings in SAFEOP and OP - e.g. in OFF state. If this is prevented by deactivation of the watchdog monitoring in the module, the outputs can be switched or set also in the SAFEOP state.

Operational (Op)

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

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

Boot

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

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

3.5 CoE - Interface: notes

This device has no CoE.

Detailed information on the CoE interface can be found in the EtherCAT system documentation on the Beckhoff website.

3.6 Distributed Clock

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

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

For detailed information please refer to the EtherCAT system description.

4 Mounting and wiring

4.1 Instructions for ESD protection

NOTE

Destruction of the devices by electrostatic discharge possible!

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

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

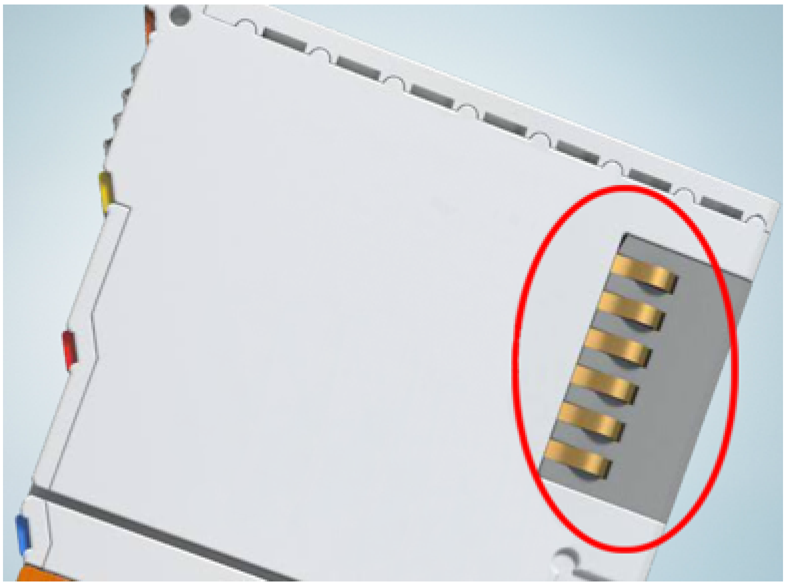


Fig. 15: Spring contacts of the Beckhoff I/O components

4.2 Installation on mounting rails

⚠ WARNING

Risk of electric shock and damage of device!

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

Assembly

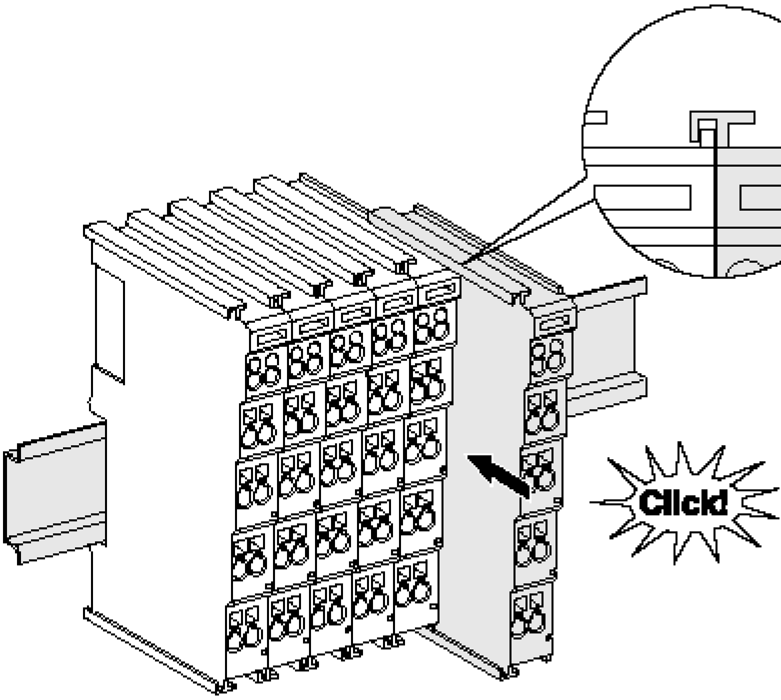


Fig. 16: Attaching on mounting rail

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

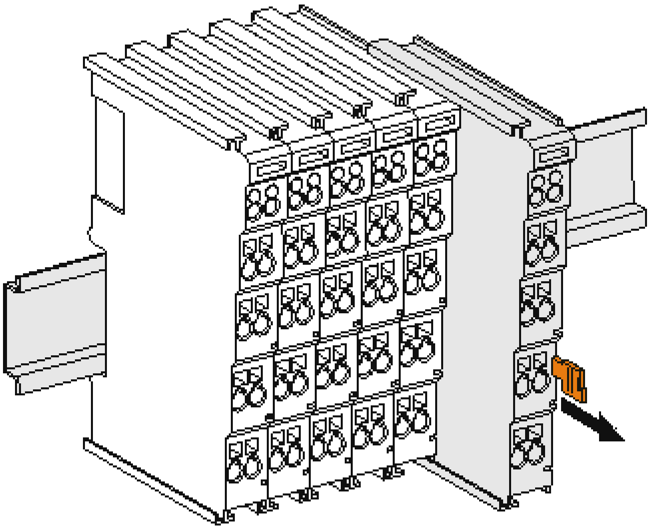


Fig. 17: Disassembling of terminal

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

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

Connections within a bus terminal block

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

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

Power Contacts

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

PE power contact

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

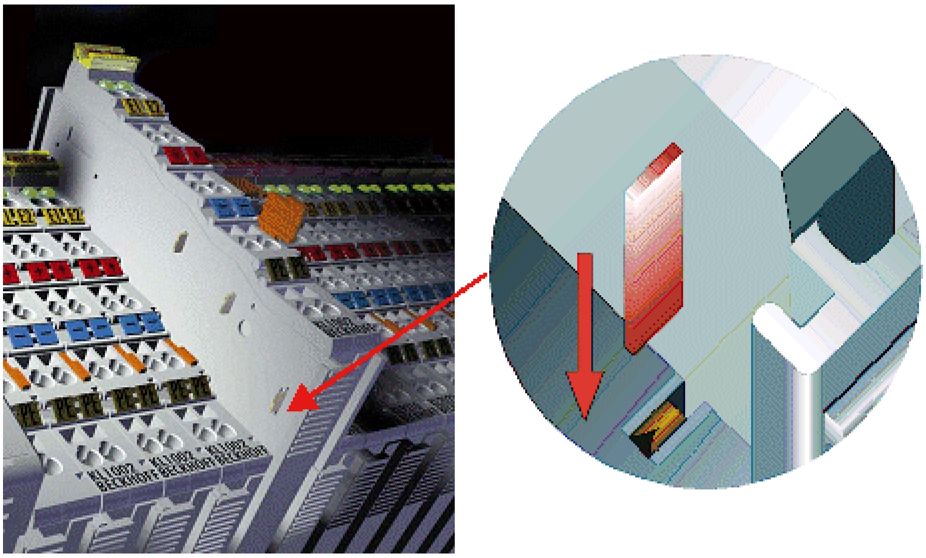


Fig. 18: Power contact on left side

NOTE

Possible damage of the device

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

⚠ WARNING

Risk of electric shock!

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

4.3 Installation instructions for enhanced mechanical load capacity

⚠ **WARNING**

Risk of injury through electric shock and damage to the device!

Bring the Bus Terminal system into a safe, de-energized state before starting mounting, disassembly or wiring of the Bus Terminals!

Additional checks

The terminals have undergone the following additional tests:

Verification	Explanation
Vibration	10 frequency runs in 3 axes
	6 Hz < f < 60 Hz displacement 0.35 mm, constant amplitude
	60.1 Hz < f < 500 Hz acceleration 5 *g*, constant amplitude
Shocks	1000 shocks in each direction, in 3 axes
	25 *g*, 6 ms

Additional installation instructions

For terminals with enhanced mechanical load capacity, the following additional installation instructions apply:

- The enhanced mechanical load capacity is valid for all permissible installation positions
- Use a mounting rail according to EN 60715 TH35-15
- Fix the terminal segment on both sides of the mounting rail with a mechanical fixture, e.g. an earth terminal or reinforced end clamp
- The maximum total extension of the terminal segment (without coupler) is:
  64 terminals (12 mm mounting with) or 32 terminals (24 mm mounting with)
- Avoid deformation, twisting, crushing and bending of the mounting rail during edging and installation of the rail
- The mounting points of the mounting rail must be set at 5 cm intervals
- Use countersunk head screws to fasten the mounting rail
- The free length between the strain relief and the wire connection should be kept as short as possible. A distance of approx. 10 cm should be maintained to the cable duct.

4.4 Connection

4.4.1 Connection system

⚠ **WARNING**

Risk of electric shock and damage of device!

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

Overview

The Bus Terminal system offers different connection options for optimum adaptation to the respective application:

- The terminals of ELxxxx and KLxxxx series with standard wiring include electronics and connection level in a single enclosure.

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

Standard wiring (ELxxxx / KLxxxx)

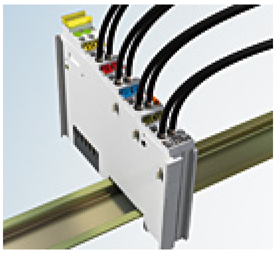


Fig. 19: Standard wiring

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

Pluggable wiring (ESxxxx / KSxxxx)

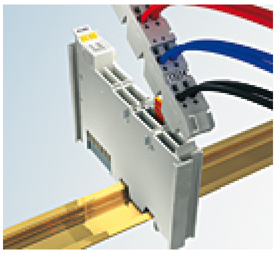


Fig. 20: Pluggable wiring

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

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

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

Conductor cross sections between 0.08 mm^2 and 2.5 mm^2 can continue to be used with the proven spring force technology.

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

High Density Terminals (HD Terminals)

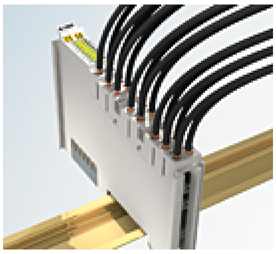


Fig. 21: High Density Terminals

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

Wiring HD Terminals

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

Ultrasonically "bonded" (ultrasonically welded) conductors

Ultrasonically "bonded" conductors

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

4.4.2 Wiring

⚠ WARNING

Risk of electric shock and damage of device!

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

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

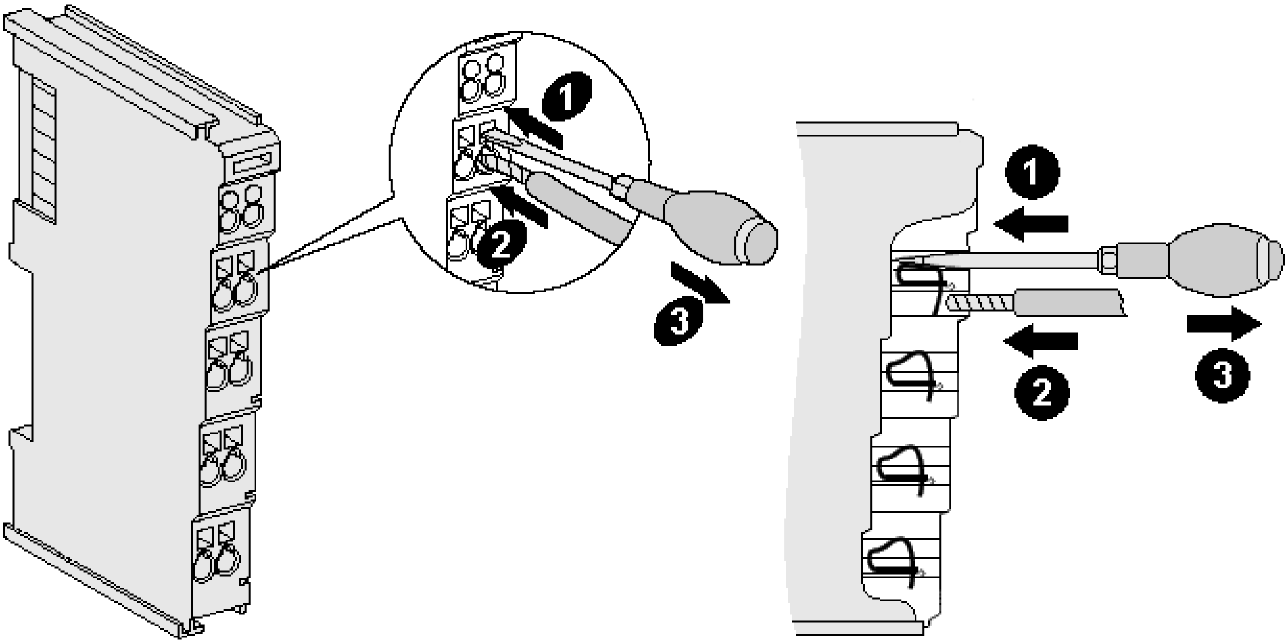


Fig. 22: Connecting a cable on a terminal point

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

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

See the following table for the suitable wire size width.

Terminal housing	ELxxxx, KLxxxx	ESxxxx, KSxxxx
Wire size width (single core wires)	0.08 ... 2.5 mm^2	0.08 ... 2.5 mm^2
Wire size width (fine-wire conductors)	0.08 ... 2.5 mm^2	0,08 ... 2.5 mm^2
Wire size width (conductors with a wire end sleeve)	0.14 ... 1.5 mm^2	0.14 ... 1.5 mm^2
Wire stripping length	8 ... 9 mm	9 ... 10 mm

High Density Terminals (HD Terminals [▸ 32]) 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

4.5 Installation positions

NOTE

Constraints regarding installation position and operating temperature range

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

Optimum installation position (standard)

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

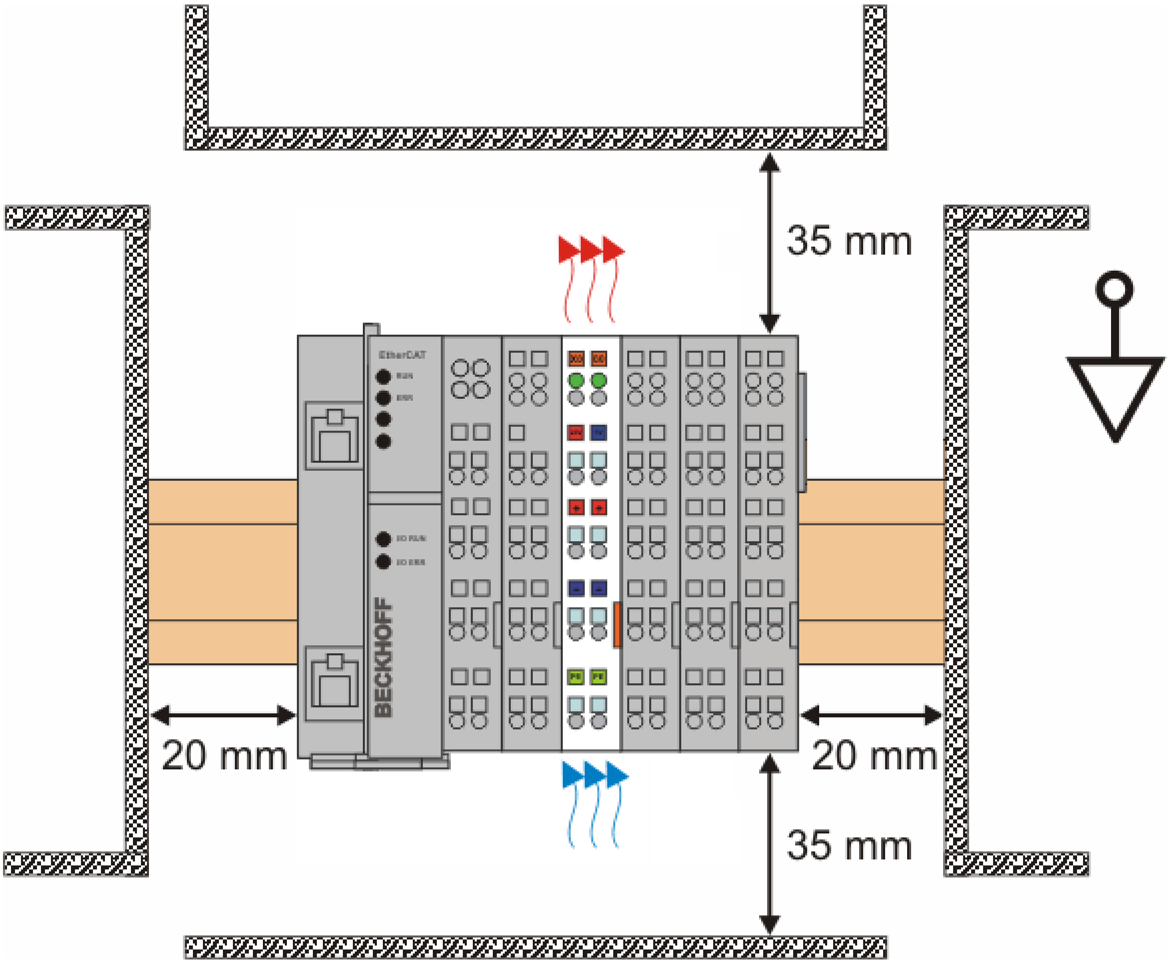


Fig. 23: Recommended distances for standard installation position

Compliance with the distances shown in Fig. *Recommended distances for standard installation position* is recommended.

Other installation positions

All other installation positions are characterized by different spatial arrangement of the mounting rail - see Fig *"Other installation positions".*

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

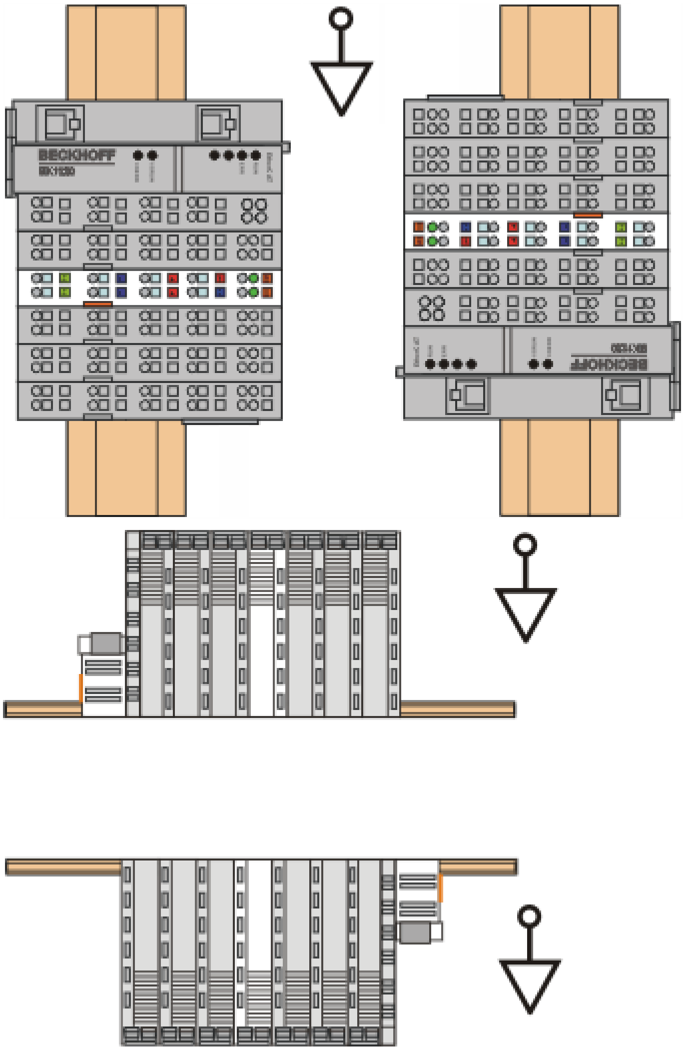


Fig. 24: Other installation positions

4.6 Positioning of passive Terminals

Hint for positioning of passive terminals in the bus terminal block

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

Examples for positioning of passive terminals (highlighted)

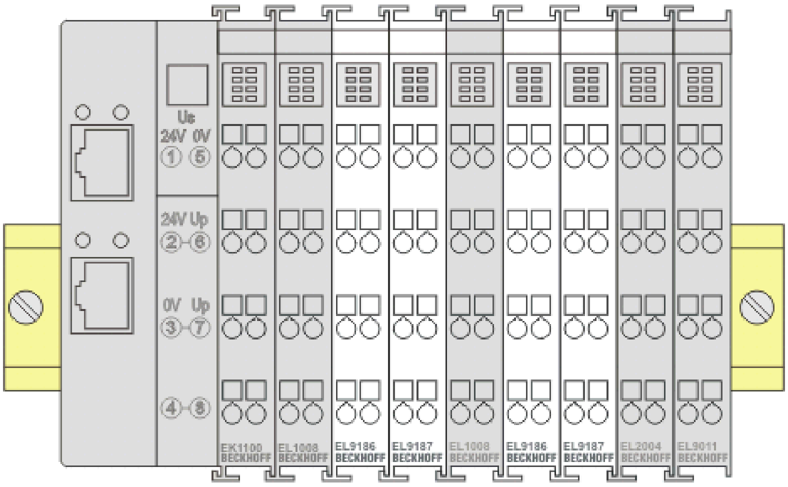


Fig. 25: Correct positioning

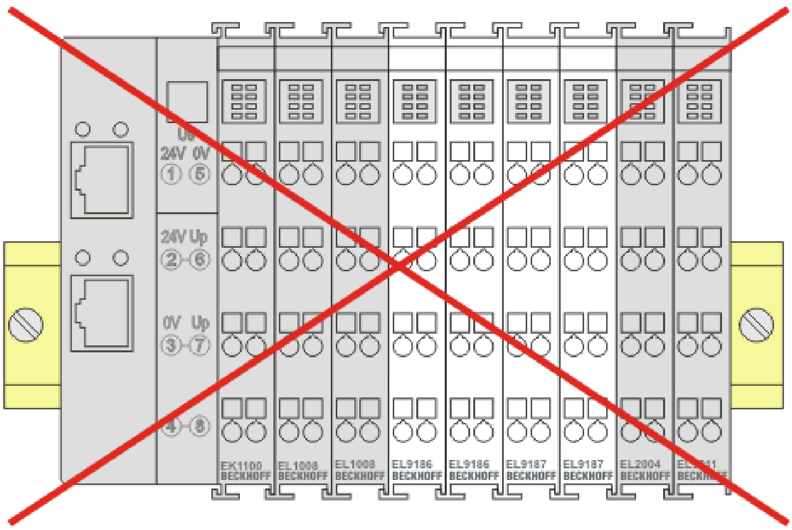


Fig. 26: Incorrect positioning

4.7 ATEX - Special conditions (standard temperature range)

⚠ WARNING

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

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

Standards

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

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

Marking

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

II 3G KEMA 10ATEX0075 X Ex nA IIC T4 Gc Ta: 0 … +55°C
II 3D KEMA 10ATEX0075 X Ex tc IIC T135°C Dc Ta: 0 ... +55°C
(only for fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9)

or

II 3G KEMA 10ATEX0075 X Ex nC IIC T4 Gc Ta: 0 … +55°C
II 3D KEMA 10ATEX0075 X Ex tc IIC T135°C Dc Ta: 0 ... +55°C
(only for fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9)

4.8 ATEX - Special conditions (extended temperature range)

⚠ WARNING

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

- The certified components are to be installed in a suitable housing that guarantees a protection class of at least IP54 in accordance with EN 60079-15! The environmental conditions during use are thereby to be taken into account!
- For dust (only the fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9): The equipment shall be installed in a suitable enclosure providing a degree of protection of IP54 according to EN 60079-0 for group IIIA or IIIB and IP6X for group IIIC, taking into account the environmental conditions under which the equipment is used.
- If the temperatures during rated operation are higher than 70°C at the feed-in points of cables, lines or pipes, or higher than 80°C at the wire branching points, then cables must be selected whose temperature data correspond to the actual measured temperature values!
- Observe the permissible ambient temperature range of -25 to 60°C for the use of Beckhoff fieldbus components with extended temperature range (ET) in potentially explosive areas!
- Measures must be taken to protect against the rated operating voltage being exceeded by more than 40% due to short-term interference voltages!
- The individual terminals may only be unplugged or removed from the Bus Terminal system if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The connections of the certified components may only be connected or disconnected if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The fuses of the KL92xx/EL92xx power feed terminals may only be exchanged if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- Address selectors and ID switches may only be adjusted if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!

Standards

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

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

Marking

The Beckhoff fieldbus components with extended temperature range (ET) certified according to the ATEX directive for potentially explosive areas bear the following marking:

II 3G KEMA 10ATEX0075 X Ex nA IIC T4 Gc Ta: -25 … +60°C
II 3D KEMA 10ATEX0075 X Ex tc IIC T135°C Dc Ta: -25 ... +60°C
(only for fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9)

or

II 3G KEMA 10ATEX0075 X Ex nC IIC T4 Gc Ta: -25 … +60°C
II 3D KEMA 10ATEX0075 X Ex tc IIC T135°C Dc Ta: -25 ... +60°C
(only for fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9)

4.9 Continuative documentation about explosion protection

Explosion protection for terminal systems

Pay also attention to the continuative documentation

Notes on the use of the Beckhoff terminal systems in hazardous areas according to ATEX and IECEx

that is available for download on the Beckhoff homepage https:\\www.beckhoff.com!

4.10 UL notice

Application

Beckhoff EtherCAT modules are intended for use with Beckhoff's UL Listed EtherCAT System only.

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).

For devices with Ethernet connectors

Not for connection to telecommunication circuits.

Basic principles

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

4.11 EL1202 - LEDs and pin assignment

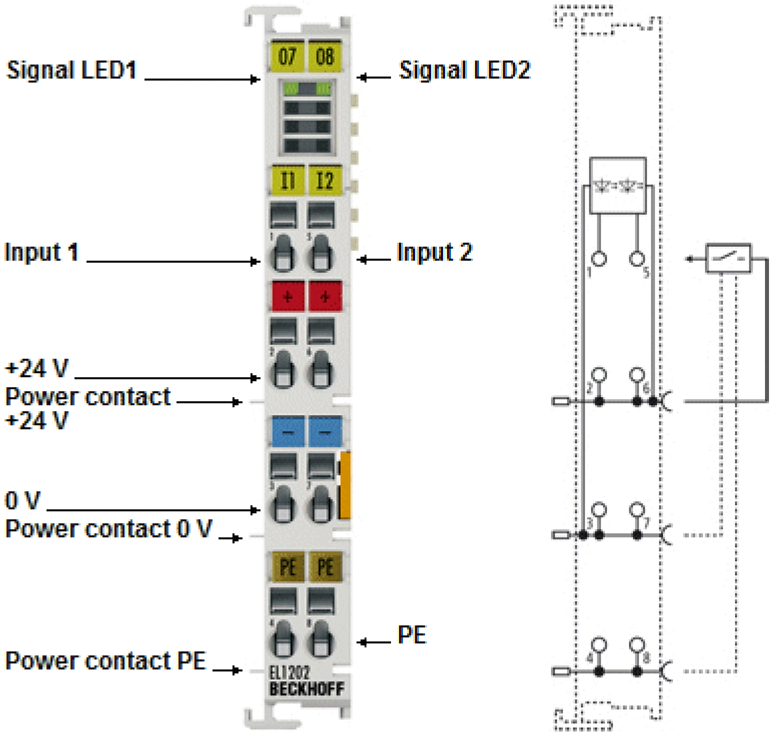


Fig. 27: EL1202

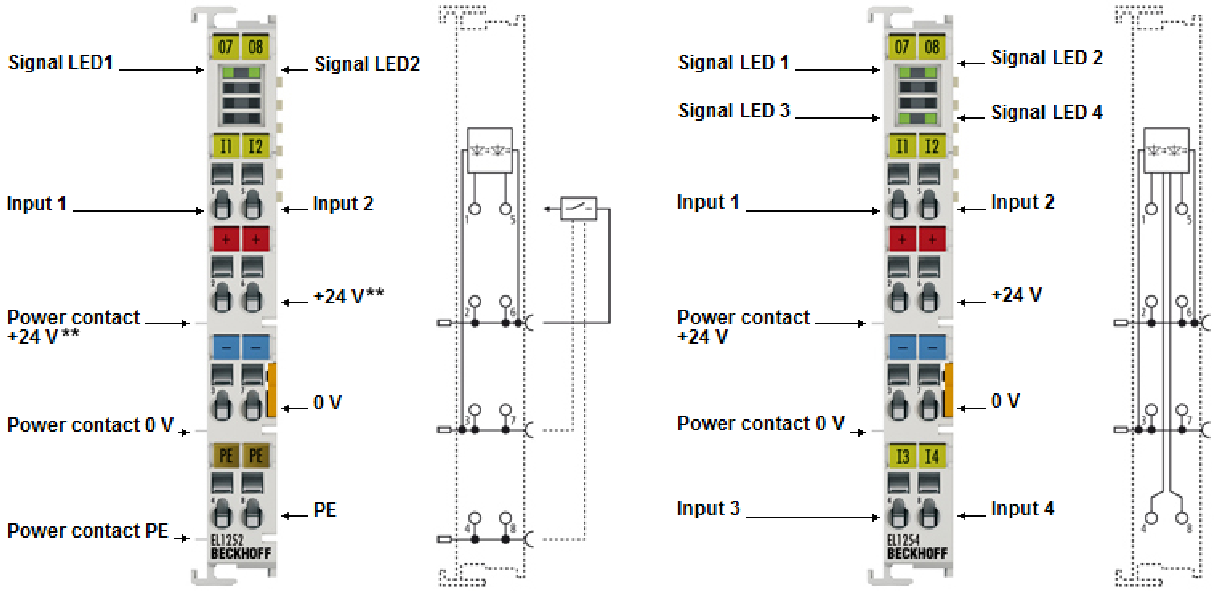
LEDs

LED	Color	Meaning	
INPUT 1 INPUT 2	green	off	There is no input signal at the respective input
		on	+24 V input signal at the respective input

EL1202 pin assignment

Terminal point		Description
Name	No.	
Input 1	1	Input 1
+ 24 V	2	+24 V (internally connected to terminal point 6 and positive power contact)
0 V	3	0 V (internally connected to terminal point 7 and negative power contact)
PE	4	PE contact
Input 2	5	Input 2
+ 24 V	6	+24 V (internally connected to terminal point 2 and positive power contact)
0 V	7	0 V (internally connected to terminal point 3 and negative power contact)
PE	8	PE contact

4.12 EL1252, EL1254 - LEDs and pin assignment



**) +5 V for EL1252-0050

Fig. 28: EL1252, EL1254

LEDs

LED	Color	Meaning	
INPUT 1 INPUT 2 INPUT 3* INPUT 4*	green	off	There is no input signal at the respective input
		on	Input signal at the respective input

*) EL1254 only

NOTE

EL1252-0050: damage to the terminal possible!

In the project planning of the Bus Terminal system, observe the different potentials on the power contacts (24 V_{DC} and 5 V_{DC}) and if necessary use separation terminals (EL9080) to isolate the potentials!

EL1252, EL1254 pin assignment

Terminal point		Description
Name	No.	
Input 1	1	Input 1
+ 24 V (EL1252-0000, EL1254) + 5 V (EL1252-0050)	2	+24 V (internally connected to terminal point 6 and positive power contact) for EL1252-0000 and EL1254 +5 V (internally connected to terminal point 6 and positive power contact) for EL1252-0050
0 V	3	0 V (internally connected to terminal point 7 and negative power contact)
PE (EL1252) Input 3 (EL1254)	4	PE contact for EL1252 Input 3 for EL1254
Input 2	5	Input 2
+ 24 V (EL1252-0000, EL1254) + 5 V (EL1252-0050)	6	+24 V (internally connected to terminal point 6 and positive power contact) for EL1252-0000 and EL1254 +5 V (internally connected to terminal point 6 and positive power contact) for EL1252-0050
0 V	7	0 V (internally connected to terminal point 3 and negative power contact)
PE (EL1252) Input 4 (EL1254)	8	PE contact for EL1252 Input 4 for EL1254

5 Commissioning

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:

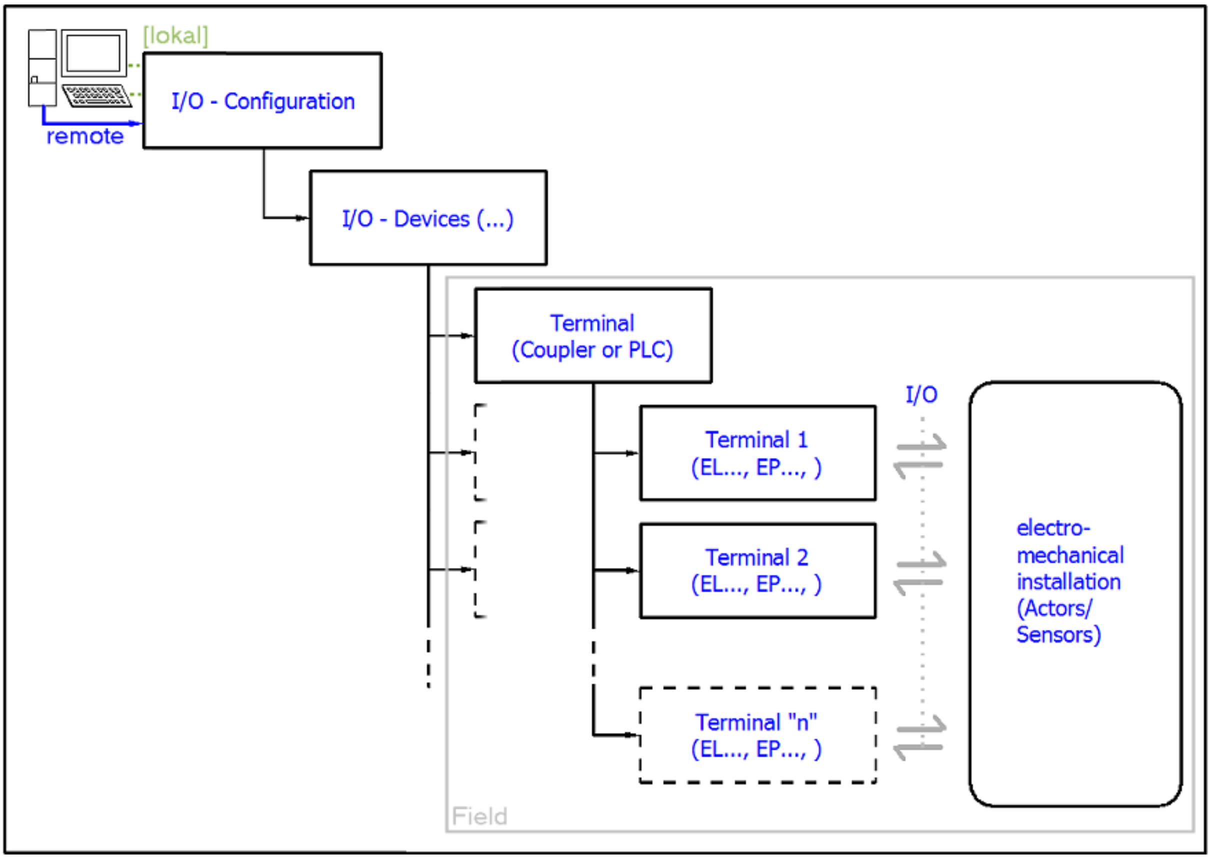


Fig. 29: Relationship between user side (commissioning) and installation

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)

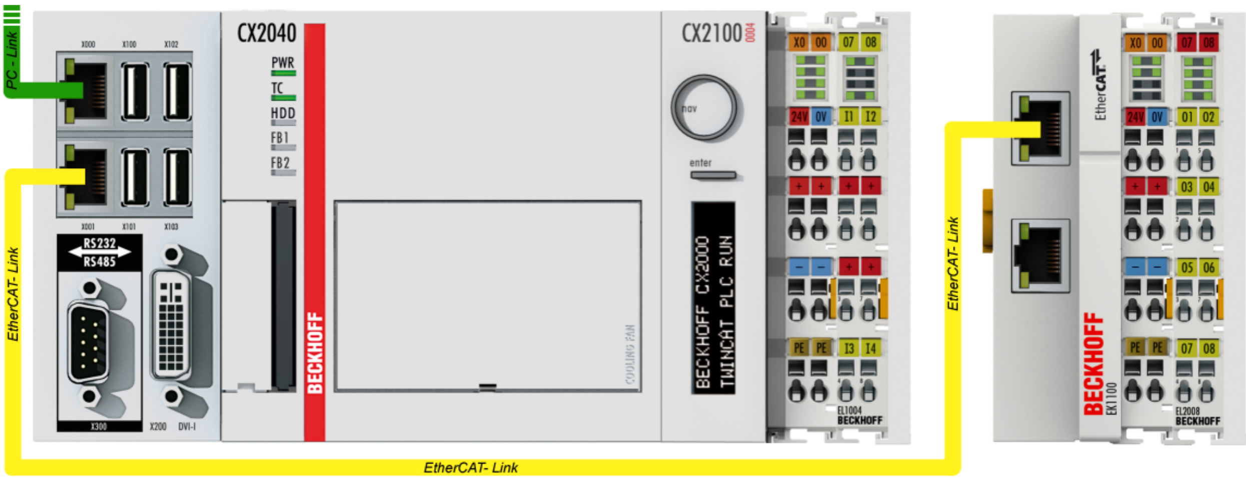


Fig. 30: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

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:

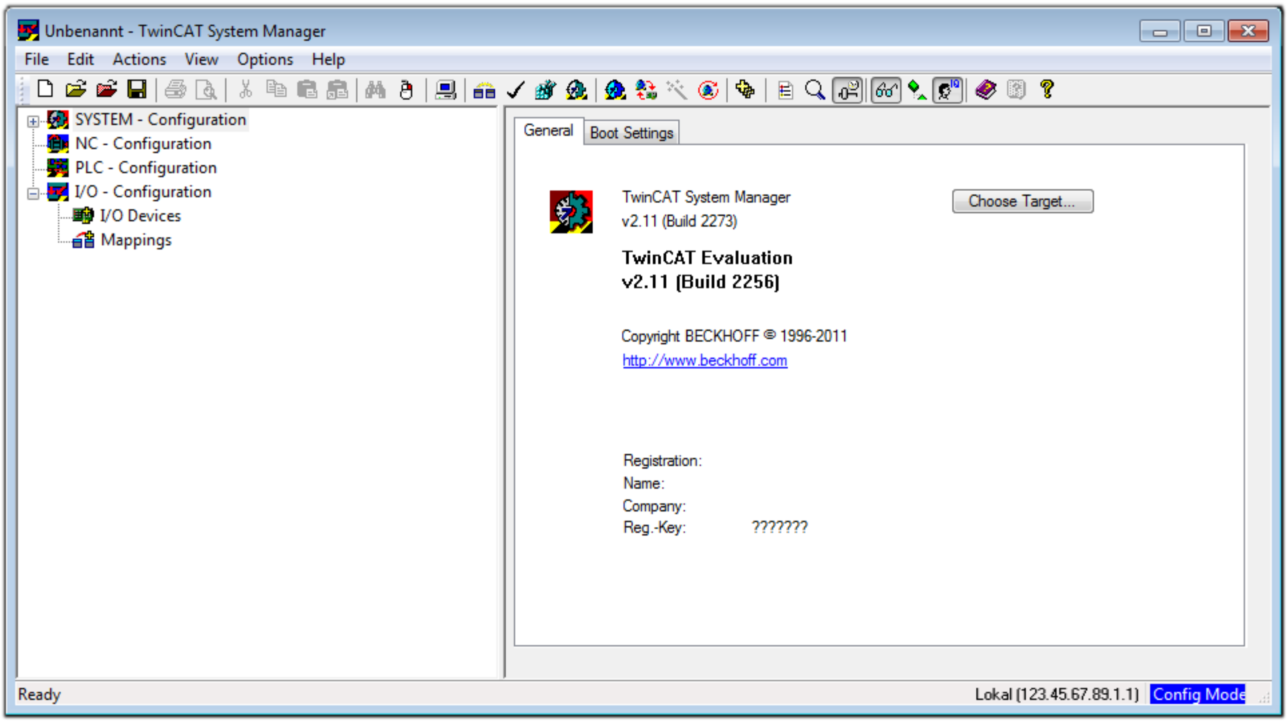


Fig. 31: Initial TwinCAT 2 user interface

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 [▶ 47]".

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:

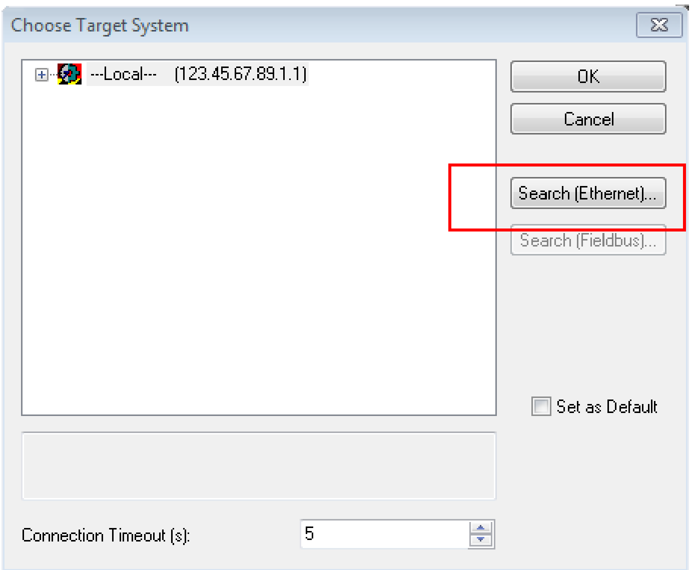


Fig. 32: Selection of the target system

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.

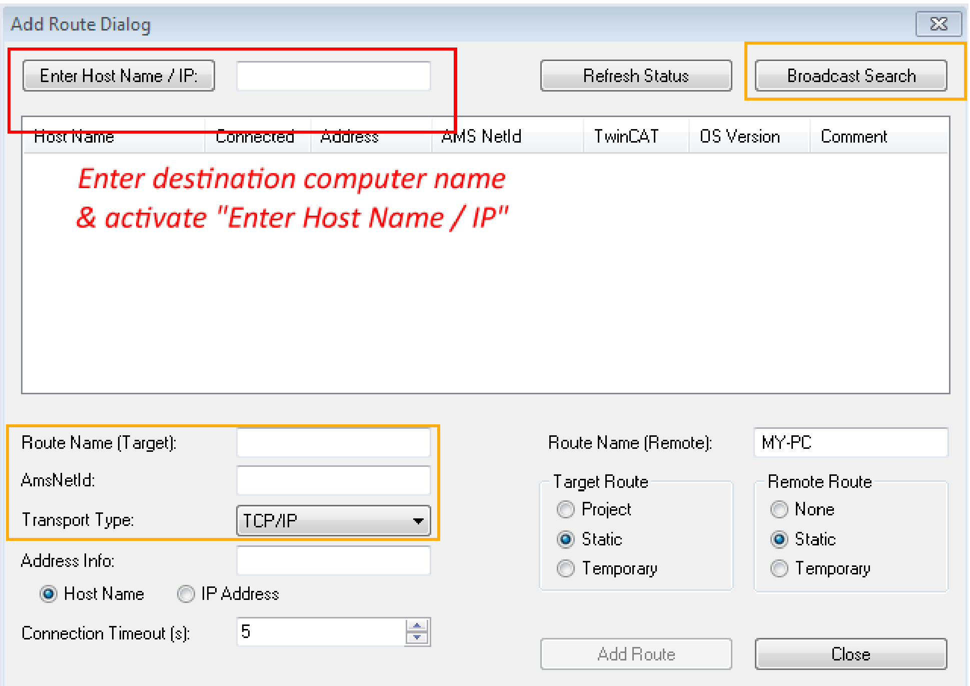
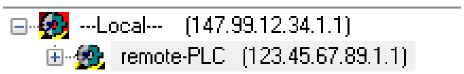


Fig. 33: Specify the PLC for access by the TwinCAT System Manager: selection of the target system

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).

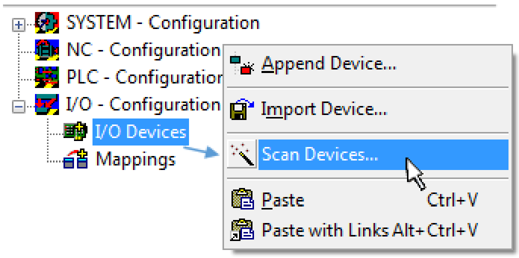


Fig. 34: Select “Scan Devices...”

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

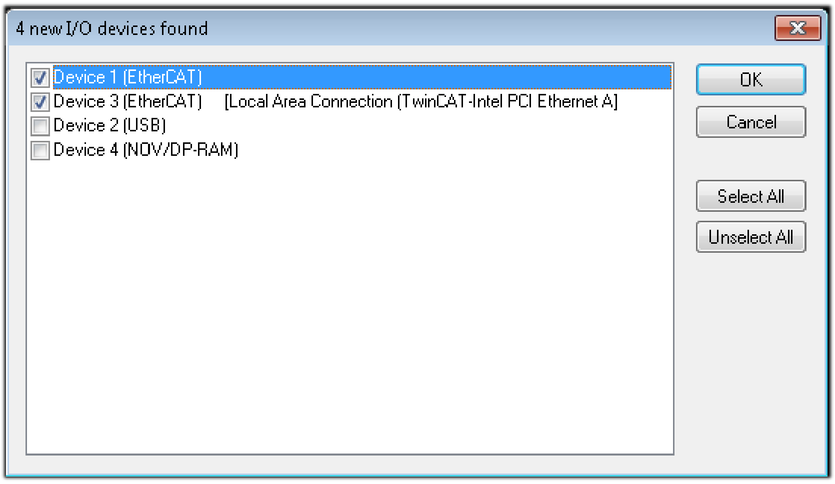


Fig. 35: Automatic detection of I/O devices: selection the devices to be integrated

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 [▶ 43] described at the beginning of this section, the result is as follows:

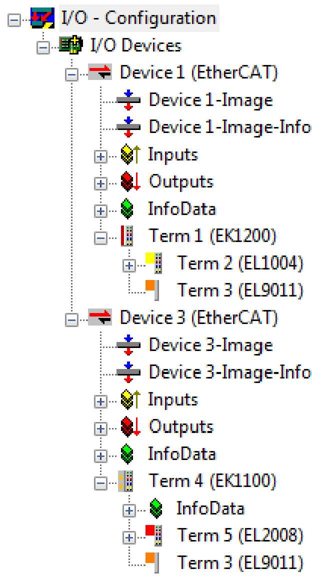


Fig. 36: Mapping of the configuration in the TwinCAT 2 System Manager

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:

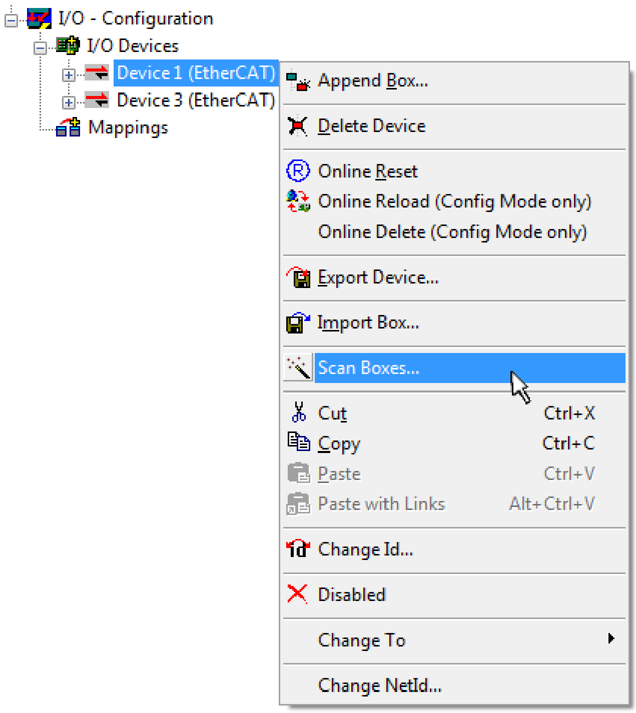


Fig. 37: Reading of individual terminals connected to a device

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:

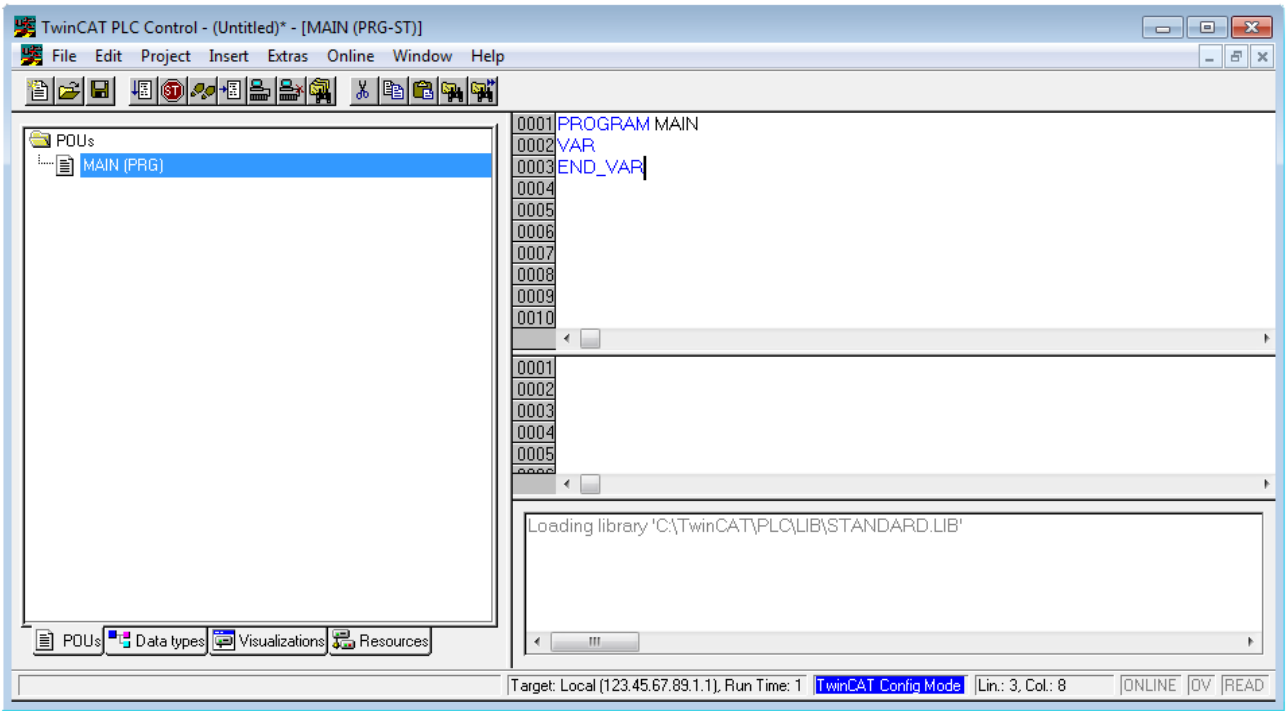


Fig. 38: TwinCAT PLC Control after startup

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

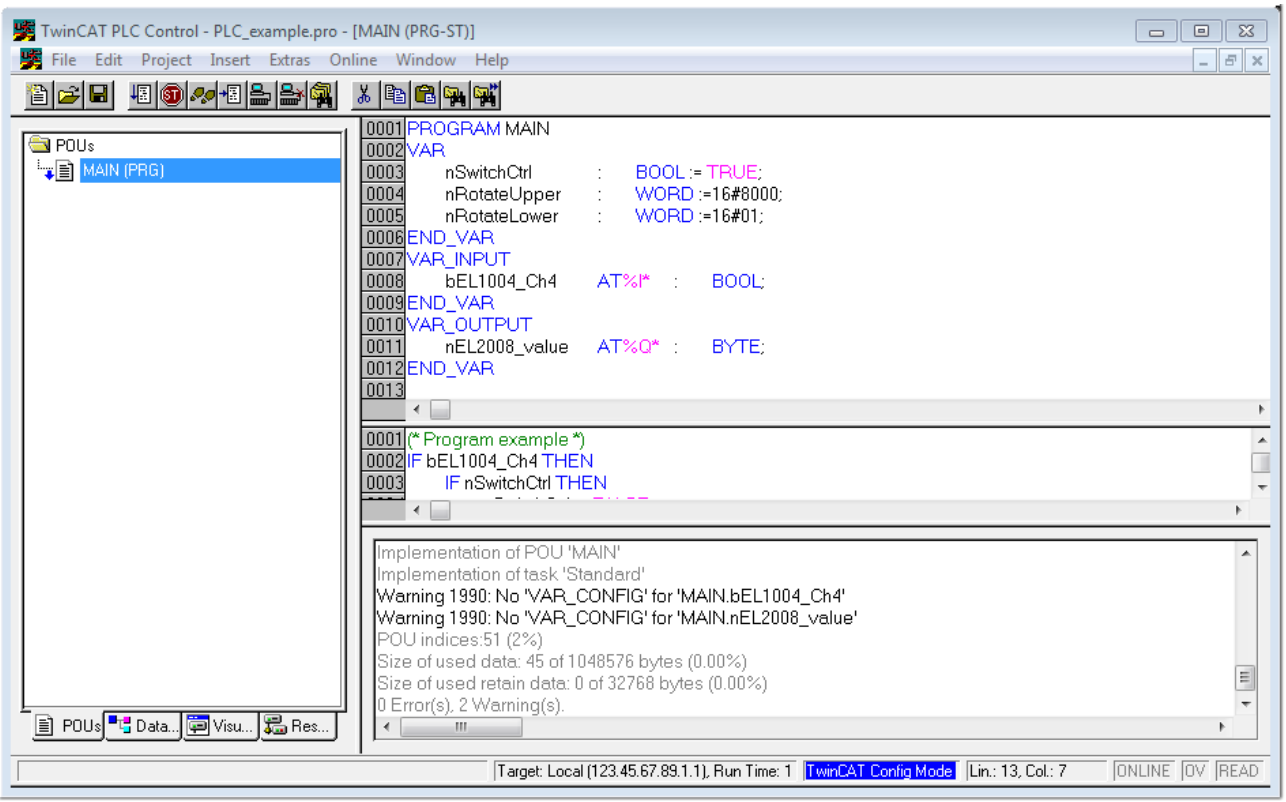


Fig. 39: Sample program with variables after a compile process (without variable integration)

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…”:

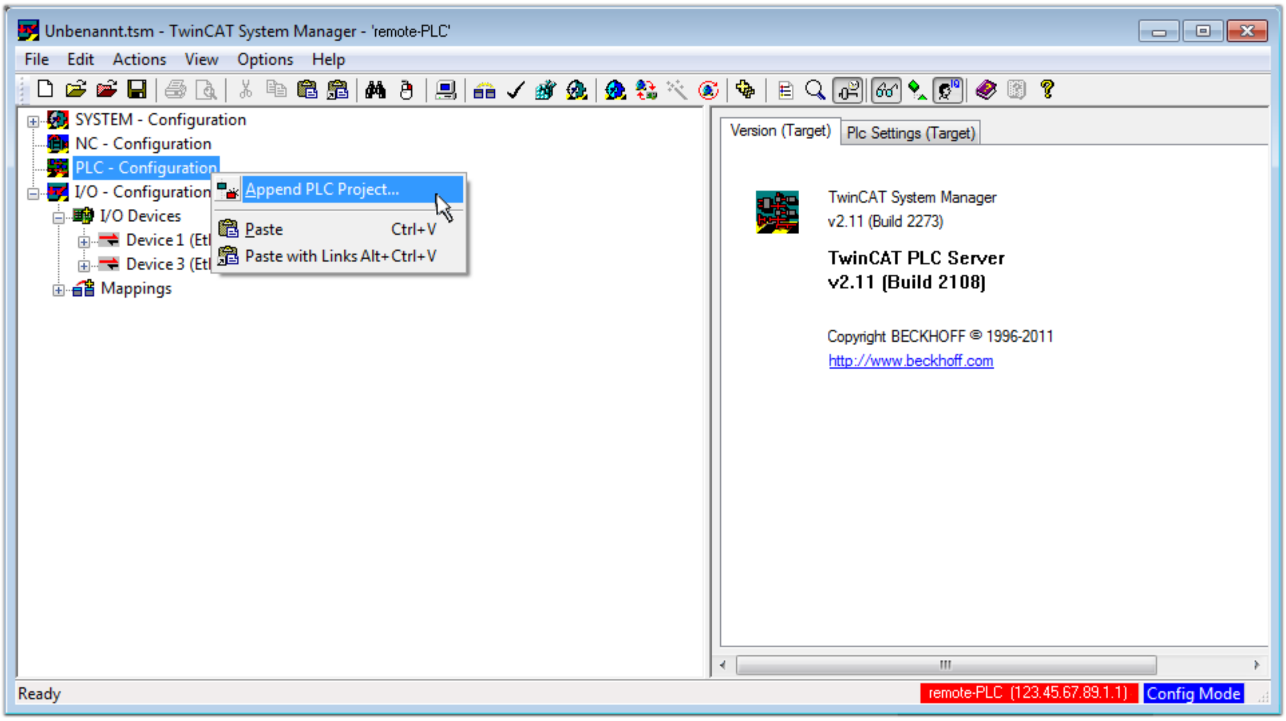


Fig. 40: Appending the TwinCAT PLC Control project

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:

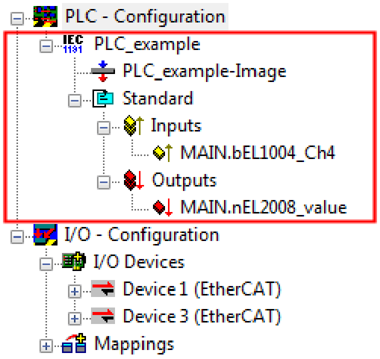


Fig. 41: PLC project integrated in the PLC configuration of the System Manager

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”:

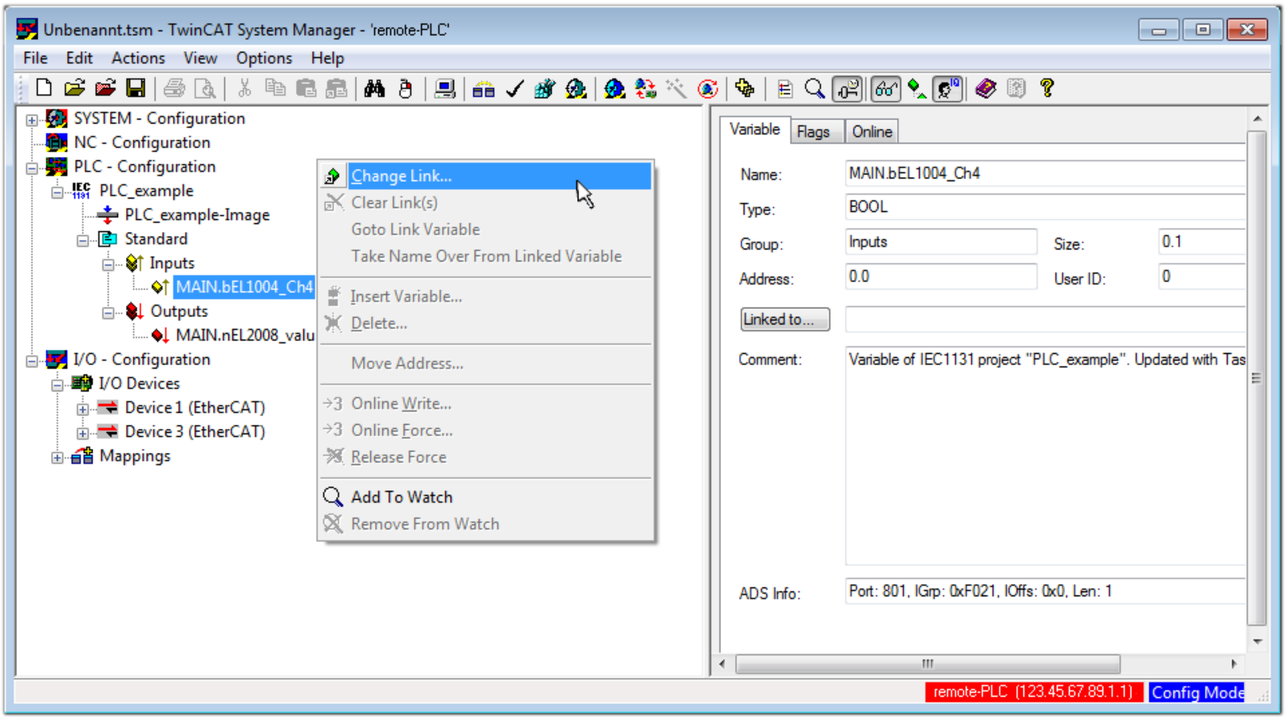


Fig. 42: Creating the links between PLC variables and process objects

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

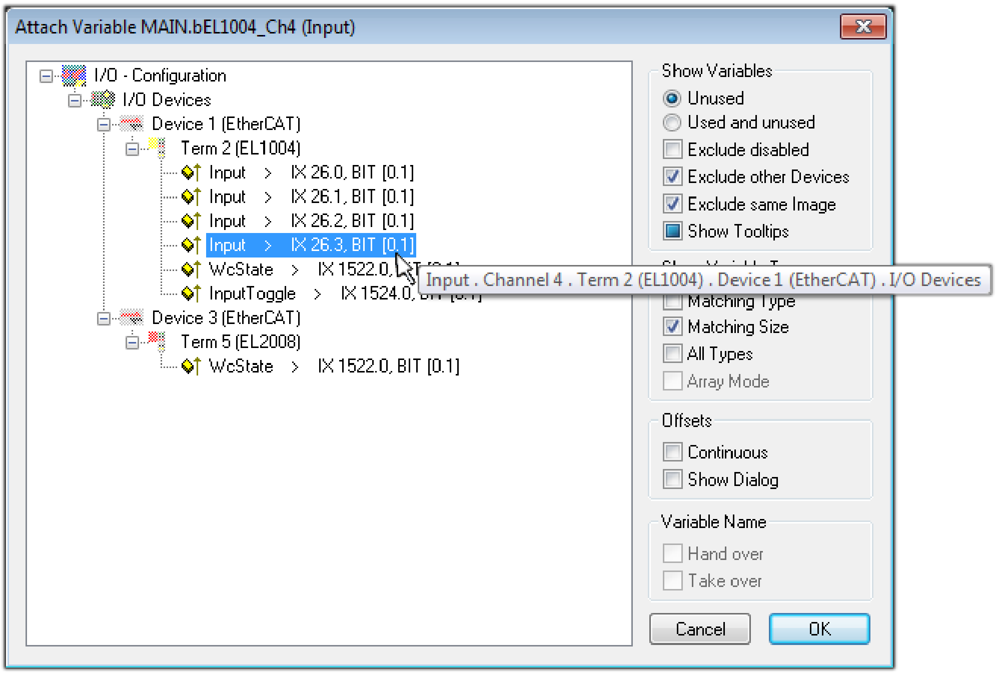


Fig. 43: Selecting PDO of type BOOL

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:

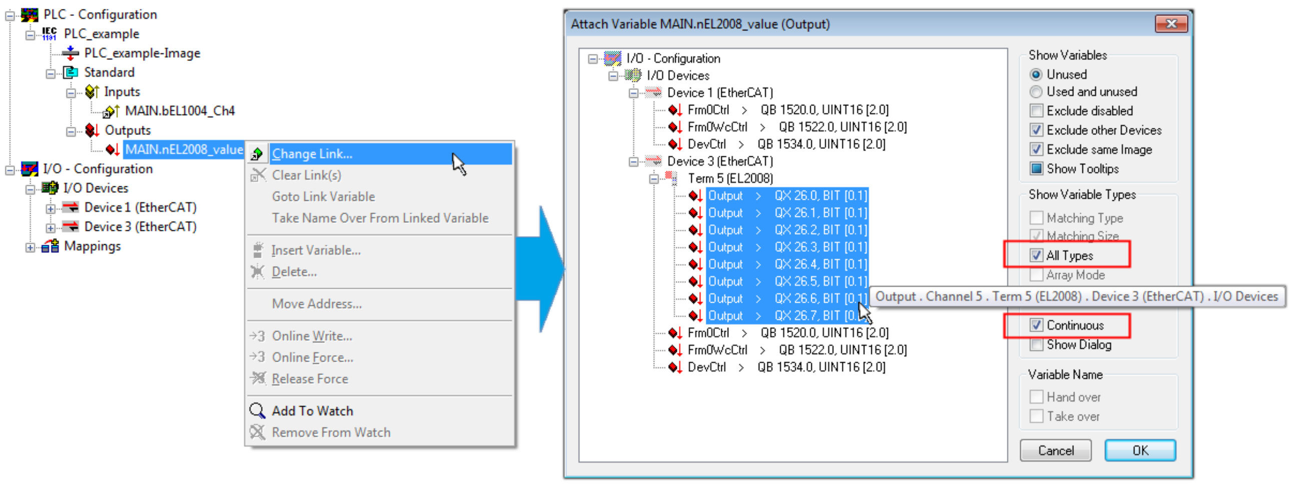


Fig. 44: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

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:

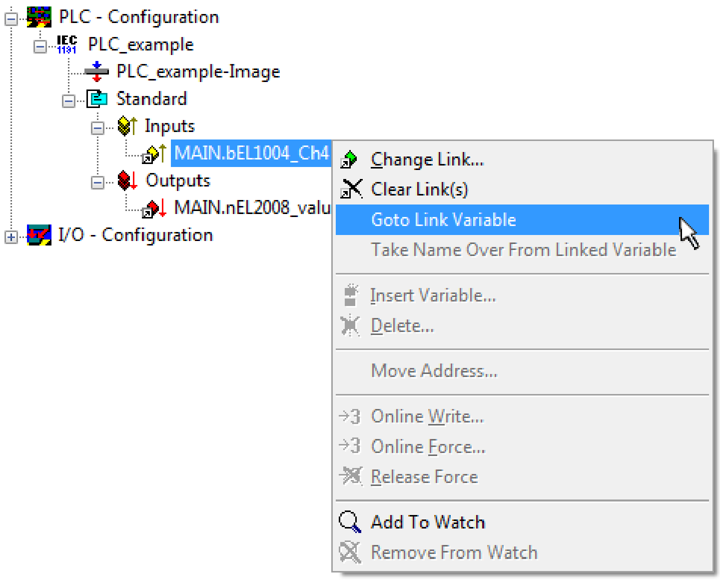
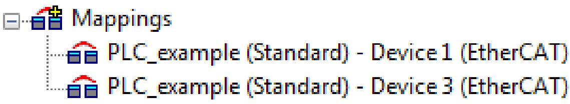


Fig. 45: Application of a “Goto Link” variable, using “MAIN.bEL1004_Ch4” as a sample

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…”:

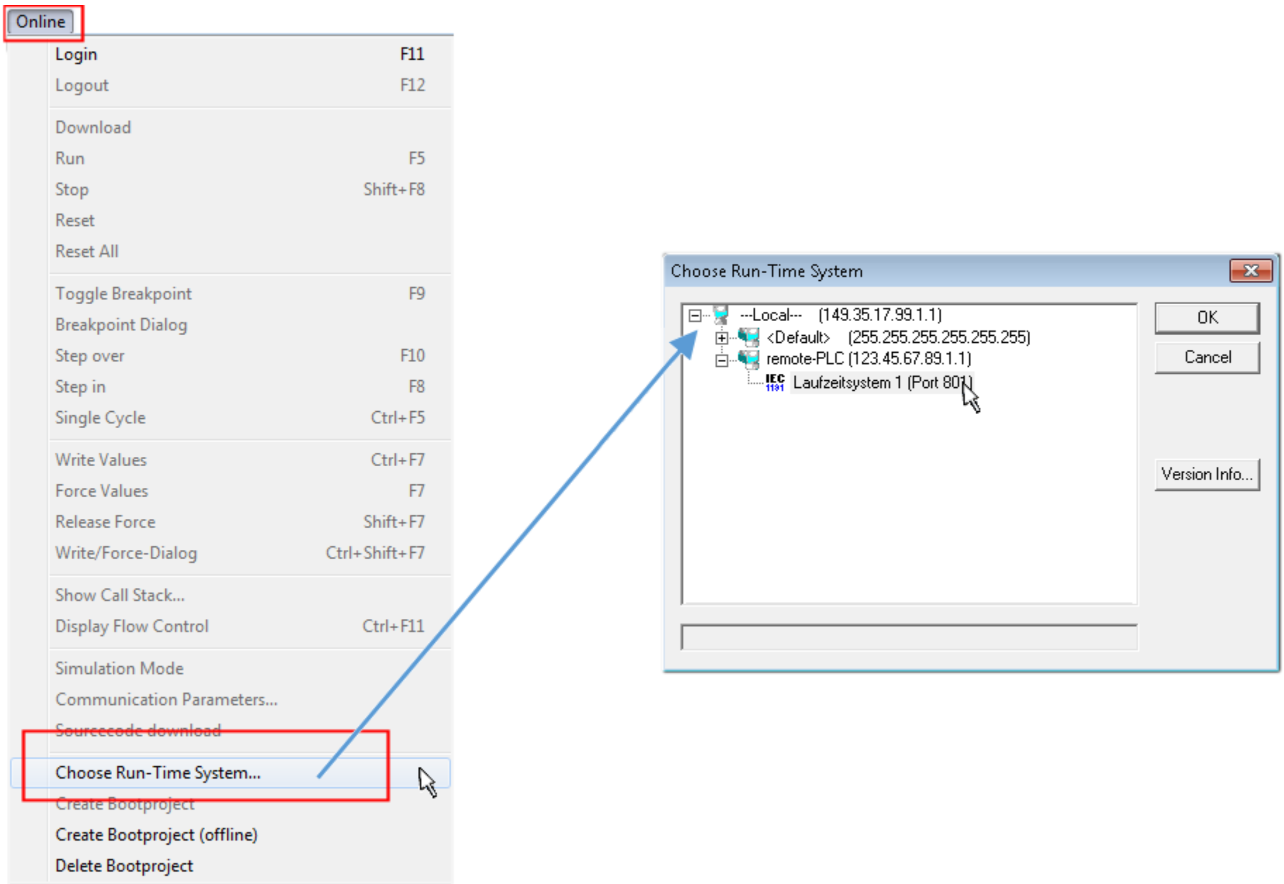
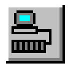


Fig. 46: Choose target system (remote)

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:

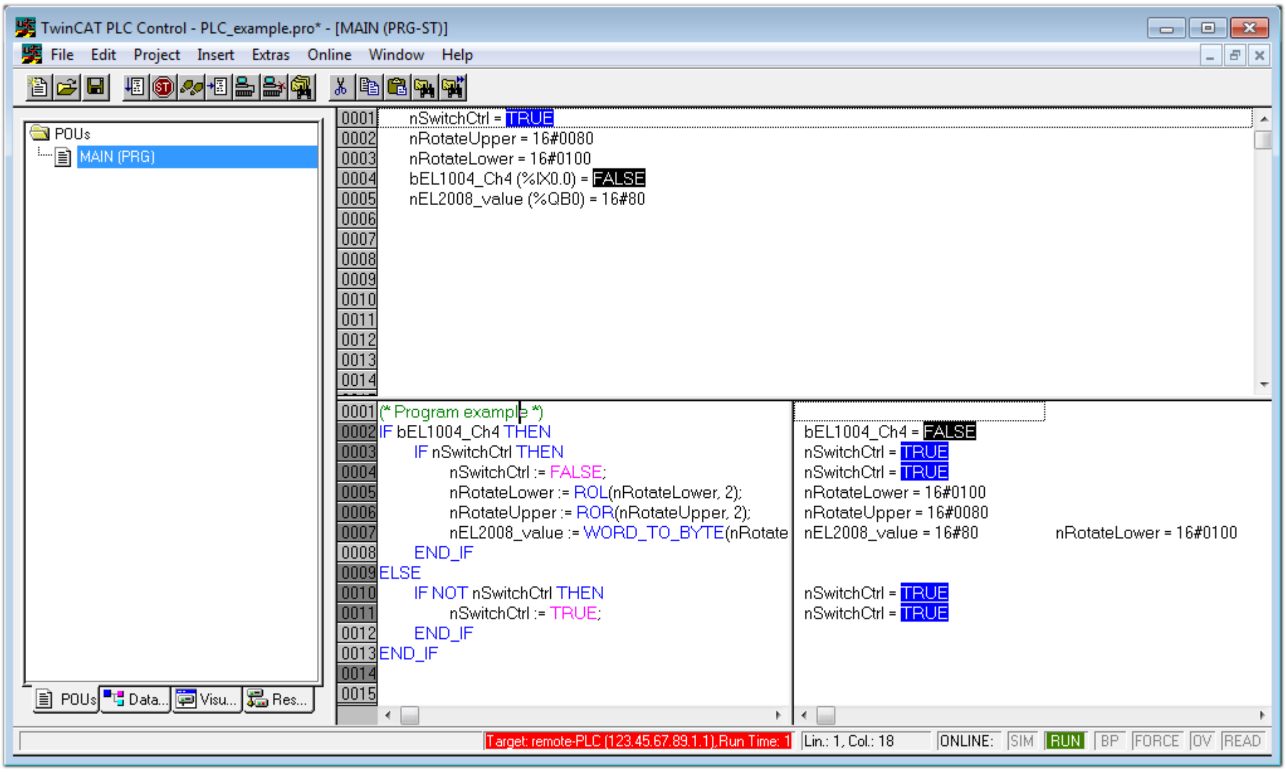


Fig. 47: PLC Control logged in, ready for program startup

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:

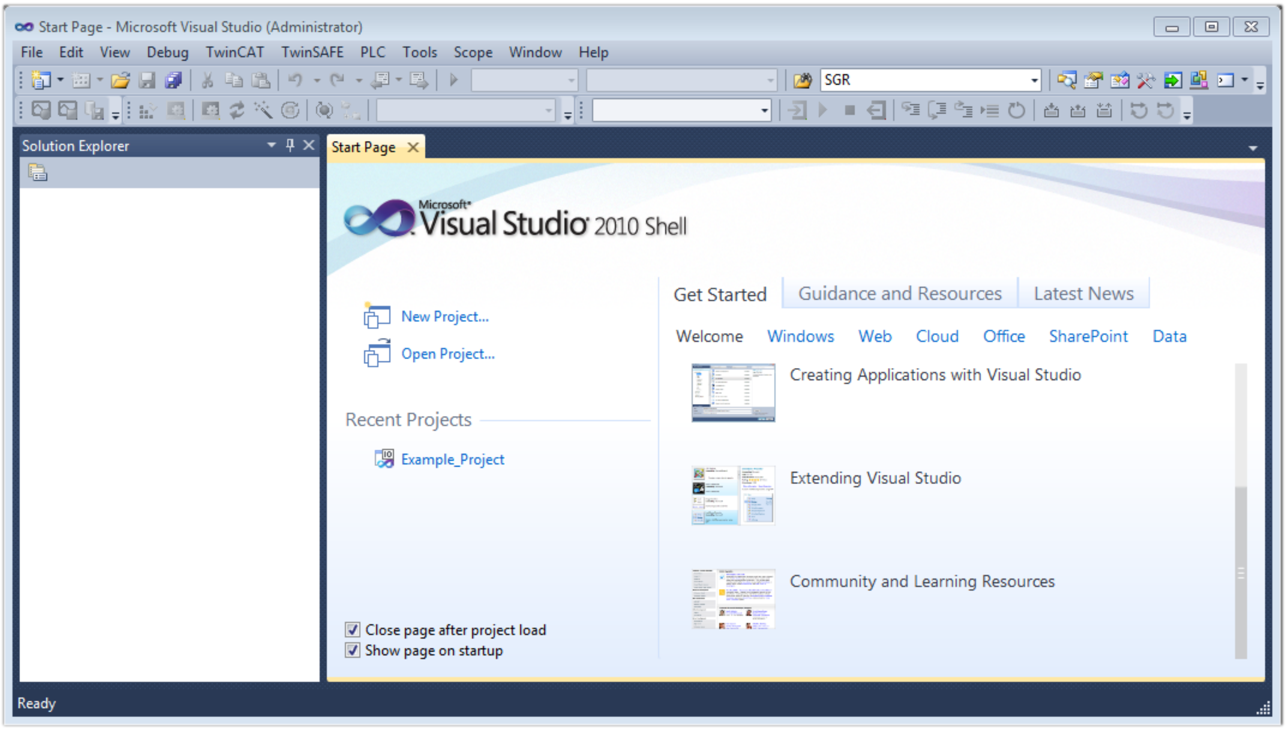


Fig. 48: Initial TwinCAT 3 user interface

First create a new project via 

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

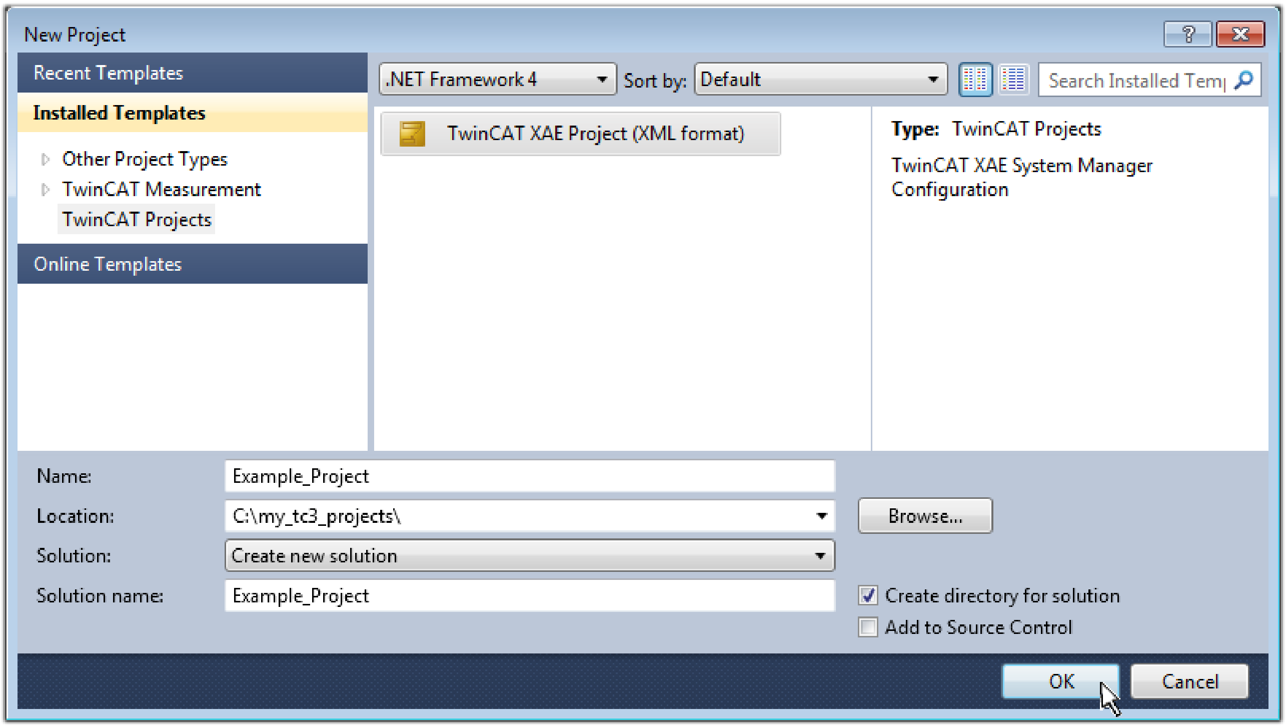


Fig. 49: Create new TwinCAT project

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

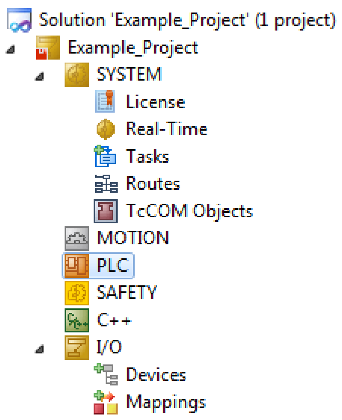
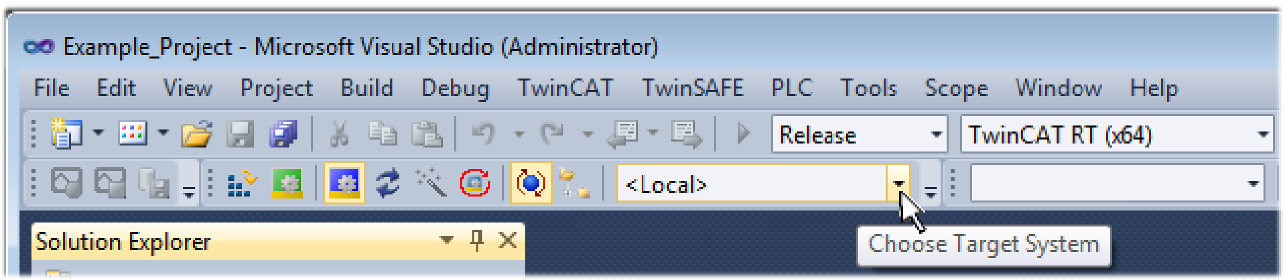


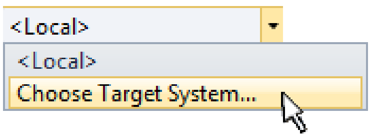
Fig. 50: New TwinCAT3 project in the project folder explorer

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 [▶ 58]".

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:

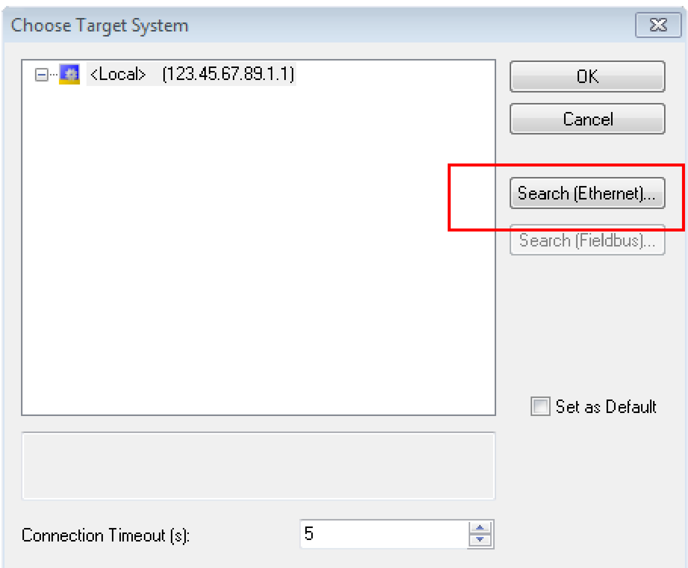


Fig. 51: Selection dialog: Choose the target system

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.

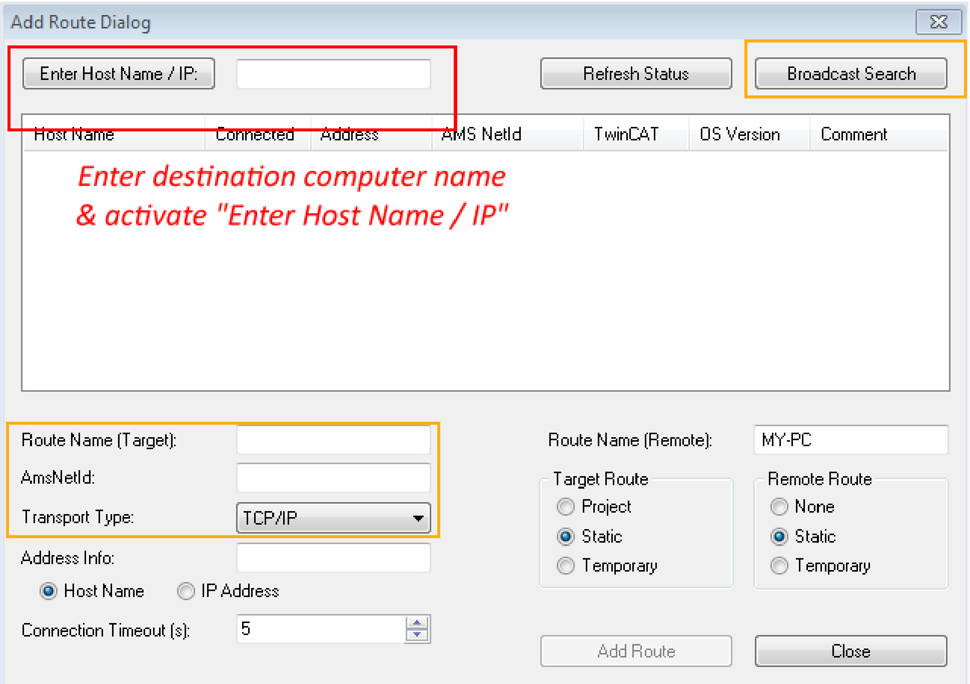
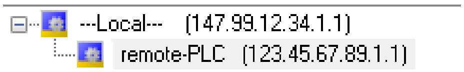


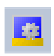
Fig. 52: Specify the PLC for access by the TwinCAT System Manager: selection of the target system

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)”.

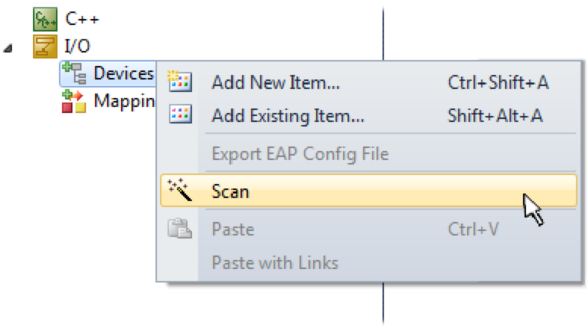


Fig. 53: Select “Scan”

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

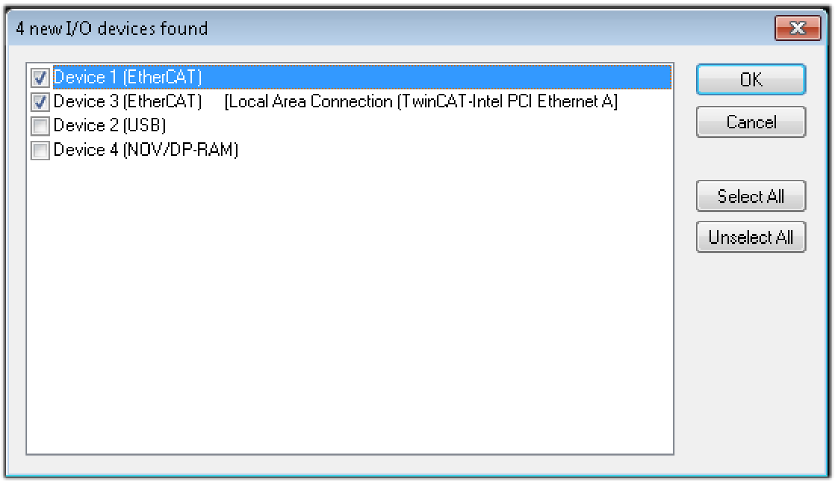


Fig. 54: Automatic detection of I/O devices: selection the devices to be integrated

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 [▶ 43] described at the beginning of this section, the result is as follows:

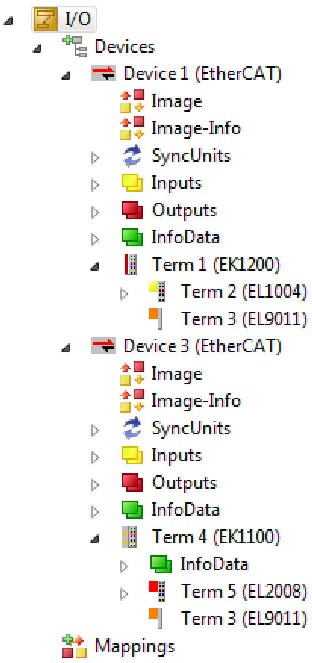


Fig. 55: Mapping of the configuration in VS shell of the TwinCAT3 environment

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:

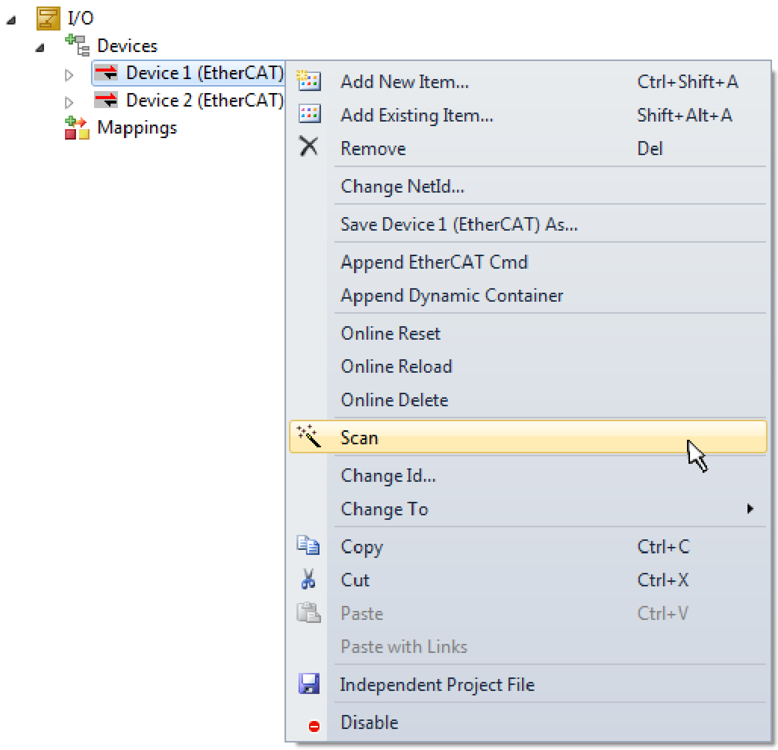


Fig. 56: Reading of individual terminals connected to a device

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….”:

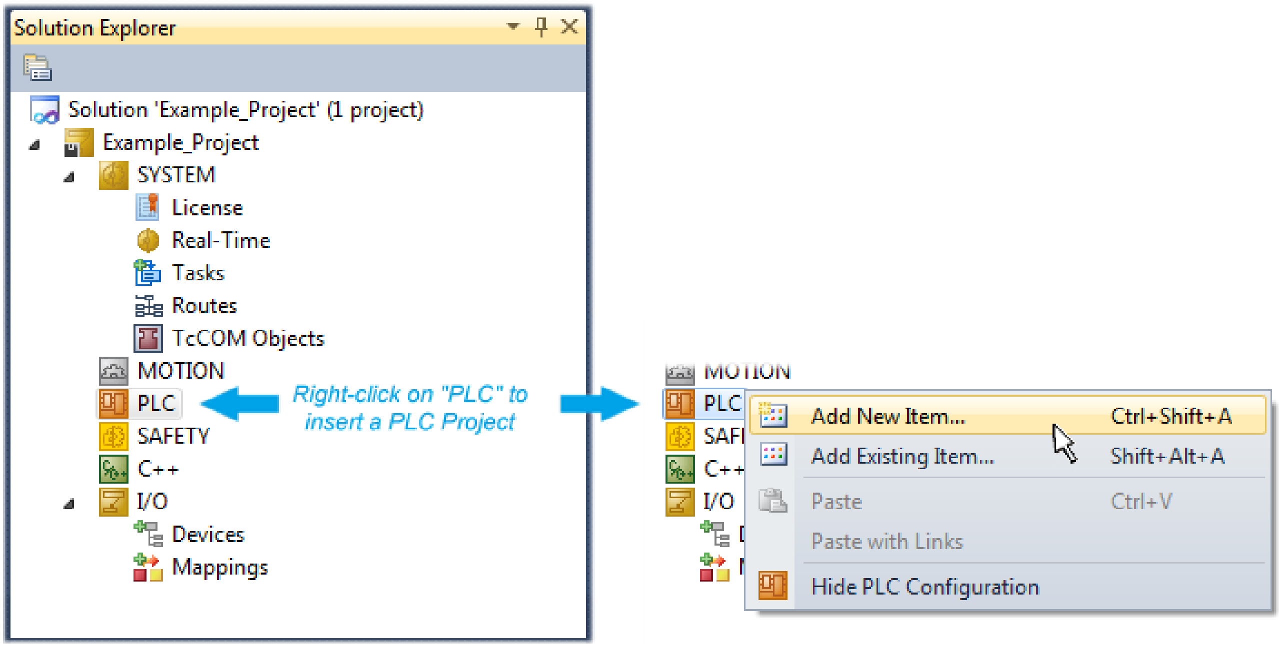


Fig. 57: Adding the programming environment in “PLC”

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

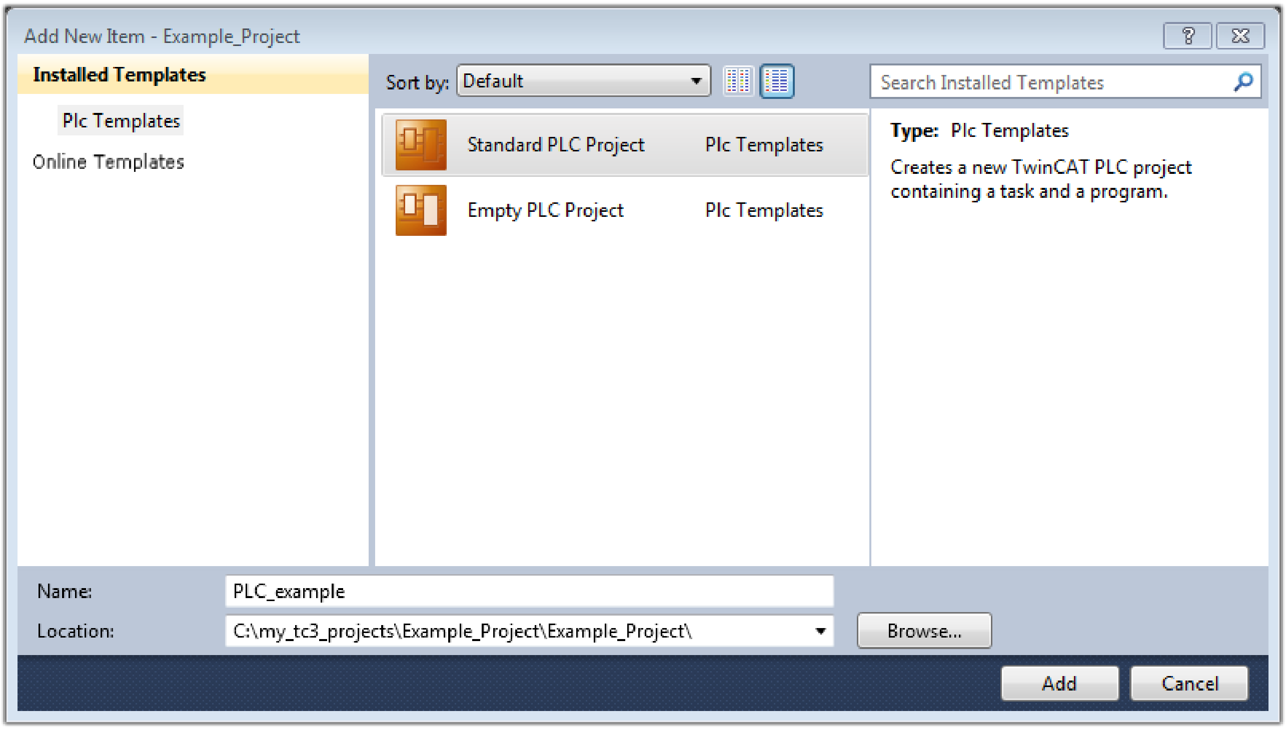


Fig. 58: Specifying the name and directory for the PLC programming environment

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:

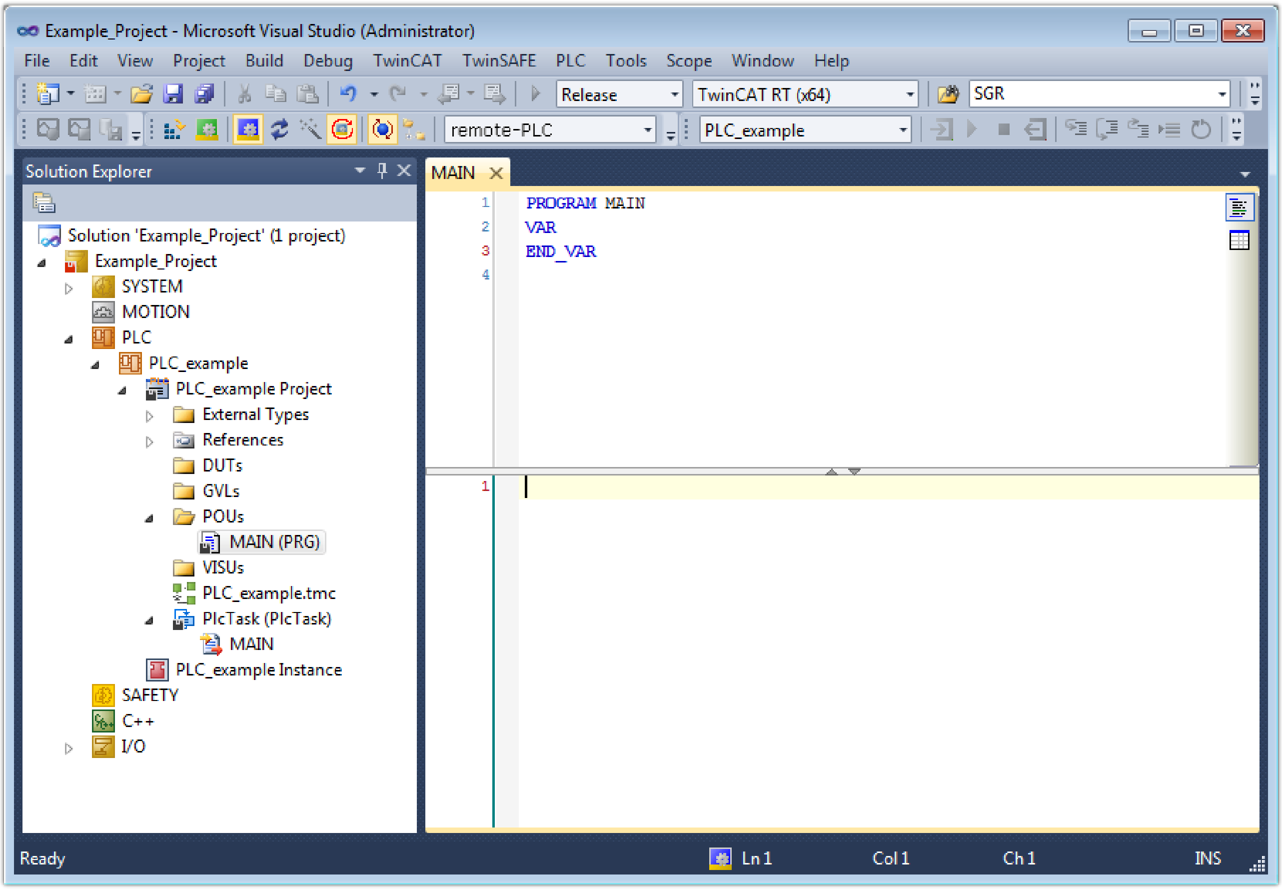


Fig. 59: Initial “Main” program of the standard PLC project

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

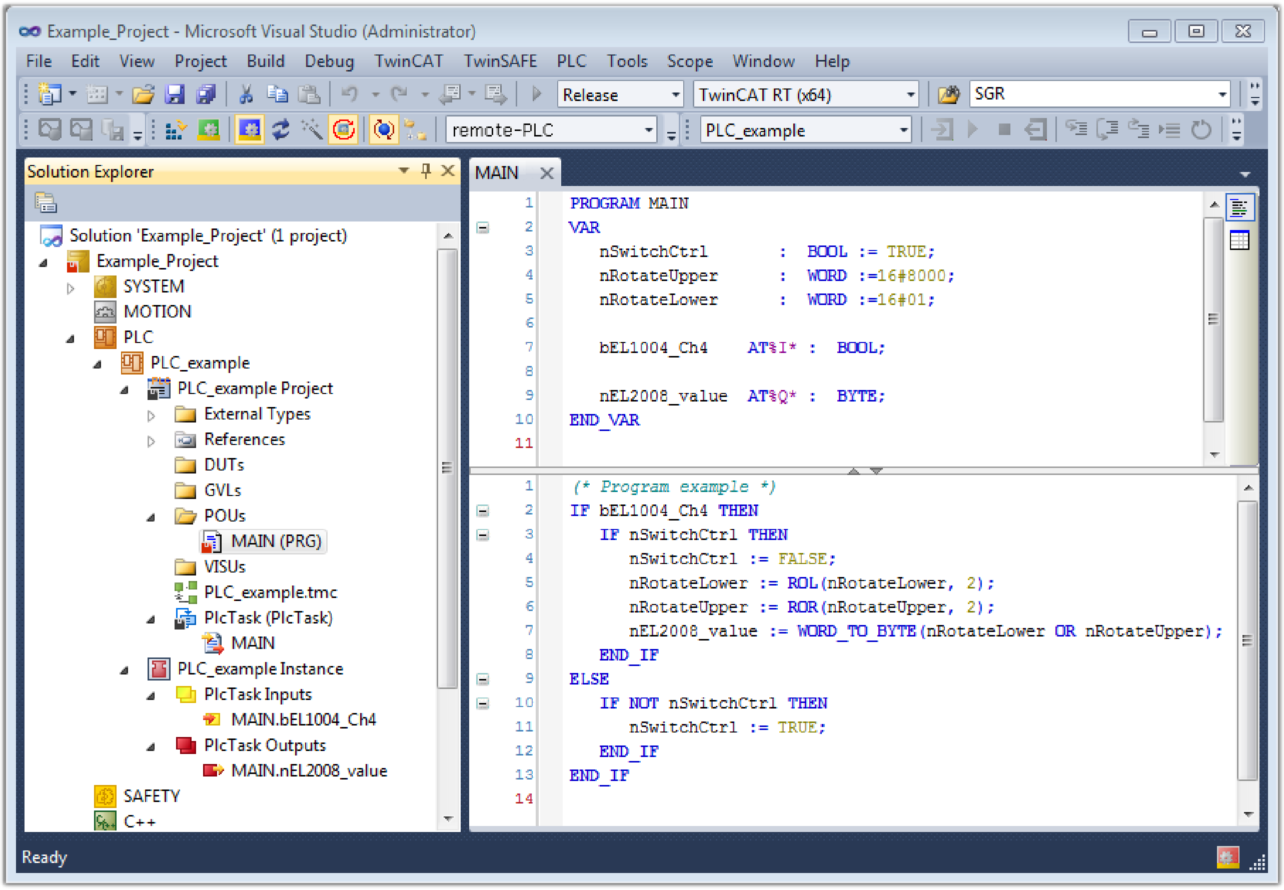


Fig. 60: Sample program with variables after a compile process (without variable integration)

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

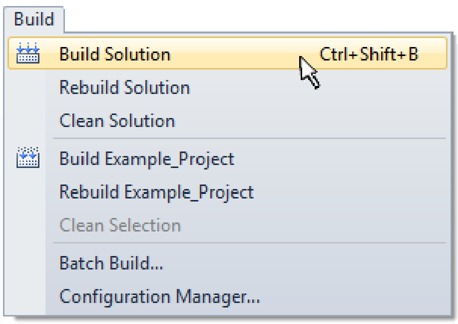
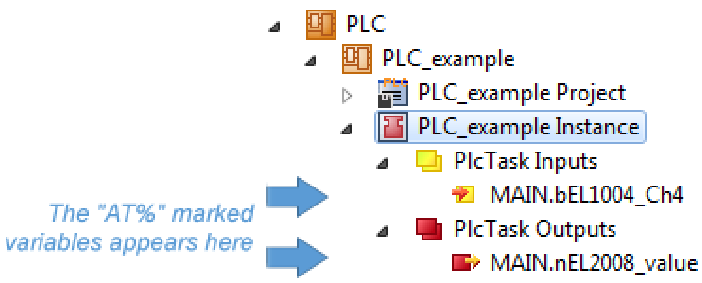


Fig. 61: Start program compilation

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:

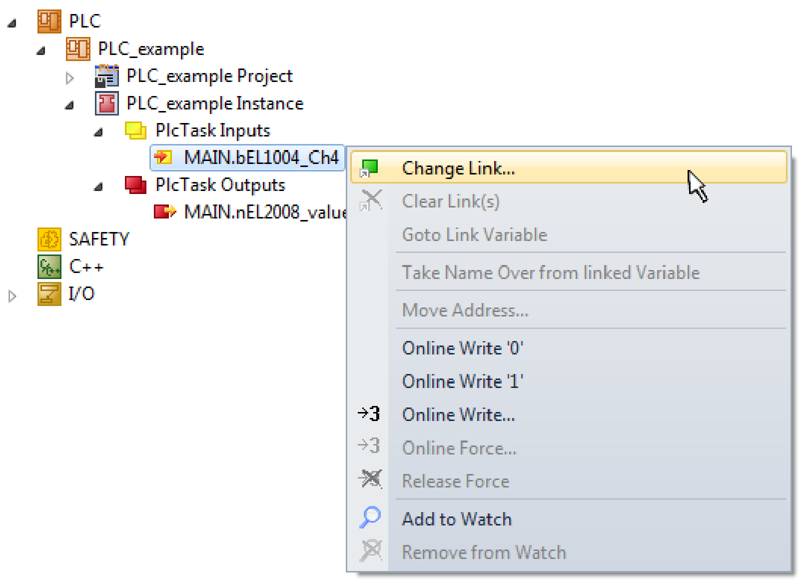


Fig. 62: Creating the links between PLC variables and process objects

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

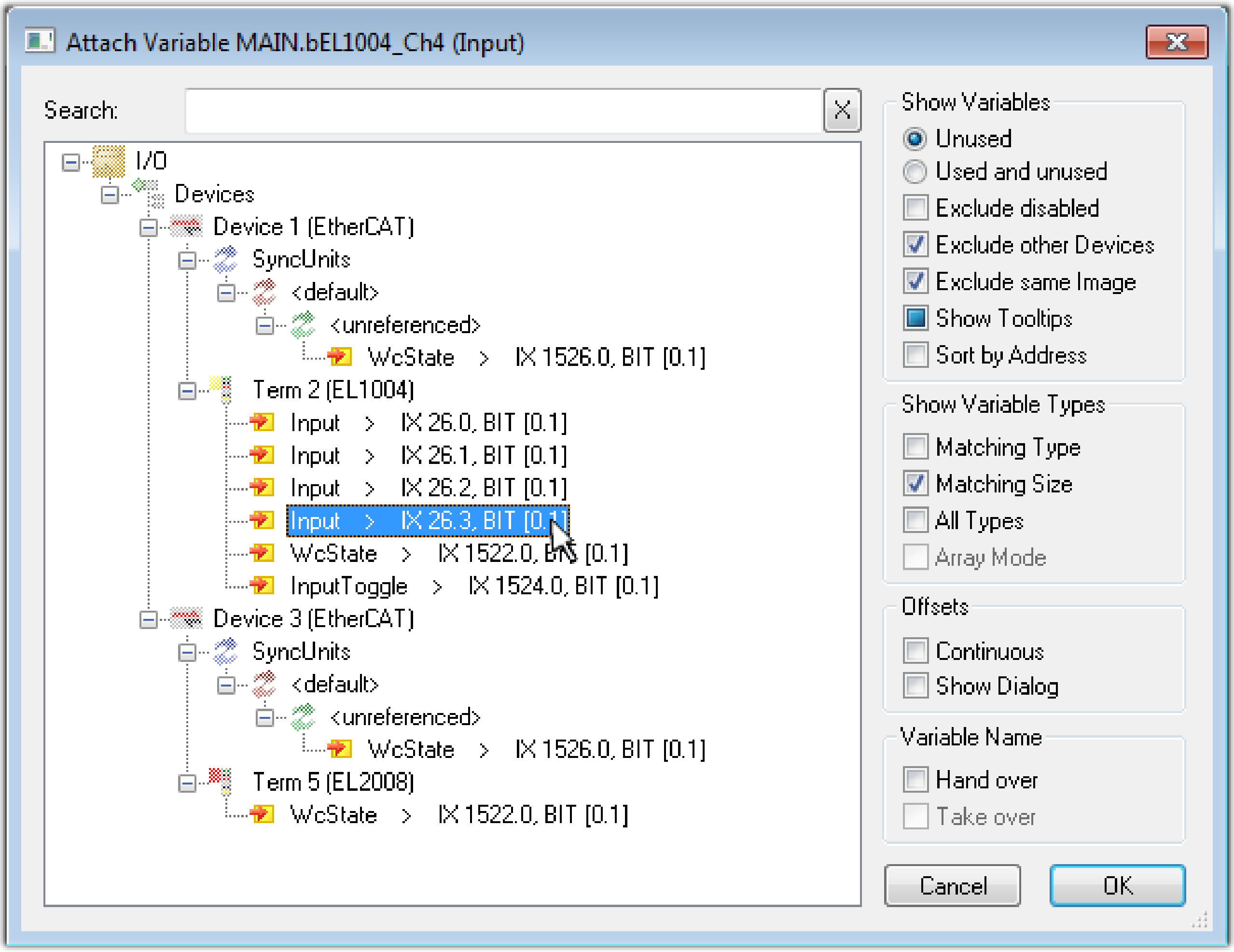


Fig. 63: Selecting PDO of type BOOL

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:

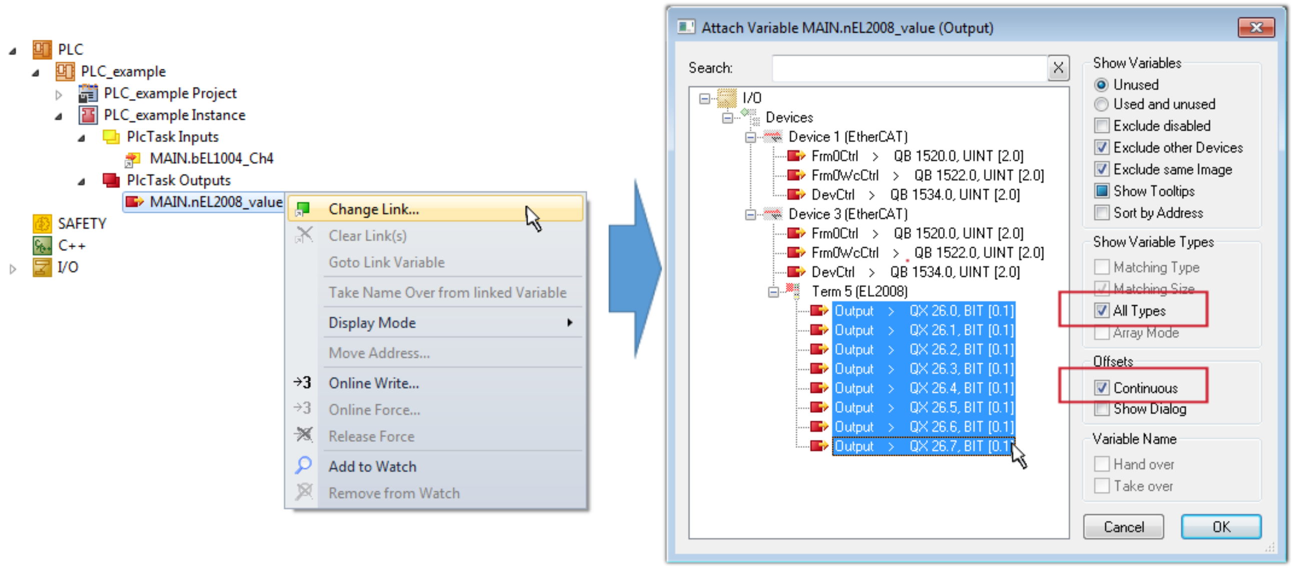


Fig. 64: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

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:

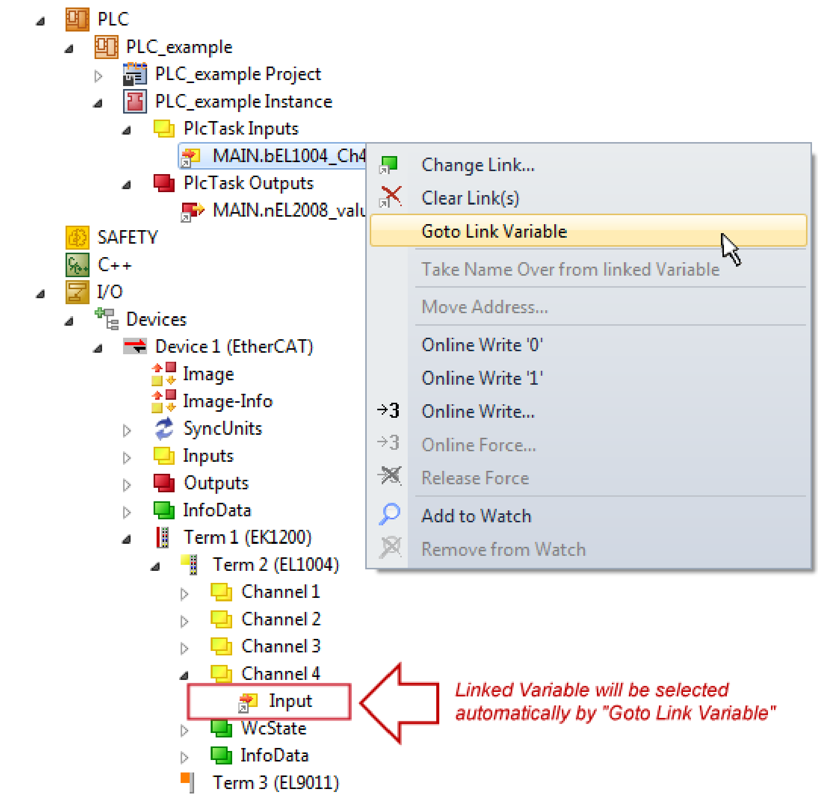


Fig. 65: Application of a “Goto Link” variable, using “MAIN.bEL1004_Ch4” as a sample

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.

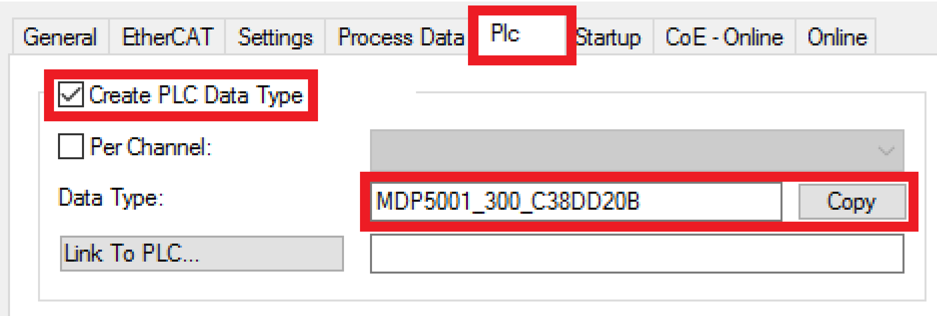


Fig. 66: Creating a PLC data type

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

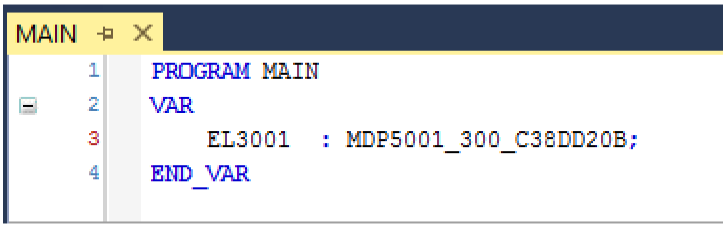


Fig. 67: Instance_of_struct

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.

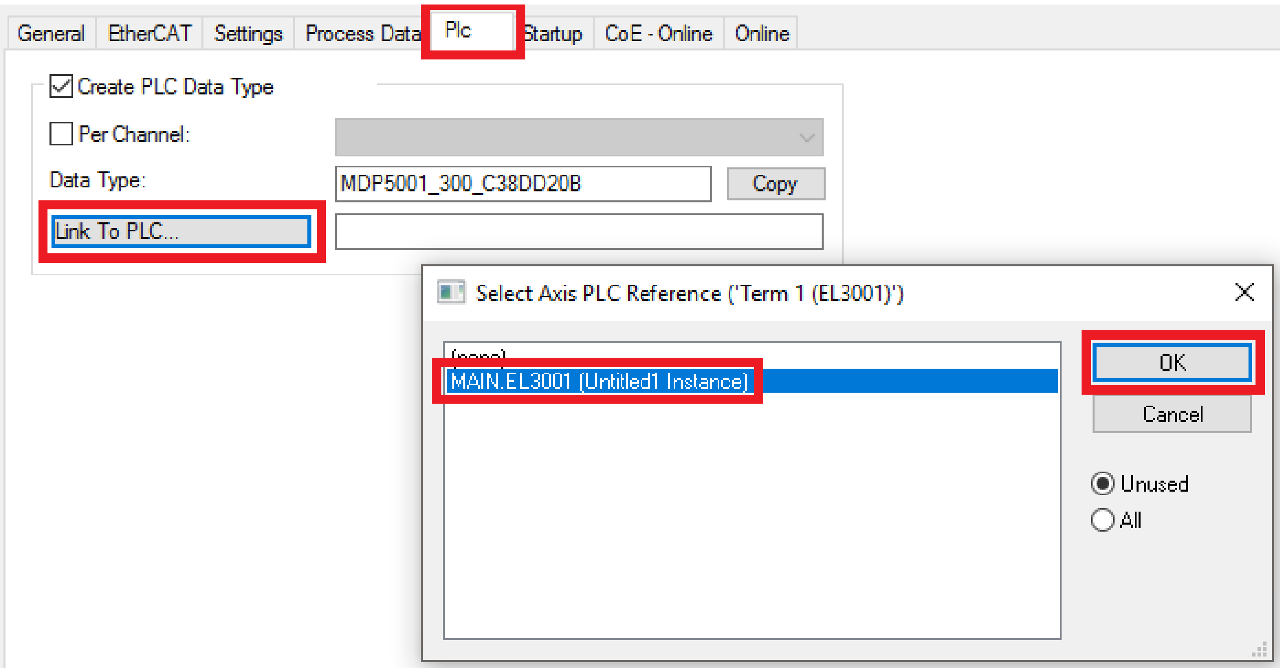


Fig. 68: Linking the structure

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

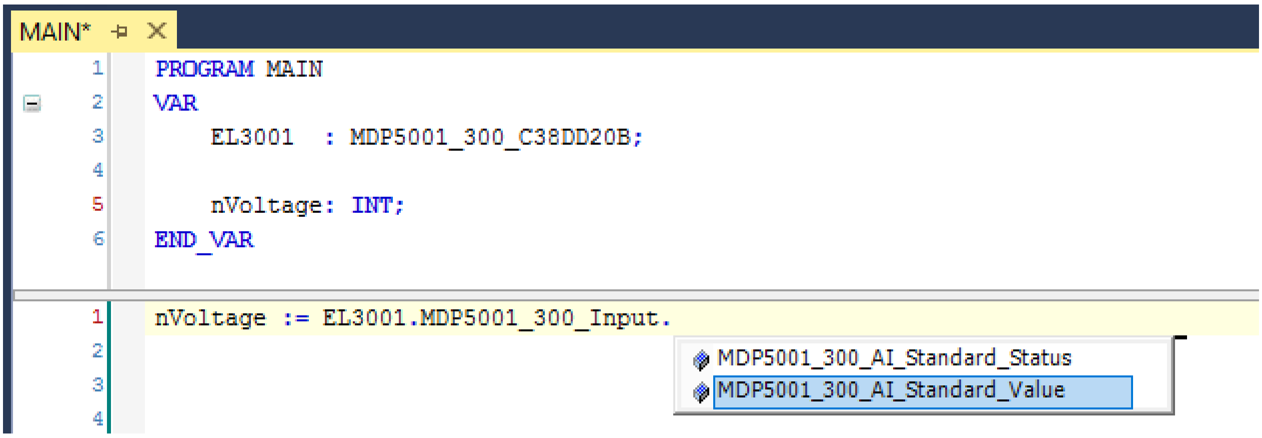
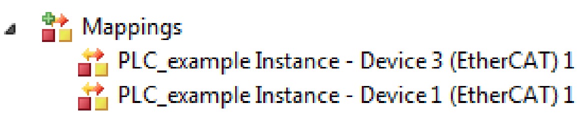


Fig. 69: Reading a variable from the structure of the process data

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:

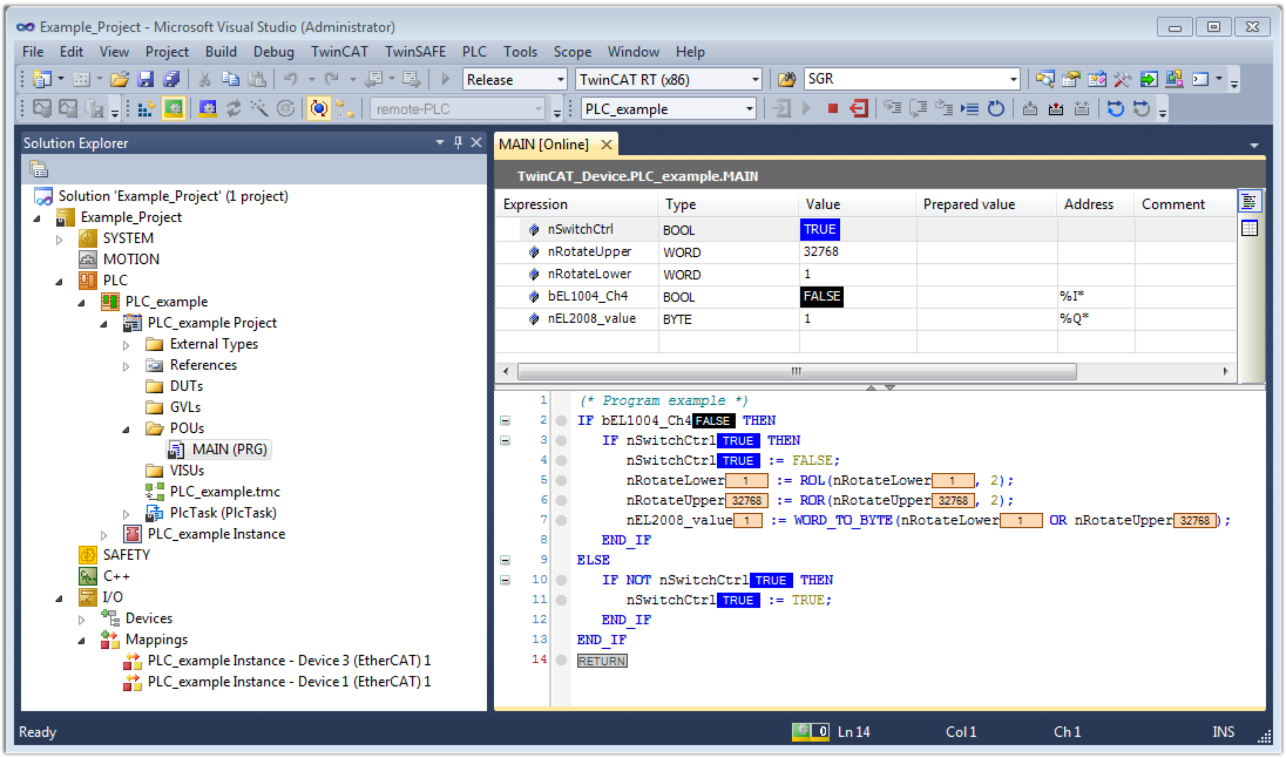


Fig. 70: TwinCAT development environment (VS shell): logged-in, after program startup

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.

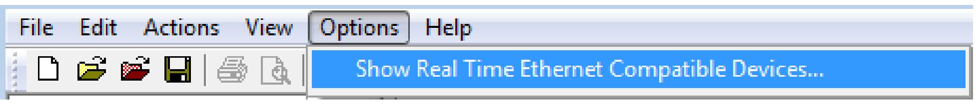


Fig. 71: System Manager “Options” (TwinCAT 2)

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

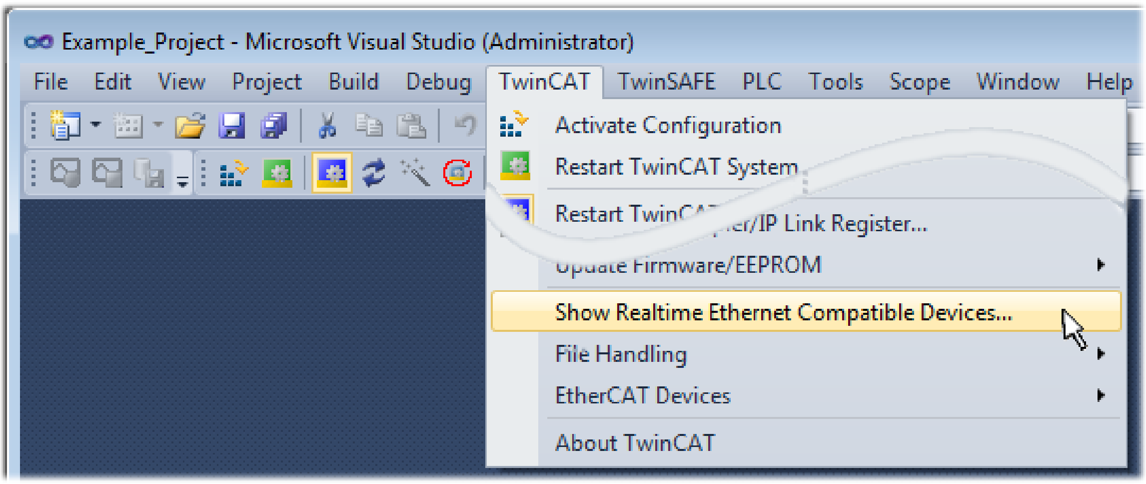


Fig. 72: Call up under VS Shell (TwinCAT 3)

The following dialog appears:

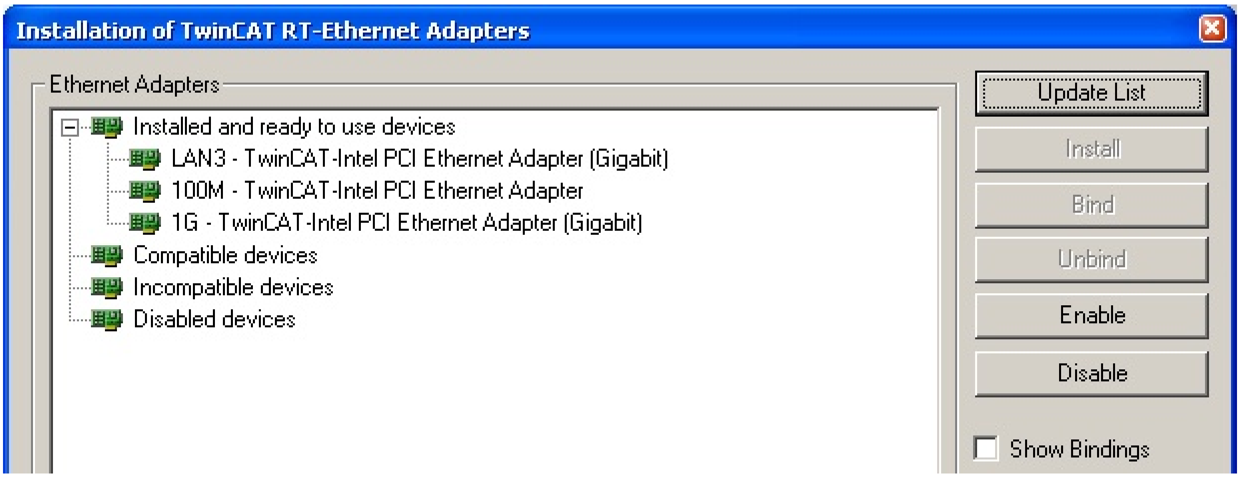


Fig. 73: Overview of network interfaces

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

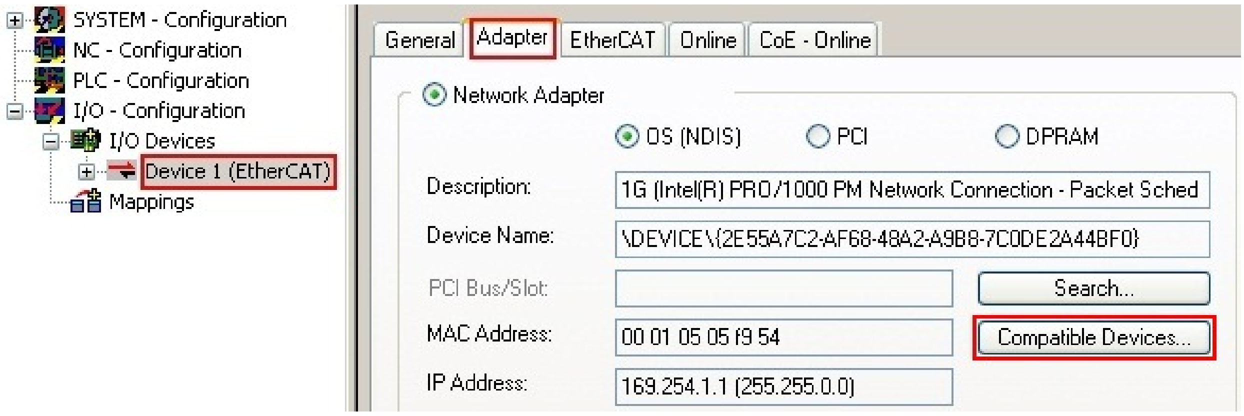
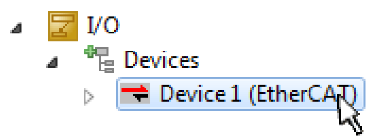


Fig. 74: EtherCAT device properties(TwinCAT 2): click on “Compatible Devices…” of tab “Adapte””

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)

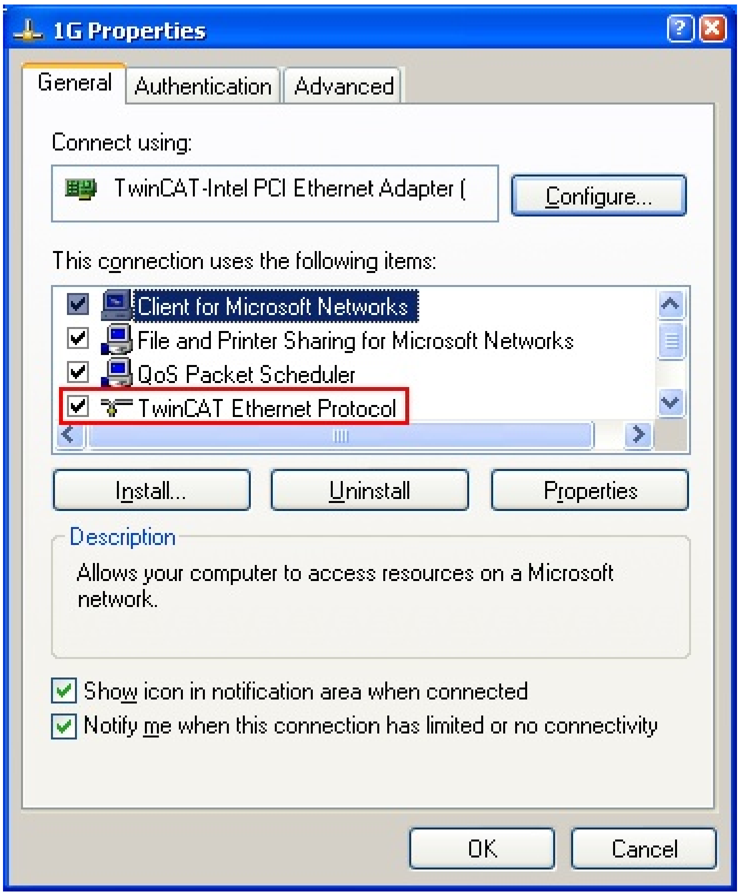


Fig. 75: Windows properties of the network interface

A correct setting of the driver could be:

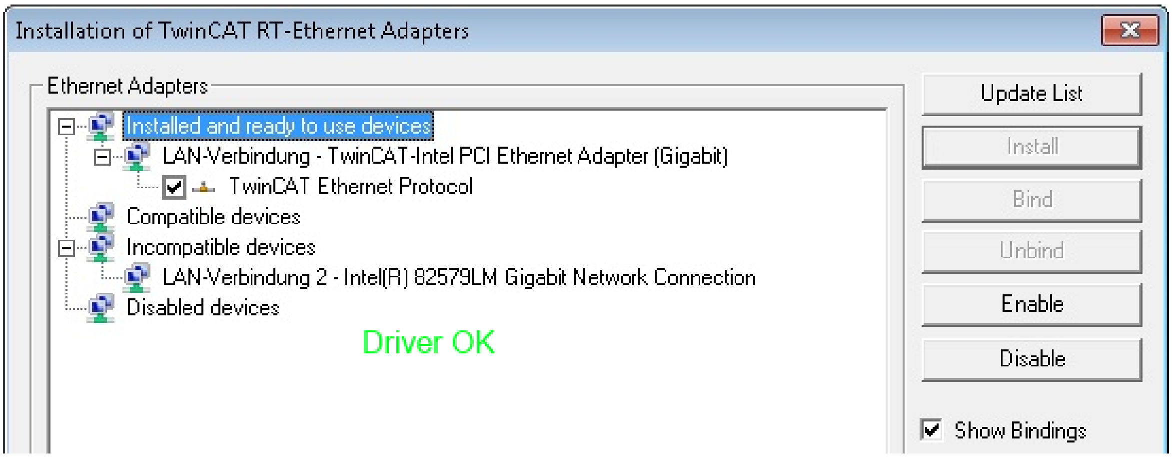


Fig. 76: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

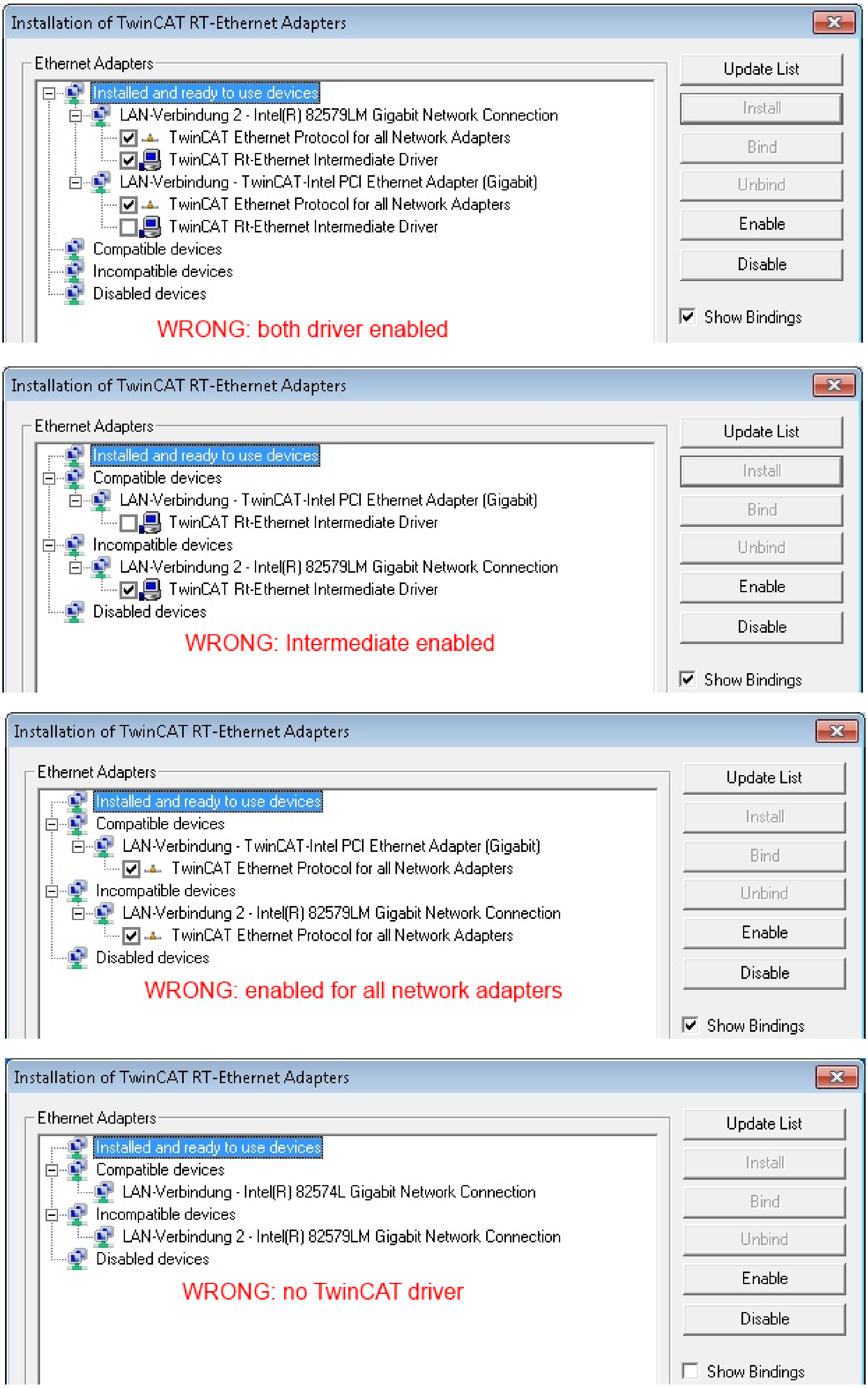


Fig. 77: Incorrect driver settings for the Ethernet port

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.

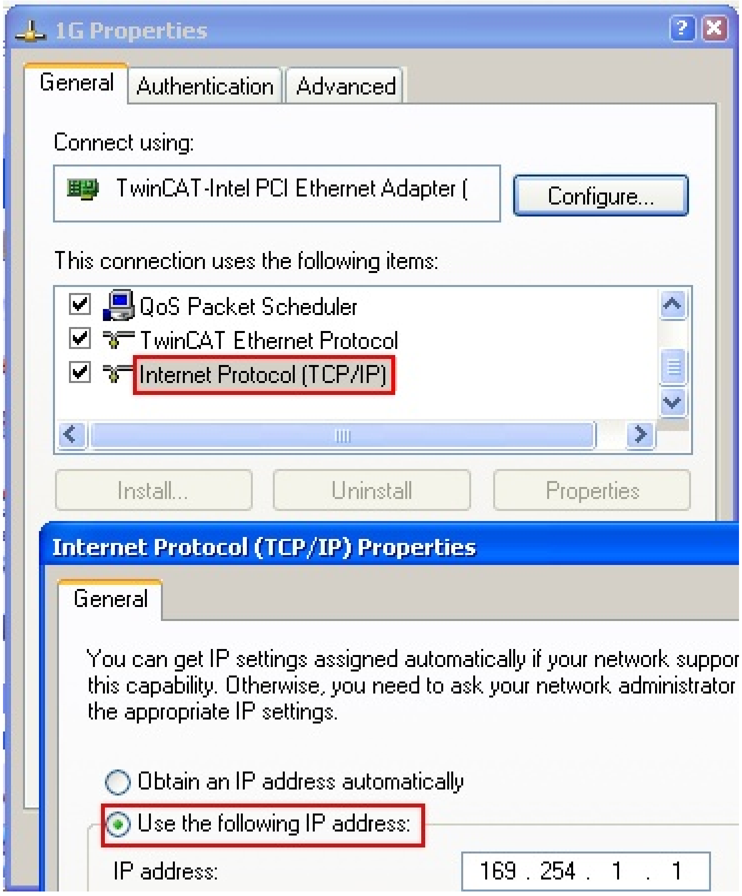


Fig. 78: TCP/IP setting for the Ethernet port

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 [▸ 78] 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”

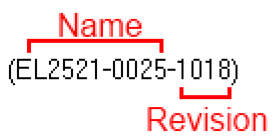


Fig. 79: Identifier structure

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 [▸ 8].

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.

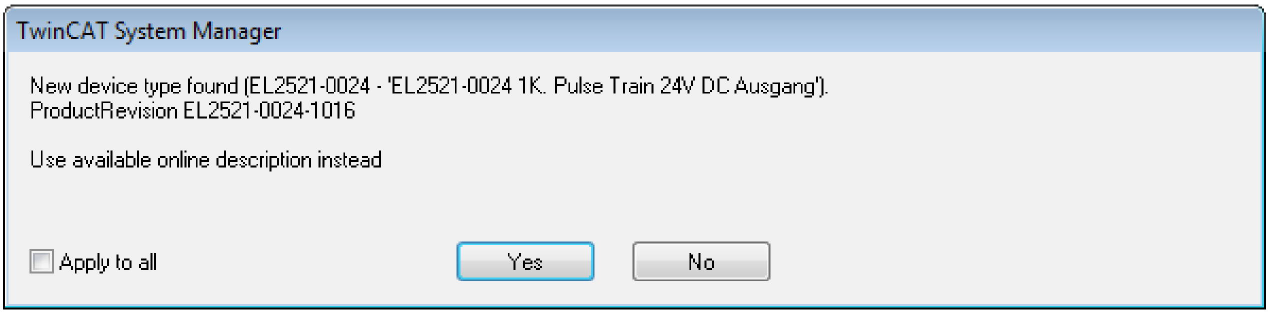


Fig. 80: OnlineDescription information window (TwinCAT 2)

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

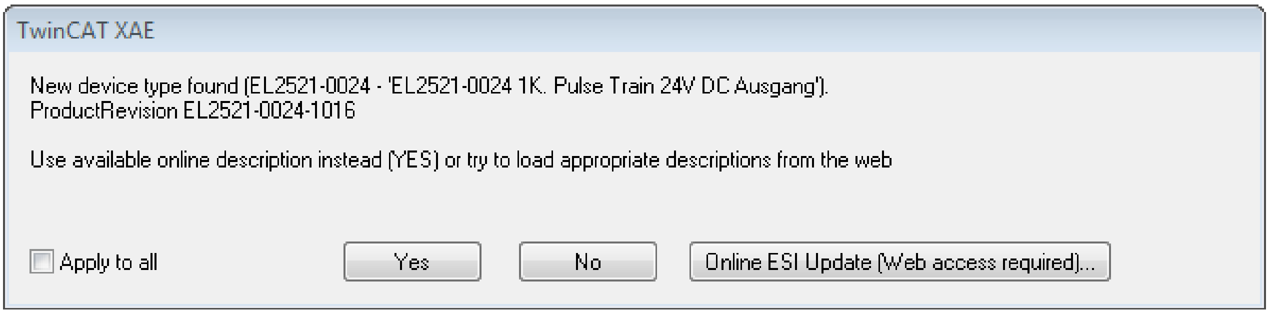


Fig. 81: Information window OnlineDescription (TwinCAT 3)

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 [▶ 79]”.

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

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

Fig. 82: 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)*.

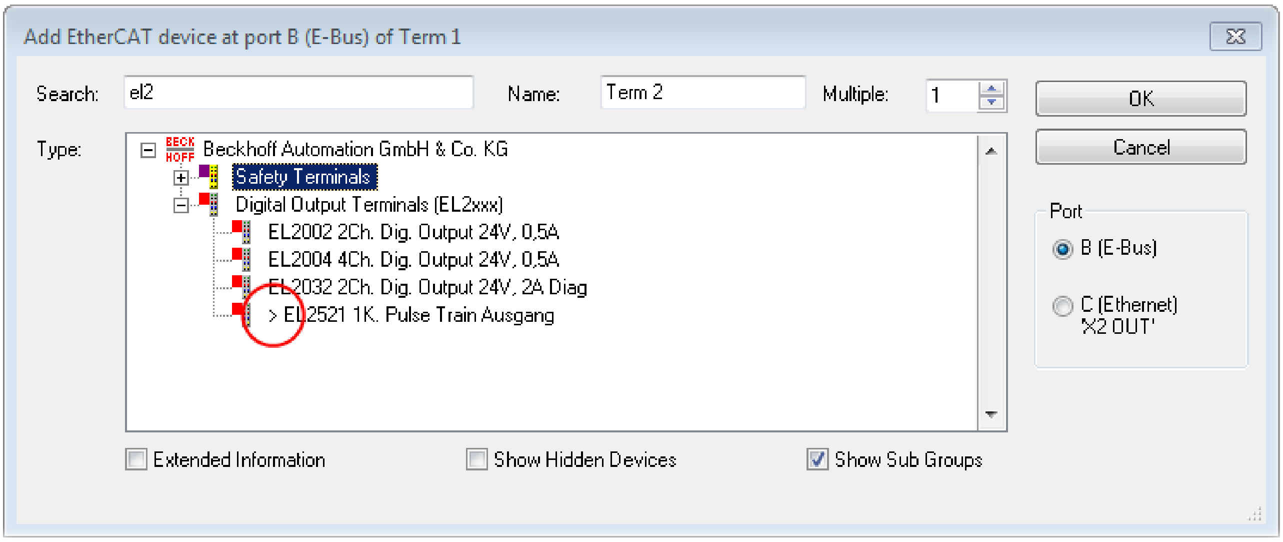


Fig. 83: Indication of an online recorded ESI of EL2521 as an example

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.

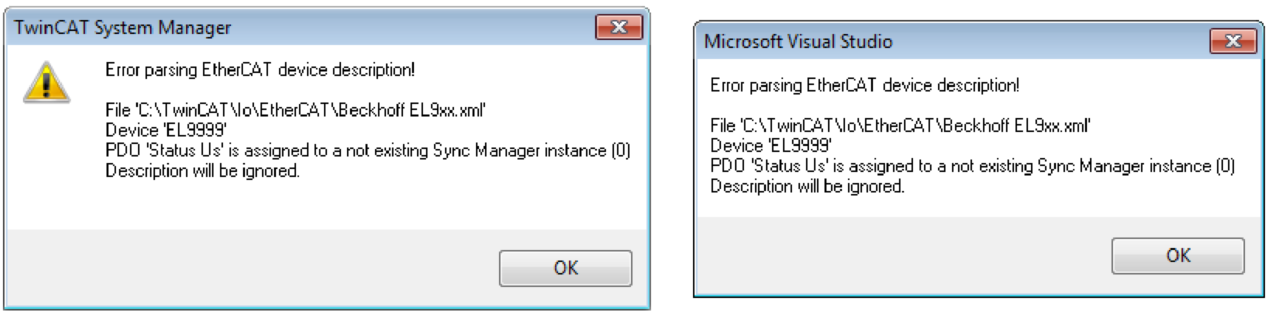


Fig. 84: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

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:

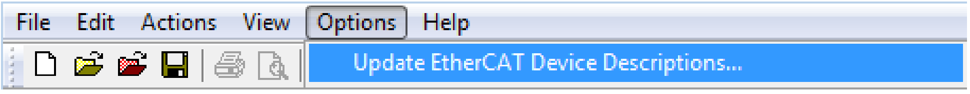


Fig. 85: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:

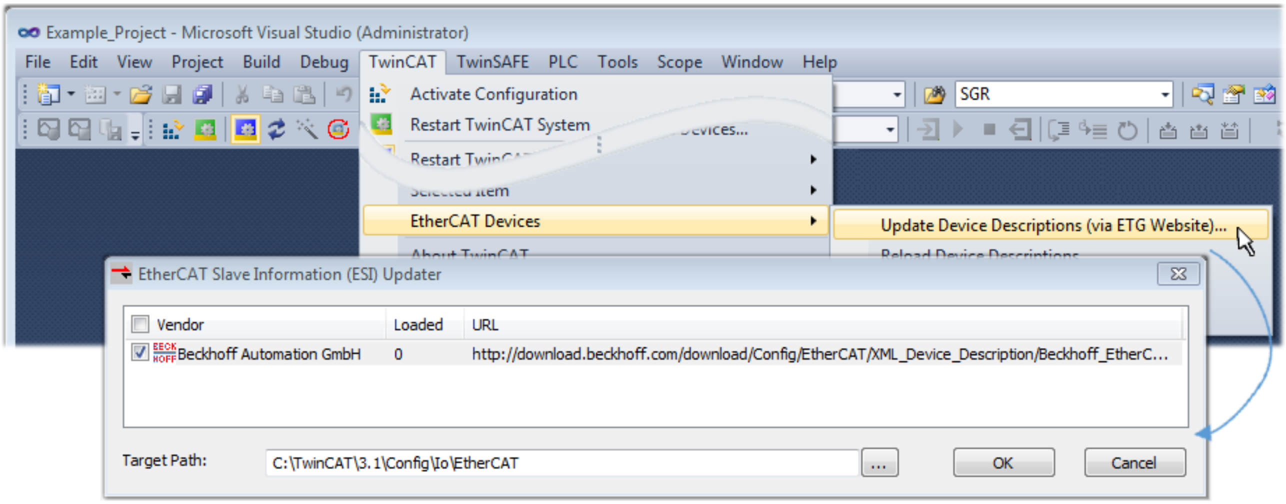


Fig. 86: Using the ESI Updater (TwinCAT 3)

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” [▶ 74].

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 [▶ 84] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 85]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 88]

The scan with existing configuration [▶ 89] 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.

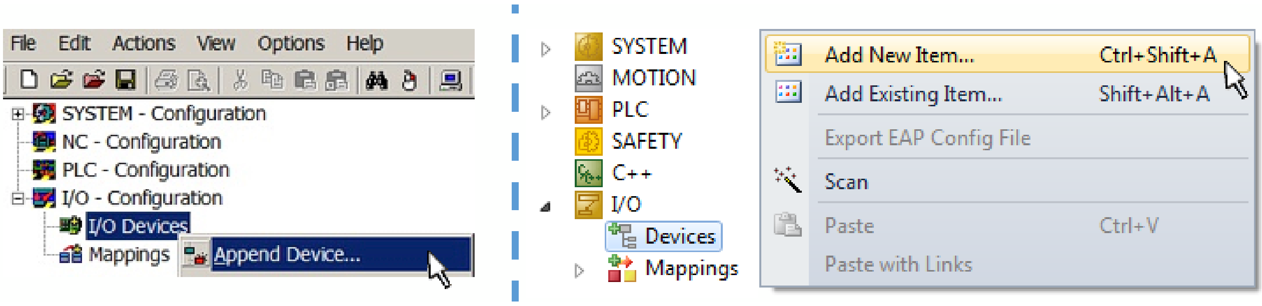


Fig. 87: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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”.

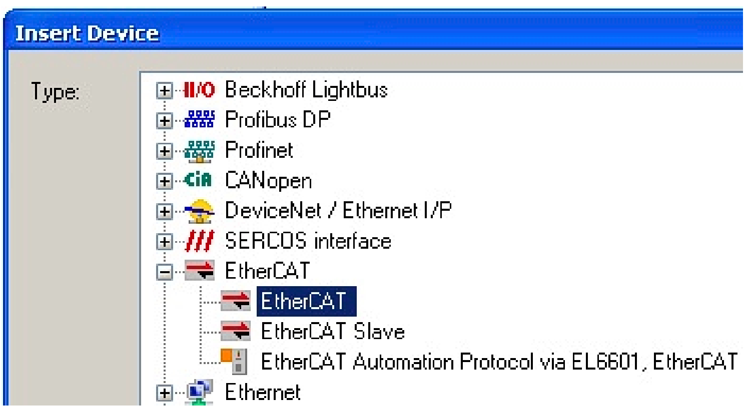


Fig. 88: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

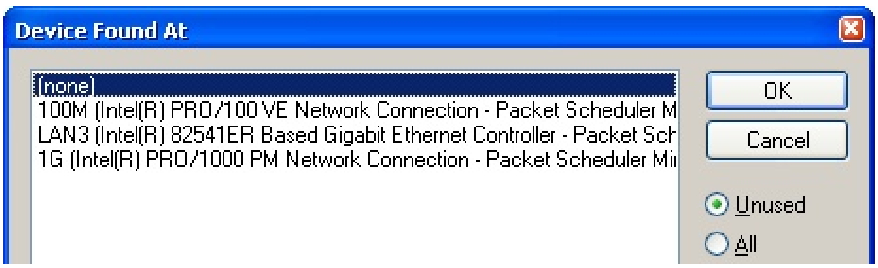


Fig. 89: Selecting the Ethernet port

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)”.

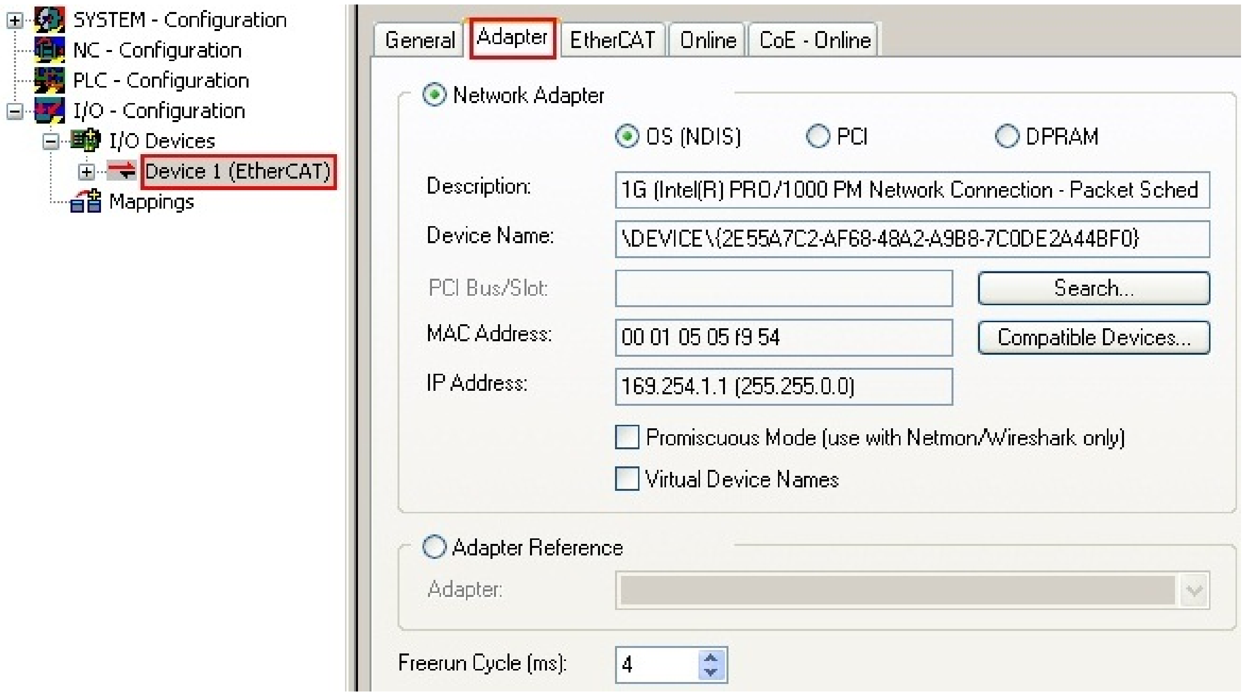
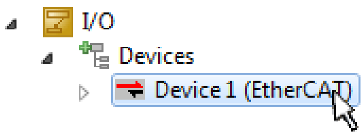


Fig. 90: EtherCAT device properties (TwinCAT 2)

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 [▶ 69].

Defining EtherCAT slaves

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

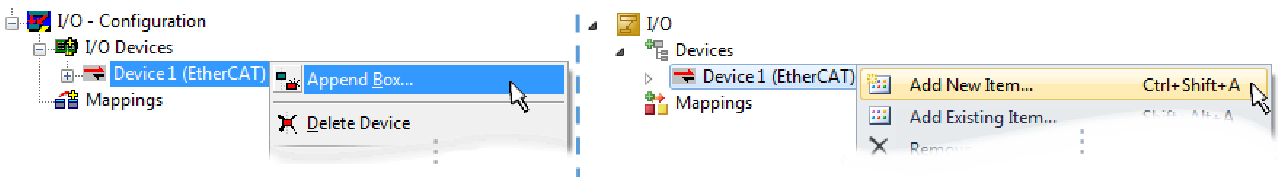


Fig. 91: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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).

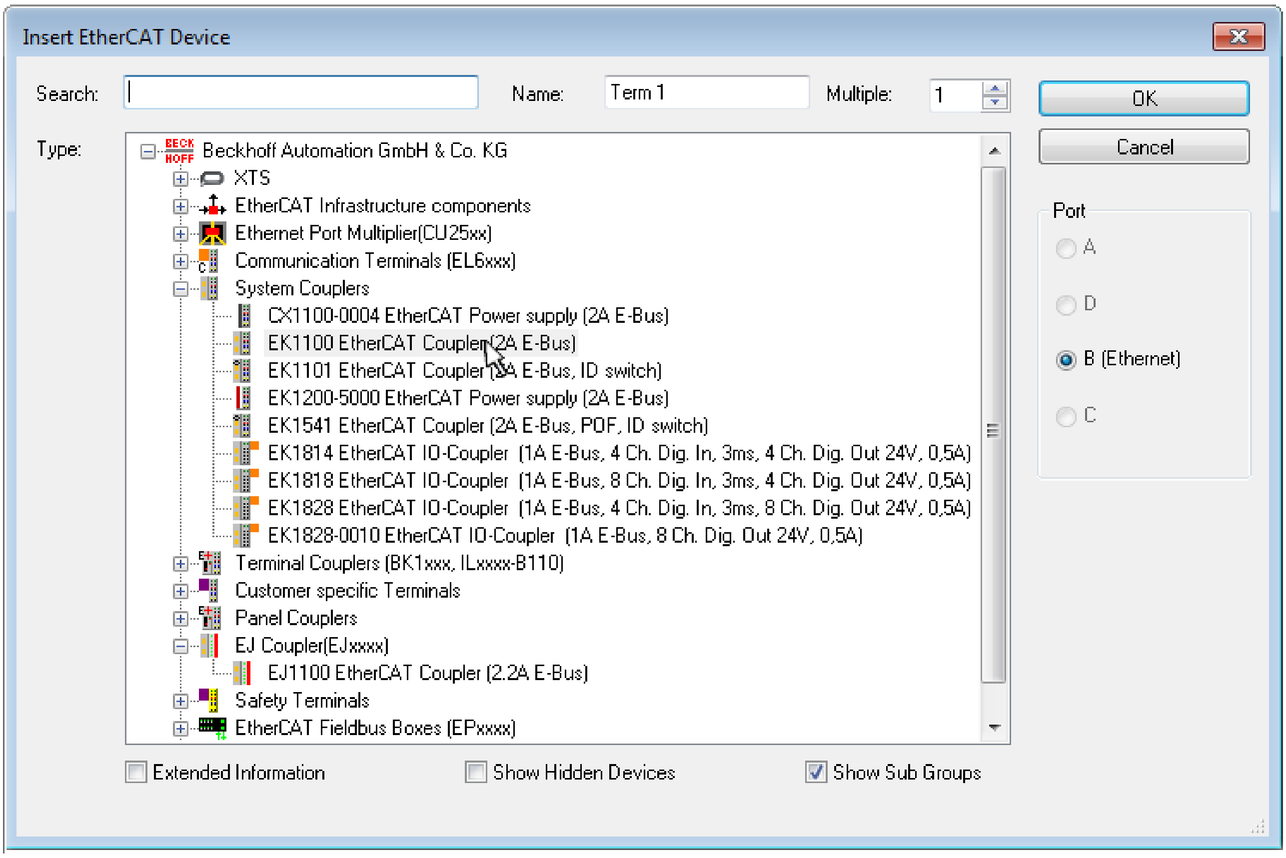


Fig. 92: Selection dialog for new EtherCAT device

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”.

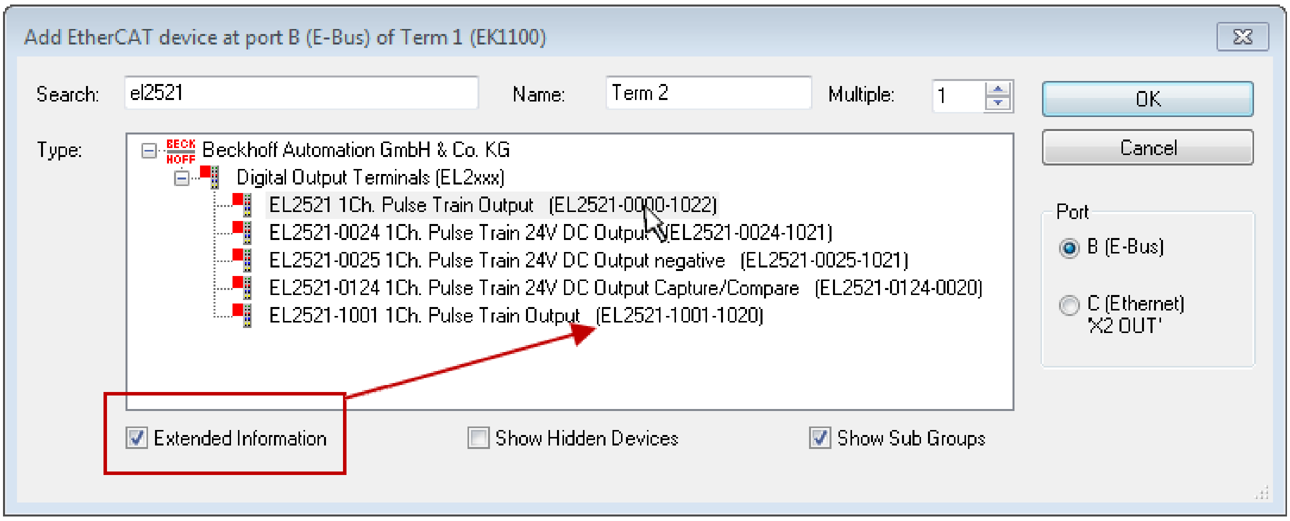


Fig. 93: Display of device revision

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”.

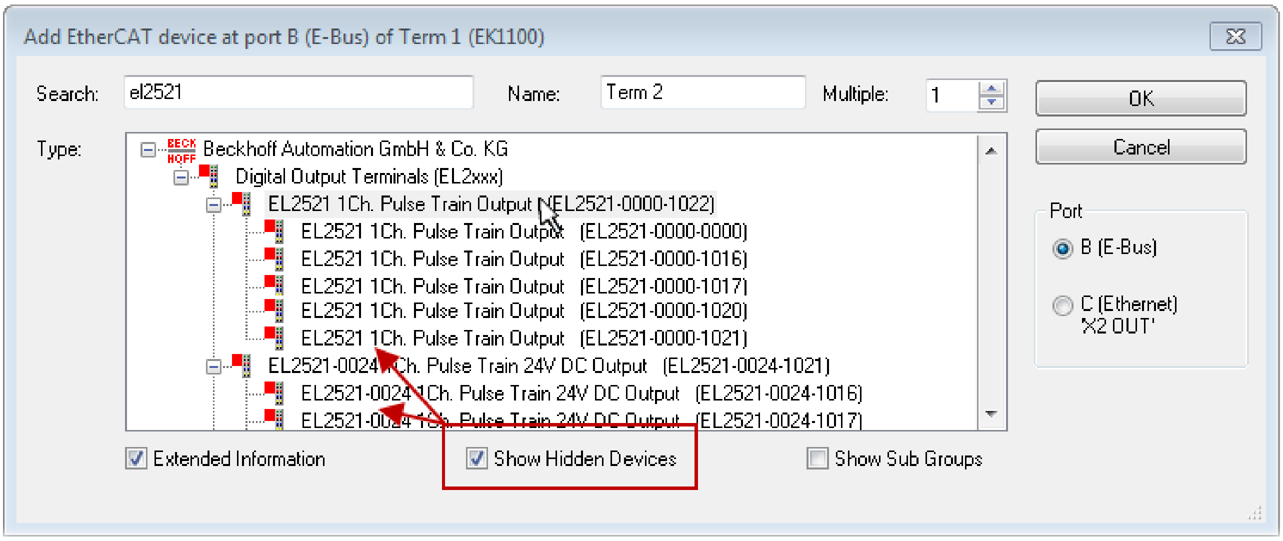


Fig. 94: Display of previous revisions

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.

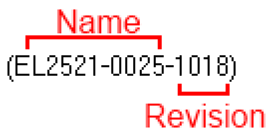


Fig. 95: Name/revision of the terminal

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, ...

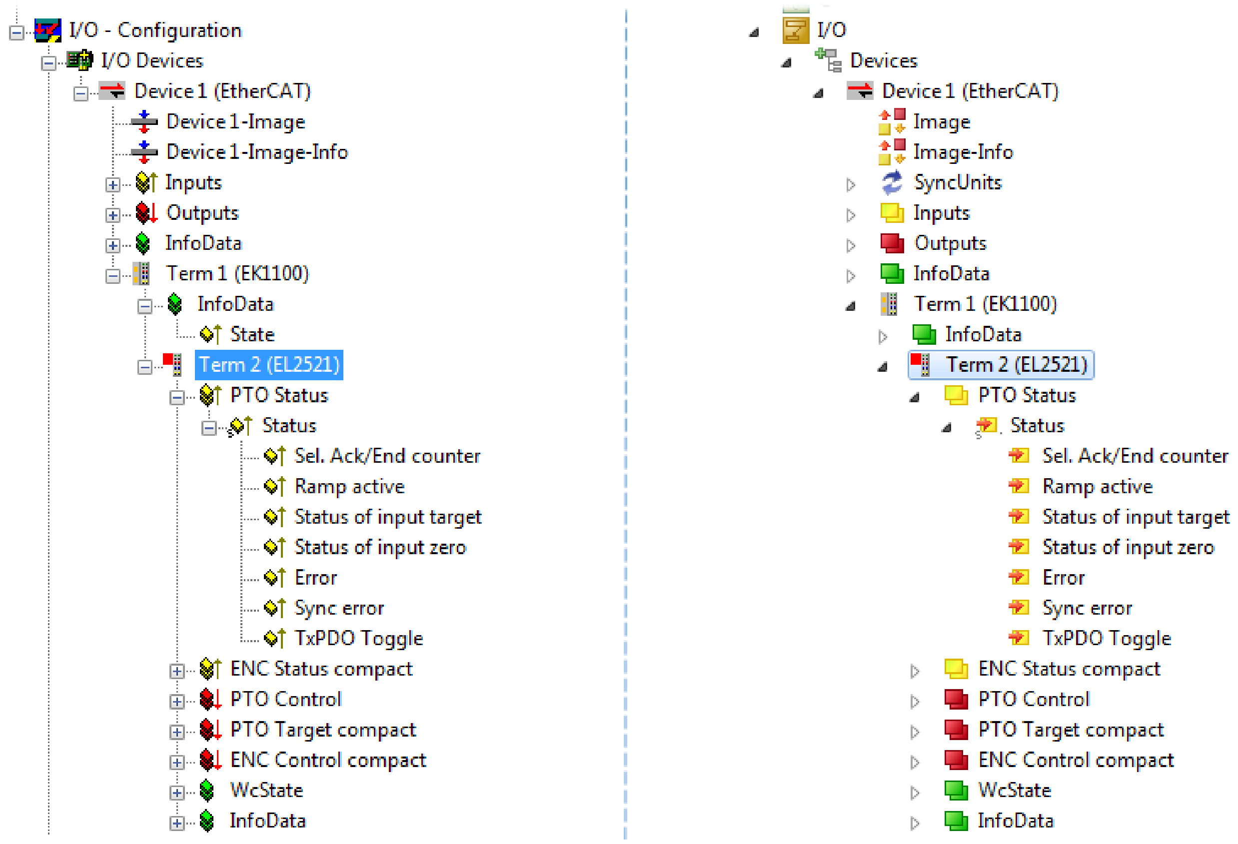


Fig. 96: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)

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:  .
- 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.

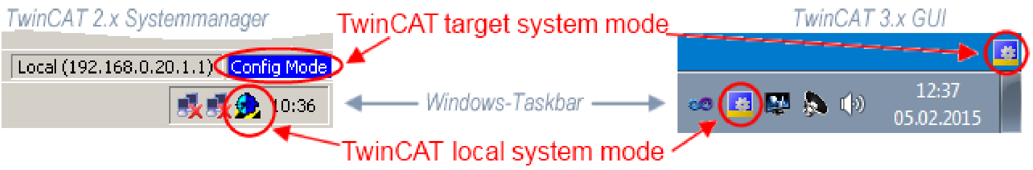


Fig. 97: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

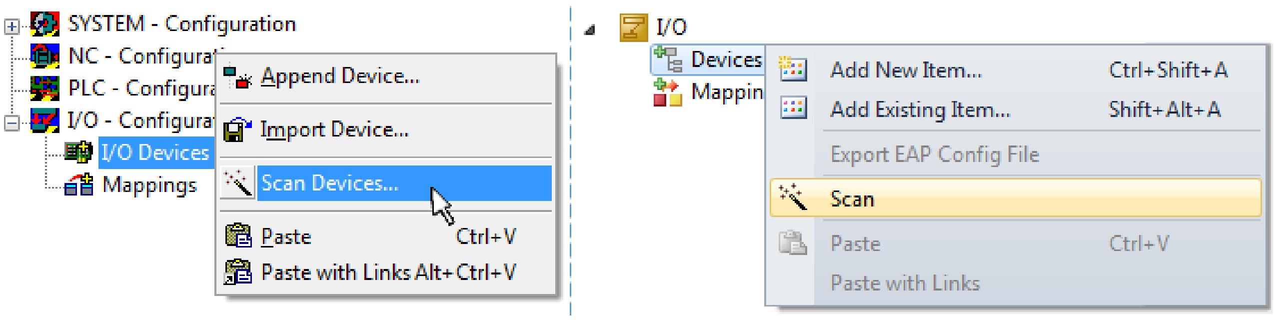


Fig. 98: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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.

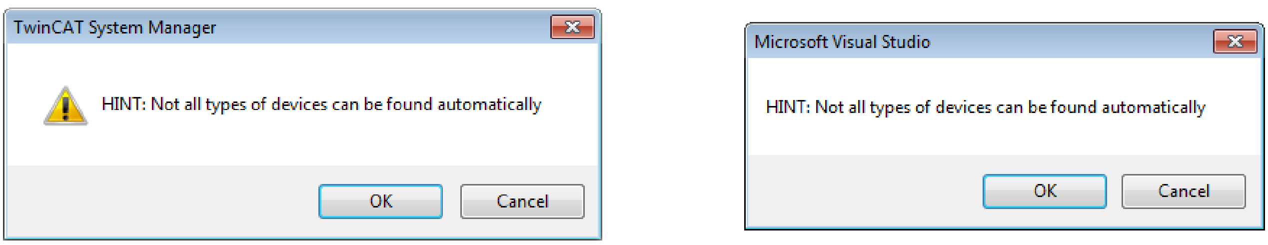


Fig. 99: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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” .

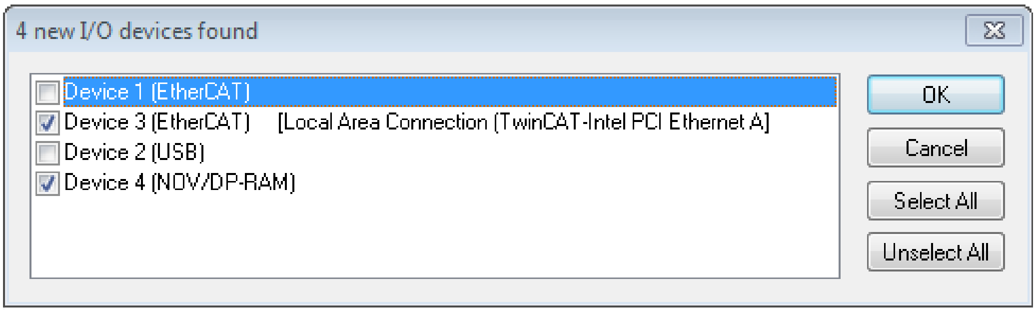


Fig. 100: Detected Ethernet devices

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 [▶ 69].

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.

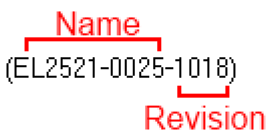


Fig. 101: Example default state

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 [▶ 89] 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:

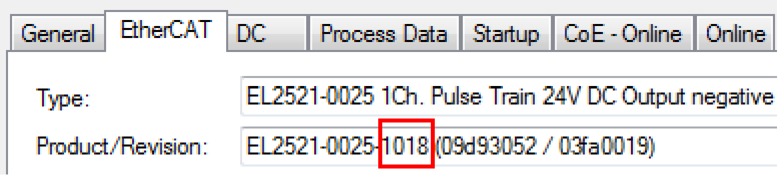


Fig. 102: Installing EthetCAT terminal with revision -1018

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 [▸ 89] 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:

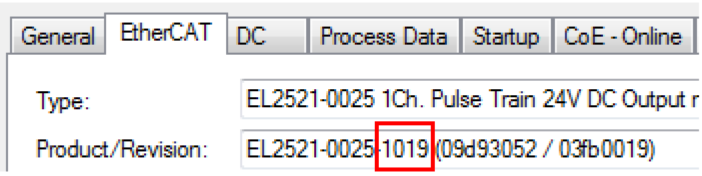


Fig. 103: Detection of EtherCAT terminal with revision -1019

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

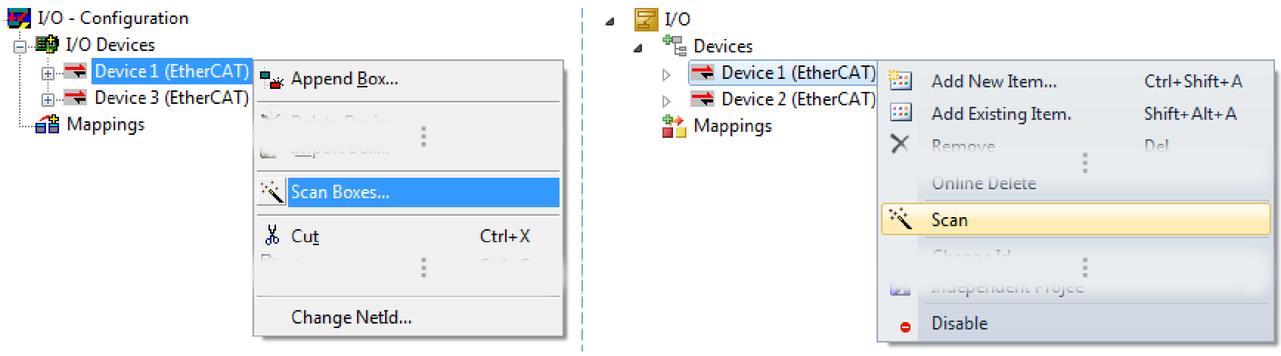


Fig. 105: Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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. 106: *Scan progressexemplary by TwinCAT 2*

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

Fig. 107: 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).

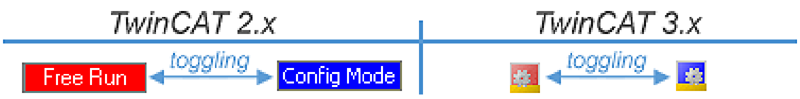


Fig. 108: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar

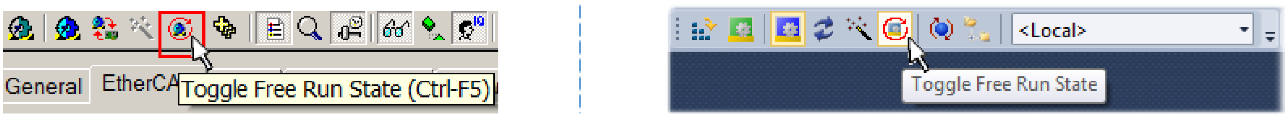


Fig. 109: TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3)

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

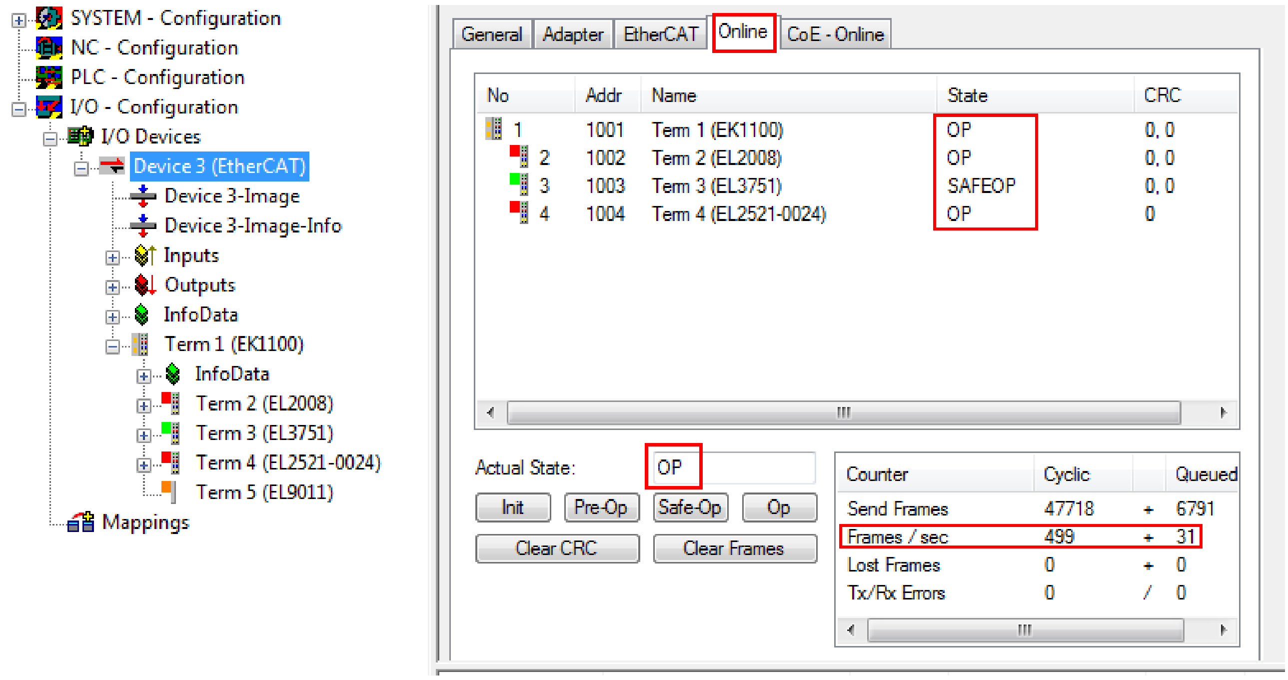


Fig. 110: Online display example

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 [▶ 79].

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.

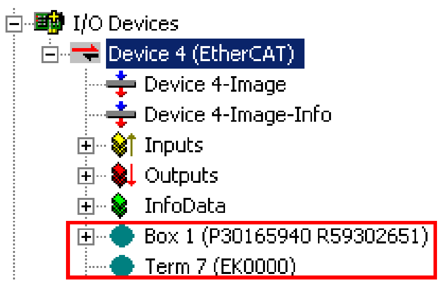


Fig. 111: Faulty identification

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. 112: 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.

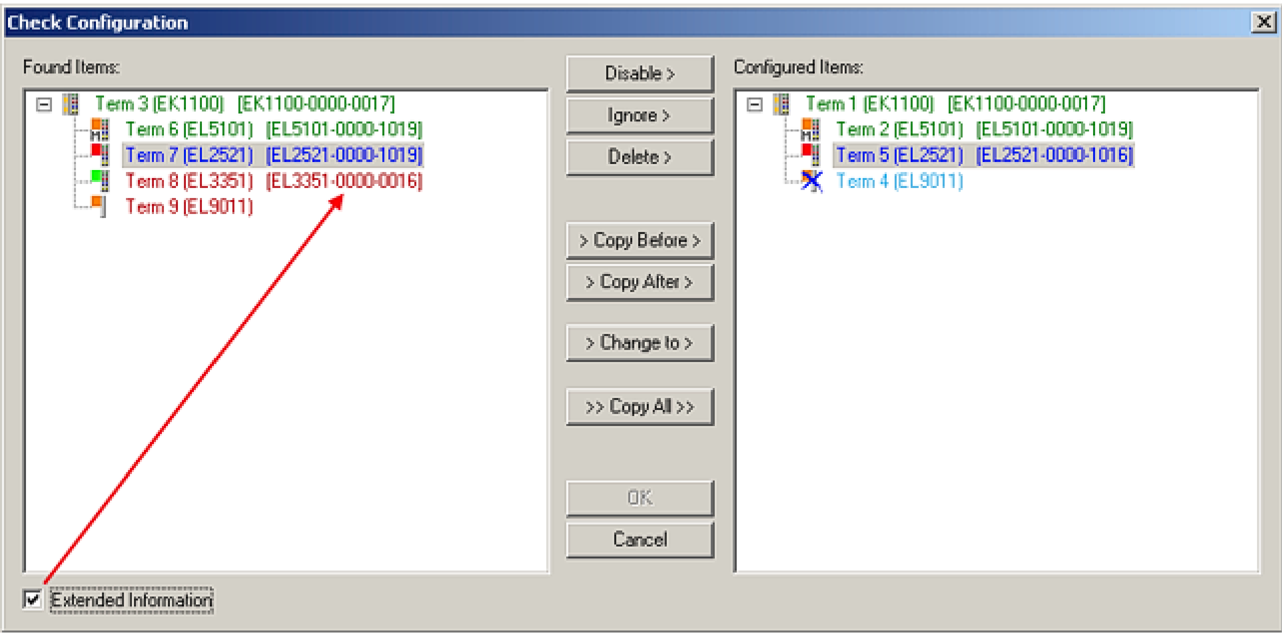


Fig. 113: Correction dialog

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.

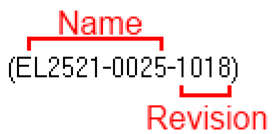


Fig. 114: Name/revision of the terminal

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, ...

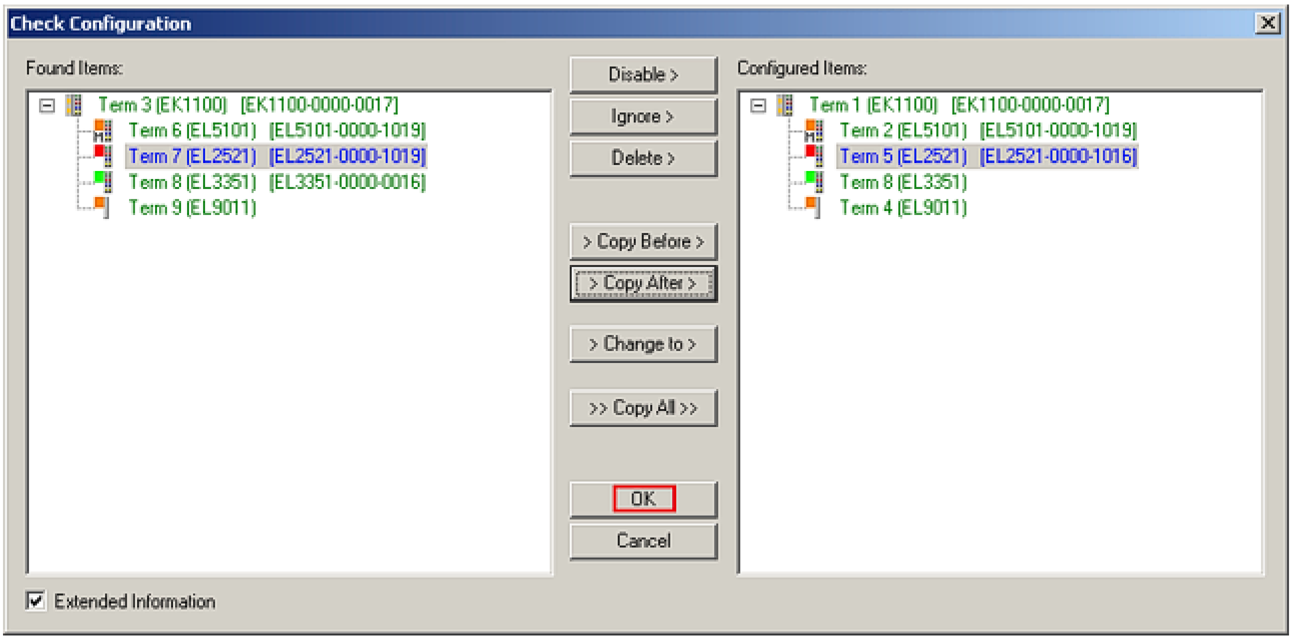


Fig. 115: Correction dialog with modifications

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.

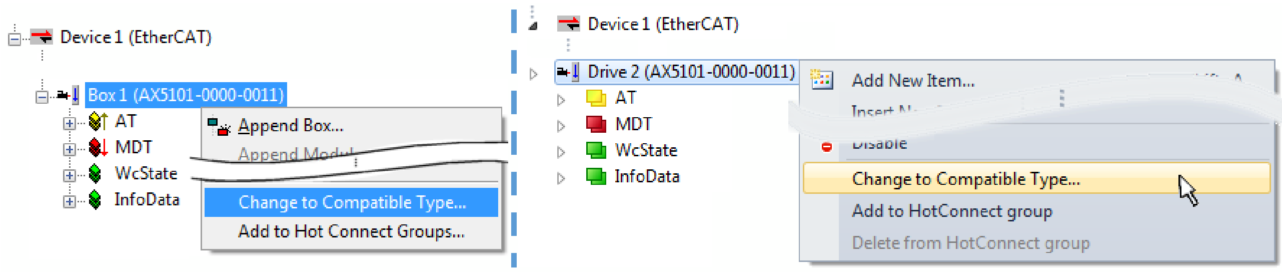


Fig. 116: Dialog “Change to Compatible Type…” (left: TwinCAT 2; right: TwinCAT 3)

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

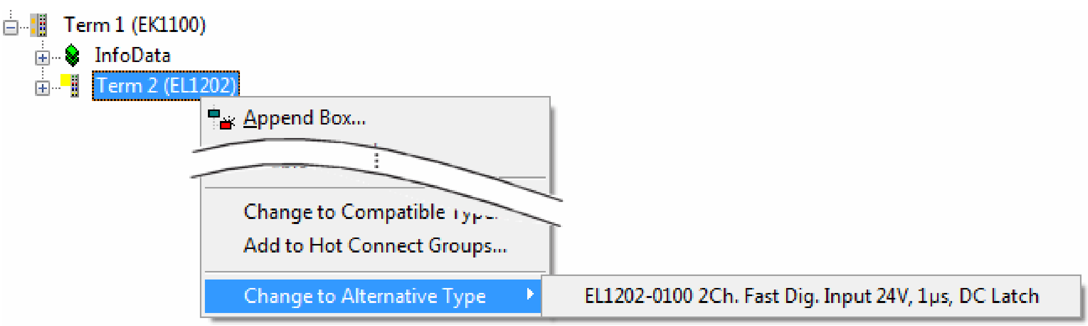


Fig. 117: TwinCAT 2 Dialog Change to Alternative Type

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).

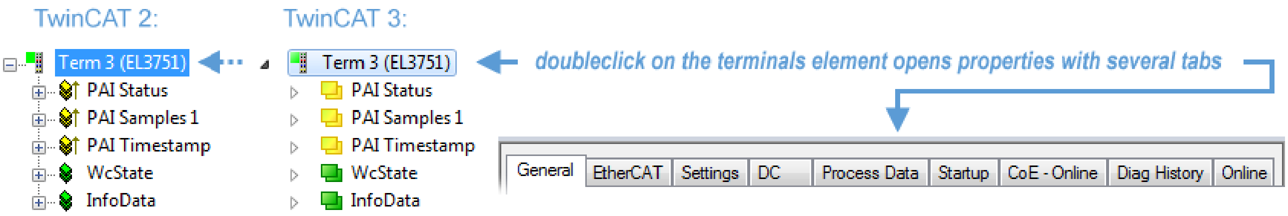


Fig. 118: Branch element as terminal EL3751

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

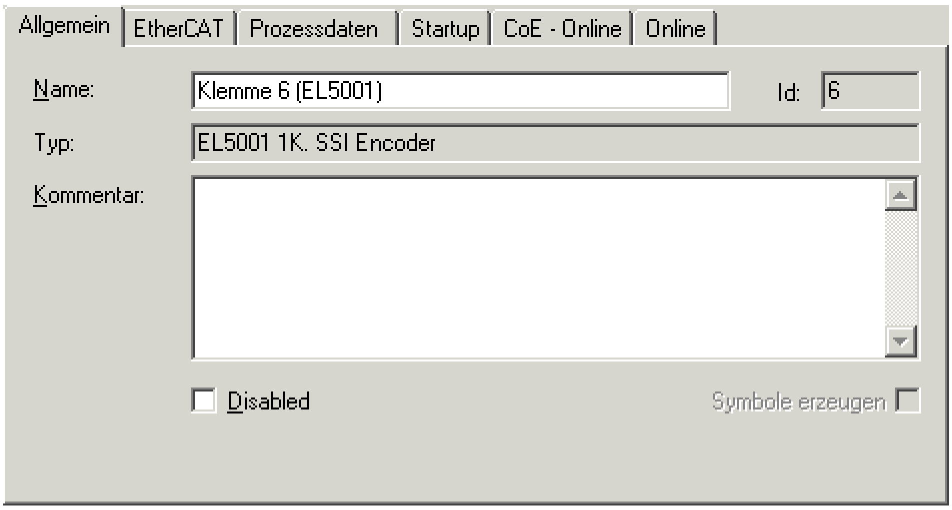


Fig. 119: “General” tab

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

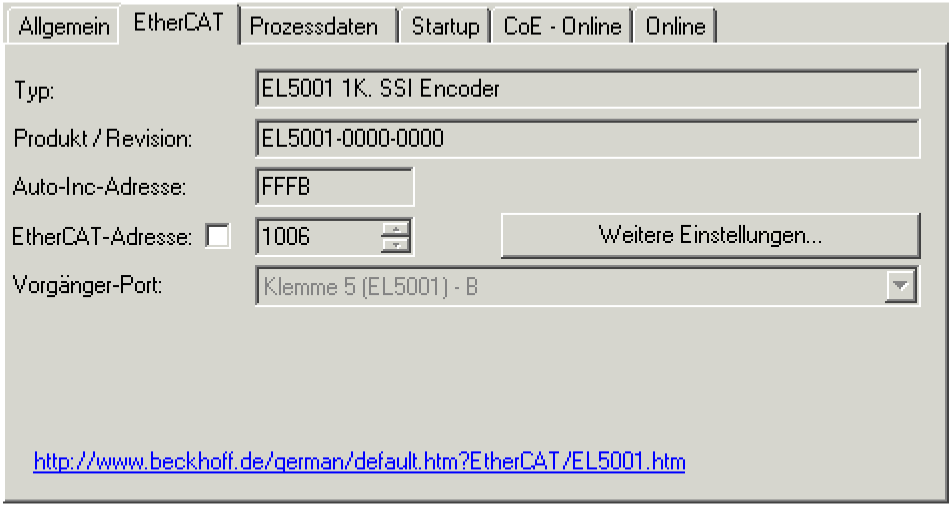


Fig. 120: "EtherCAT" tab

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.

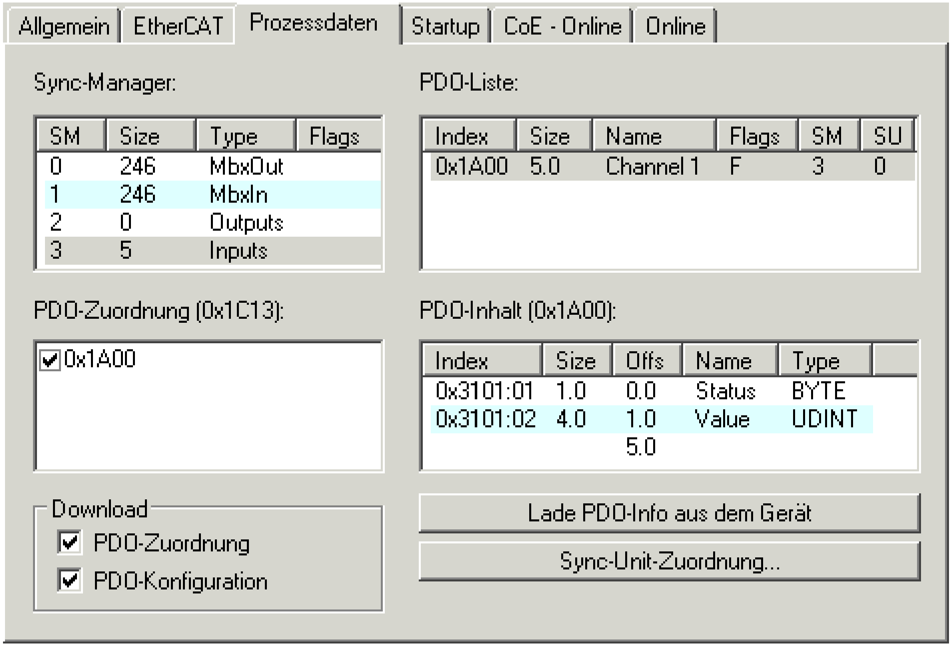


Fig. 121: “Process Data” tab

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”).

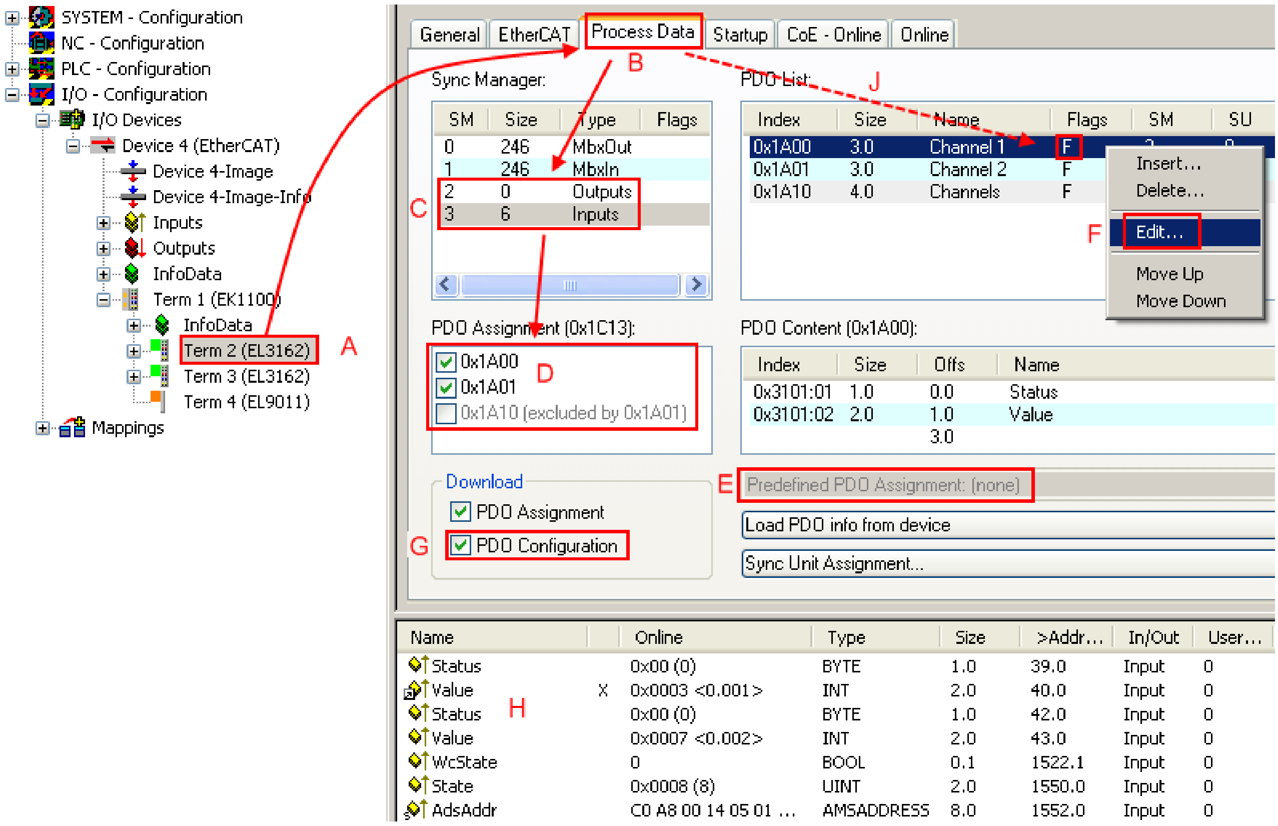


Fig. 122: Configuring the process data

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 [▶ 100] 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.

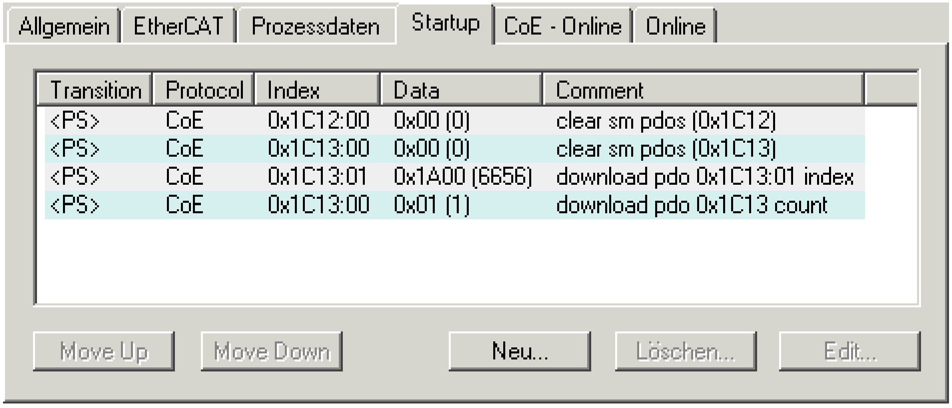


Fig. 123: “Startup” tab

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.

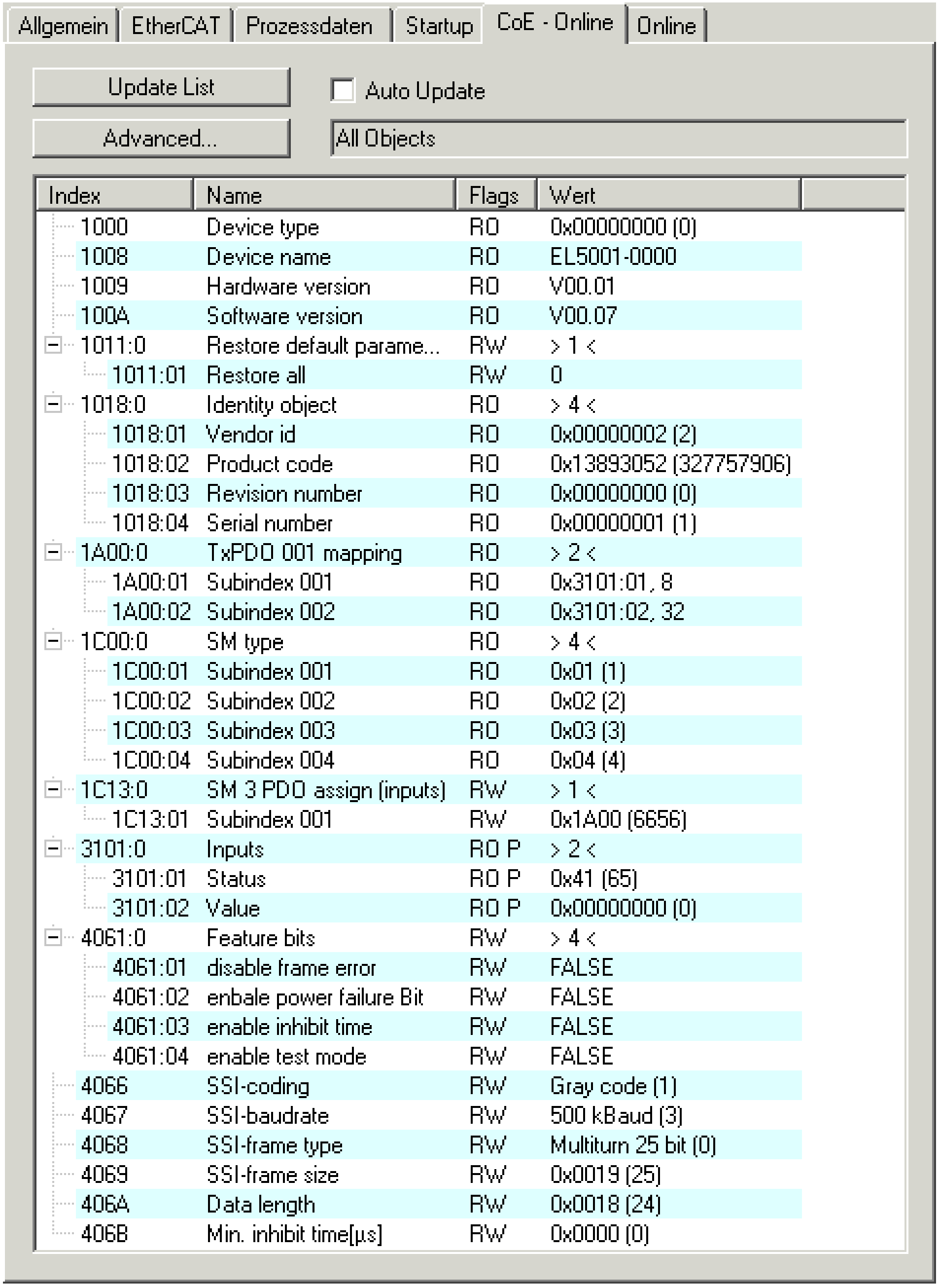


Fig. 124: “CoE - Online” tab

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.

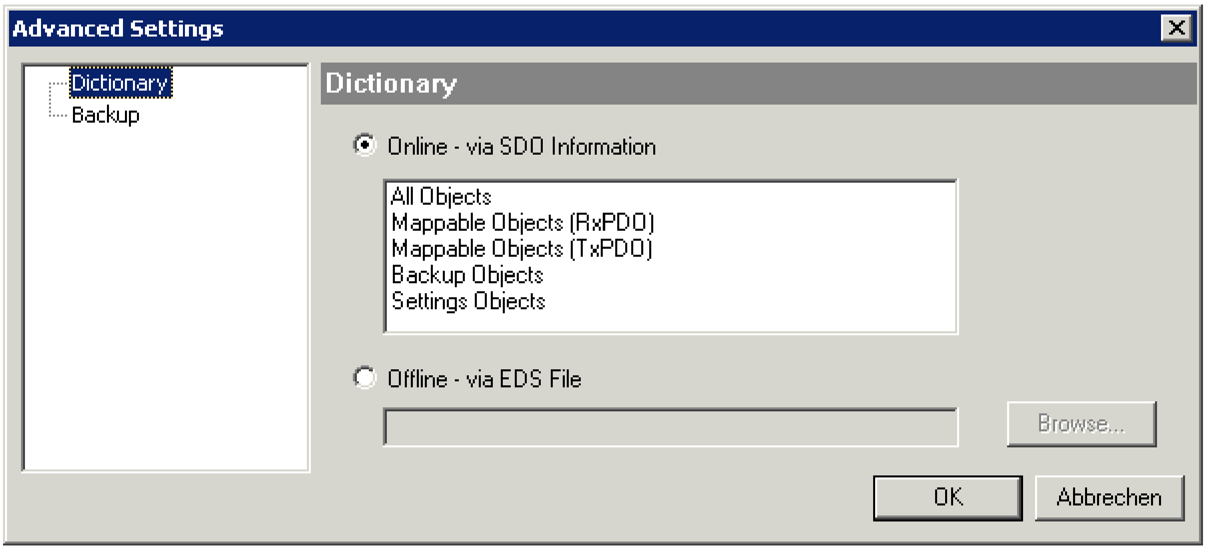


Fig. 125: Dialog “Advanced settings”

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

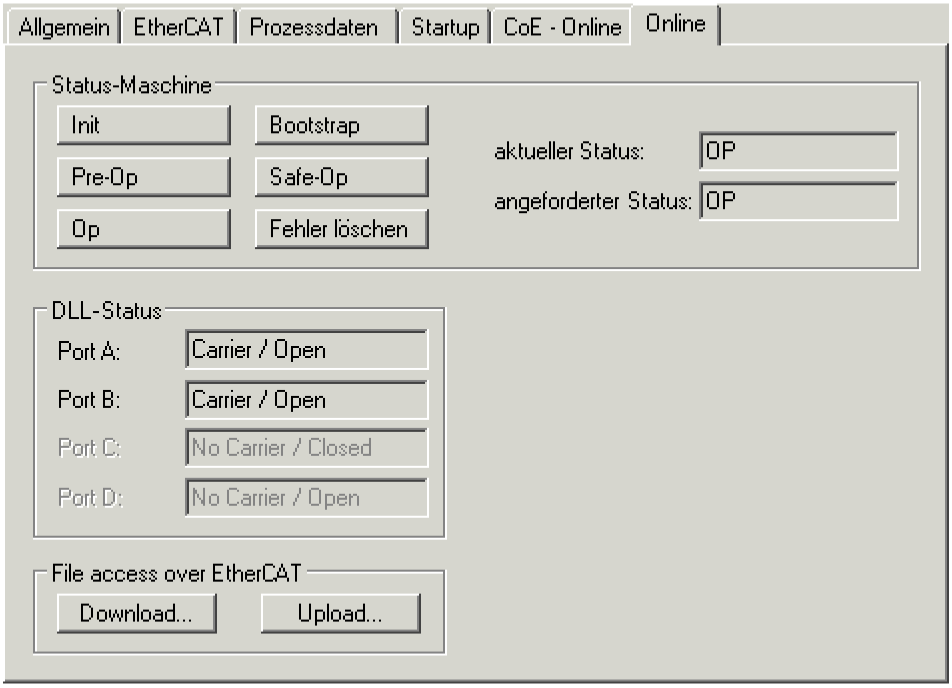


Fig. 126: “Online” tab

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)

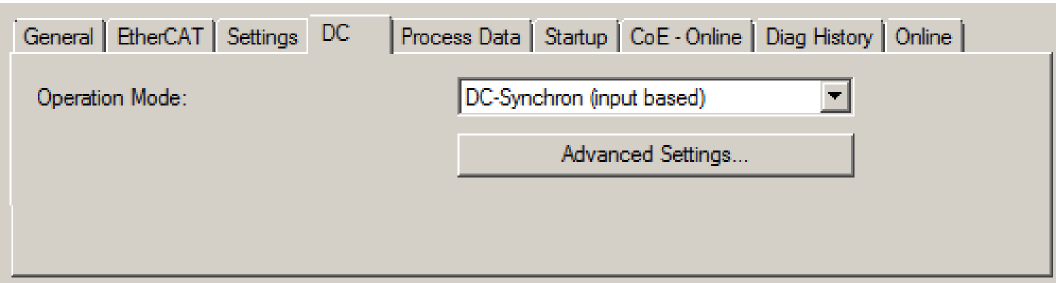


Fig. 127: "DC" tab (Distributed Clocks)

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 [▸ 98]),

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 [▸ 95] 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.

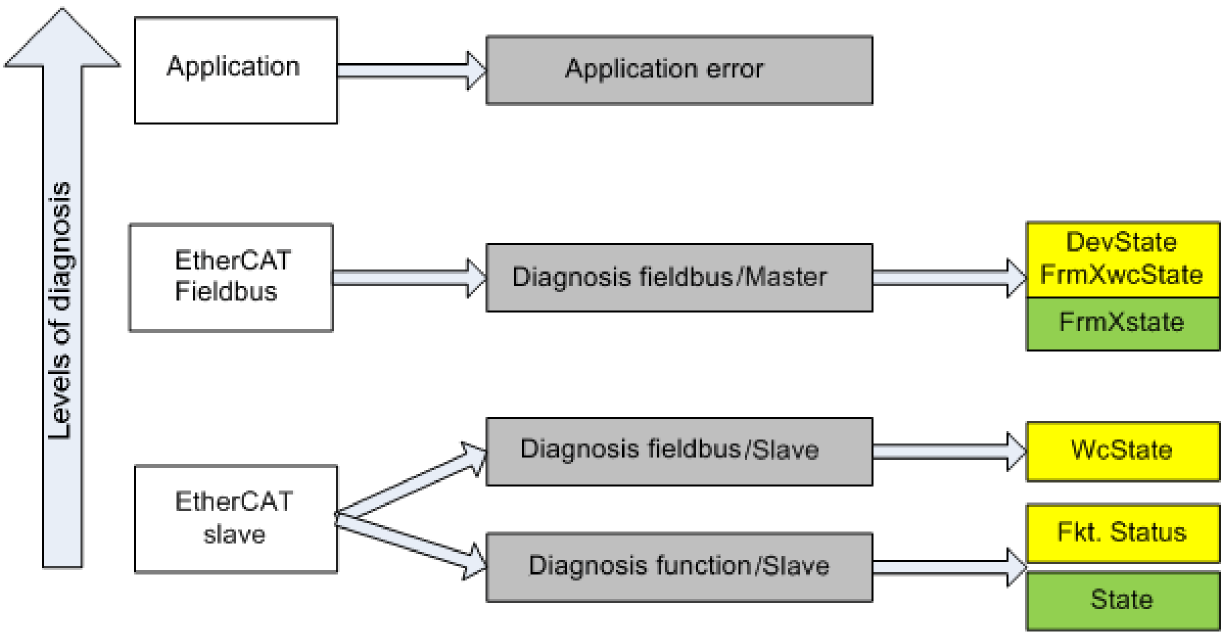


Fig. 128: Selection of the diagnostic information of an EtherCAT Slave

In general, an EtherCAT Slave offers

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

  as well as

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

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

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

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

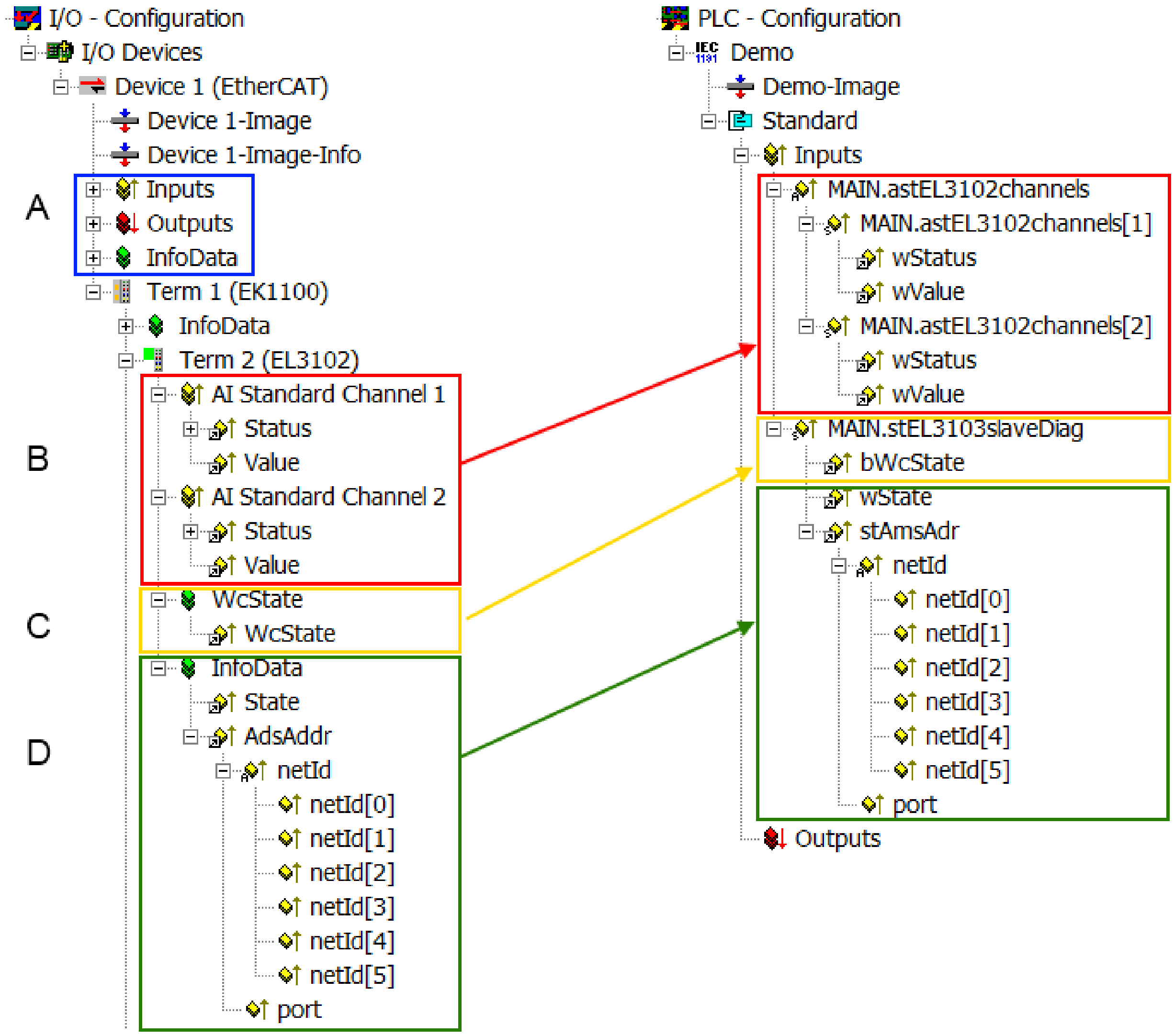


Fig. 129: Basic EtherCAT Slave Diagnosis in the PLC

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*:

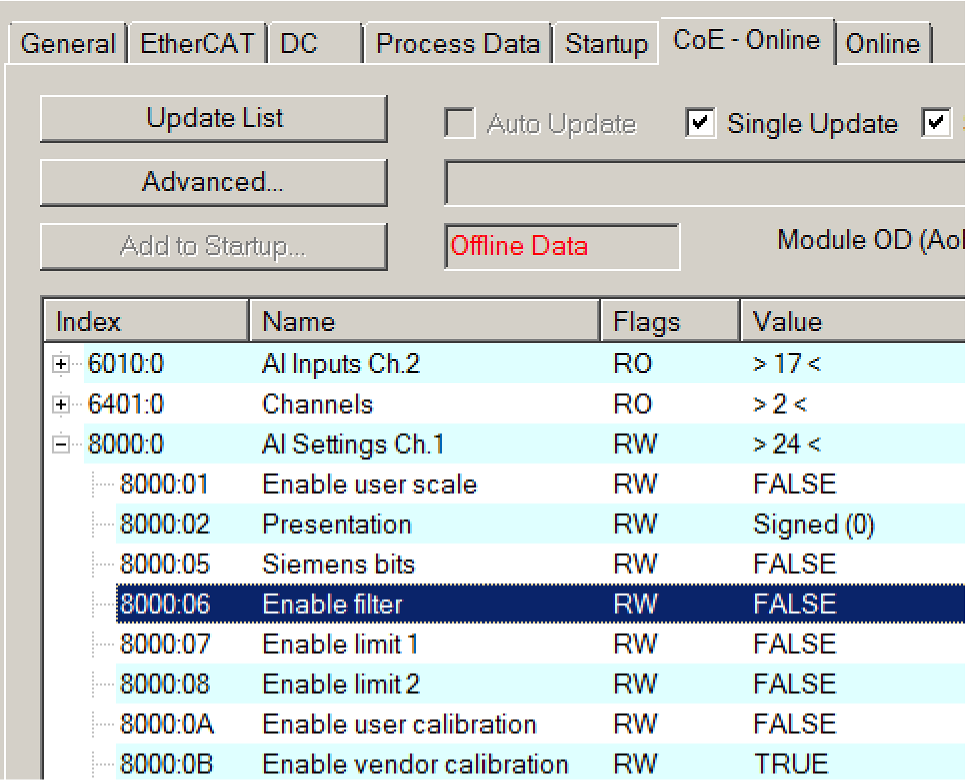


Fig. 130: EL3102, CoE directory

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.

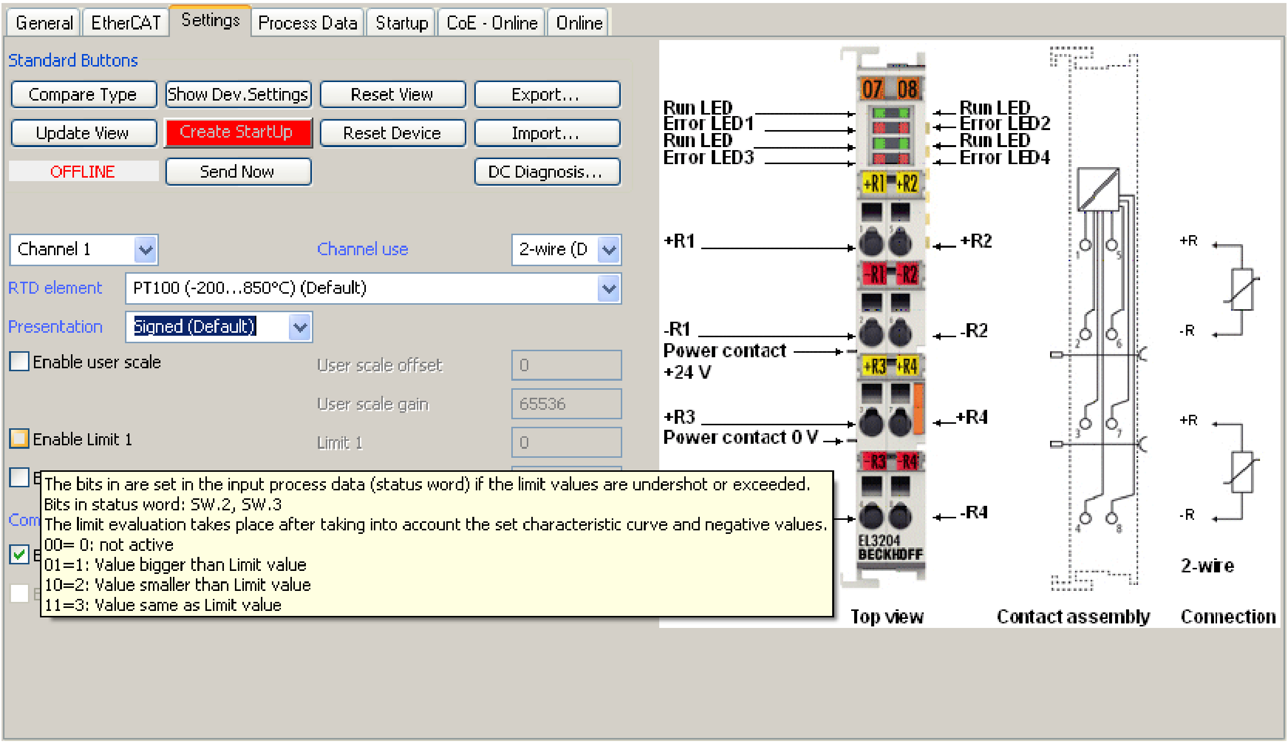


Fig. 131: Example of commissioning aid for a EL3204

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 [▶ 22]" 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.

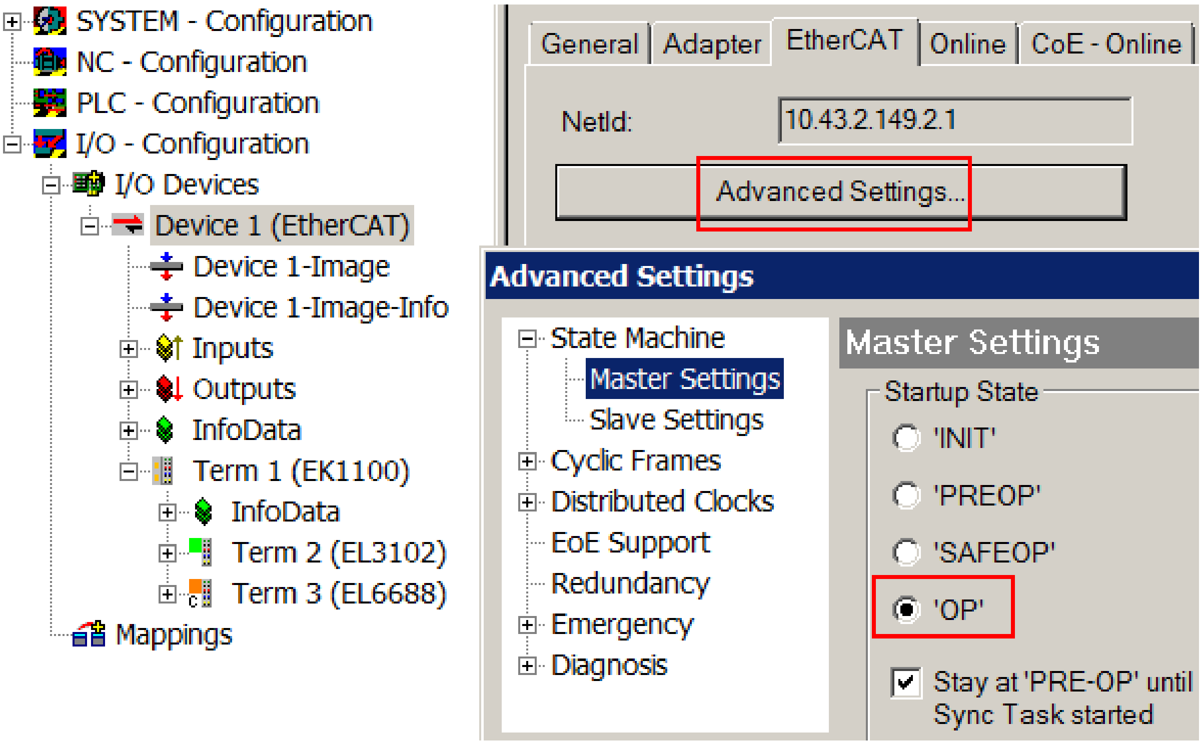


Fig. 132: Default behaviour of the System Manager

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

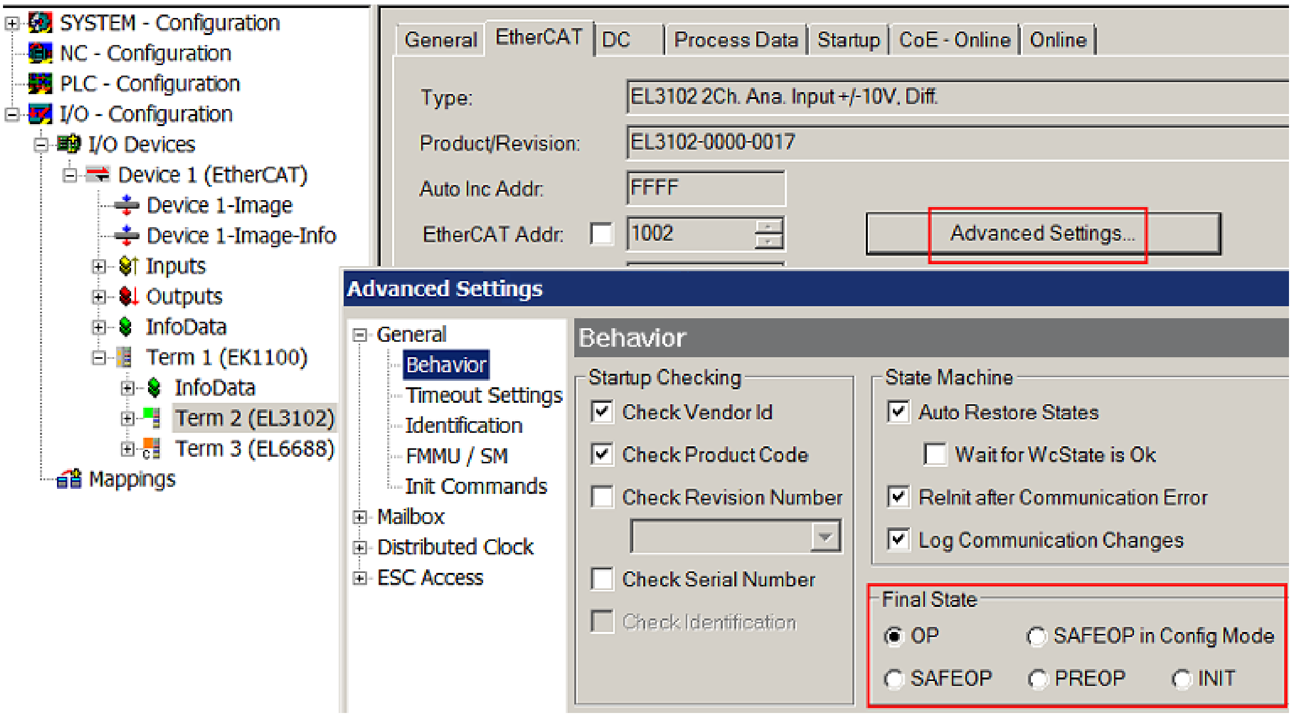


Fig. 133: Default target state in the Slave

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.

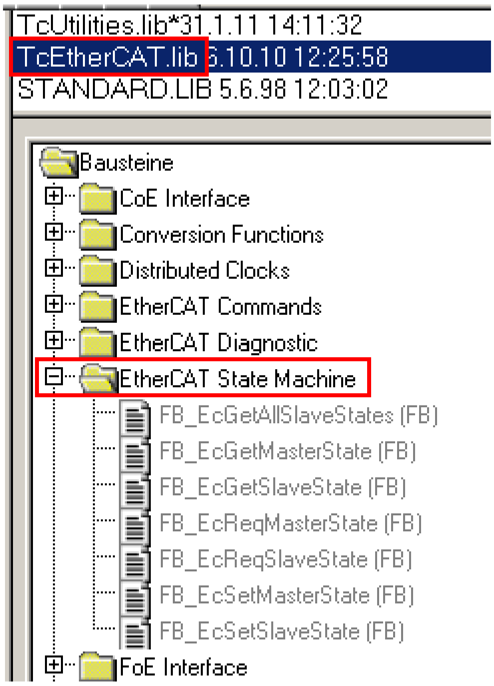
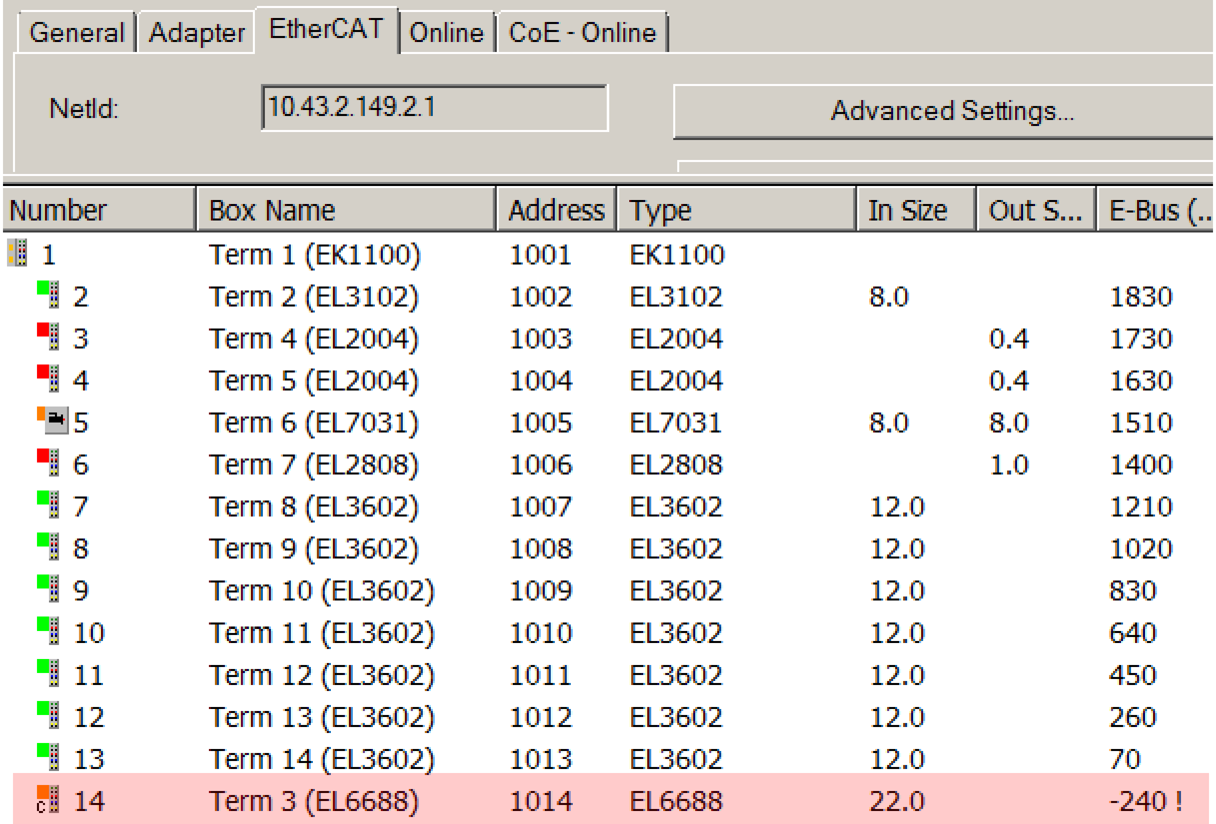


Fig. 134: PLC function blocks

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. 135: 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:

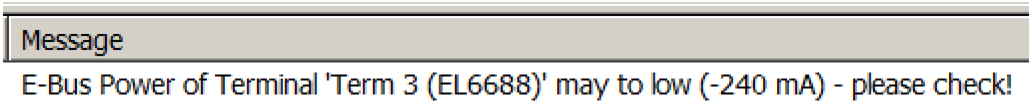


Fig. 136: Warning message for exceeding E-Bus current

NOTE

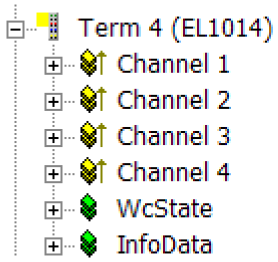
Caution! Malfunction possible!

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

5.4 Configuration with the TwinCAT System Manager - digital input and output terminals

(with TwinCAT from version 2.10.0 (build 1305), example based on EL1014)

Click in the left-hand window of the TwinCAT System Manager on the tree branch of the terminal that you wish to configure (in the example: terminal 4 EL1014).



In the right-hand window of the TwinCAT System Manager, various tabs are now available for configuring the terminal.

General tab

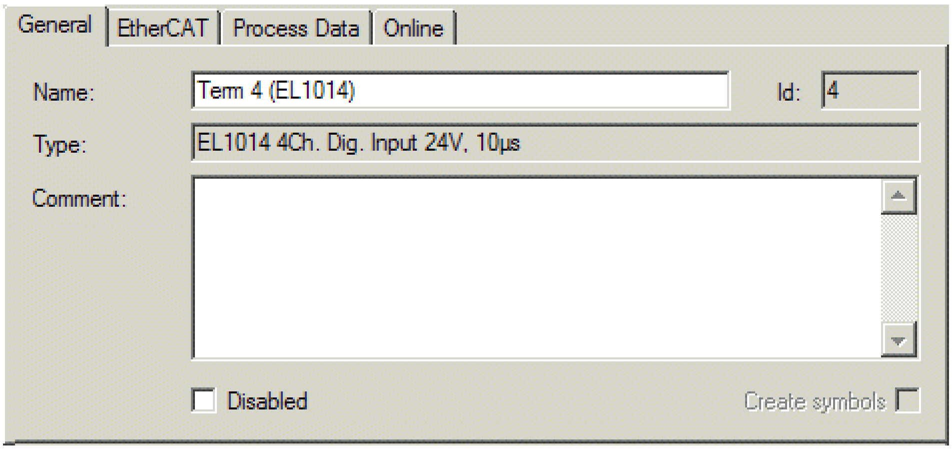


Fig. 137: General tab

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

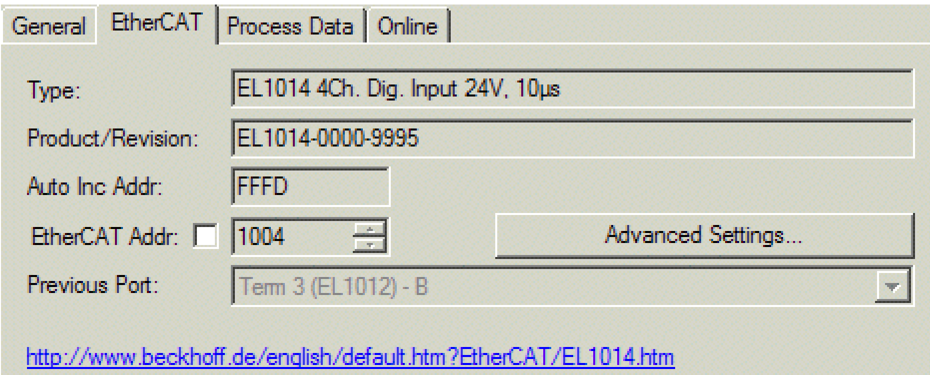


Fig. 138: EtherCAT tab

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 (PDO). Digital input terminals (EL10xx) and digital output terminals (EL20xx) feature a fixed PDO allocation.

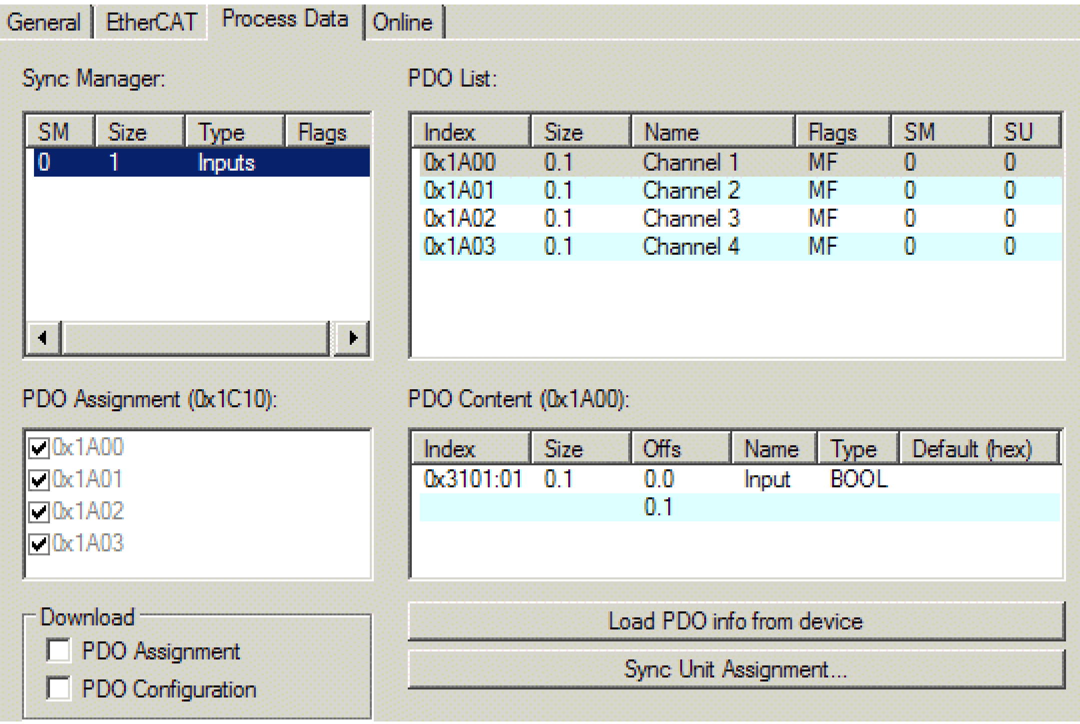


Fig. 139: Process Data tab

Sync Manager

Lists the configuration of the Sync Manager (SM).
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.

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.

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.

Download: PDO assignment, PDO configuration

The download function for the PDO Assignment and configuration is only supported by intelligent terminals.

Online tab

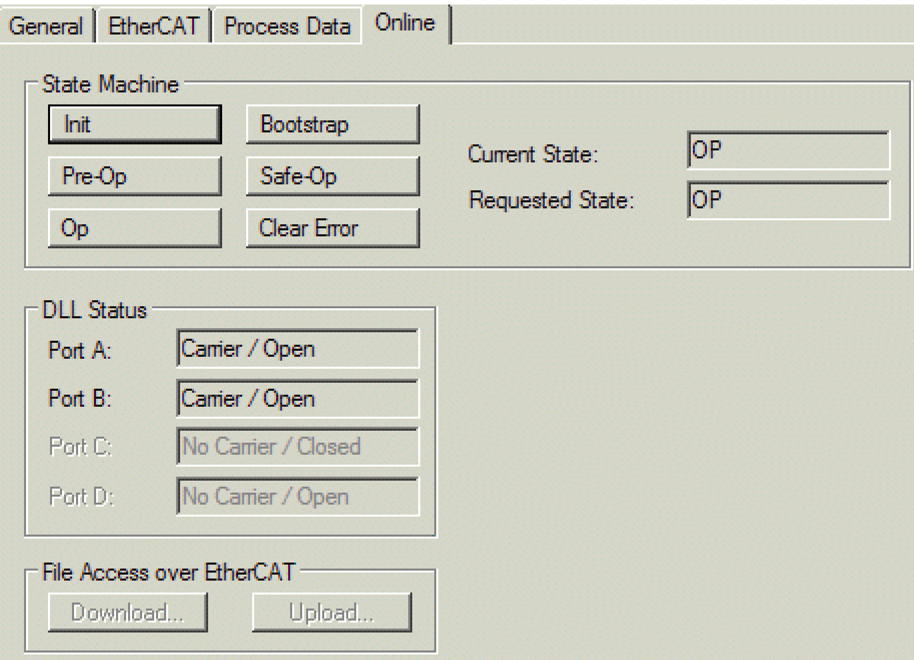


Fig. 140: Online tab

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.

Table 1: File Access over EtherCAT

Download	This is inactive for non-intelligent terminals.
Upload	This is inactive for non-intelligent terminals.

5.5 Switching characteristics

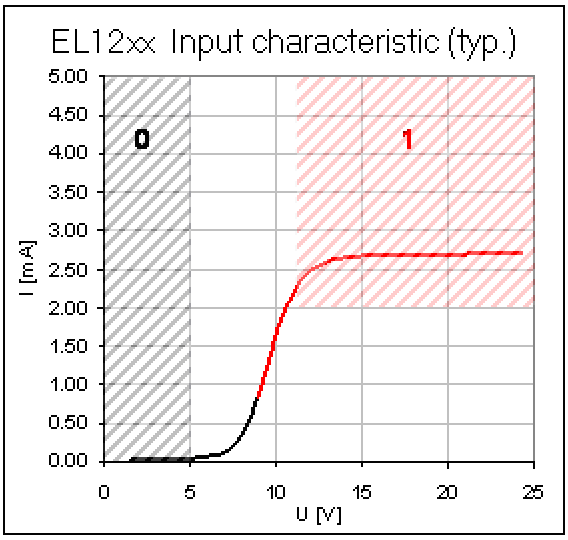


Fig. 141: Input curve on basis of EN 61131-2, type 3; typical measurement
(Beckhoff retains the right to make changes without warning!)

The sensor/signal generator must be able to generate a sufficiently steep signal edge. The power supply unit used should have sufficient reserves/buffers so that the signal reaches the terminal with sufficient edge steepness despite capacitive/inductive supply cable losses.

5.6 Sensitivity of the input

The input circuit of the EL12xx is optimized for fast signal changes and the shortest possible signal detection. The time required for a signal change as a rising/falling edge from the terminal point at the front of the terminal to the logic of the central processing unit (ESC) is specified for the EL12xx series at $T_{ON}/T_{OFF} < 1$ µs, both for rising (T_{ON}) and falling edge (T_{OFF}). Due to this low absolute cycle time, the temperature drift of the cycle time is also very low.

It should be considered that the input circuitry has little or no filtering, depending on the type. It is optimized for fastest signal transmission from the input to the evaluation unit. Fast level changes/pulses in the µs range, e.g. due to possible EMC influences, thus arrive at the evaluation unit unfiltered/undamped and are possibly visible as a change of state of the input.
If necessary, shielded cables should be used to exclude environmental influences.

5.7 EL1202

5.7.1 Delivery state

It is not necessary to make any particular settings when operation of the EL1202 is first started.

5.7.2 Functioning

The EL1202 series currently offers two different operating modes:

EL1202-0000: fast 2-channel 24 V input terminal – the reading of the inputs is triggered by the communication frame (delivery state). No other settings can be made at the terminal.

EL1202-0100: fast, 2-channel, 24 V input terminal - reading the inputs is triggered in a highly stable cycle by the local DC (distributed clock), and can be configured.

The EL1202 is configured by Beckhoff ex stock as an EL1202-0000. The user must convert the terminal if operation as an EL1202-0100 is desired. This conversion is reversible.

Instruction for TwinCAT 2

Within TwinCAT 2 the EL1202(-0100) have to be selected in the configuration tree and "*Change to Alternative Type*" be chosen. This changes the EEPROM and thus the functionality of the EL1202 (Fig. *Change to Alternative Type*).

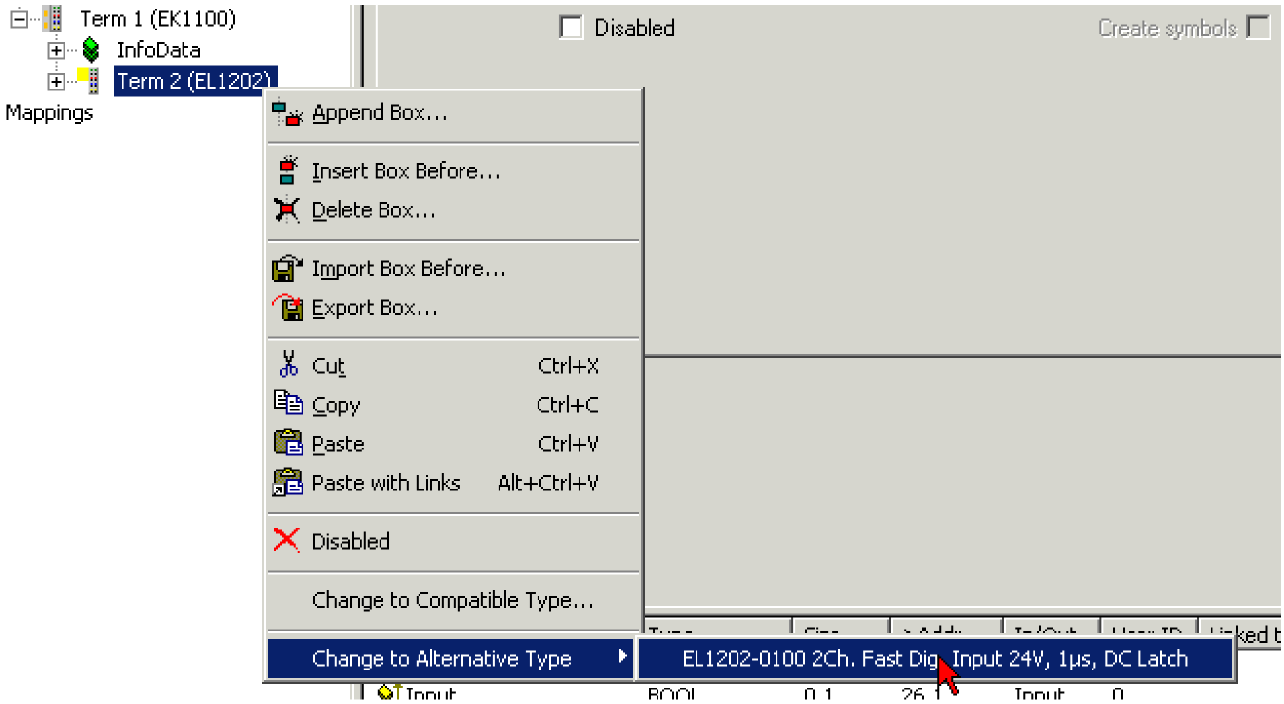


Fig. 142: Change to Alternative Type

Instruction for TwinCAT 3

Within TwinCAT 3 either the terminal in the configuration as the EEPROM of the terminal have to be reconfigured; the procedure as described for TwinCAT 2 is not possible.

- Within the configuration: add EL1202-0100 as new device; delete the EL1202-0000.
- Update EEPROM as described in chapter Firmware Update EL/ES/EM/EPxxxx [▶ 140].

Requirements for the master

It is only possible to use the EL1202-0100 with EtherCAT masters (such as Beckhoff TwinCAT) that support distributed clocks (DC). In order to switch between the operating modes, the failure-safe configuration in the terminal must be changed (as from TwinCAT 2.10 Build 1316 this can be done on the terminal's “EtherCAT” tab).

The absence of an input filter means that electrical pulses are not lengthened. In order to be recognized as TRUE or FALSE, an electrical signal must therefore be present at the evaluation unit at the time when it reads the input into its registers. This time depends on the version of the terminal and on the configuration that it has.

5.7.3 EL1202-0000 - EtherCAT settings

“EtherCAT” tab

In the EL1202, optimization for fast signal transfer, with a propagation time T_{ON}/T_{OFF} < 1 µs, means that the terminal reads the voltage present at the inputs as soon as the EtherCAT communication frame reaches the ESC. The signals that occur between the two communication cycles are ignored. The EL1202 appears as a normal, 2-channel, digital 24 V input terminal without any other special features. DC is switched off, which means that this terminal cannot operate as a DC master clock. No other settings can be made on this terminal.

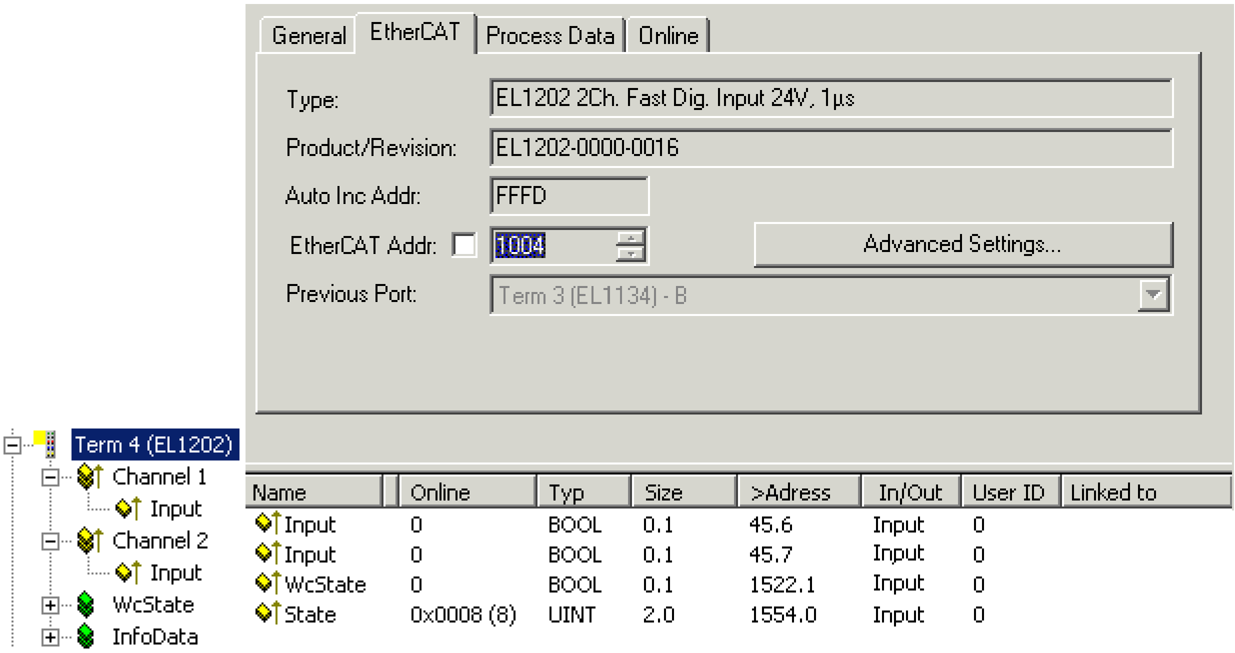


Fig. 143: EtherCAT tab, terminal EL1202-0000

5.7.4 EL1202-0100 - EtherCAT settings

Libraries for 64-bit processing

A selection of functions for handling 64-bit numbers is available under Beckhoff TwinCAT in the TcUtilities.lib library. Longer execution times are required here than is the case with standard, 32-bit data types. A data type with a width of 64-bits is defined in TcEthercat.lib as T_DCTIME or in TcUtilities.lib as T_LARGE_INTEGER.

“EtherCAT” tab

“EtherCAT” tab: the EL1202-0100 terminal appears as a 2-channel digital input terminal with further process data:

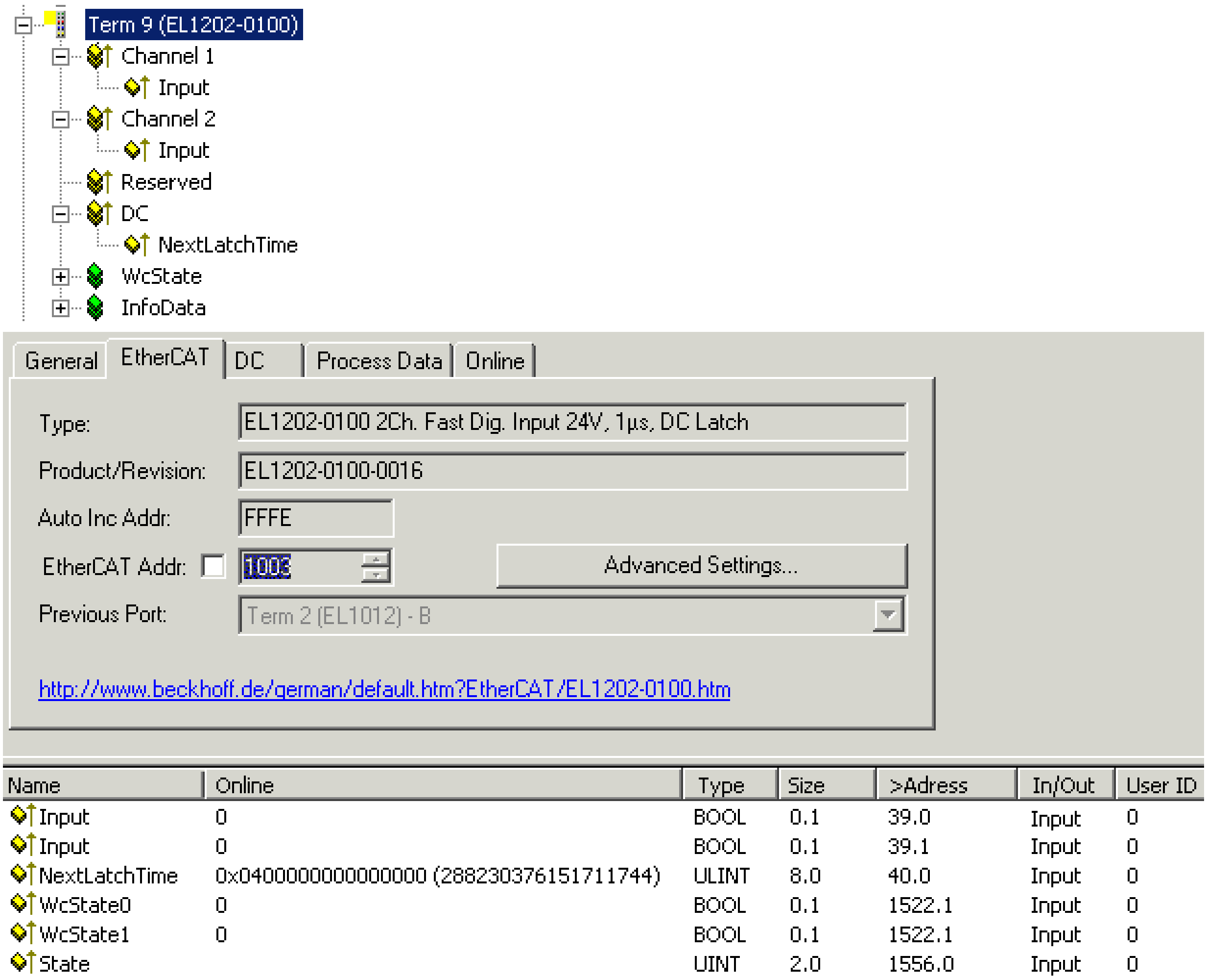


Fig. 144: EtherCAT tab, terminal EL1202-0100

DC is normally switched on, and the EL1202-0100 can therefore be used as a DC master clock.

“Input Channel 1” and “Input Channel 2”

The state of the inputs at the time of the last query is given here (0/1). The two inputs can only be latched at the same time.

“Reserved”

Reserved for future use.

“NextLatchTime”

Gives the next time when the two inputs will be latched. The status of “Input Channel0” and “Input Channel1” given in the process data image is therefore determined at the previously specified “NextLatchTime”.

Example:

- The process data of an EL1202-0100 in an EtherCAT cycle is:
  “Input Channel1” = 1
- “Input Channel2” = 0
- “NextLatchTime” = 288.230.376.151.711.744 (ns)

  With a SYNC0 cycle time of 1 ms (= 1,000,000 ns) the quoted status of “Input Channel 1” and “Input Channel 2” is determined for time 237,628,613,985,878,208 - 1,000,000 = 237,628,613,984,878,208 (64-bit processing).

5.7.5 Configuration of the process data

“Process data” tab

Sync Manager 0 (SM0), PDO assignment 0x1C10

“Channel 1” (0x1A00), “Channel 2” (0x1A01) and “Reserved” (0x1A02) can not be changed

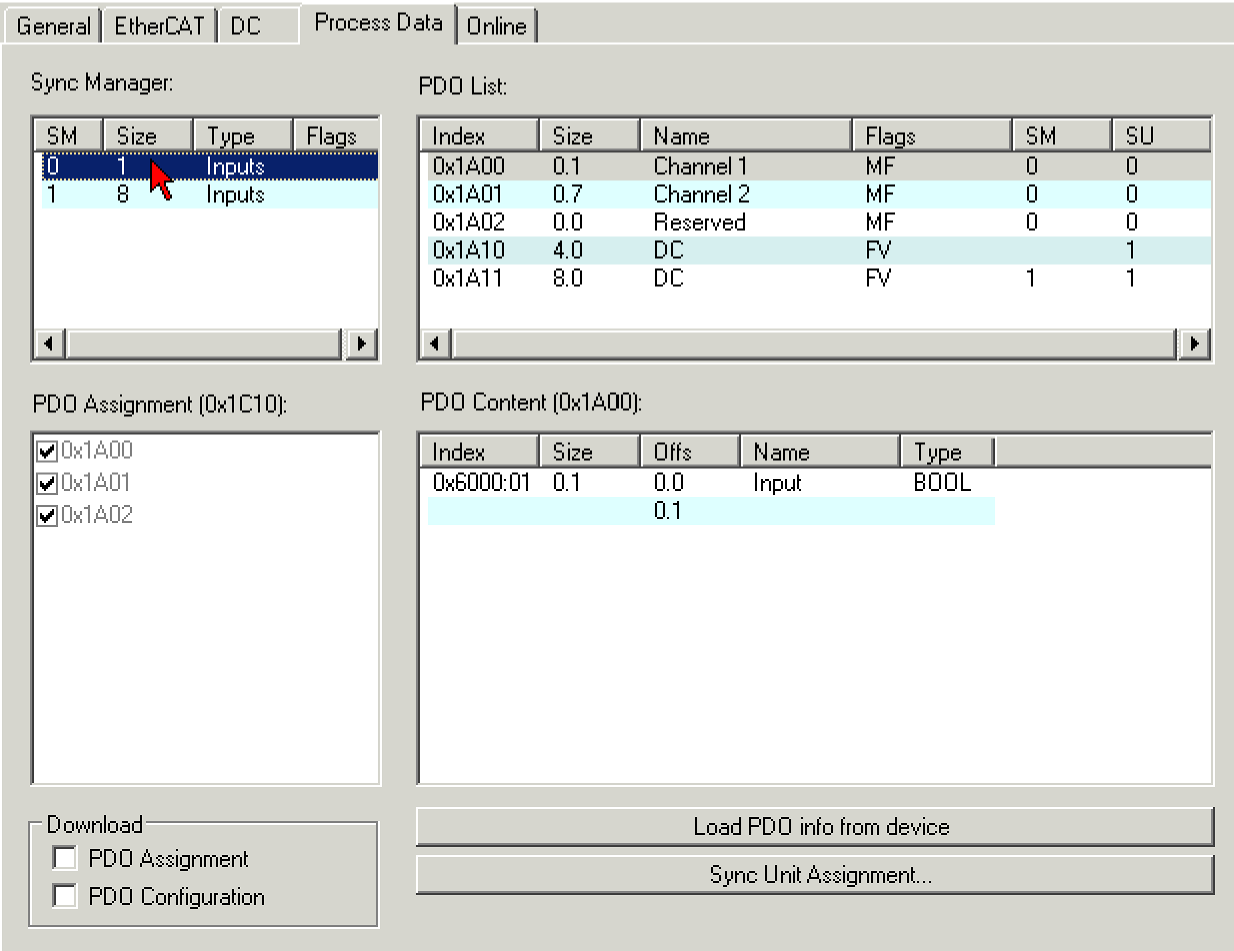


Fig. 145: Process data tab, PDO input assignment, terminal EL1202-0100

Sync Manager 1 (SM1), PDO assignment 0x1C11

The specification of the “NextLatchTime” can be changed from 64-bit representation (PDO 0x1A11) to 32-bit representation (PDO 0x1A10). PDO 0x1A11 must first be deactivated, as the two PDOs are mutually exclusive.

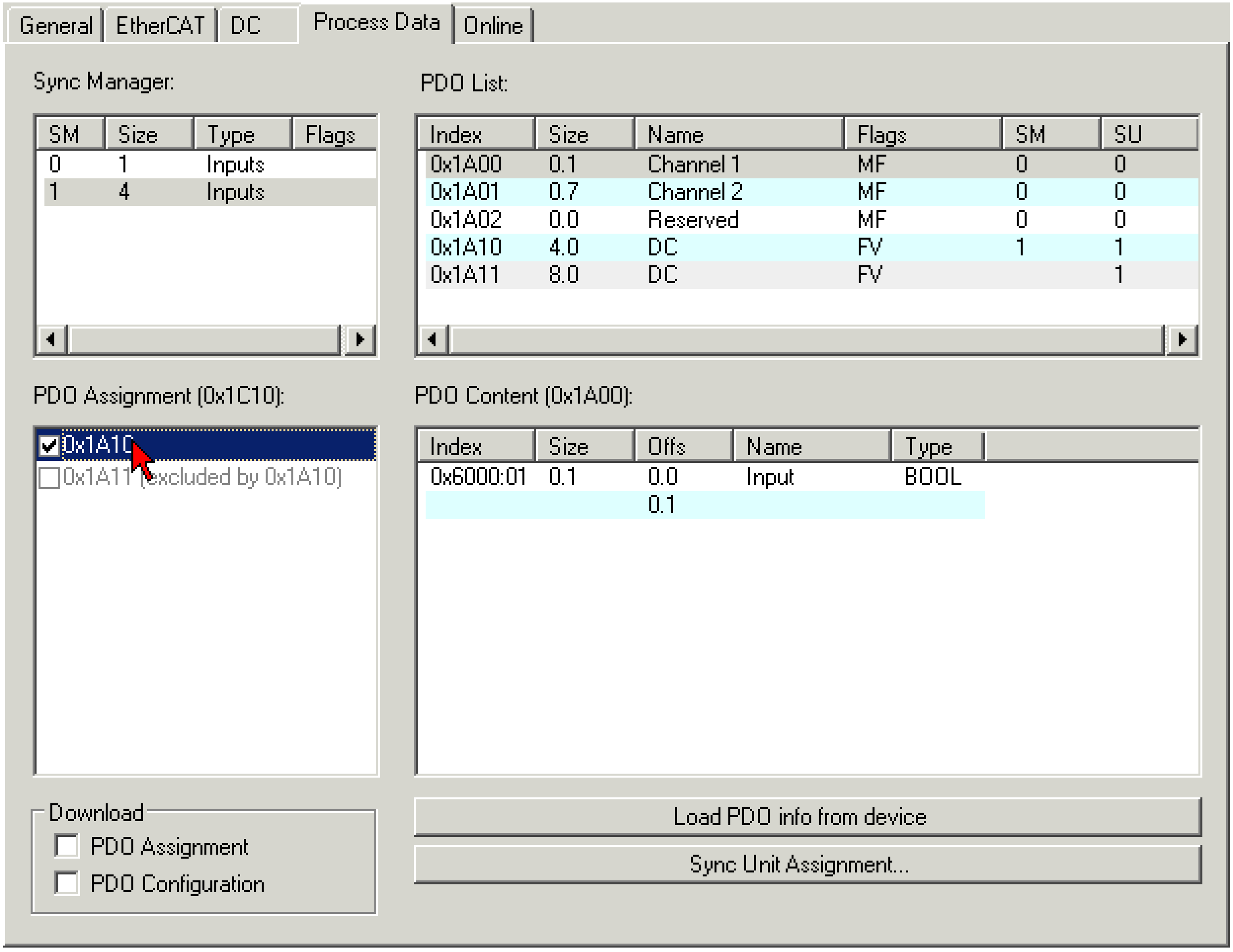


Fig. 146: Process data tab, changing over the “NextLatchTime”, terminal EL1202-0100

5.7.6 Distributed Clock settings

“DC” tab

The DC/Advanced Settings tab can modify the behavior of the EL1202-0100 in association with the local DC.

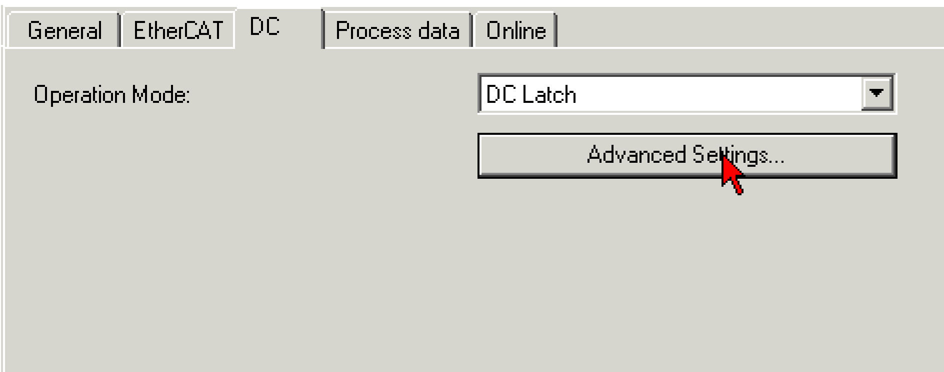


Fig. 147: DC tab, terminal EL1202-0100

General

Interrupt SYNC0 or SYNC1 is triggered when the time in the DC matches the specified value for the relevant SYNC (“NextSync0Puls” or “NextSync1Puls”). “NextSync0Puls” is also made available as process data, the “NextLatchTime”. The EtherCAT slave controller (ESC) in the terminal has the job of handling these DC operations.

Default setting

The cyclic read of the inputs is triggered by the SYNC0 pulse (interrupt) from the DC in the EL1202-0100. The standard arrangement is for the reading cycle time of the EtherCAT master to be set to the PLC cycle time being used, and therefore to the EtherCAT cycle time.
(Fig. Advanced Distributed Clock (DC) settings, EL1202-0100 terminal [▸ 120]: 4000µs = 4 ms, as TwinCAT is in configuration mode.

Use in the EL1202-0100

On starting the EtherCAT system, the first specified value is set by the EtherCAT master to “Current DC time” + SYNC0 Cycle Time + SYNC0 Shift Time (see Fig. Advanced Distributed Clock (DC) settings, EL1202-0100 terminal [▸ 120]). The EtherCAT slave controller now continues to check its local DC clock for agreement with the specified value. If the specified value matches the DC time, SYNC0 is triggered, which in turn causes reading of the inputs to begin immediately. The specified value is then counted on: “new specified value” = “old specified value” + SYNC0 cycle time. This process now continues from here. The EtherCAT master continuously adjusts all the DC clocks in the EtherCAT system. Further information will be found under www.beckhoff.de

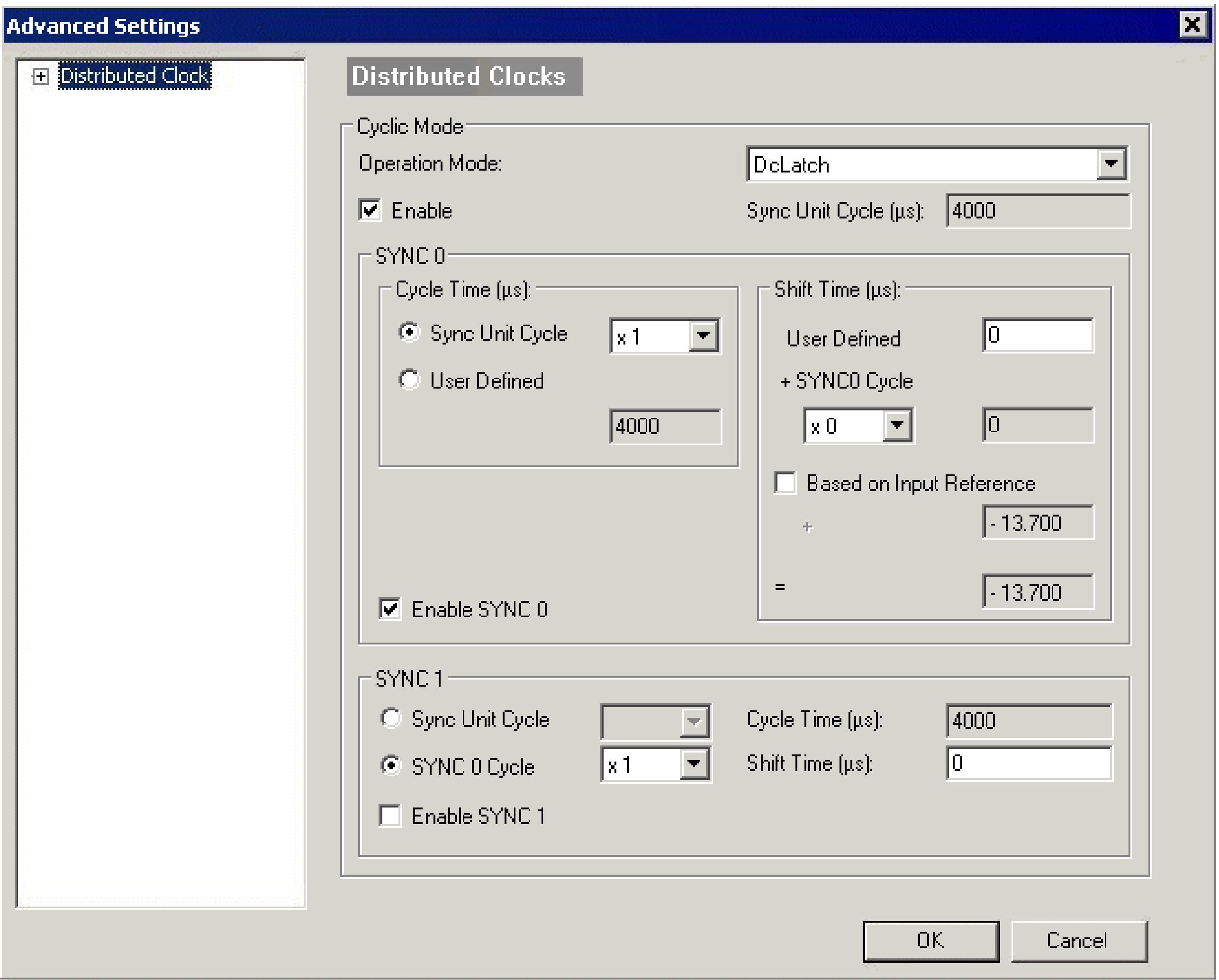


Fig. 148: Additional settings for the Distributed Clock (DC), terminal EL1202-0100

Distributed Clock settings

- **Cyclic mode/enable**
  Switches the DC on.
- **Operating mode**
  Only DC latch is possible.

- **SYNC0**
  Sync unit cycle: a multiple of the bus cycle time. The inputs are read periodically 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 displace the SYNC0 pulse of this EL1202-0100 in relation to other terminals in nanosecond steps. If the inputs of multiple EL1202 terminals are to be read at the same time, the same value must be entered here.
- **Based on input reference**
  If this option is activated, then an additional "input shift" is added to the SYNC0 shift configured locally for the terminal. 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, Advanced Settings, Distributed Clock [▶ 121]). As a result, *all* the input terminals in the system (EL1xxx, EL3xxx) 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.
- **Enable SYNC0**
  Required by the EL1202-0100.
- **SYNC1**
  Additional SYNC pulse, derived from SYNC0 or from the DC itself. Not required by the EL1202-0100.

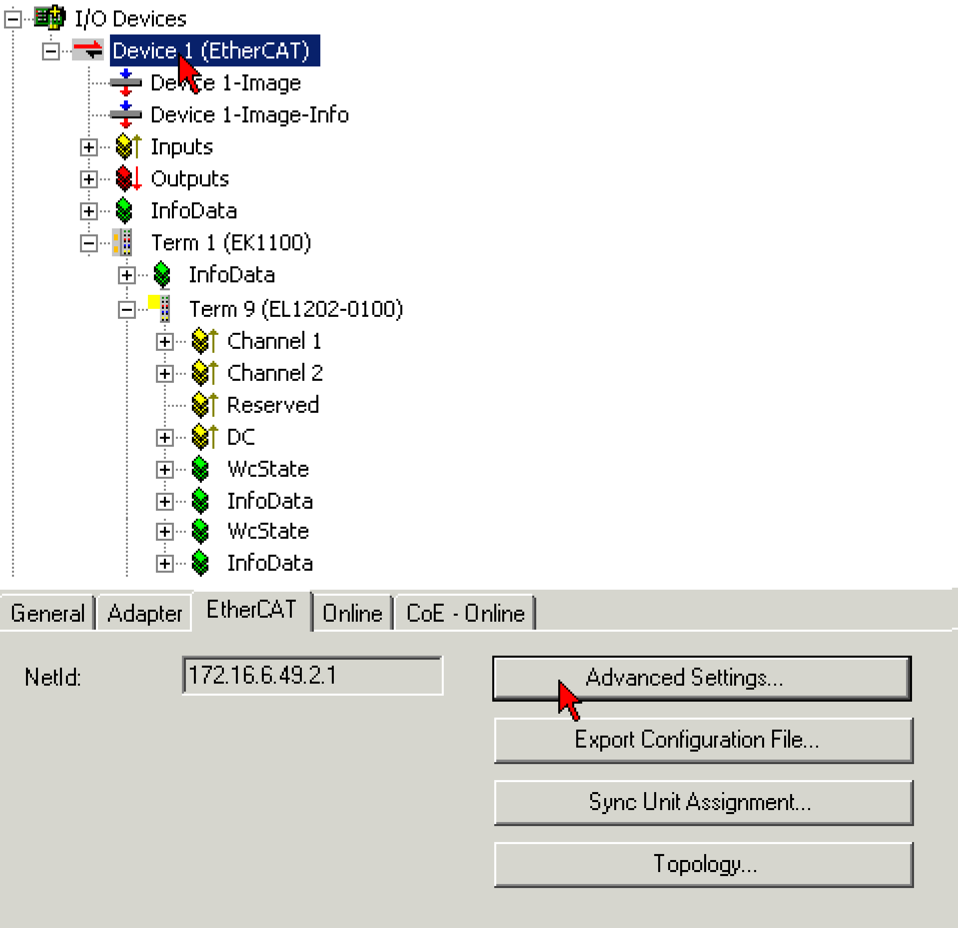


Fig. 149: EtherCAT Master, EtherCAT tab, Advanced Settings

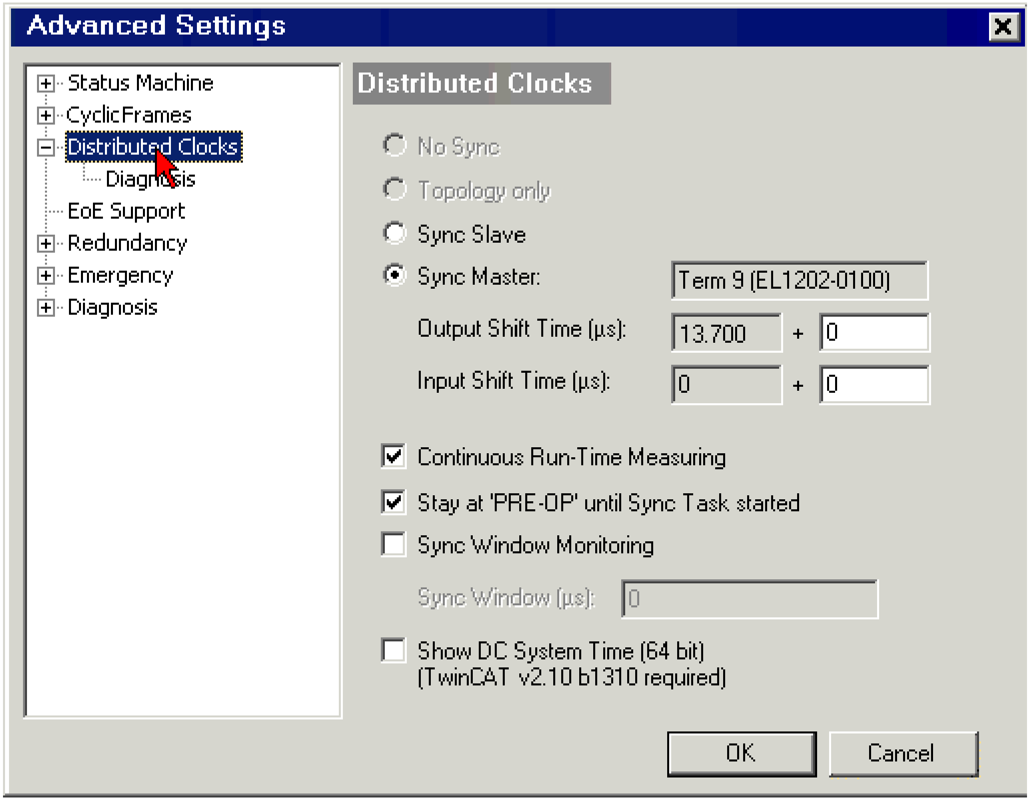


Fig. 150: EtherCAT Master, Advanced Settings, Distributed Clock

5.8 EL1252, EL1254

5.8.1 EL1252, EL1254 - Delivery state

It is not necessary to make any particular settings when operation of the EL1252 or EL1254 is first started.

TwinCAT version note

The terminals can be used without restriction under Beckhoff TwinCAT versions from build 1315 up. The online description of the terminal from EL1252-0000-0017 is to be used exclusively under earlier TwinCAT 2.10 builds. As an alternative, you can download the latest suitable XML file from the download area of the Beckhoff web site (at https://www.beckhoff.de/english/download/default.htm?id=69071317099797) and install it in accordance with the installation instructions.

5.8.2 EL1252, EL1254 - Functioning

General features

The EL1252, EL1254 provides not only the input level, but also the last time stamp at which a channel underwent a rising or falling edge. The status byte for each channel provides information about changes in the latch processes.

Calibration

The time delay between the physically real edge of the signal voltage at the terminal input and the time stamp in the EtherCAT slave controller (ESC) is kept below 1 µs by the optimized electronics, but is not indefinitely short. If the user has a particular need for precision, calibration must be carried out, and the precise time delay measured in accordance with the environmental conditions.

The terminal provides the following information for each channel (see Fig. *TwinCAT tree, EL1252*):

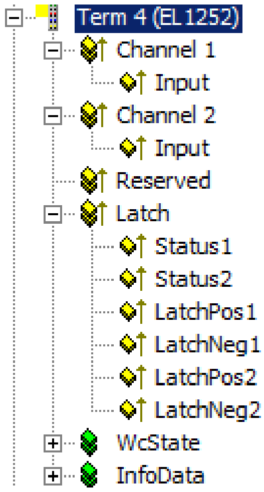


Fig. 151: TwinCAT tree, EL1252

- **Status1/Status2**
  Depending on the terminal and setting in SingleEvent-Mode [▶ 132] changes in the state of the channel in each of the status bytes Status1/Status2 are displayed in the bits 0.1and 2. The bits 3 to 7 are reserved and not recommended for evaluation.

Status Byte

EL1252	0	0->1	1	1->0
ContinuousMode (default)	0 (b#00000000)	0 (b#00000000)	0 (b#00000000)	0 (b#00000000)
SingleEventMode	0 (b#00000000)	1 (b#00000001)	1 (b#00000001)	2 (b#00000010)
EL1254	0	0->1	1	1->0
ContinuousMode (default)	0 (b#00000000)	4 (b#00000100)	4 (b#00000100)	0 (b#00000000)
SingleEventMode	0 (b#00000000)	5 (b#00000101)	4 (b#00000100)	0 (b#00000000)

- In SingleEventMode changes 0 -> 1/1 -> 0 are displayed for only one EtherCAT cycle, because reading the time LatchPosY/LatchNegY also resets the displayed changes.
- **LatchPos1/LatchPos2**
  The time of the first/last rising signal edge (in the period of time between the most recent fieldbus cycle and the current read of the terminal by an EtherCAT frame)
- **LatchNeg1/LatchNeg2**
  The time of the first/last falling signal edge (in the period of time between the most recent fieldbus cycle and the current read of the terminal by an EtherCAT frame)
- **Channel 1 Input / Channel 2 Input**
  Input level 0 or 1

The times of the edges are made available in the form of 64-bit times, i.e. as 8 bytes of process data based on the Distributed Clock of the terminal.
They are constructed as follows: LatchPosXXXY (XXX: POS/NEG, rising or falling edge; Y: channel, 1 or 2).

Temporal consistency of the input data

The input data of the EL1252, EL1254 can be functionally divided into two groups. This is also visible from their storage location in the ESC registers:
- Input channels (Channel x Input) and Distributed Clocks status bytes
- Distributed Clocks latch times.

The two data blocks are updated at different “speeds” - whilst the input data is read in and made available **upon the arrival** of the Ethernet frame at the EtherCAT slave, the DC latch times can still be updated **during the processing of the datagram** on account of the external latch event. Therefore, the latch times can still change up until a few ns before being read into the Ethernet frame - this ensures that it is really the latest possible status of the inputs that is mapped into the frame.
Example: Low/0V is present on channel 1 and the collecting Ethernet frame arrives at the EL1252/EL1254. During frame processing, but before the DC latch data is read in, a rising edge occurs that leads to LatchPos1 being updated - this updated time is then written to the collecting datagram. The two pieces of information, “rising edge” vs “signal level”, are thus inconsistent in this case.
Recommendation: In order to precisely analyze the input channels, especially in the case of short pulses at the inputs, exclusively the 64-bit time stamp LatchNeg/LatchPos should be evaluated.

5.8.3 EL1252, EL1254 - Process data

5.8.3.1 EL1252 - Process data

The extent of the process data that is made available can be changed via the “Process data” tab (see Fig. *Process data tab, EL1252*).

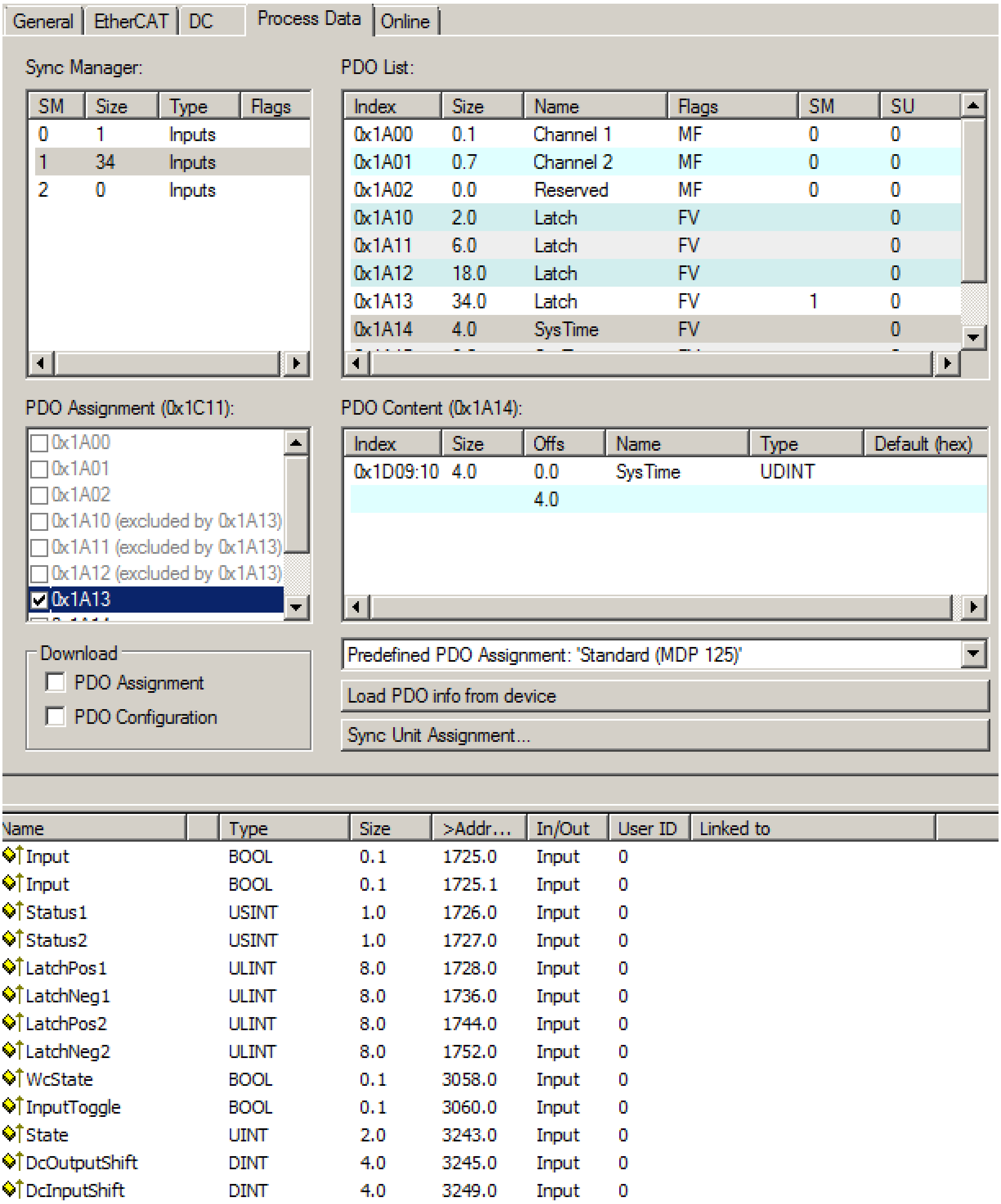


Fig. 152: Process data tab, EL1252

Sync Manager (SM)

In order to configure the process data, select the required Sync Manager (SM0..SM2 are available) in the upper left-hand “Sync Manager” box (fig. Process data tab, EL1252 [▶ 125]). The process data assigned to this Sync Manager can then be switched on or off in the “PDO Assignment” box underneath. Restarting the EtherCAT system, or reloading the configuration in configuration mode (F4), causes the EtherCAT communication to restart, and the process data is transferred from the terminal.

It is of crucial importance that an optional PDO is assigned to the correct SyncManager, otherwise the EtherCAT start will result in incomplete data exchange or individual variables "do not work". In the following described by means of the example 0x1A15 (belonging to the SM0):

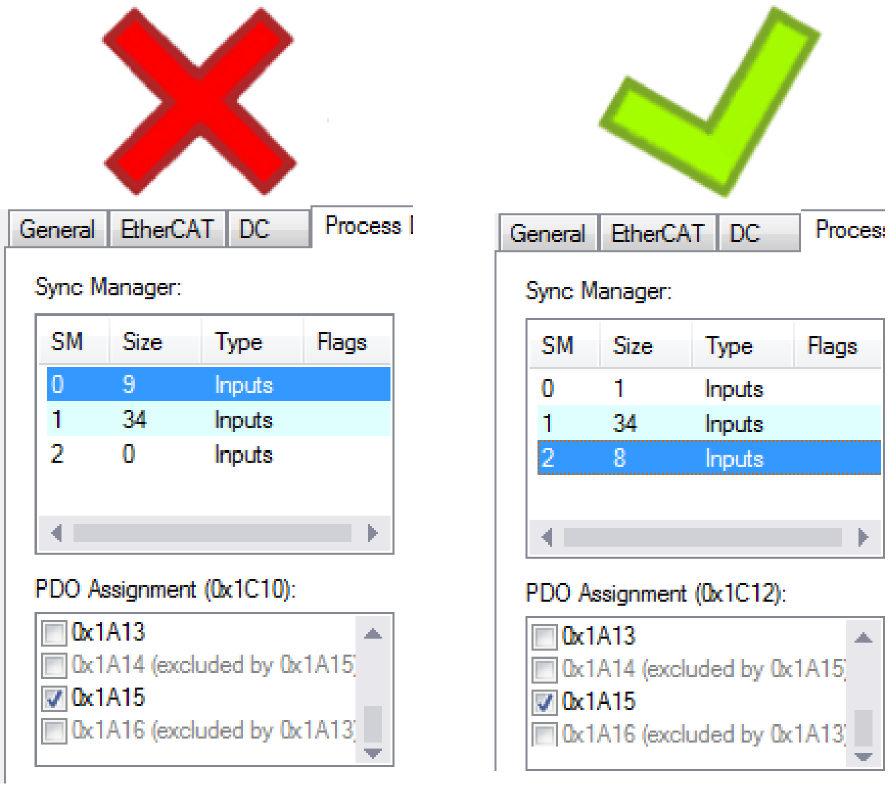


Fig. 153: Incorrect PDO assignment (left) and correct PDO assignment (right) by means of the example 0x1A15

SM0, PDO assignment 0x1C10 (not changeable)			
Index	**Size (byte.bit)**	**Name**	**PDO content**
0x1A00 (default)	0.1	Channel 1	Input 1
0x1A01(default)	0.1	Channel 2	Input 2
0x1A02 (default)	0.6	Reserved	reserved

SM01, PDO assignment 0x1C11 (PDOs are mutually exclusive)			
Index	**Size (byte.bit)**	**Name**	**PDO content**
0x1A10	2.0	Latch	Status1; Status2
0x1A11	6.0	Latch	Status1; LatchPos1
0x1A12	18.0	Latch	Status1; LatchPos1; LatchNeg1
0x1A13 (default)	34.0	Latch	Status1/Status2; LatchPos1/LatchPos2; LatchNeg1/LatchNeg2
0x1A16	34.0	Latch	LatchPos1/LatchPos2; LatchNeg1/Latch-Neg2
0x1A17*	26.0	LatchPos64	LatchPos1/LatchPos2;
0x1A18*	34.0	LatchNeg64	LatchNeg1/LatchNeg2

*) from Rev. -0022

SM2, *PDO assignment 0x1C12 (PDOs are mutually exclusive)*			
Index	**Size (byte.bit)**	**Name**	**PDO content**
0x1A14	4.0	SysTime	32 bit representation
0x1A15 (default)	8.0	SysTime	64 bit representation

Extent of process data

The standard configuration of the EL2152 is shown in Fig. *Standard extent of process data, EL1252*:

Name	Type	Size	In/Out
Input	BOOL	0.1	Input
Input	BOOL	0.1	Input
Status1	USINT	1.0	Input
Status2	USINT	1.0	Input
LatchPos1	ULINT	8.0	Input
LatchNeg1	ULINT	8.0	Input
LatchPos2	ULINT	8.0	Input
LatchNeg2	ULINT	8.0	Input
WcState	BOOL	0.1	Input
InputToggle	BOOL	0.1	Input
State	UINT	2.0	Input
DcOutputShift	DINT	4.0	Input
DcInputShift	DINT	4.0	Input

Fig. 154: Standard extent of process data EL1252

- PDO 0x1A00, 0x1A01, 0x1A02 (Channel 1, Input; Channel 2, Input)
- PDO 0x1A13 (Status1/Status2; LatchPos1/LatchPos2; LatchNeg1/LatchNeg2)

Libraries for 64-bit processing

A selection of functions for handling 64-bit numbers is available under Beckhoff TwinCAT in the TcUtilities.lib library. Longer execution times are required here than is the case with standard, 32-bit data types. A data type with a width of 64-bits is defined in TcEthercat.lib as T_DCTIME or in TcUtilities.lib as T_LARGE_INTEGER.

5.8.3.2 EL1252 - Predefined PDO Assignment

The "Predefined PDO Assignment" enables a simplified selection of the process data (from Rev. - 0018). The desired function is selected on the lower part of the "Process Data" tab. As a result, all necessary PDOs are automatically activated and the unnecessary PDOs are deactivated.
If channels are not to be used, it is recommended to deactivate the respective channels in order to avoid possible error messages from the terminal.

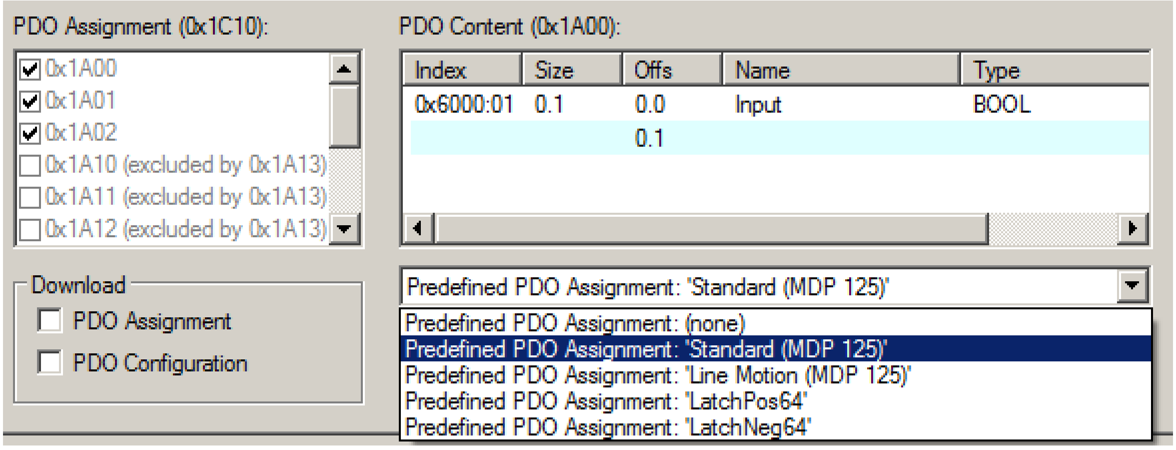


Fig. 155: EL1252 - Selection of predefined PDOs

There is a choice of four PDO assignments:

Name	SM0, PDO assignment	SM1, PDO assignment
Standard (MDP125)	0x1A00 (Channel 1, Input) 0x1A01 (Channel 2, Input) 0x1A02 (Reserved)	0x1A13 (Latch)
Line Motion (MDP125)	0x1A00 (Channel 1, Input) 0x1A01 (Channel 2, Input) 0x1A02 (Reserved)	0x1A16 (Latch)
LatchPos64*	0x1A00 (Channel 1, Input) 0x1A01 (Channel 2, Input) 0x1A02 (Reserved)	0x1A17 (LatchPos64)
LatchNeg64*	0x1A00 (Channel 1, Input) 0x1A01 (Channel 2, Input) 0x1A02 (Reserved)	0x1A18 (LatchNeg64)

*) from Rev. -0022

5.8.3.3 EL1254 - Process data

The extent of the process data that is made available can be changed via the “Process data” tab (see Fig. *Process data tab, EL1254*).

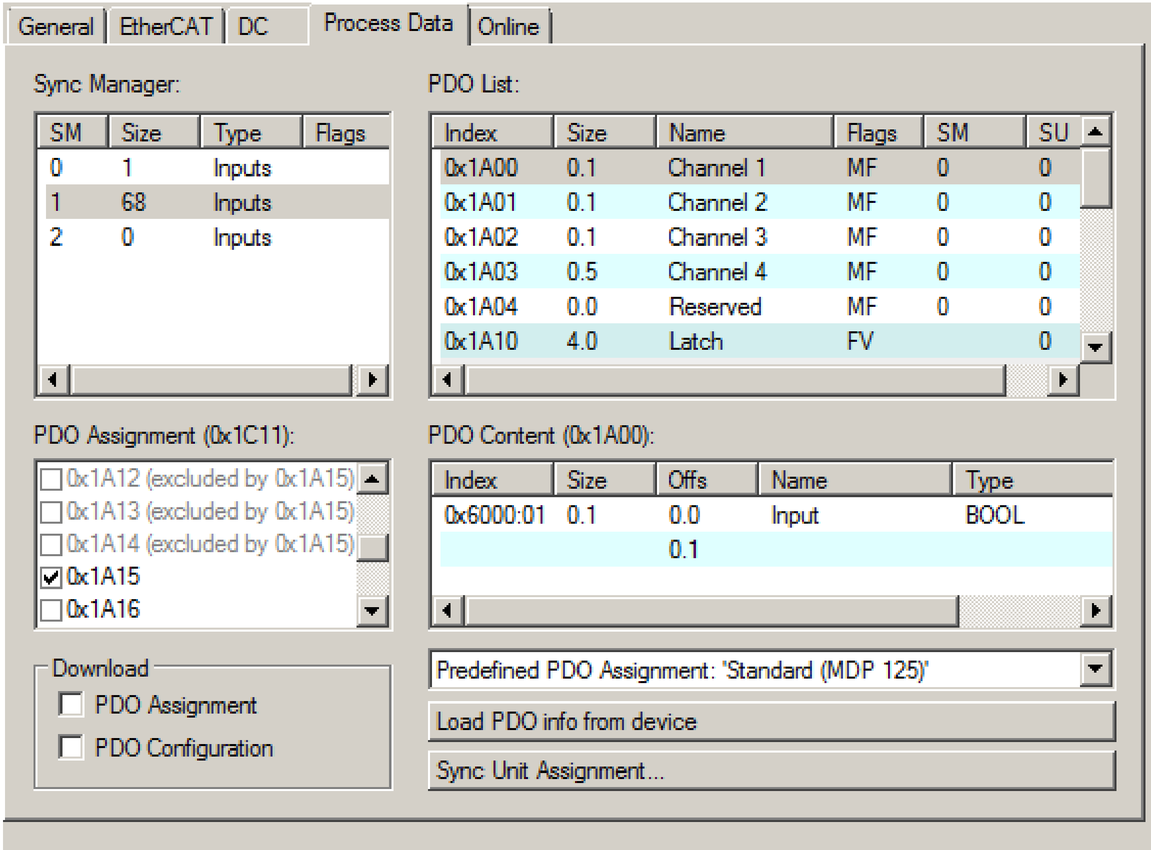


Fig. 156: Process data tab, EL1254

Sync Manager (SM)

In order to configure the process data, select the required Sync Manager (SM0..SM2 are available) in the upper left-hand “Sync Manager” box (fig. Process data tab, EL1254 [▸ 129]). The process data assigned to this Sync Manager can then be switched on or off in the “PDO Assignment” box underneath. Restarting the EtherCAT system, or reloading the configuration in configuration mode (F4), causes the EtherCAT communication to restart, and the process data is transferred from the terminal.

It is of crucial importance that an optional PDO is assigned to the correct SyncManager, otherwise the EtherCAT start will result in incomplete data exchange or individual variables "do not work". In the following described by means of the example 0x1A17 (belonging to the SM0):

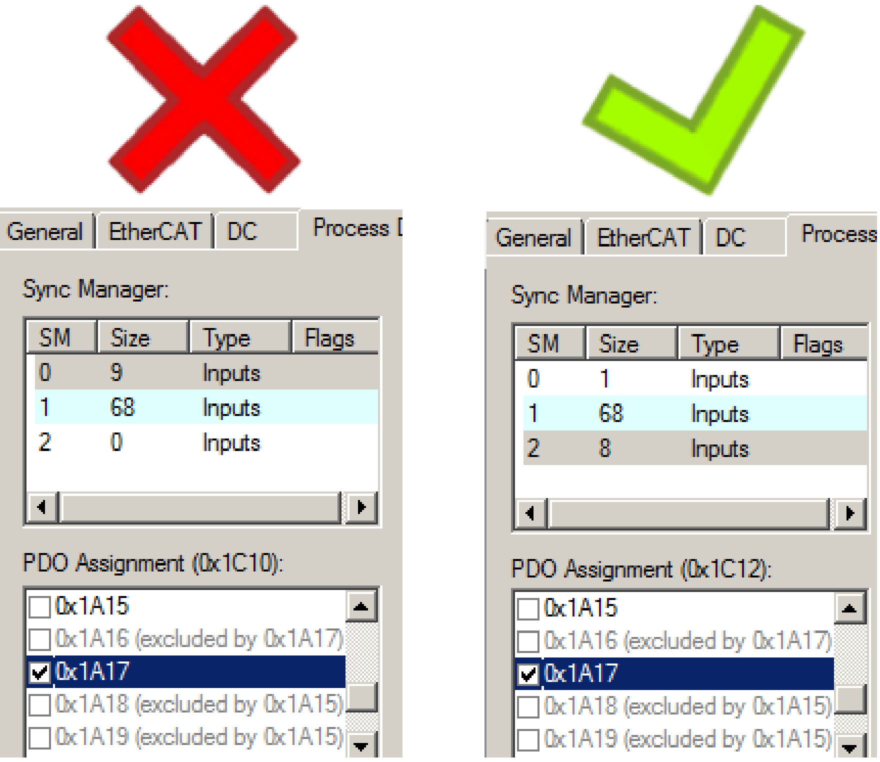


Fig. 157: Incorrect PDO assignment (left), correct PDO assignment (right) by means of the example 0x1A17

SM0, PDO assignment 0x1C10 (not changeable)			
Index	**Size (byte.bit)**	**Name**	**PDO content**
0x1A00 (default)	0.1	Channel 1	Input 1
0x1A01(default)	0.1	Channel 2	Input 2
0x1A02 (default)	0.1	Channel 3	Input 3
0x1A03 (default)	0.5	Channel 4	Input 4
0x1A04 (default)	0.0	Reserved	Reserved

SM01, PDO assignment 0x1C11 (PDOs are mutually exclusive)			
Index	**Size (byte.bit)**	**Name**	**PDO content**
0x1A10	4.0	Latch	Status1; Status2, Status3, Status4
0x1A11	6.0	Latch	Status1; LatchPos1
0x1A12	18.0	Latch	Status1;LatchPos1;LatchNeg1
0x1A13	34.0	Latch	Status1/Status2; LatchPos1/LatchPos2; LatchNeg1/LatchNeg2
0x1A14	51.0	Latch	Status1; Status2, Status3 LatchPos1/LatchPos2/LatchPos3; LatchNeg1/LatchNeg2/LatchNeg3
0x1A15 (default)	68.0	Latch	Status1; Status2, Status3, Status4 LatchPos1/LatchPos2/LatchPos3/LatchPos4; LatchNeg1/LatchNeg2/LatchNeg3/LatchNeg4
0x1A18	34.0	Latch	LatchPos1LatchPos2; LatchNeg1/LatchNeg2
0x1A19	52.0	Latch	LatchPos1LatchPos2/LatchPos3; LatchNeg1/LatchNeg2/LatchNeg3
0x1A1A	68.0	Latch	LatchPos1/LatchPos2/LatchPos3/LatchPos4; LatchNeg1/LatchNeg2/LatchNeg3/LatchNeg4
0x1A1B	60.0	LatchPos64	LatchPos1/LatchPos2/LatchPos3/LatchPos4;
0x1A1C	68.0	LatchNeg64	LatchNeg1/LatchNeg2/LatchNeg3/LatchNeg4

SM2, *PDO assignment 0x1C12 (PDOs are mutually exclusive)*			
Index	**Size (byte.bit)**	**Name**	**PDO content**
0x1A16	4.0	SysTime	32 bit representation
0x1A17	8.0	SysTime	64 bit representation

Extent of process data

The standard configuration of the EL2154 is shown in Fig. *Standard extent of process data, EL1254*:

Name	Type	Size	>Addr...	In/Out
Input	BOOL	0.1	1760.0	Input
Input	BOOL	0.1	1760.1	Input
Input	BOOL	0.1	1760.2	Input
Input	BOOL	0.1	1760.3	Input
Status3	USINT	1.0	1761.0	Input
Status4	USINT	1.0	1762.0	Input
Status1	USINT	1.0	1763.0	Input
Status2	USINT	1.0	1764.0	Input
LatchPos1	ULINT	8.0	1765.0	Input
LatchNeg1	ULINT	8.0	1773.0	Input
LatchPos2	ULINT	8.0	1781.0	Input
LatchNeg2	ULINT	8.0	1789.0	Input
LatchPos3	ULINT	8.0	1797.0	Input
LatchNeg3	ULINT	8.0	1805.0	Input
LatchPos4	ULINT	8.0	1813.0	Input
LatchNeg4	ULINT	8.0	1821.0	Input
WcState	BOOL	0.1	3058.0	Input
InputToggle	BOOL	0.1	3060.0	Input
State	UINT	2.0	3253.0	Input
DcOutputShift	DINT	4.0	3255.0	Input
DcInputShift	DINT	4.0	3259.0	Input

Fig. 158: Standard extent of process data EL1254

- PDO 0x1A00 (Channel 1, Input), 0x1A01 (Channel 2, Input), 0x1A02 (Channel 3, Input), 0x1A03 (Channel 4, Input), 0x1A04 (reserved)
- PDO 0x1A15 (Status1/Status2/Status3/Status4; LatchPos1/LatchPos2/LatchPos3, LatchPos4; LatchNeg1/LatchNeg2/LatchNeg3/LatchNeg4)

Libraries for 64-bit processing

A selection of functions for handling 64-bit numbers is available under Beckhoff TwinCAT in the TcUtilities.lib library. Longer execution times are required here than is the case with standard, 32-bit data types. A data type with a width of 64-bits is defined in TcEthercat.lib as T_DCTIME or in TcUtilities.lib as T_LARGE_INTEGER.

5.8.3.4 EL1254 - Predefined PDO Assignment

The "Predefined PDO Assignment" enables a simplified selection of the process data. The desired function is selected on the lower part of the "Process Data" tab. As a result, all necessary PDOs are automatically activated and the unnecessary PDOs are deactivated.
If channels are not to be used, it is recommended to deactivate the respective channels in order to avoid possible error messages from the terminal.

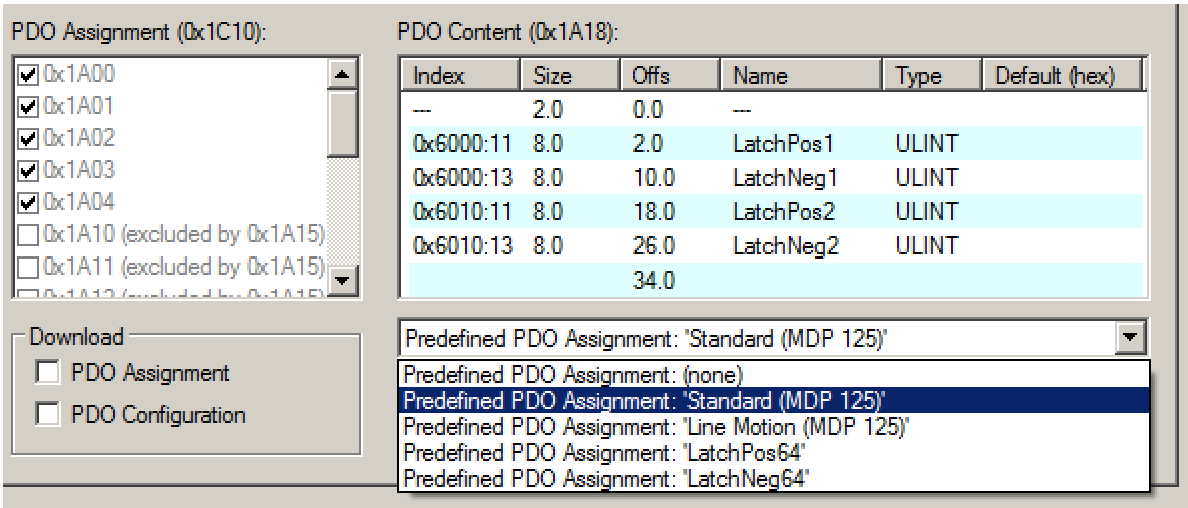


Fig. 159: EL1254 - Selection of predefined PDOs

There is a choice of four PDO assignments:

Name	**SM0, PDO assignment**	**SM1, PDO assignment**
Standard (MDP125)	0x1A00 (Channel 1, Input) 0x1A01 (Channel 2, Input) 0x1A02 (Channel 3, Input) 0x1A03 (Channel4, Input) 0x1A04 (Reserved)	0x1A15 (Latch)
Line Motion (MDP125)	0x1A00 (Channel 1, Input) 0x1A01 (Channel 2, Input) 0x1A02 (Channel 3, Input) 0x1A03 (Channel4, Input) 0x1A04 (Reserved)	0x1A1A (Latch)
LatchPos64	0x1A00 (Channel 1, Input) 0x1A01 (Channel 2, Input) 0x1A02 (Channel 3, Input) 0x1A03 (Channel4, Input) 0x1A04 (Reserved)	0x1A1B (LatchPos64)
LatchNeg64	0x1A00 (Channel 1, Input) 0x1A01 (Channel 2, Input) 0x1A02 (Channel 3, Input) 0x1A03 (Channel4, Input) 0x1A04 (Reserved)	0x1A1B (LatchNeg64)

5.8.3.5 Timestamp function

For each channel and for each fieldbus cycle, the terminal can only store **one** edge event as rising or falling together with its timestamp. If more than one signal change occurs at the terminal input between two fieldbus cycles, a setting can be made on the DC tab “Advanced Settings -> Distributed Clocks ->Latch” to determine whether the **first** or **last** signal change occurring within an interrogation cycle is to be stored in the LatchPosXXXY variable. This value is then fetched by the EtherCAT frame.

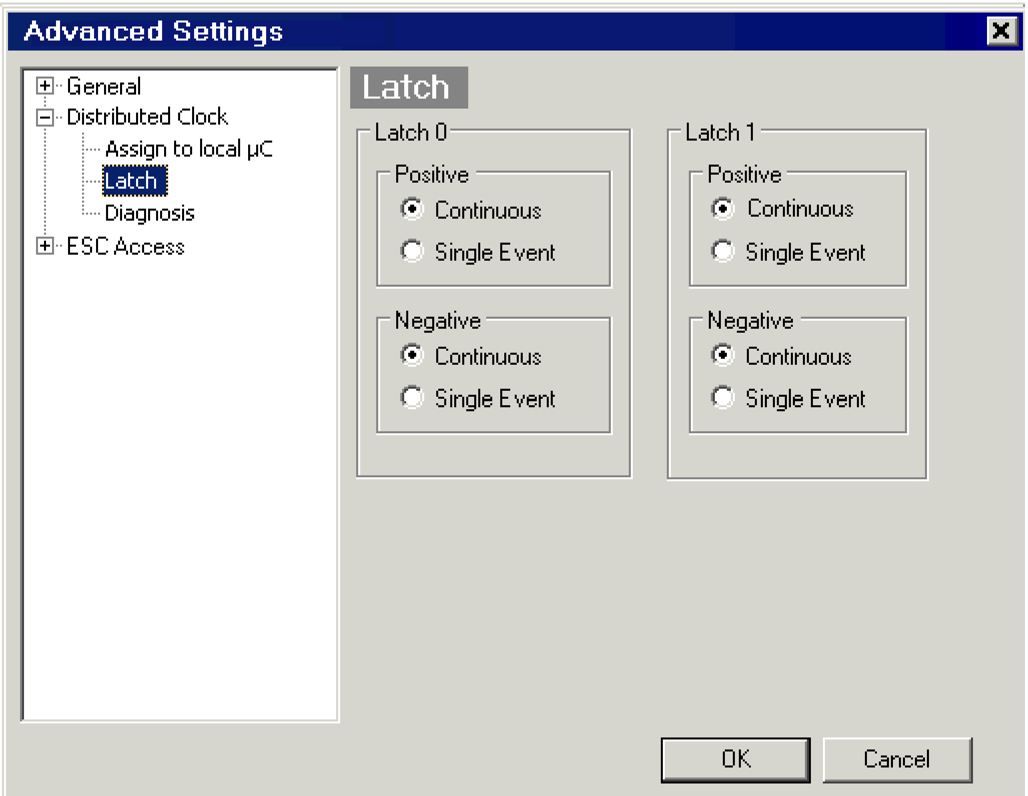


Fig. 160: Advanced settings - latch

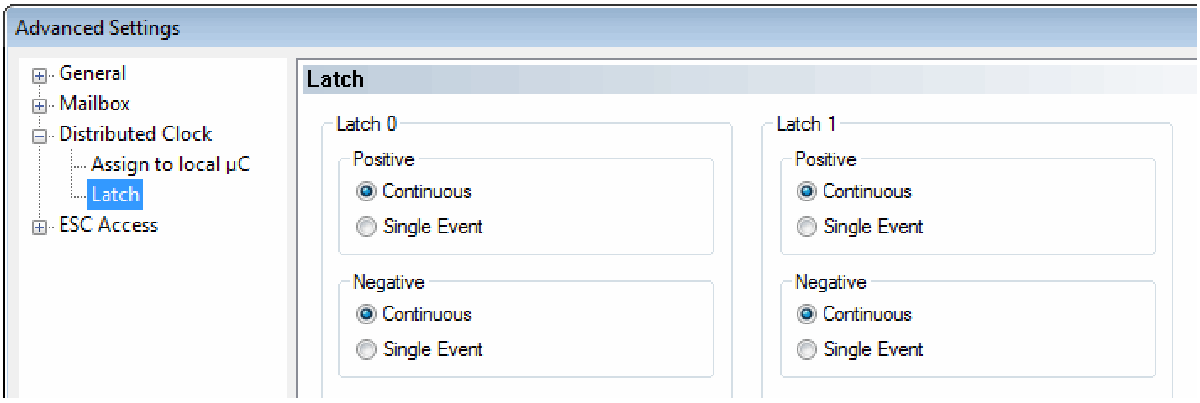
- **"Single Event"**
  If the EL1252/EL1254 is set to store the **first** signal change, the time of the first change in the signal to occur after an EtherCAT frame has read the terminal is stored in LatchPosXXXY. This time stamp then remains unchanged until the next read by an EtherCAT frame; it is not overwritten. Only after being read by an EtherCAT frame is the value free to be overwritten by another signal change.
  In this mode, the flags in the status byte also provide an indication.
- **"Continuous" (default)**
  If the EL1252/EL1254 is set to record the **last** signal change, the value in LatchPosXXXY is continuously updated whenever a change occurs in the input signal, regardless of reads by the EtherCAT frame.
  In this mode the status byte does not contain any edge flags for each channel.

5.8.3.6 Status Byte

The first two bits of the status byte for each channel indicate whether a rising or falling edge has been detected or latched. The other bits are intended for future purpose and shall not be evaluated.

Bit	SB.7	SB.6	SB.5	SB.4	SB.3	SB.2	SB.1	SB.0
Name	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Event falling edge 1→0	Event rising edge 0→1

The flags will signal only, if the terminal selected in the system tree is set to "Single Event" instead of "Continuous" under
„EtherCAT→Advanced Settings": Distributed Clock/ Latch for the relevant event:



The flags are not reset until the respective LatchPosXXXY variables are then read.

Performance

Please be sure that the technical possibilities offered by the EL1252/EL1254 are exploited fully by providing high input signal quality, avoiding contact bounce and line interference, and by providing an input signal with sufficiently short rise and fall times!

5.8.4 Example programs

Using the sample programs

This document contains sample applications of our products for certain areas of application. The application notes provided here are based on typical features of our products and only serve as examples. The notes contained in this document explicitly do not refer to specific applications. The customer is therefore responsible for assessing and deciding whether the product is suitable for a particular application. We accept no responsibility for the completeness and correctness of the source code contained in this document. We reserve the right to modify the content of this document at any time and accept no responsibility for errors and missing information.

Example 1: Output and evaluation of pulses

In this sample program an EL2252 generates periodically different pulses on both channels, which can be measured in real-time with an EL1252.

Connection diagram:

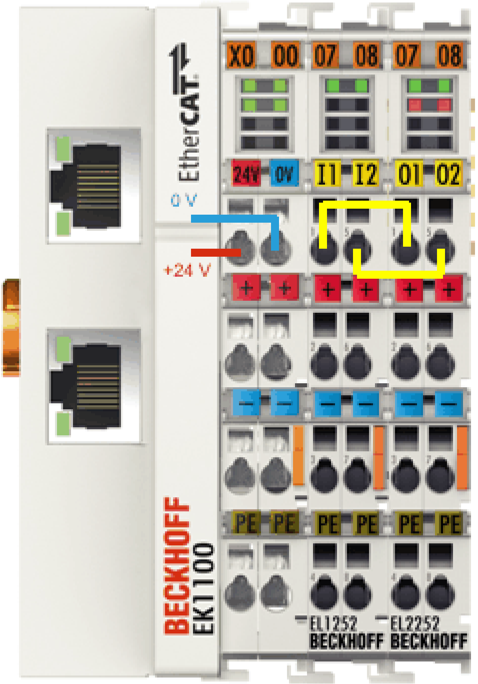


Fig. 161: Wiring for example program 1

Starting the example program

The application examples have been tested with a test configuration and are described accordingly. Certain deviations when setting up actual applications are possible.

The following hardware and software were used for the test configuration:

- TwinCAT 2
    - TwinCAT master PC with Windows XP Professional SP 3, TwinCAT version 2.10 (Build 1335) and INTEL PRO/100 VE Ethernet adapter
- TwinCAT 3
    - TwinCAT-Master-PC with Windows 7 (64 Bit) SP 1, TwinCAT version 3.1 (Build 4022) and TwinCAT-Intel PCI Ethernet adapter (Gigabit)
- Beckhoff EK1100 EtherCAT coupler, EL1252, EL2252 and EL9011 terminals.

5.8.4.1 Sample program TwinCAT 2

https://infosys.beckhoff.com/content/1033/el1202_el1252/Resources/zip/4318611851.zip

Please continue with further descriptions in sectionTwinCAT Quickstart, TwinCAT 2.TwinCAT Quickstart, TwinCAT 2 [▶ 45].

Procedure for starting the program

- After clicking the Download button, save the zip file locally on your hard disk, and unzip the *.TSM (configuration) and the *.PRO (PLC program) files into a temporary working folder.
- The *.pro file can be opened by double click or by the TwinCAT PLC Control application with menu selection “File/ Open”. The *.tsm file is provided for the TwinCAT System Manager (to review or overtake configurations).
- Connect the hardware in accordance with fig. Connection for sample program [▶ 134] and connect the Ethernet adapter of your PC to the EtherCAT coupler (further information on this can be found in the corresponding coupler manuals)

- Select the local Ethernet adapter (with real-time driver, if applicable) under System configuration, I/O configuration, I/O devices, Device (EtherCAT); then on the "Adapter" tab choose "Search...", select the appropriate adapter and confirm (see Fig. *Searching the Ethernet adapter* + *Selection and confirmation of the Ethernet adapter*).

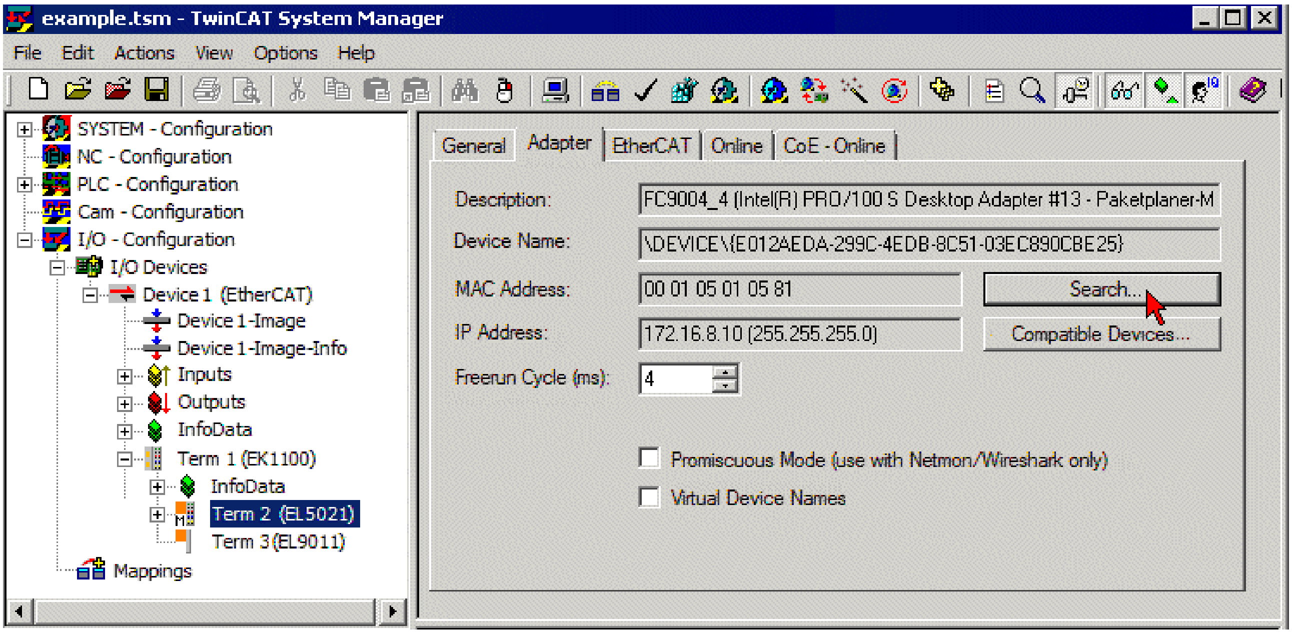


Fig. 162: *Searching the Ethernet adapter*

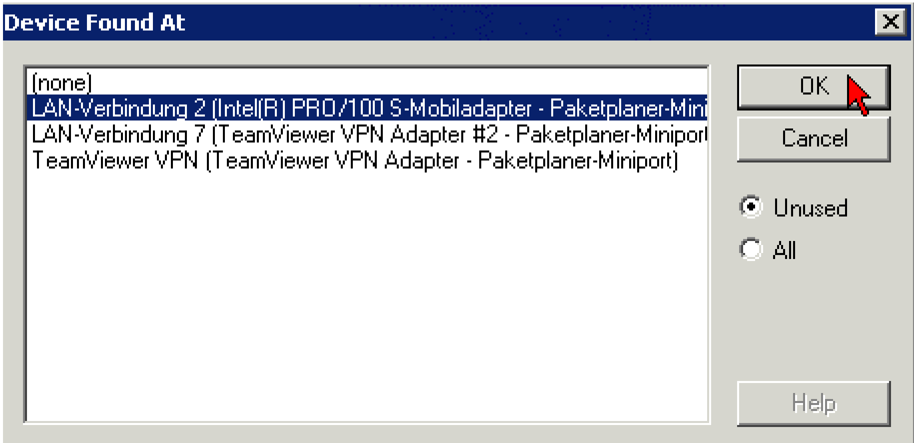


Fig. 163: Selection and confirmation of the Ethernet adapter

- Activate and confirm the configuration (Fig. *Activation of the configuration* + *Confirming the activation of the configuration*)

Fig. 164: Activation of the configuration

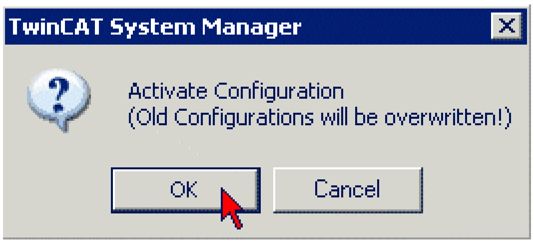


Fig. 165: Confirming the activation of the configuration

- Confirm new variable mapping, restart in RUN mode (Fig. *Generate variable mapping* + *Restarting TwinCAT in RUN mode*)

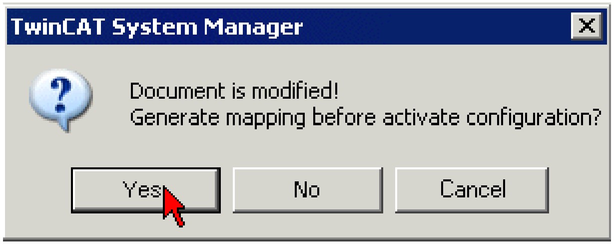


Fig. 166: Generating variable mapping

Fig. 167: Restarting TwinCAT in RUN mode

- In TwinCAT PLC, under the “Project” menu, select “Rebuild all” to compile the project (Fig. *Compile project*)

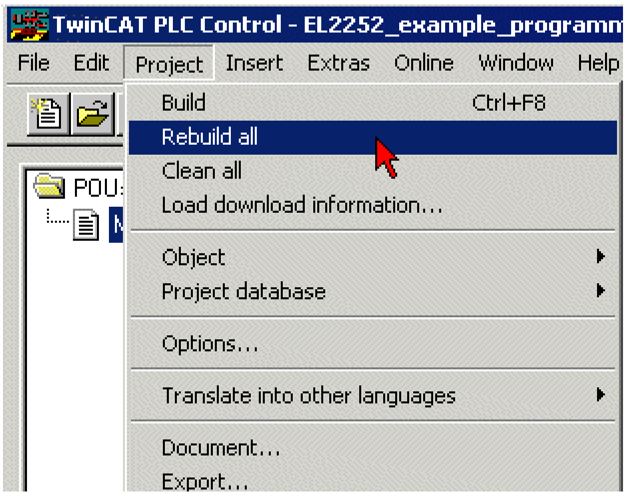


Fig. 168: Compile project

- In TwinCAT PLC: log in with the “F11” button, confirm loading the program (Fig. *Confirming program start*), run the program with the “F5” button

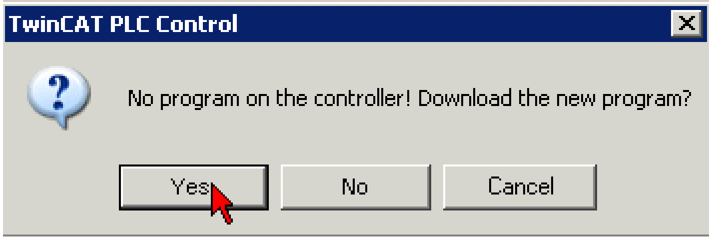


Fig. 169: Confirming program start

5.8.4.2 Sample program TwinCAT 3

https://infosys.beckhoff.com/content/1033/el1202_el1252/Resources/zip/5510184331.zip

Preparations for starting the sample programs (tnzip file / TwinCAT 3)

- Click on the download button to save the Zip archive locally on your hard disk, then unzip the *.tnzip archive file in a temporary folder.

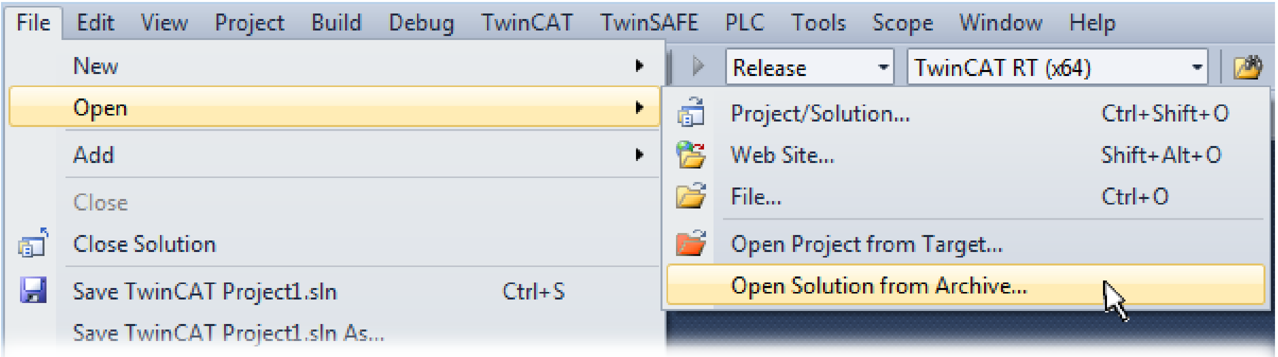


Fig. 170: Opening the *. tnzip archive

- Select the .tnzip file (sample program).
- A further selection window opens. Select the destination directory for storing the project.
- For a description of the general PLC commissioning procedure and starting the program please refer to the terminal documentation or the EtherCAT system documentation.
- The EtherCAT device of the example should usually be declared your present system. After selection of the EtherCAT device in the “Solutionexplorer” select the “Adapter” tab and click on “Search...”:

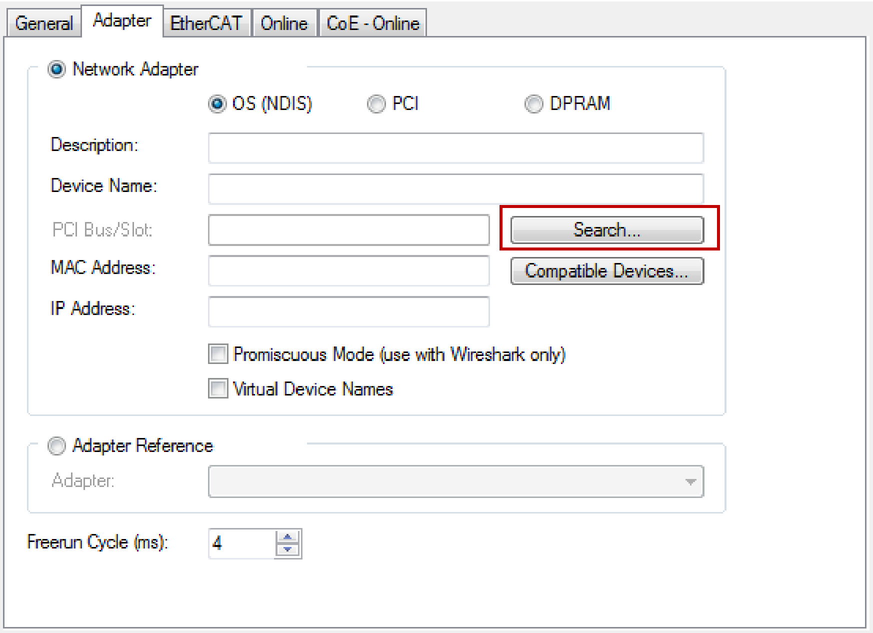
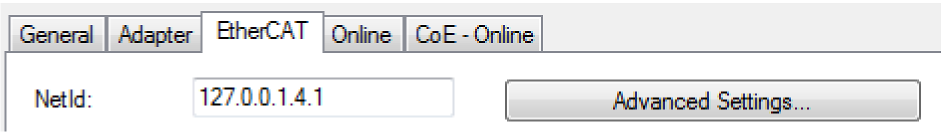


Fig. 171: Search of the existing HW configuration for the EtherCAT configuration of the example

- Checking NetId: the “EtherCAT” tab of the EtherCAT device shows the configured NetId:

  The first 4 numbers have to be identical with the project NetId of the target system. The project NetId can be viewed within the TwinCAT environment above, where a pull down menu can be opened to choose a target system (by clicking right in the text field). The number blocks are placed in brackets there next to each computer name of a target system.
- Modify the NetId: By right clicking on “EtherCAT device” within the solution explorer a context menu opens where “Change NetId...” have to be selected. The first four numbers of the NetId of the target computer have to be entered; the both last values are 4.1 usually.
  Example:
  - NetId of project: myComputer (123.45.67.89.1.1)
  - Entry via „Change NetId...“: 123.45.67.89.4.1

Also see more hints in section:
Commissioning, TwinCAT Quickstart, TwinCAT 3, Startup [▶ 55].

6 Appendix

6.1 Firmware compatibility

The terminals of the EL12xx series have no firmware.

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

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

Storage locations

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

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

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

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

Simplified update by bundle firmware

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

- for the firmware to be in a packed format: recognizable by the file name, which also contains the revision number, e.g. ELxxxx-xxxx_REV0016_SW01.efw
- for password=1 to be entered in the download dialog. If password=0 (default setting) only the firmware update is carried out, without an ESI update.
- for the device to support this function. The function usually cannot be retrofitted; it is a component of many new developments from year of manufacture 2016.

Following the update, its success should be verified

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

NOTE

Risk of damage to the device!

✓ Note the following when downloading new device files

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

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

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

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

6.2.1 Device description ESI file/XML

NOTE

Attention regarding update of the ESI description/EEPROM

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

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

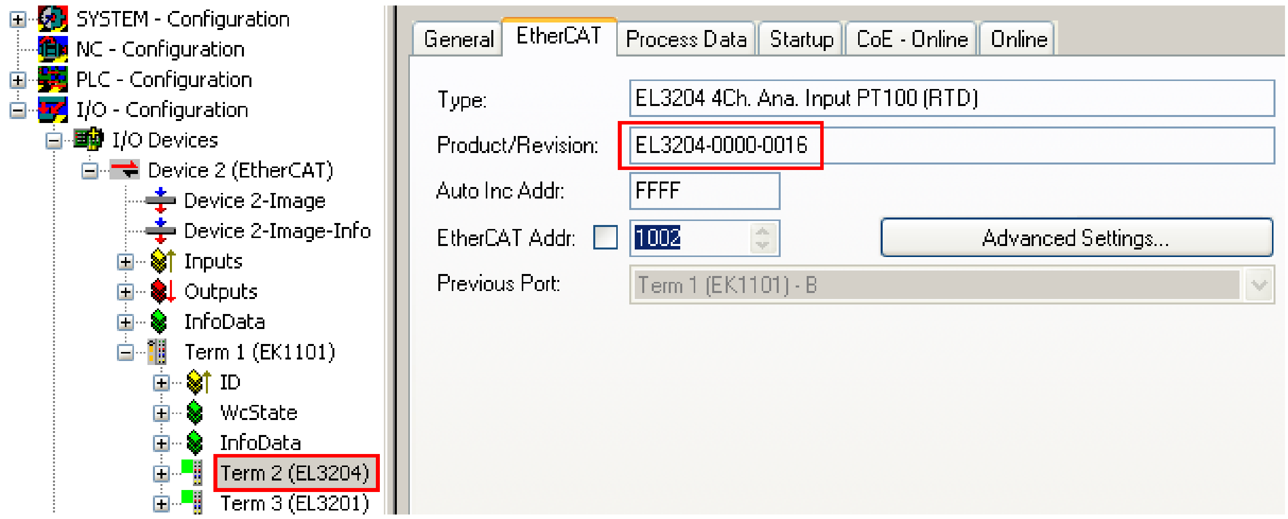


Fig. 172: Device identifier consisting of name EL3204-0000 and revision -0016

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

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

Update of XML/ESI description

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

Display of ESI slave identifier

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

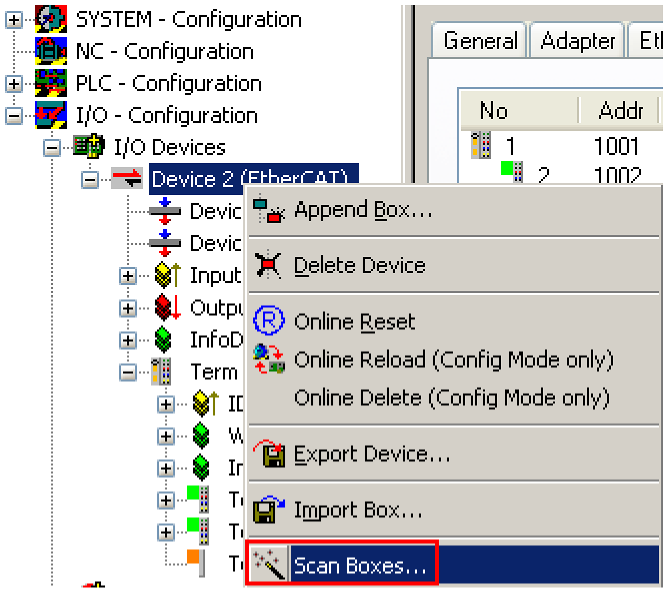


Fig. 173: Scan the subordinate field by right-clicking on the EtherCAT device

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

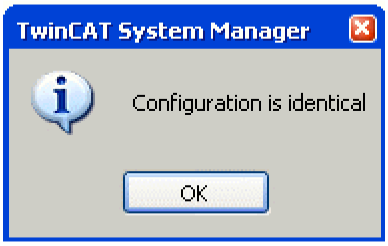


Fig. 174: Configuration is identical

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

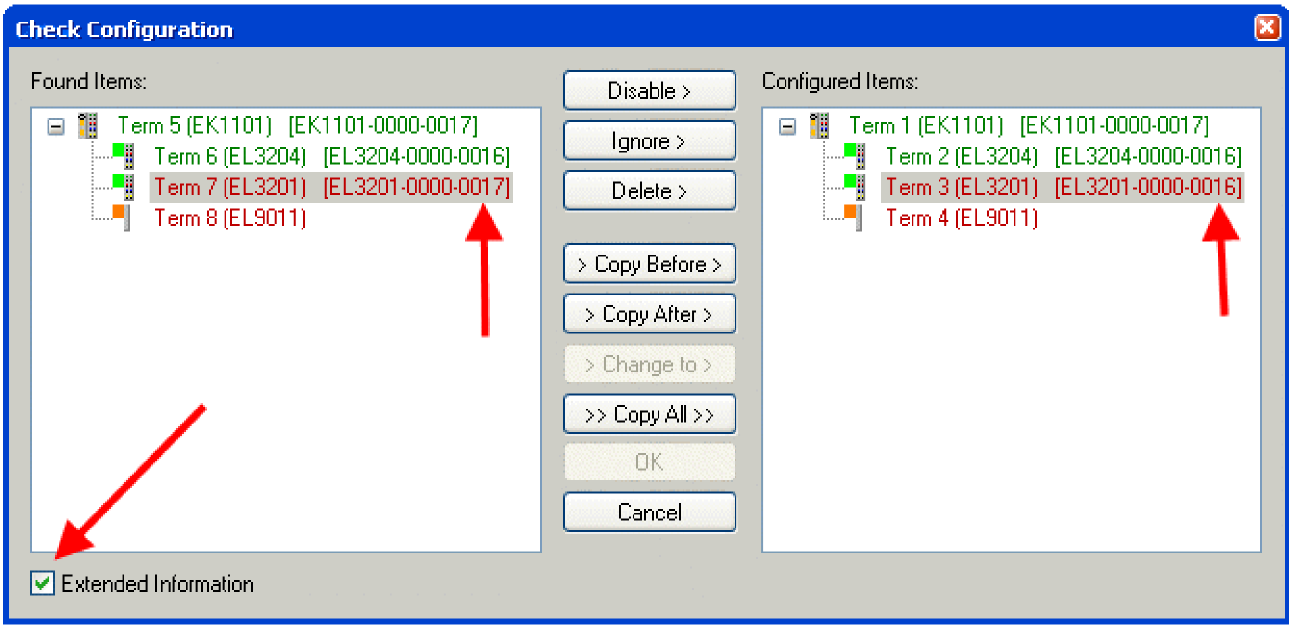


Fig. 175: Change dialog

In this example in Fig. *Change dialog*, an EL3201-0000-**0017** was found, while an EL3201-0000-**0016** was configured. In this case the configuration can be adapted with the *Copy Before* button. The *Extended Information* checkbox must be set in order to display the revision.

Changing the ESI slave identifier

The ESI/EEPROM identifier can be updated as follows under TwinCAT:

- Trouble-free EtherCAT communication must be established with the slave.
- The state of the slave is irrelevant.
- Right-clicking on the slave in the online display opens the *EEPROM Update* dialog, Fig. *EEPROM Update*

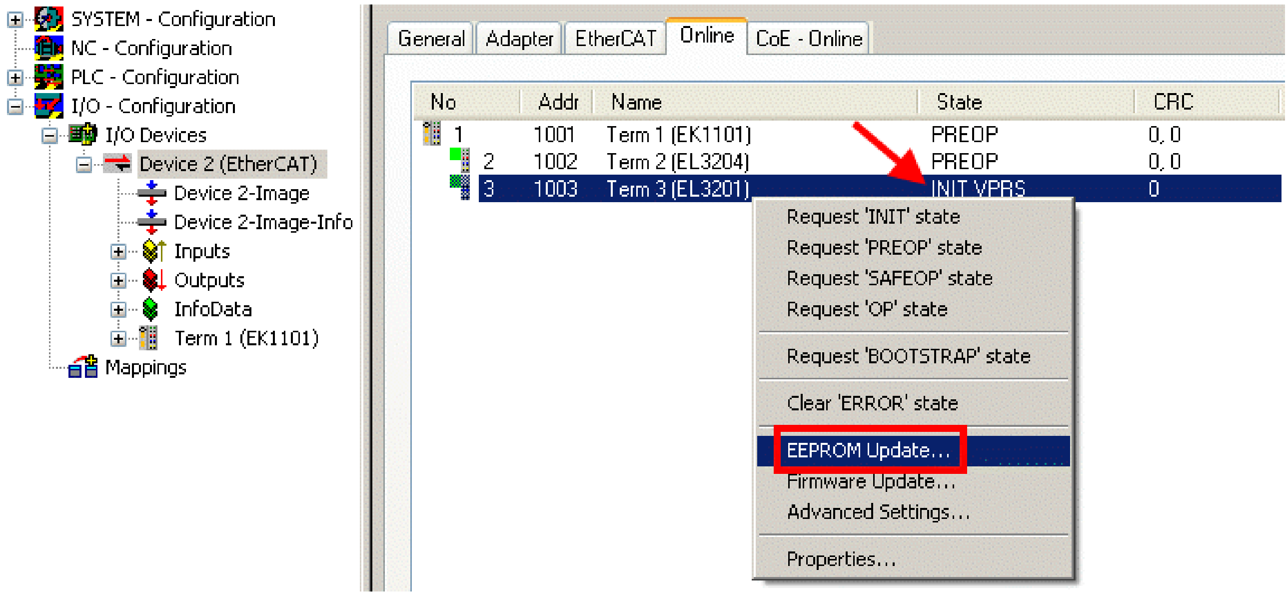


Fig. 176: EEPROM Update

The new ESI description is selected in the following dialog, see Fig. *Selecting the new ESI.* The checkbox *Show Hidden Devices* also displays older, normally hidden versions of a slave.

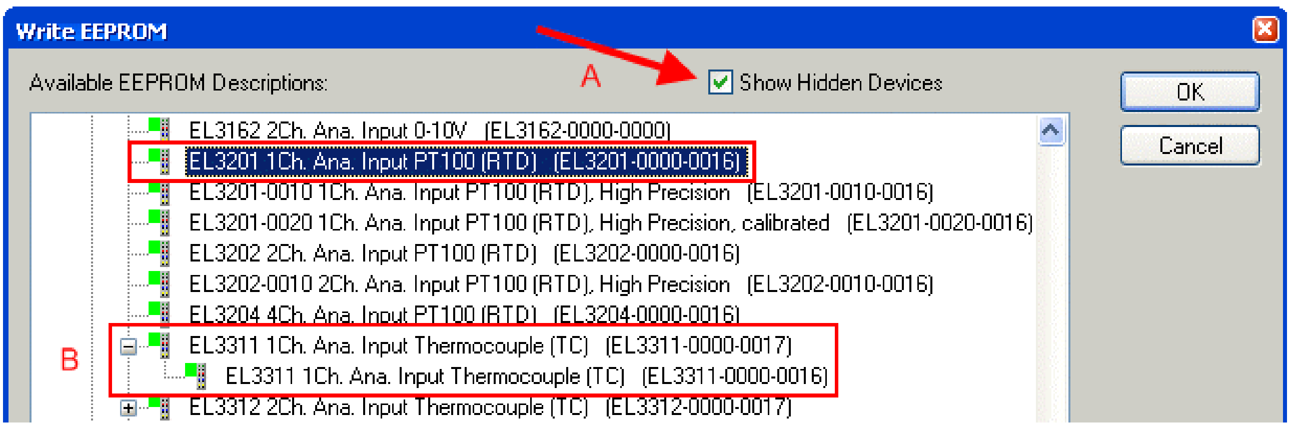


Fig. 177: Selecting the new ESI

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

The change only takes effect after a restart.

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

6.2.2 Firmware explanation

Determining the firmware version

Determining the version on laser inscription

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

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

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

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

Determining the version via the System Manager

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

CoE Online and Offline CoE

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

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

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

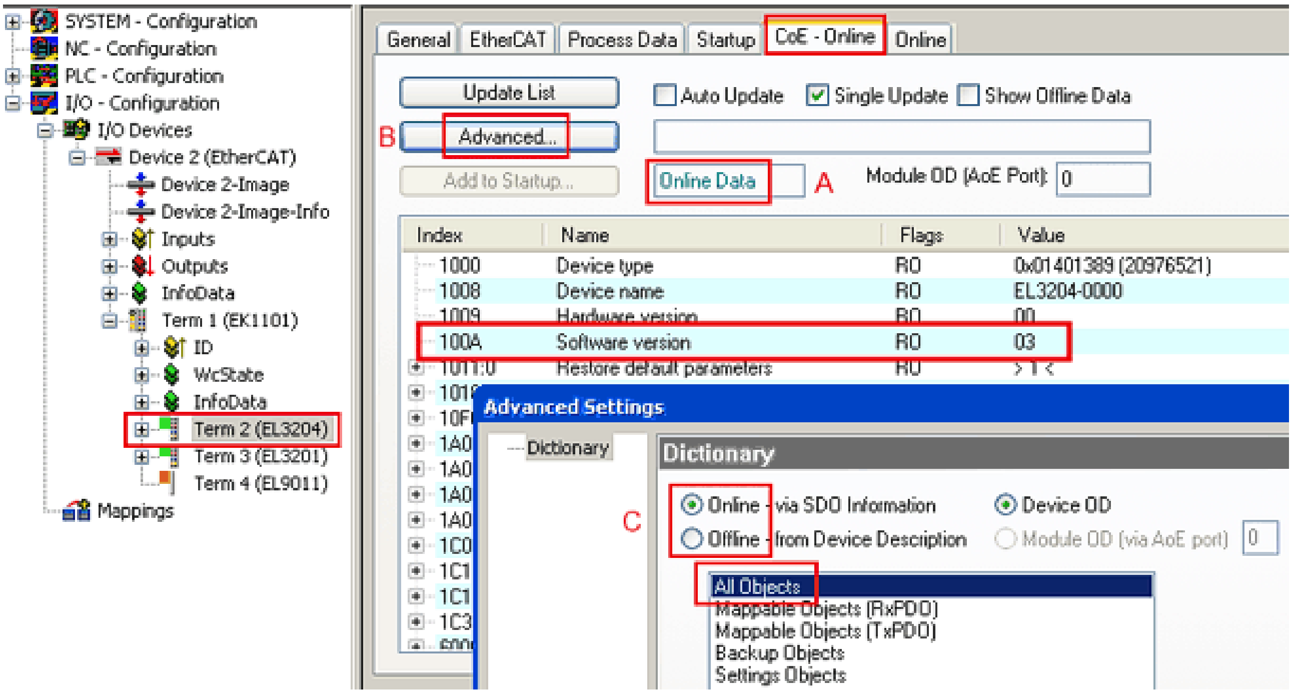


Fig. 178: Display of EL3204 firmware version

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

6.2.3 Updating controller firmware *.efw

CoE directory

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

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

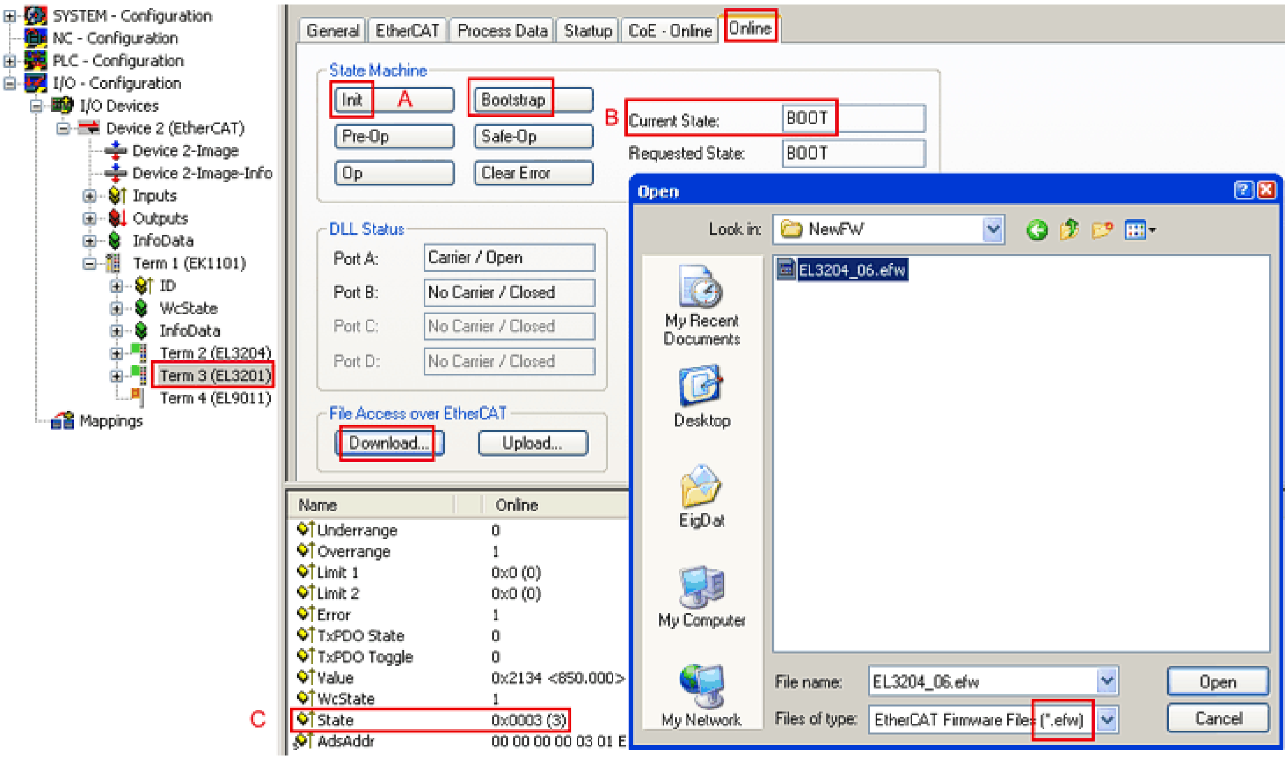
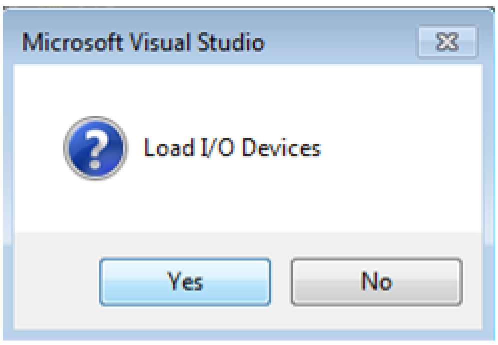
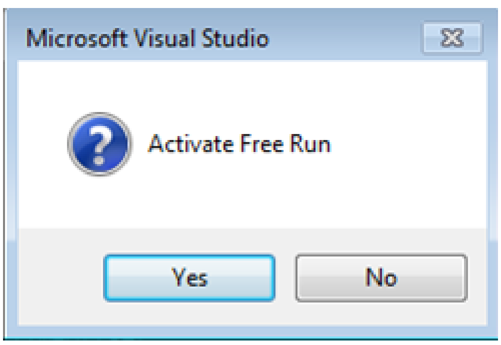


Fig. 179: Firmware Update

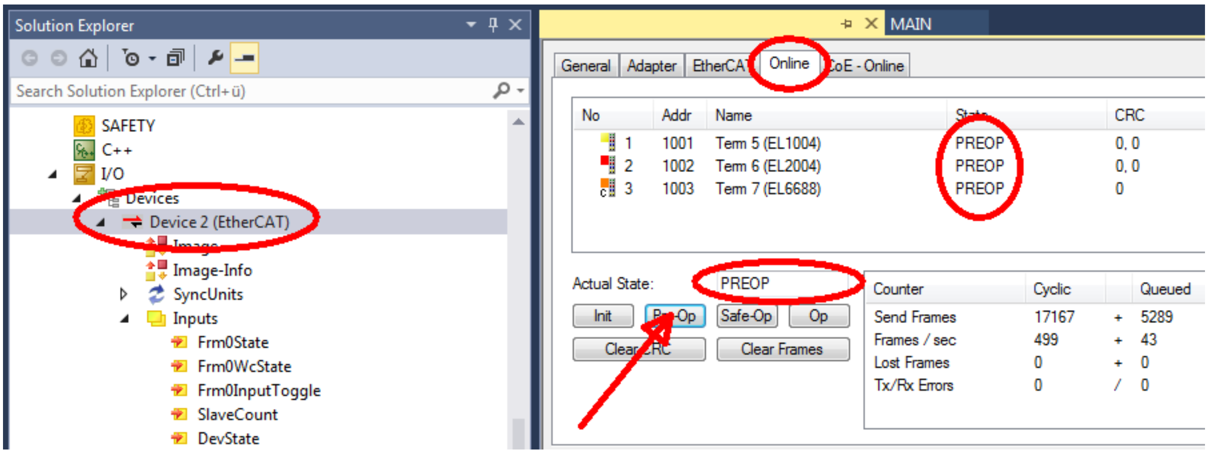
Proceed as follows, unless instructed otherwise by Beckhoff support. Valid for TwinCAT 2 and 3 as EtherCAT master.

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

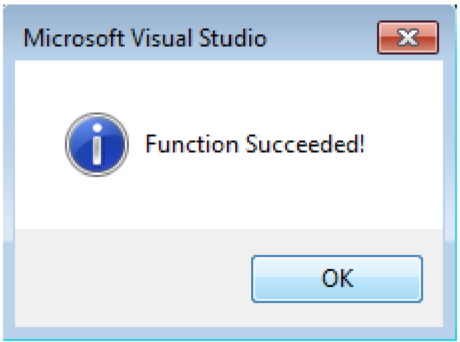




- Switch EtherCAT Master to PreOP



- Switch slave to INIT (A)
- Switch slave to BOOTSTRAP
- Check the current status (B, C)
- Download the new *efw file (wait until it ends). A pass word will not be neccessary usually.



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

6.2.4 FPGA firmware *.rbf

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

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

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

Determining the version via the System Manager

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

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

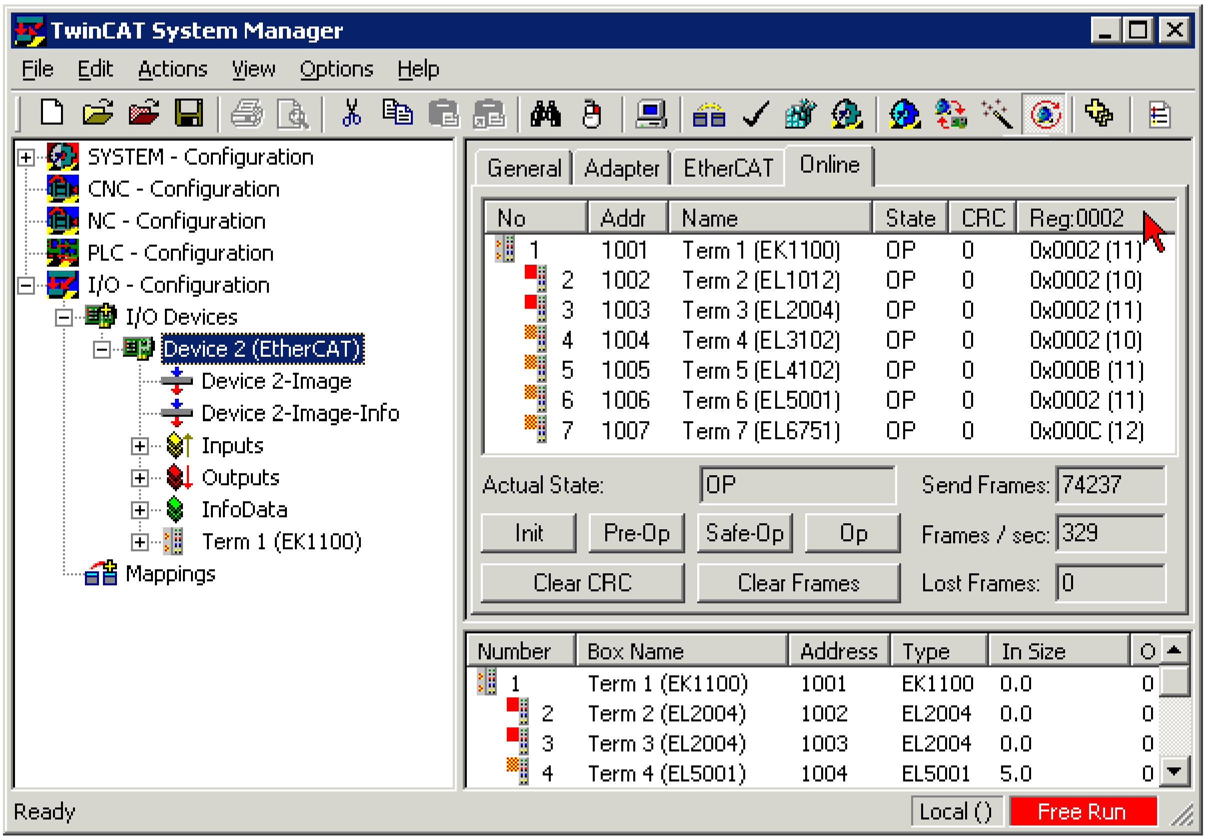


Fig. 180: FPGA firmware version definition

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

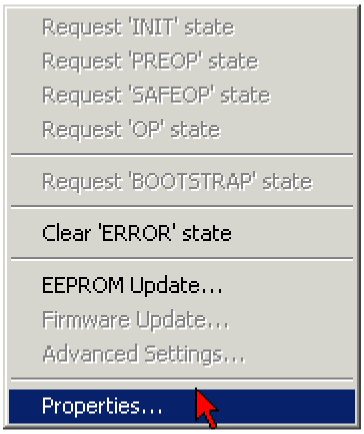


Fig. 181: Context menu *Properties*

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

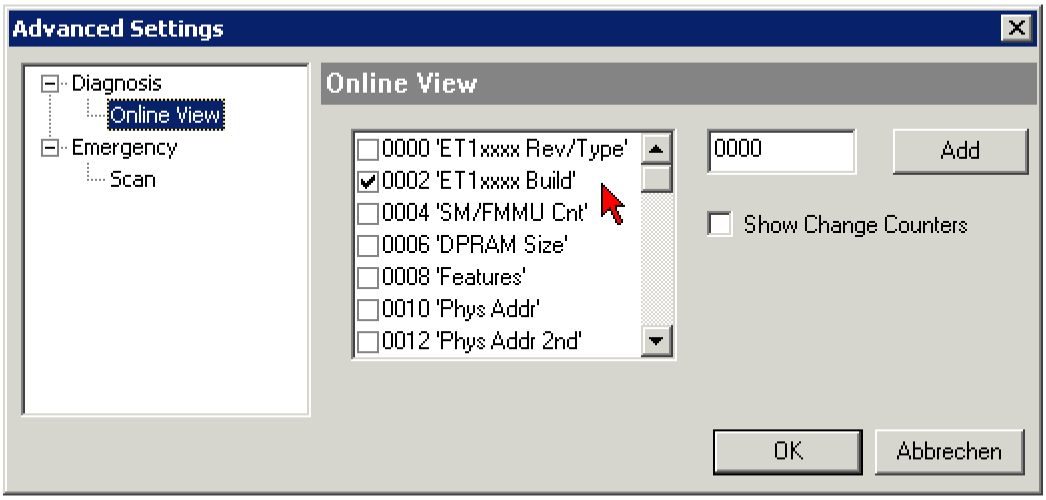


Fig. 182: Dialog *Advanced Settings*

Update

For updating the FPGA firmware

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

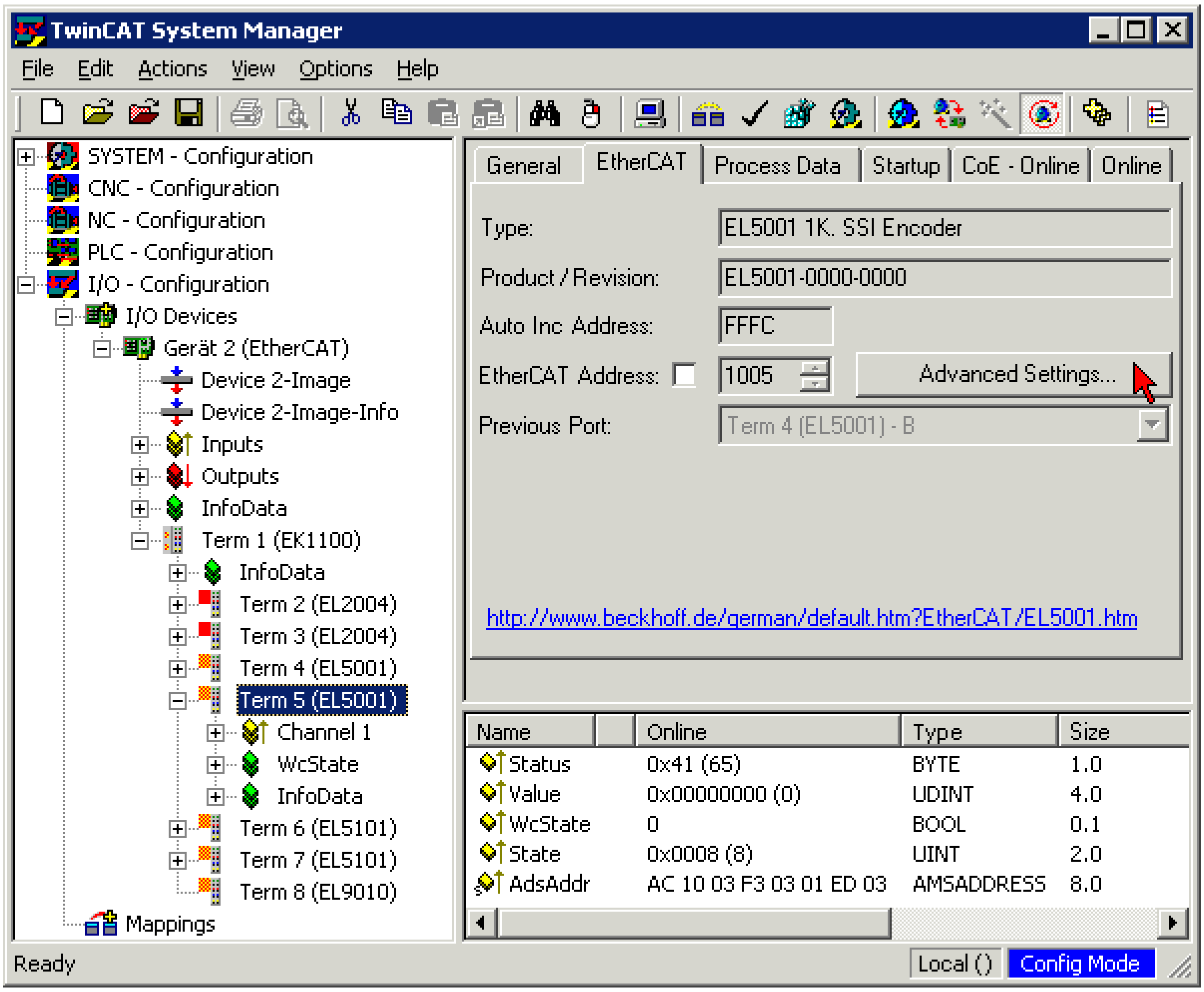
Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

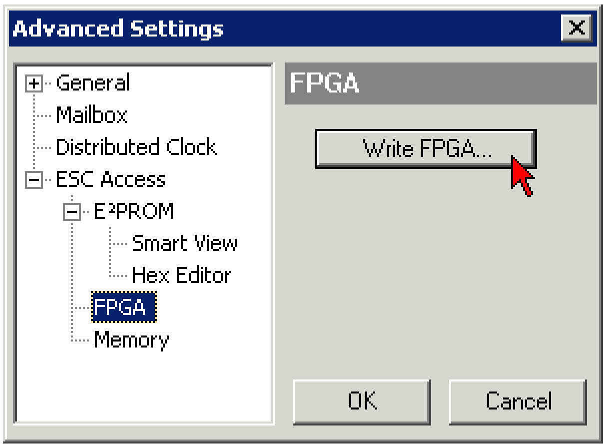
The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

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

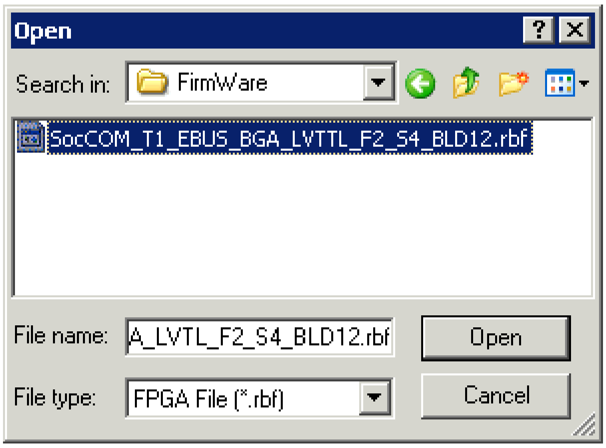
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and
click the *Advanced Settings* button in the *EtherCAT* tab:



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



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



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

NOTE

Risk of damage to the device!

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

6.2.5 Simultaneous updating of several EtherCAT devices

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

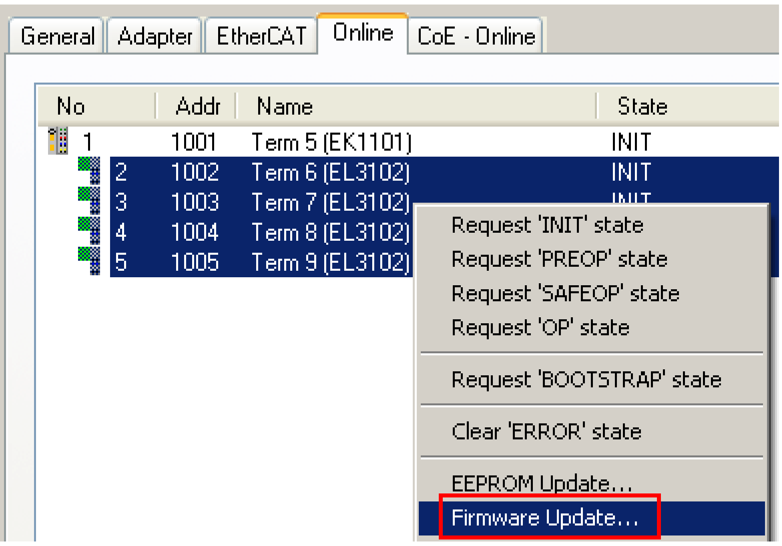


Fig. 183: Multiple selection and firmware update

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

6.3 Restoring the delivery state

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

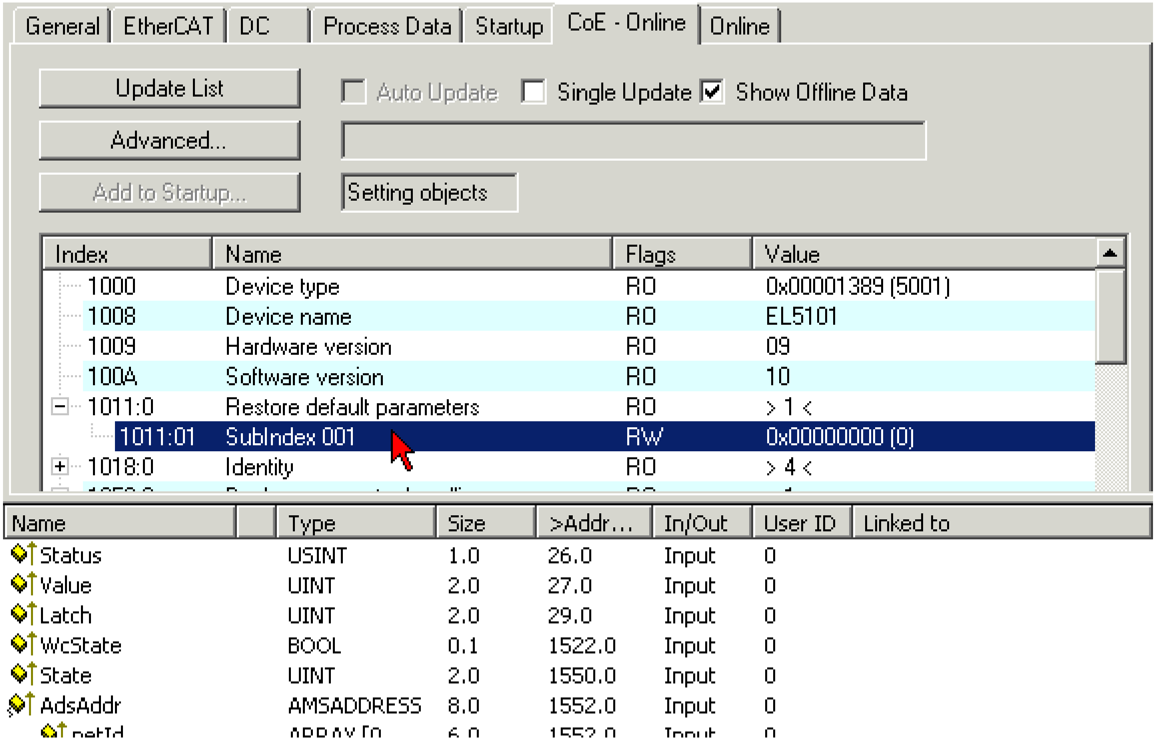


Fig. 184: Selecting the *Restore default parameters* PDO

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

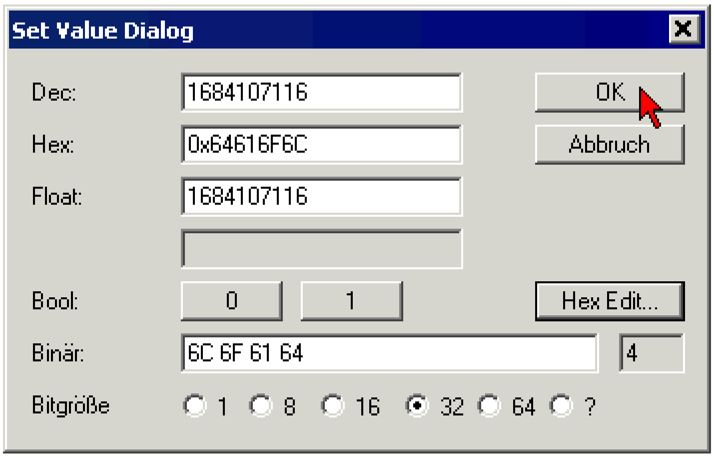


Fig. 185: Entering a restore value in the Set Value dialog

Alternative restore value

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

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:
http://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

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

Beckhoff Support

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

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

Hotline:	+49 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

List of Illustrations

More Information:
www.beckhoff.com/english/ethercat/overview.htm

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