BECKHOFF New Automation Technology

Documentation | EN

EL2596, EL2596-0010

1 Channel LED Strobe Control Terminals

2023-05-05 | Version: 1.3

Table of contents

1 Foreword

1.1 Notes on the documentation

Intended audience

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

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

Disclaimer

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

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

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

Trademarks

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

Patent Pending

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

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.
The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.
Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

1.2 Guide through documentation

NOTE

Further components of documentation

This documentation describes device-specific content. It is part of the modular documentation concept for Beckhoff I/O components. For the use and safe operation of the device / devices described in this documentation, additional cross-product descriptions are required, which can be found in the following table.

Title	Description
EtherCAT System Documentation (PDF)	• System overview • EtherCAT basics • Cable redundancy • Hot Connect • EtherCAT devices configuration
Infrastructure for EtherCAT/Ethernet (PDF)	Technical recommendations and notes for design, implementation and testing
Software Declarations I/O (PDF)	Open source software declarations for Beckhoff I/O components

The documentations can be viewed at and downloaded from the Beckhoff website (www.beckhoff.com) via:

- the “Documentation and Download” area of the respective product page,
- the Download finder,
- the Beckhoff Information System.

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
1.3	• Update chapter “Foreword” • Update structure
1.2	• Update chapter “Technical data” • Update chapter “Mounting and wiring” • Update chapters “Notes on commissioning”, “Current control” and “Determination of the output parameters” • Update revision status • Update structure
1.1	• Addenda, EL2596-0010
1.0	• 1st public issue
0.2 - 0.9	• Complements, corrections
0.1	• provisional documentation for EL2596-xxxx

1.5 Version identification of EtherCAT devices

1.5.1 General notes on marking

Designation

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

- family key
- type
- version
- revision

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

Notes

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

1.5.2 Version identification of EL terminals

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

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

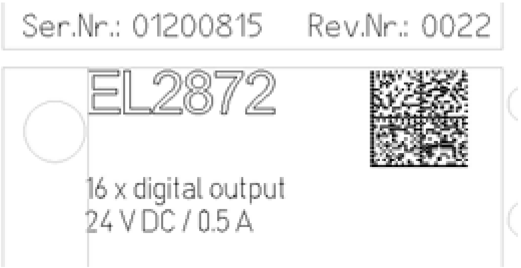


Fig. 1: EL2872 with revision 0022 and serial number 01200815

1.5.3 Beckhoff Identification Code (BIC)

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

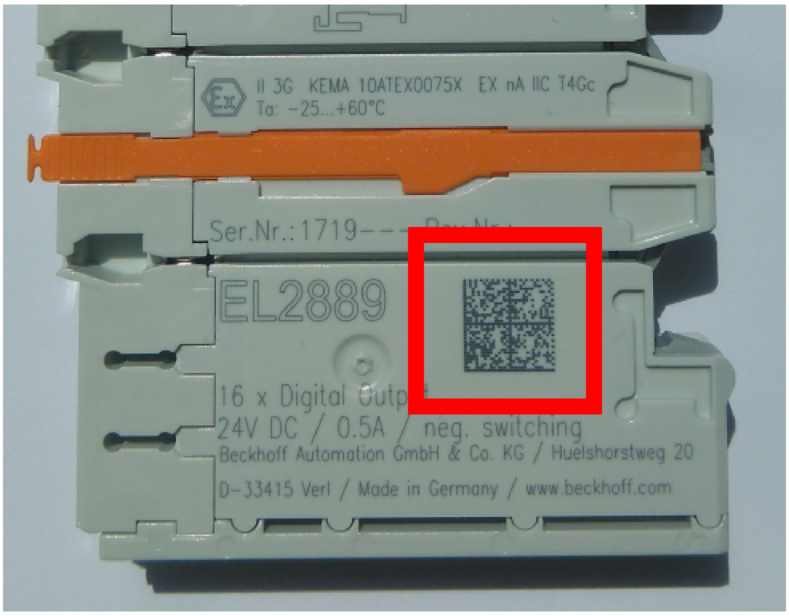

II 3G KEMA 10ATEX0075X EX nA IIC T4Gc
Ta: -25...+60°C
Ser.Nr.:1719---
EL2889
16 x Digital Output
24V DC / 0.5A / neg. switching
Beckhoff Automation GmbH & Co. KG / Huelshorstweg 20
D-33415 Verl / Made in Germany / www.beckhoff.com

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

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

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

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

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

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

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

Posi-tion	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	**Beckhoff order number**	1P	8	1P072222
2	Beckhoff Traceability Number (BTN**)**	**Unique serial number, see note below**	SBTN	12	SBTNk4p562d7
3	Article description	**Beckhoff article description, e.g. EL1008**	1K	32	1KEL1809
4	Quantity	**Quantity in packaging unit, e.g. 1, 10, etc.**	Q	6	Q1
5	Batch number	Optional: Year and week of production	2P	14	2P401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	51S678294
7	Variant number	Optional: Product variant number on the basis of standard products	30P	32	30PF971, 2*K183
...					

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

Structure of the BIC

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

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

Accordingly as DMC:

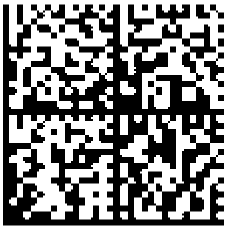


Fig. 3: Example DMC **1P**072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

BTN

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

NOTE

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

1.5.4 Electronic access to the BIC (eBIC)

Electronic BIC (eBIC)

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

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

K-bus devices (IP20, IP67)

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

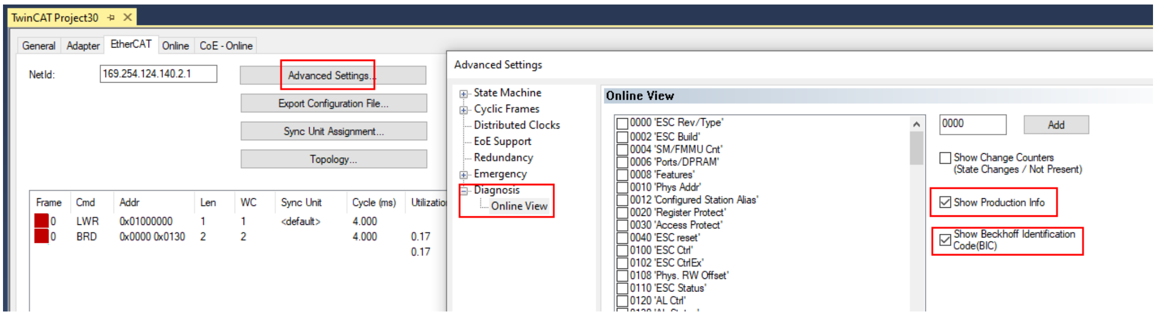
EtherCAT devices (IP20, IP67)

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

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

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

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



NetId: 169.254.124.140.2.1

Frame	Cmd	Addr	Len	WC	Sync Unit	Cycle (ms)	Utilizatio
0	LWR	0x01000000	1	1	<default>	4.000	
0	BRD	0x0000 0x0130	2	2		4.000	0.17
							0.17

Online View

□ 0000 'ESC Rev/Type'
□ 0002 'ESC Build'
□ 0004 'SM/FMMU Cnt'
□ 0006 'Ports/DPRAM'
□ 0008 'Features'
□ 0010 'Phys Addr'
□ 0012 'Configured Station Alias'
□ 0020 'Register Protect'
□ 0030 'Access Protect'
□ 0040 'ESC reset'
□ 0100 'ESC Ctrl'
□ 0102 'ESC CtrlEx'
□ 0108 'Phys. RW Offset'
□ 0110 'ESC Status'
□ 0120 'AL Ctrl'

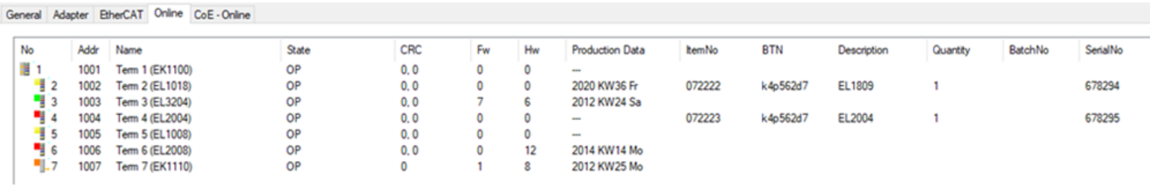
0000

□ Show Change Counters (State Changes / Not Present)

☑ Show Production Info

☑ Show Beckhoff Identification Code(BIC)

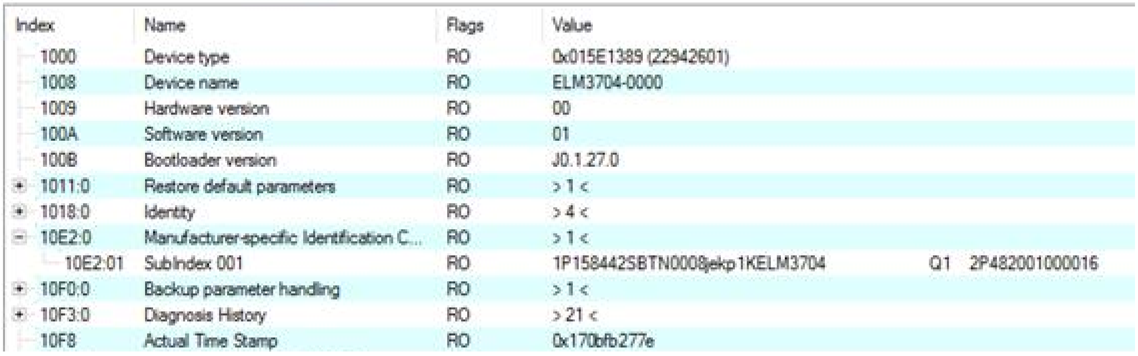
- The BTN and its contents are then displayed:



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

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

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



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

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

Profibus/Profinet/DeviceNet… Devices

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

2 Product overview

2.1 Introduction

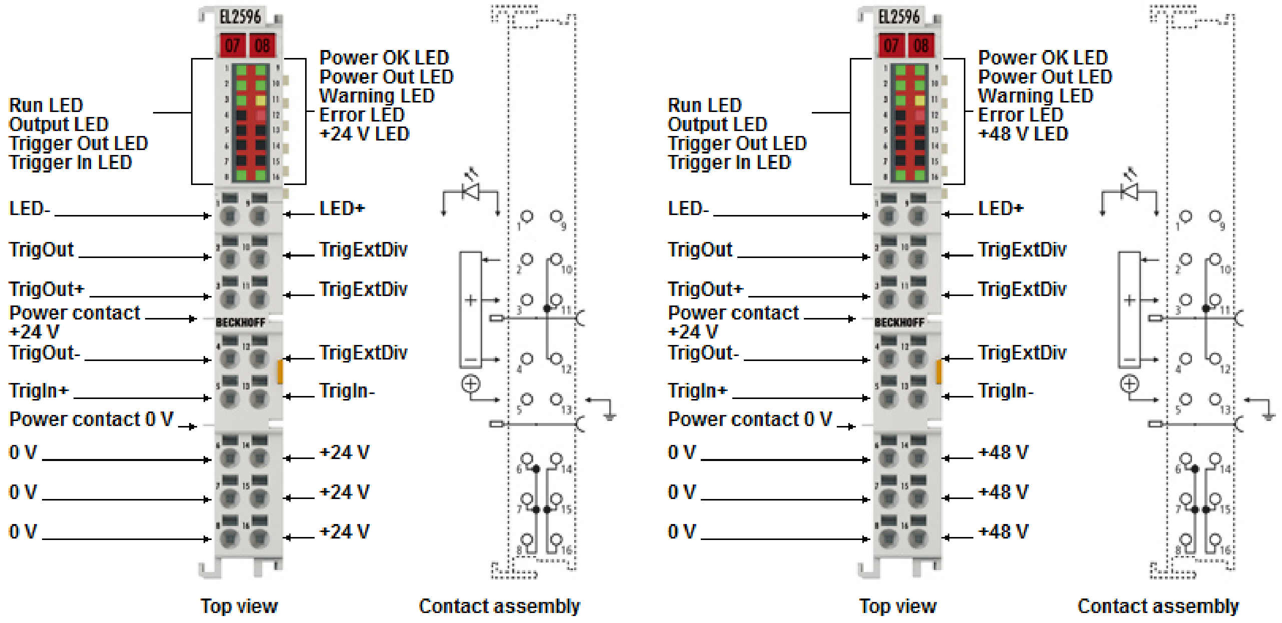


Fig. 4: 1-channel LED strobe control terminals EL2596 (24 V), EL2596-0010 (48 V)

1-channel LED strobe control terminals

The EL2596-xxxx EtherCAT terminals contain an adjustable power supply source for LEDs and are designed for controlling one or more LEDs connected in series in continuous light or pulse mode. The user specifies the desired set current. The integrated power supply unit then provides the necessary forward voltage depending on the connected LEDs. For this purpose, the EL2596-xxxx terminals have a step-down power supply unit that provides the desired output voltage. LEDs with a forward voltage up to just below the EL2596 input voltage can be operated. The high-quality and fast current and voltage control also enables constant illumination of line scan cameras.

The current can be switched quickly for short-term lighting, so that even the shortest flashes of light are possible. As with the EL2252, the flash time itself can be set by timestamp via distributed clock. The operation of the LED with a short-term overdriven current, so-called “overdriving”, is also possible as long as the output voltage remains below the supply voltage.

The EL2596-xxxx features extensive trigger options. Trigger input and output can be used at the same time:

- Trigger output to trigger cameras
- Trigger input (parameterizable) to be triggered by a camera or sensor

The integration of the LED controller EL2596-xxxx into the EtherCAT system allows

- single flash control from the controller with regard to the start and end of each individual light flash up to the kHz range
- the changing of current/voltage parameters during operation
- intensive diagnosis of terminal and LED such as input current/voltage, output current/voltage and terminal temperature: If a parameterizable load corridor is exited, e.g. due to a load error, the terminal switches itself off after a warning to protect the load. This can be reset.

For operation of the 48 V variant EL2596-0010, a ZB8610 fan cartridge is mandatory to ensure that the terminal does not exceed the maximum allowed internal temperature.

Quick links

- Technical data [▶ 18]
- Commissioning [▶ 129]
- Process data [▶ 194]
- Object description and parameterization [▶ 216]
- EtherCAT basics

2.2 Technical data

Device functions	EL2596	EL2596-0010
Application recommendation	Standard terminal for the vision application on lighting up to 24 V_{DC}	Standard terminal for the vision application on lighting up to 48 V_{DC}
Connection technology	2-wire	
Number of outputs	1	
Input voltage U_{IN}	24 V_{DC} (-15 %/+20 %) Use only a regulated power supply unit for the supply of power	48 V_{DC} (-15 %/+20 %) Use only a regulated power supply unit for the supply of power
Load type	LED (ohmic), potential-free	
Special features	Operation modes: Constant voltage, constant current, PWM; extensive real-time diagnostics; voltage divider connection option, TriggerOut; LED continuous operation possible, operation of a multi-color Common Anode LED	

LED output	EL2596	EL2596-0010
Output voltage U_{OUT}	Continuous light: 0 ... (U_{IN} - 0.5 V) Pulsed mode (0 ... 2 A): 0 ... (U_{IN} – 2 V) Pulsed mode (3 A*): 0 ... (U_{IN} – 3 V) *linear behavior of the maximum output voltage between 2 A and 3 A	
Output current I_{OUT}	In pulse mode: 0 mA ... 3 A (depending on output voltage and duty cycle) In continuous light mode: 0 mA …1.2 A For EL2596-0010 only for operation with fan (ZB8610).	
Output power P_{Out}	max. 14.4 W	
Control accuracy of output current I_{OUT}	< ±3 mA in Current Control and Current Control PWM mode corresponds to < ±0.1% of the output end value 3 A	
Pulse duration at the LED output	> 25 µs .. 10 s (lower values on request)	
Switching times	T_{ON}: < 1 µs typ., T_{OFF}: < 1 µs typ.	

Trigger input (from camera)	EL2596	EL2596-0010
Number	1	
Type	Electrically isolated, 3 mA typ., 0 … 30 V_{DC}, Sensitivity of the trigger level can be switched in the CoE: High = 10 V, Low = 5 V (from HW02: 4 V)	
Maximum input frequency at the trigger input	20 kHz	
Input filter of trigger input	5 µs	
Blind time of the trigger input	0 µs (< FW04) or 20 µs…65535 µs (from FW04)	

Trigger output (to the camera)	EL2596	EL2596-0010
Number	1	
Type	Electrically isolated, max. 10 mA push-pull, 10 … 24 V_{DC} (voltage can be changed by externally connectable voltage divider [▶ 180])	
Pulse duration at trigger output	25 µs ... 10 s	
Pulse delay in relation to the trigger input	2 µs … 10 s	

Supply and potentials	EL2596	EL2596-0010
Power supply for the electronics	via the E-bus	
Current consumption via E-bus	typ. 240 mA	typ. 265 mA
Electrical isolation	500 V (E-bus/field voltage)	
Current consumption power contacts	- (Power contacts are only passed through)	

Communication	EL2596	EL2596-0010
Configuration	via TwinCAT System Manager	
Distributed clocks	yes	
Accuracy Distributed Clocks	<< 1 µs	
min. cycle time in distributed clocks mode	200 µs	

Environmental conditions	EL2596	EL2596-0010
Permissible ambient temperature range during operation	0°C ... + 55°C	
Permissible ambient temperature range during storage	-25°C ... + 85°C	
Permissible relative air humidity	95 %, no condensation	

General data	EL2596	EL2596-0010
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)	
Weight	approx. 57 g	
Mounting	on 35 mm mounting rail according to EN 60715	
Installation position	see Prescribed installation position [▶ 52]	

Standards and approvals	EL2596	EL2596-0010
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27	
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4	
Protection class	IP20	
Approvals/markings*)	CE, EAC, UKCA	

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

2.3 LEDs and connection

2.3.1 EL2596

EL2596 - Connection

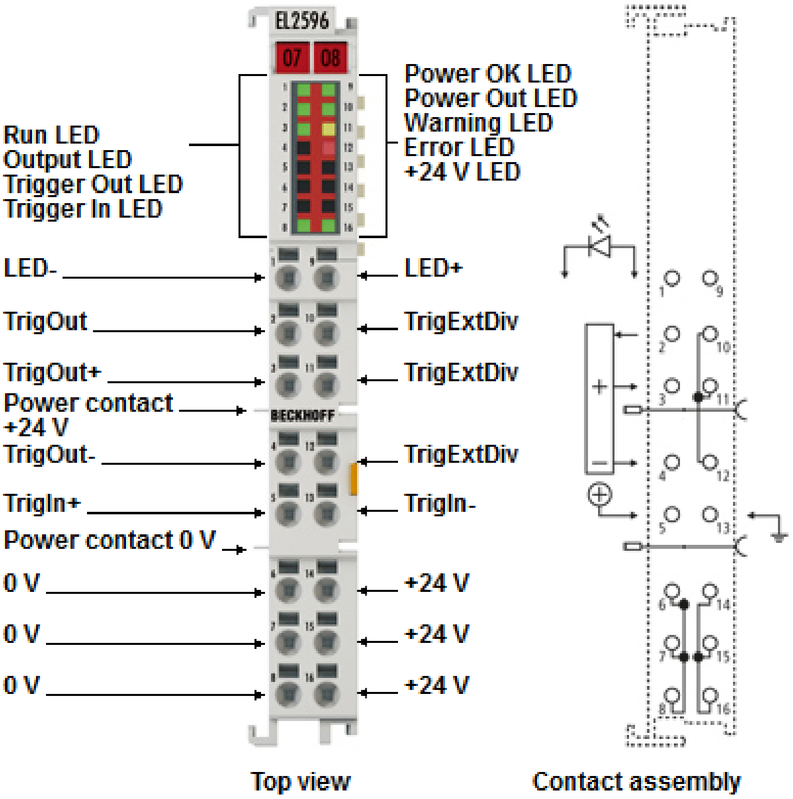


Fig. 5: EL2596 - connection

Terminal point	No.	Comment
LED-	1	- Output for the LED
TrigOut	2	Trigger output signal
TrigOut+	3	Connection for the + voltage supply for the trigger output
TrigOut-	4	Connection for the - voltage supply for the trigger output
TrigIn+	5	+ trigger input
0 V	6	0 V input voltage (no supply via power contacts)
0 V	7	0 V input voltage (no supply via power contacts)
0 V	8	0 V input voltage (no supply via power contacts)
LED+	9	+ output for the LED
TrigExtDiv	10	Connection option for a voltage divider for adjusting the output voltage for the trigger output
TrigExtDiv	11	Connection option for a voltage divider for adjusting the output voltage for the trigger output
TrigExtDiv	12	Connection option for a voltage divider for adjusting the output voltage for the trigger output
TrigIn-	13	- trigger input
+24 V	14	24 V input voltage (no supply via power contacts)
+24 V	15	24 V input voltage (no supply via power contacts)
+24 V	16	24 V input voltage (no supply via power contacts)

EL2596 - LEDs

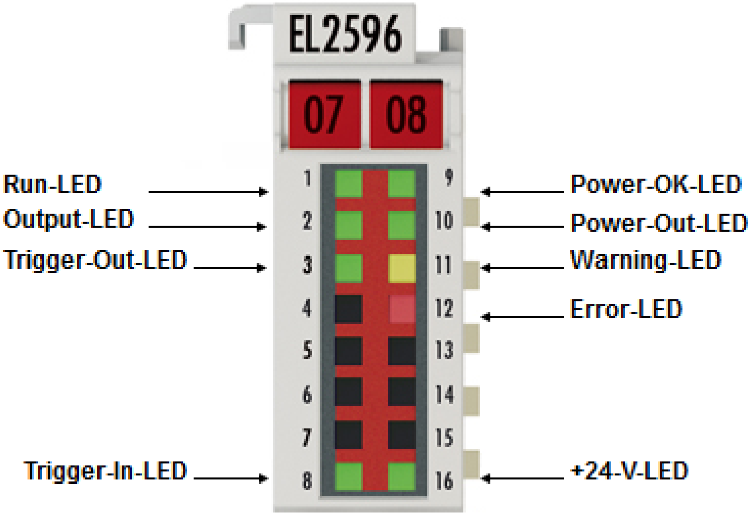


Fig. 6: EL2596 - LEDs

LED	Color	Meaning	
RUN	green	This LED indicates the terminal's operating state:	
		off	State of the EtherCAT State Machine: **INIT** = initialization of the terminal
		flashing	State of the EtherCAT State Machine: **PREOP =** function for mailbox communication and different default settings set
		Single flash	State of the EtherCAT State Machine: **SAFEOP** = verification of the sync manager channels and the distributed clocks. Outputs remain in safe state
		on	State of the EtherCAT State Machine: **OP =** normal operating state; mailbox and process data communication is possible
		flickering	State of the EtherCAT State Machine: **BOOTSTRAP** = function for terminal firmware updates
OUTPUT	green	The LED output is active.	
TRIGGER OUT	green	The trigger output is active.	
TRIGGER IN	green	The trigger input is wired and active.	
POWER OK	green	The input voltage is within the configured range.	
POWER OUT	green	The LED drive circuit is ready to operate.	
WARNING	yellow	off	no defect
		on	• Input voltage is outside the configured range • Output voltage is outside the configured range • Internal temperature of 80 °C exceeded • ...
ERROR	red	off	no defect
		on	• Configuration error, e.g.: • no input voltage connected • Internal temperature of 100 °C exceeded • Short circuit • ...
+24 V	green	Input voltage > 10 V is electrically applied, without further diagnostics	

2.3.2 EL2596-0010

EL2596-0010 - connection

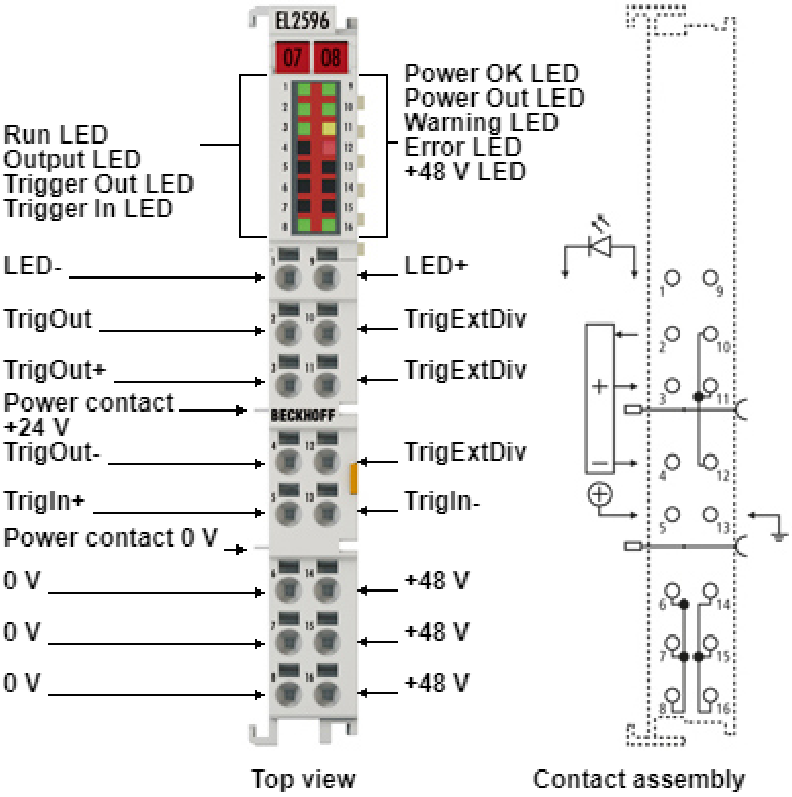


Fig. 7: EL2596-0010 - connection

Terminal point	No.	Comment
LED-	1	- Output for the LED
TrigOut	2	Trigger output signal
TrigOut+	3	Connection for the + voltage supply for the trigger output
TrigOut-	4	Connection for the - voltage supply for the trigger output
TrigIn+	5	+ trigger input
0 V	6	0 V input voltage (no supply via power contacts)
0 V	7	0 V input voltage (no supply via power contacts)
0 V	8	0 V input voltage (no supply via power contacts)
LED+	9	+ output for the LED
TrigExtDiv	10	Connection option for a voltage divider for adjusting the output voltage for the trigger output
TrigExtDiv	11	Connection option for a voltage divider for adjusting the output voltage for the trigger output
TrigExtDiv	12	Connection option for a voltage divider for adjusting the output voltage for the trigger output
TrigIn-	13	- trigger input
+48 V	14	48 V input voltage (no supply via power contacts)
+48 V	15	48 V input voltage (no supply via power contacts)
+48 V	16	48 V input voltage (no supply via power contacts)

EL2596-0010 - LEDs

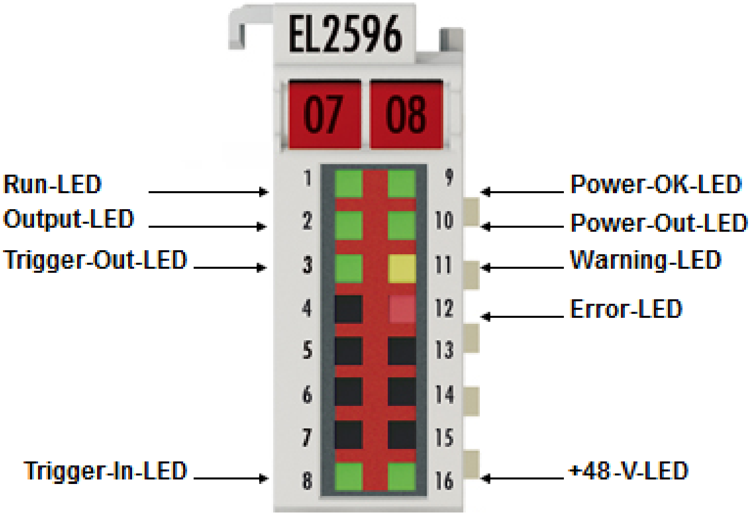


Fig. 8: EL2596-0010 - LEDs

LED	Color	Meaning	
RUN	green	This LED indicates the terminal's operating state:	
		off	State of the EtherCAT State Machine: **INIT** = initialization of the terminal
		flashing	State of the EtherCAT State Machine: **PREOP =** function for mailbox communication and different default settings set
		Single flash	State of the EtherCAT State Machine: **SAFEOP** = verification of the sync manager channels and the distributed clocks. Outputs remain in safe state
		on	State of the EtherCAT State Machine: **OP =** normal operating state; mailbox and process data communication is possible
		flickering	State of the EtherCAT State Machine: **BOOTSTRAP** = function for terminal firmware updates
OUTPUT	green	The LED output is active.	
TRIGGER OUT	green	The trigger output is active.	
TRIGGER IN	green	The trigger input is wired and active.	
POWER OK	green	The input voltage is within the configured range.	
POWER OUT	green	The LED drive circuit is ready to operate.	
WARNING	yellow	off	no defect
		on	• Input voltage is outside the configured range • Output voltage is outside the configured range • Internal temperature of 80 °C exceeded • ...
ERROR	red	off	no defect
		on	• Configuration error, e.g.: • no input voltage connected • Internal temperature of 100 °C exceeded • Short circuit • ...
+48 V	green	Input voltage is within the specified range	

2.4 Basics of LED technology

Some basic information on light emitting diode technology (LED) is given in the following. This information is of a basic nature; therefore, please check the extent to which it applies to your application.

Beckhoff offer several products for the control of LEDs in various operating modes, including

- EL2595, EL2596: single-channel power source as an EtherCAT Terminal, also for flashing operation
- EL2564: PWM terminal for voltage operation as an EtherCAT Terminal

For others, see www.beckhoff.de

2.4.1 Definition

An LED (**l**ight-**e**mitting **d**iode) converts electrical energy into light. An LED consists of a semiconductor p-n junction. Like a conventional semiconductor diode, an LED is forward-biased and reverse-biased. When applying a voltage in the forward direction, surplus electrons in the semiconductor recombine with the electron holes. The LED becomes conductive and energy is given off in the form of light. The energy and thus the light color depend on the semiconducting material used.

The electrical circuit symbol for an LED shows a diode with two arrows. These arrows symbolize the emitted light.

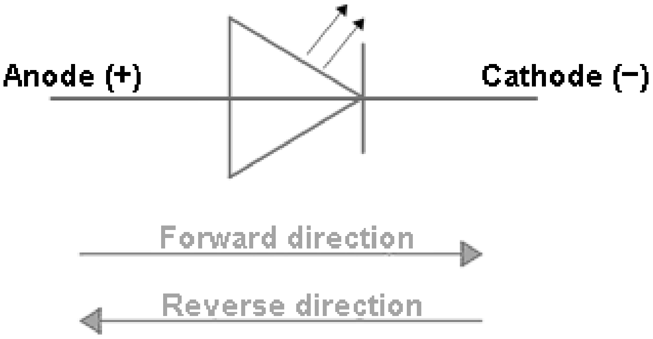


Fig. 9: Circuit symbol for an LED

2.4.2 Structure

Simple standard LEDs usually consist of an LED chip, a reflector with contact to the cathode (-), a gold wire as contact to the anode (+) and a plastic lens.

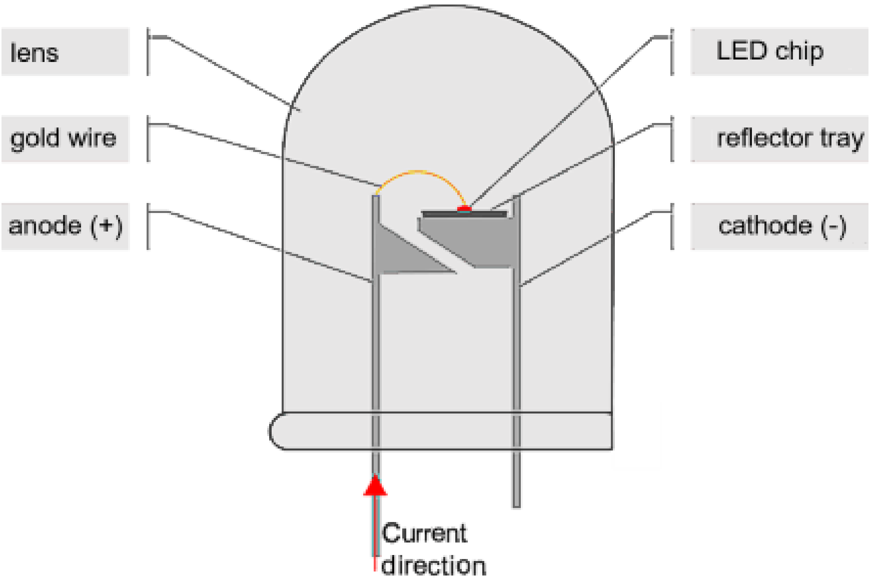


Fig. 10: Classic structure of a single-color LED

However, the structure shown above is just an example. In addition to the structure shown, there are also LEDs in high-power or SMD versions, for example.

The LED chip consists in principle of two layers. One layer has surplus electrons (n-type doping). The second, p-doped layer on the other hand has an electron deficiency; a majority of electron holes exists. This different charge distribution is achieved through the purposeful contamination (doping) of the pure semiconducting material, where other atoms such as boron or silicon are added to the semiconducting material. The illustration below shows this simplified structure of an LED chip. The emitted light is only directed by a reflector or a lens.

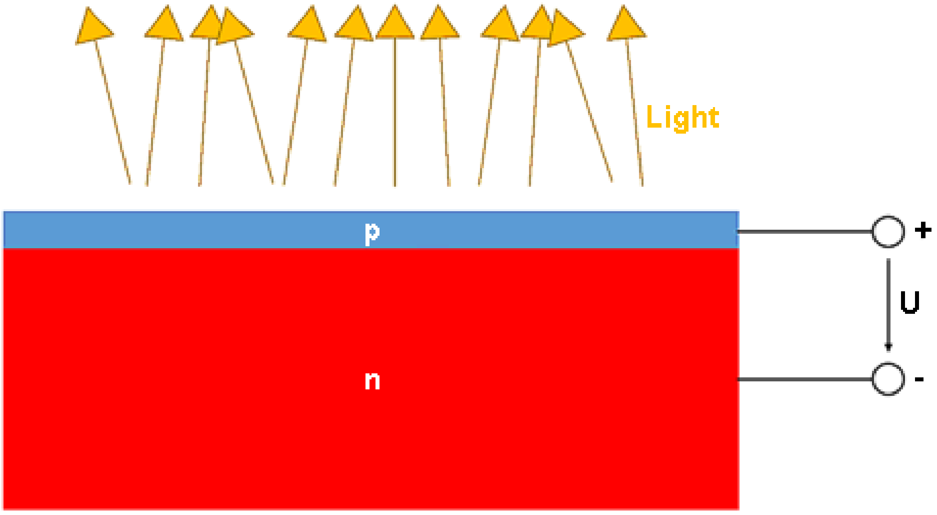


Fig. 11: Example of the structure of an LED chip

2.4.3 Properties

Forward current I_F [mA]

The forward current of an LED is the current flowing through the LED in forward direction from the anode (+) to the cathode (-). For the maximum forward current, a distinction can be made between the maximum current in continuous light mode and in pulse mode. The maximum forward current is usually higher in pulse mode than in continuous light mode.

Nominal current I_N [mA]

If the LED is operated with a forward current equal to the nominal current, the LED has the characteristics specified in its data sheet, e.g. the nominal brightness. Operation with I_F greater than I_N reduces the service life of the LED due to increased heat generation. Common nominal currents for LEDs are 20 mA, 350 mA and 1000 mA.

Conducting voltage U_D [V]

The conducting voltage indicates the level of electrical voltage required for the LED to become conductive. When the conducting voltage is applied between the anode (+) and the cathode (-), a current flows through the LED in forward direction. The conducting voltage level of an LED depends on the semiconducting material. Typical values for the conducting voltage of various LEDs are, for example, 1.6 V for red and 2.6 V for blue emitting LEDs (see Colors [▸ 32]).

Forward voltage U_F [V]

The forward voltage of an LED is the voltage applied in the forward direction between the anode (+) and the cathode (-). The forward voltage is a function of the forward current $U_F = f(I_F)$. This dependence is strongly non-linear. The relationship between U_F and I_F is shown as an example in the chapter Characteristic curve [▸ 27].

Reverse voltage U_R [V]

The reverse voltage is the electrical voltage applied to the LED in reverse direction. Data sheets usually indicate the maximum reverse voltage. This maximum reverse voltage must not be exceeded, otherwise the LED can be irreversibly damaged. A typical value for the reverse voltage of an LED is 5 V.

Typ. wavelength λ [nm]

The typical wavelength is the wavelength of the emitted light at the nominal current.

2.4.4 Characteristic curve

The characteristic curve of an LED is strongly non-linear. An LED is non-conductive if no external voltage is applied. The LED starts to conduct when the applied forward voltage U_F is at least as high than the conducting voltage U_D and the band gap is overcome by the electrons. The forward current is not proportional to the applied forward voltage. A small change in voltage can cause a large change in current. A small voltage change can lead to a strong change in light emission due to the proportionality of luminous flux and current intensity. This means that LEDs must generally be operated with a current limiter of some form or other, otherwise even slight fluctuations in the applied voltage can destroy the LED.

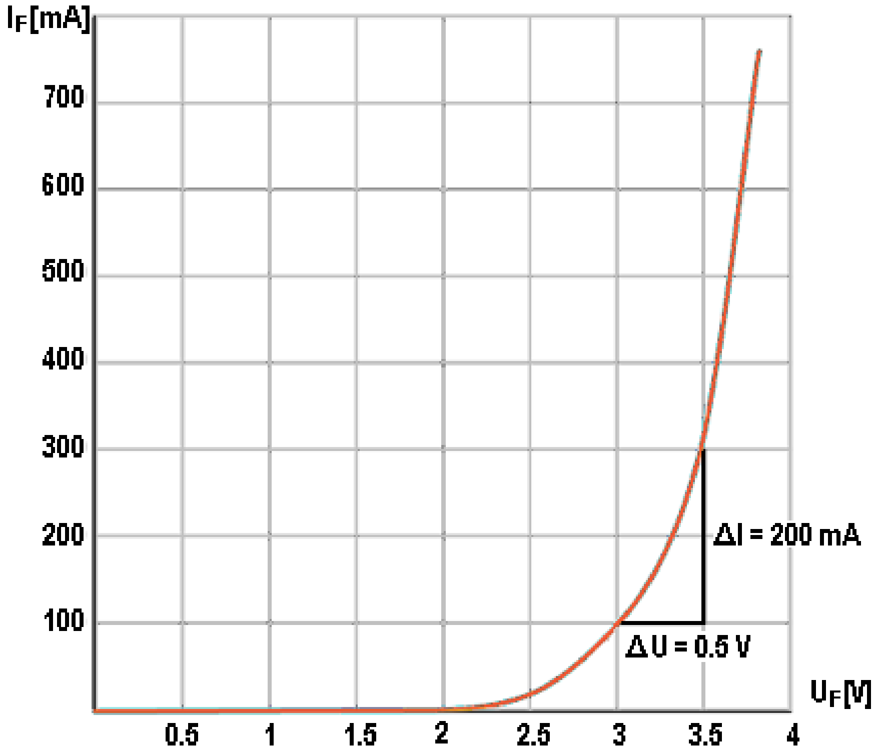


Fig. 12: Example characteristic curve of an LED

A gradient triangle is drawn on the example characteristic curve. On the basis of this gradient triangle it can be seen that a small change in the voltage of 0.5 V from 3 V to 3.5 V results in a large change in the current intensity of 200 mA from 100 mA to 300 mA. In this example case, a voltage change of less than 17 % results in a current change of 300 %.

This example shows that small voltage fluctuations lead to large changes in the current through the LED and thus to large changes in the luminous flux.

2.4.5 Control

There are two common types of control of LEDs: Current-controlled without series resistor and voltage-controlled with a series resistor. Each control method has advantages and disadvantages for certain use cases, which are explained below. Depending on the use case, a decision must be made as to which method of control is to be used.

1. Voltage mode

Voltage mode, e.g. with a battery or a power supply unit, is a simple and cost-effective way of controlling LEDs. All that is needed is an additional series resistor R_V. Due to the linear behavior of an ohmic resistance, R_V makes the overall circuit much less sensitive to voltage changes, resulting in robust LED control. Due to the ohmic resistance, however, the power loss of the control increases and is given off in the form of heat ($P_V = R_V \cdot I_{LED}^2$).

The current I_{LED} through the LED results from the ratio $I_{LED} = U/R_V$.

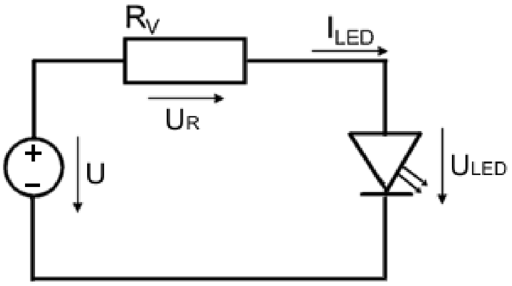


Fig. 13: Voltage-controlled LED with series resistor

The series resistor R_V is calculated as follows:

$R_V = U_R / I_{LED}$

The current through the LED is known. The voltage U_R, which is dropped across the series resistor to be calculated, is missing. This voltage is formed from the operating voltage minus the voltage dropped across the LED. The voltage U_{LED} that is dropped across the LED at I_{LED} can be read from the U/I characteristic curve for the LED in the data sheet.

$U_R = U - U_{LED}$

If the brightness of an LED with a series resistor is to be adjusted, the applied voltage must be reduced (darker) or increased (brighter).

However, the disadvantage of this type of control is that the luminous flux cannot be controlled precisely. As described at the beginning, a small change in voltage leads to a big change in current and thus to a big change in the luminous flux. In the case of voltage control, therefore, fluctuations in the supply voltage may have a direct influence on the luminous flux of the LED. It should also be borne in mind that the electrical characteristics of the resistor and the LED are temperature dependent.

- **Advantages:** simple design, simple control, the brightness of the LED can be set directly via the voltage
- **Disadvantages:** additional resistance, resulting in waste heat

2. Current mode

An LED can be operated directly if a power source (electronic circuit) is used instead of a voltage source (e.g. battery). With current control, the luminous flux of the LED can be adjusted directly via the specified current value, without resistance. Fluctuations in the supply voltage thus have no influence on the luminous flux of the LED. The luminous flux is constant and reproducible with current control. Current control is thus recommended, for example, in machine vision applications.

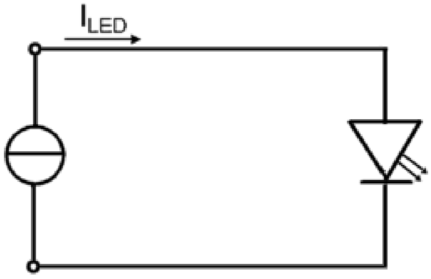


Fig. 14: Current-operated LED

- **Advantages:** no additional components; the brightness of the LED can be adjusted directly via the current
- **Disadvantages:** complex power source may be required

Glowing LED due to a leakage current

Even when the LED is switched off, a small current flow through the LED may occur, depending on the control circuit. Due to this leakage current, the LED glows noticeably in some cases.

Further information about the control of LEDs with Beckhoff components can also be found in the corresponding Application Note.

https://download.beckhoff.com/download/document/Application_Notes/DK9222-0620-0065.pdf

2.4.6 Operation modes

There are two operation modes for LEDs: continuous operation and pulse operation. Each operation mode has advantages and disadvantages, therefore a decision has to be made about which mode to use depending on the use case.

1. Continuous operation

An LED circuit can be designed for continuous operation. The LED is then switched on continuously. In this operation mode the current through the LED may not exceed the nominal current.

- **Advantages:** Simpler and cheaper circuit
- **Disadvantages:** Only a small part of the maximum possible luminous flux of the LED can be used. Continuous operation generates higher waste heat, leading to faster aging of the LED.

Continuous operation can take place in various ways:

a. Current and voltage output

The continuous switching on of a voltage or current (depending on the selected control method) leads to a continuous light. The description, as well as the advantages and disadvantages of the two control methods can be found in the chapter Control [▶ 28].

b. Pulse width modulation (PWM)

If the constant current or constant voltage with series resistor is clocked quickly in the kHz range, this is referred to as a PWM mode. The true-color brightness can then be adjusted by adapting the duty factor of the pulse width modulation (PWM). By switching the power supply on and off with a sufficiently high frequency and a preset duty cycle (0...100 %), the flashing appears to the human eye like a continuous light. By changing the duty cycle, the current averaged over time is reduced or increased by the LED, thus adjusting the brightness.

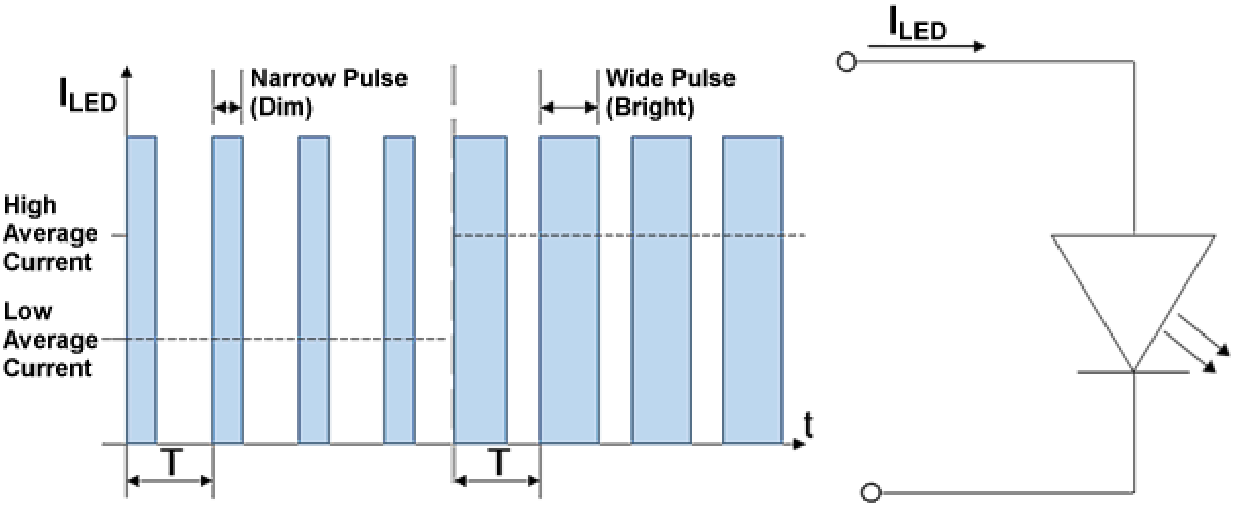


Fig. 15: Control of an LED with PWM

- **Advantages:** true-color brightness adjustment
- **Disadvantages:** supply must be able to provide rapidly increasing currents, complex supply source may be required

2. Pulse operation

In some applications an “overdrive” may be desired, as the light power in continuous operation with the nominal current is insufficient. The LED is operated briefly for a few µs to ms with a considerably higher power than in nominal operation by means of a brief and pulsating increase in the current above the nominal current. This results in briefly higher luminous fluxes. The LED can cool down again in the subsequent pause.

Overdriving leads to increased development of heat in the LEDs. The temperature of the LED chip must not increase beyond the temperature limit value during the pulse. The LED will otherwise be damaged. Following a pulse, there must be a sufficiently long pause (switch-off time) before the next pulse so that the LED can cool down. The ratio of switch-on to switch-off time is set by the duty cycle. A maximum duty cycle of 10% is often set for pulse operation. Hence, the pulse duration may not exceed 10% of the entire period. The precise values are to be taken from the manufacturer's data sheets.

$$\text{Duty Cycle} = \frac{T_{on}}{T_{on} + T_{off}} = \frac{T_{on}}{T} \leq 10\%$$

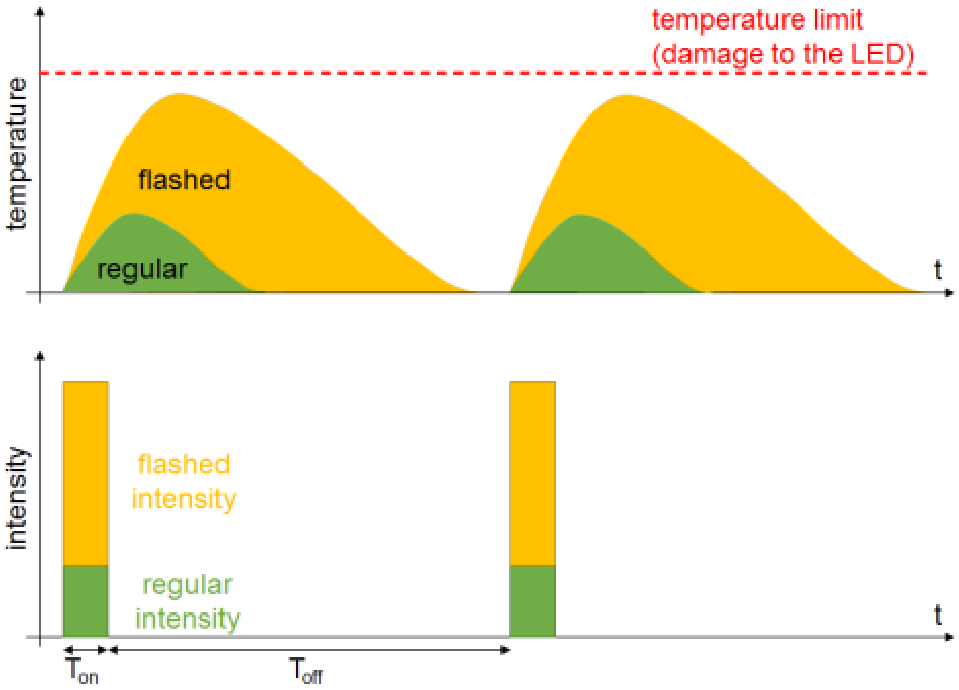


Fig. 16: Temperature and brightness as a function of time in pulse operation

- **Advantages:** The maximum luminous flux of the LED can be utilized. If the duty cycle is adhered to or undershot, the waste heat created is at the most as much as in continuous operation. If the maximum duty cycle is undershot so that the average power is less than in continuous operation, this can lead to less aging and thus to a longer service life.
- **Disadvantages:** Requires a more elaborate circuit and control solution, for example in the form of a flash controller

Here, too, implementation via a voltage or current output is possible. The output values must then be dimensioned in such a way that the output flash reaches the desired brightness. The maximum output power must always be taken into account when dimensioning the flashes.

The output of light pulses is also possible with fast PWM. A light flash with a length of 1 ms is generated with 1 kHz PWM.

2.4.7 Connection of several LEDs together

1. Series connection

The series connection of LEDs is the usual way to increase the illuminance, for example. In a series circuit, the same current flows through all consumers. It therefore makes sense if all the LEDs connected in series have the same color or, better still, are of the same type with the same characteristic values.

With a sufficiently high supply voltage, several LEDs can be connected in series. A single resistor or a current controller is then sufficient. The number of LEDs must be taken into account when calculating the series resistance, as there is a voltage drop U_{LED} across each LED, which then adds up.

2. Parallel connection

The connection of LEDs in parallel should be avoided as the U/I characteristic curve of an LED is not linear, but approximately exponential. Thus, a small change in voltage leads to a large change in current.

If two or more LEDs (with the same nominal conducting voltage) are connected in parallel, the largest current will flow through the LED with the lowest conducting voltage. As a result, this LED will be brighter and thus also warmer than the other parallel LEDs. The conducting voltage decreases as the temperature increases, as a result of which the effect is amplified and the current increases further until destruction.

Since LEDs made of different semiconducting material, i.e. with different colors, have different conducting voltages, the parallel connection of LEDs with different colors is not permitted. There is variance even in the conducting voltage of LEDs with the same color and from the same manufacturing series. When connecting LEDs in parallel, a series resistor/current controller should be used for each individual LED.

3. LED controller for pixel LEDs

The so-called pixel system is an intelligent method of LED control for several LEDs. “Pixel” LEDs are LEDs with an integrated circuit (IC). With an LED matrix or an LED strip, several LEDs are not connected classically in series; instead, each LED can receive individual signals via a bus communication. In this way, each LED can be controlled individually. These LEDs or LED strips require an LED controller, which serially transmits the communication signals with >100 kHz. Each individual LED is then assigned its own pixel controller.

2.4.8 Colors

The color of the light emitted by single-color LEDs can be set through the selection of the semiconducting material. The wavelength range of the light extends from near-infrared through the visible spectrum to the ultraviolet range. The shorter the wavelengths become, the larger the band gap of this semiconductor and the higher the conducting voltage U_D for the operation of the LED.
The following table shows sample values for colors with the associated wavelengths, possibly usable semiconducting materials and associated conducting voltages. This table merely contains example values, therefore characteristic values and materials are not fully applicable and not applicable to every LED.

Color	Wavelength λ in [nm]	Material	Conducting voltage U_D in [V]
Infrared	>760	Gallium arsenide (GaAs) Aluminum gallium arsenide (AlGaAs)	<1.6
Red	610 - 760	Aluminum gallium arsenide (AlGaAs) Gallium arsenide phosphide (GaAsP) Aluminum gallium indium phosphide (AlGaInP) Gallium phosphide (GaP)	1.6 - 1.9
Orange	590 - 610	Gallium arsenide phosphide (GaAsP) Aluminum gallium indium phosphide (AlGaInP) Gallium phosphide (GaP)	1.8 - 2.2
Yellow	570 - 590	Gallium arsenide phosphide (GaAsP) Aluminum gallium indium phosphide (AlGaInP) Gallium phosphide (GaP)	2.0 - 2.4
Green	500 - 570	Indium gallium nitride (InGaN) Gallium nitride (GaN) Gallium phosphide (GaP) Aluminum gallium indium phosphide (AlGaInP) Aluminum gallium phosphide (AlGaP)	2.2 - 2.7
Blue	450 - 500	Zinc selenide (ZnSe) Indium gallium nitride (InGaN) Silicon carbide (SiC)	2.6 - 3.3
Violet	400 - 450	Indium gallium nitride (InGaN)	3.2 - 3.6
Ultraviolet	230 - 400	Aluminum nitride (AlN) Aluminum gallium nitride (AlGaN) Aluminum gallium indium nitride (AlGaInN)	3.5 - 4.2

LEDs can generally only generate light in a small wavelength range with a width of a few tens of nanometers. White light is the sum of all colors or the sum of all wavelengths in the visible range. Therefore, colors must be mixed additively in order to generate white light with an LED. There are various methods of doing this, of which two essential methods are described below.

1. **Combination of different colored LEDs**
   Red, green and blue LEDs (RGB LEDs) can be combined with one another in a housing so that the colors mix in order to generate white light. If the LEDs are controlled appropriately, the light appears to be white. In the RGB combination of LEDs it is also possible through appropriate control of the individual LEDs to generate light of a different color with continuous color transitions.
2. **Luminescence**
   A short-wave LED (blue, violet, ultraviolet) is combined with photoluminescent dye. Photoluminescence describes the emission of light after excitation by light – usually blue or ultraviolet. The dye converts blue, higher-energy light into longer-wave light with a typically larger wavelength range. The dye used significantly influences the color temperature, so that different white tones (Cold White, Warm White) can be generated.

As the duration of use of LEDs increases, the color of the emitted light changes due to aging. These color changes proceed differently with each LED. In LEDs that emit white light by means of a photoluminescent dye, both the LED chip and the dye itself age.

2.4.9 Typical designs of multi-color LEDs

There are generally two types of LEDs, monochrome and multicolor. With monochrome LEDs, it is possible to adjust the brightness via the current in the forward direction, but the color is unchangeable as the LED is made of only one semiconducting material and therefore emits a specific wavelength. The color of the LED is not affected by the control mode. With the multi-colored LEDs, there are different types with different color options. An n-color LED consists of n individual semiconductor PN transitions combined in one housing. The individual LEDs in the multi-color LED consist of the corresponding semiconducting material, which emits the corresponding wavelength. The types RGB (red-green-blue), RGBW (red-green-blue-white) and RGBWW (red-green-blue-white-white) are common. The exact color emitted is determined by the current through the individual semiconductor transitions.

Monochrome LEDs differ in their semiconducting material, resulting in different characteristic values and colors.

The characteristic values and colors also differ with multi-color LEDs. In the case of multi-color LEDs, however, it is additionally necessary to consider how the individual monochrome LEDs are interconnected within the multi-color LED light source. Some of the possible interconnections are illustrated and explained below:

1. Inverse parallel

The “Inverse parallel” interconnection only works with two (differently colored) LEDs. With this interconnection it is possible to create different color mixtures with two LEDs.

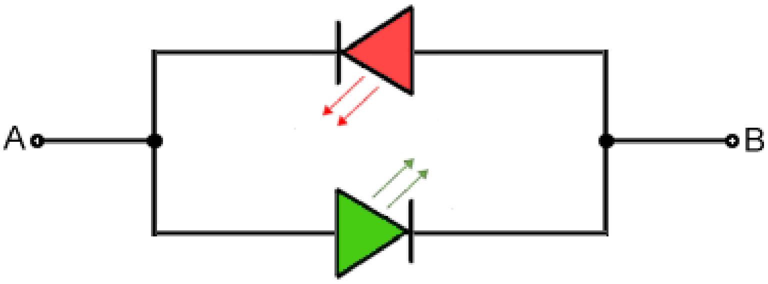


Fig. 17: Inverse parallel LEDs

When the current flows from A to B, a green light is output because the green LED is operated in the forward direction. From B to A, the red LED would light up. Since the different colors have different conducting voltages, each LED requires its own series resistor or current controller. With bidirectional current, the two LEDs would light up alternately. If the current direction changes with a significantly shorter period than the exposure time of a camera, the individual colors of the LED mix to a mixed color. To the human eye, the colors also appear mixed in the case of a quick change.

This type of arrangement of two LEDs is used, for example, to indicate polarity, e.g. for the correct connection of batteries or power supplies.

2. Common anode

The “common anode” interconnection can be used to combine any number of LEDs. This method is common with many RGB/RGBW LEDs. In addition to switching the individual LEDs on and off, only a low current may flow through some of them. This makes any color mixture possible.

All LEDs have a common positive potential at the anode (+). In order to operate an LED in the forward direction, a lower potential must be applied to the cathode connection of the desired color than to the anode connection. If the potential at the cathode (-) is higher, the LED is operated in the reverse direction. Caution: LEDs often have very low reverse voltages of only a few volts!

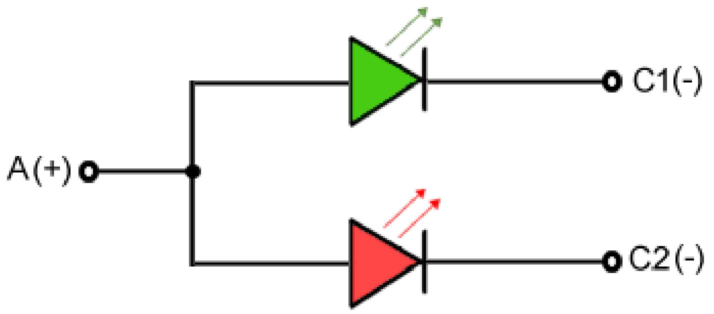


Fig. 18: Common anode LEDs

3. Common cathode

The operation of LEDs with a common cathode (-) is similar to that with a common anode (see “Common anode”). This method is used less often than common anode. Here, too, any number of LEDs can be combined in different colors. Any color can be generated by switching the LEDs on and off differently.

With the “common cathode” interconnection, all LEDs have a common negative potential. In order to switch on an LED, a higher potential must be applied to its anode (+) than to the cathode (-).

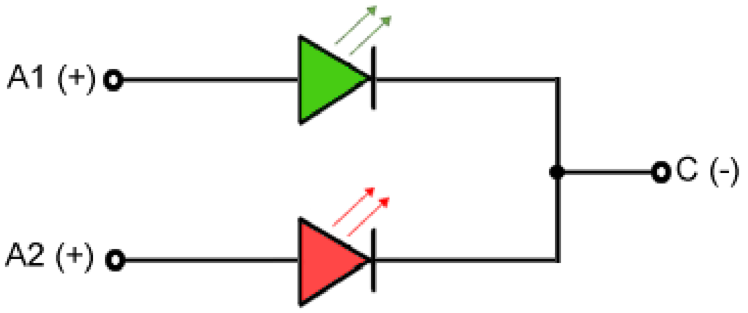


Fig. 19: Common cathode LEDs

2.4.10 Temperature and aging

As with all semiconductors, the properties of an LED are temperature-dependent. Typical changes occur mainly in the luminosity, the wavelength of the emitted light and the conducting voltage.

1. **Luminous flux**
   An increasing temperature in the LED chip leads to a reduction in the luminous flux.
2. **Wavelength λ**
   An increasing temperature in the LED chip leads to an increase in the wavelength (the extent depends on the semiconducting material)
3. **Conducting voltage U_D**
   An increasing temperature in the LED chip leads to a reduction in the conducting voltage (2 mV/°C). In contrast, the conducting voltage increases at low temperatures. A reduction in the conducting voltage leads to an increase in the current. As the current increases, the temperature of the LED chip continues to rise. This leads to a further drop in the conducting voltage.

LED circuits must be sufficiently dimensioned or cooled to prevent temperature-related changes in the current from causing damage or shortening of service life.

With falling temperatures, the current would be reduced by the increasing conducting voltage. This could lead to the required luminosity not being achieved.

The aging of LEDs is approximately exponential. The speed of aging depends on the respective semiconducting material and the operating conditions (temperature, current). If LEDs are operated at the usage limits (maximum forward voltage, maximum forward current, maximum operating temperature), the service life of the LED is shortened. The aging of LEDs is reflected in the reduction of luminosity and a change in the color temperature.

2.5 Start

For commissioning:

- mount the EL2596-xxxx as described in the chapter Installation [▶ 49]
- configure the EL2596-xxxx in TwinCAT as described in the chapter TwinCAT Quick Start [▶ 98].
- To use the LED output, observe the notes [▶ 125] and proceed with the configuration as described in the chapter Quick Start [▶ 125].
- A step-by-step guide to commissioning the various operation modes is described in the chapter Setting the operation modes [▶ 131].

3 Basics communication

3.1 EtherCAT basics

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

3.2 EtherCAT cabling – wire-bound

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

Cables and connectors

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

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

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

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

Recommended cables

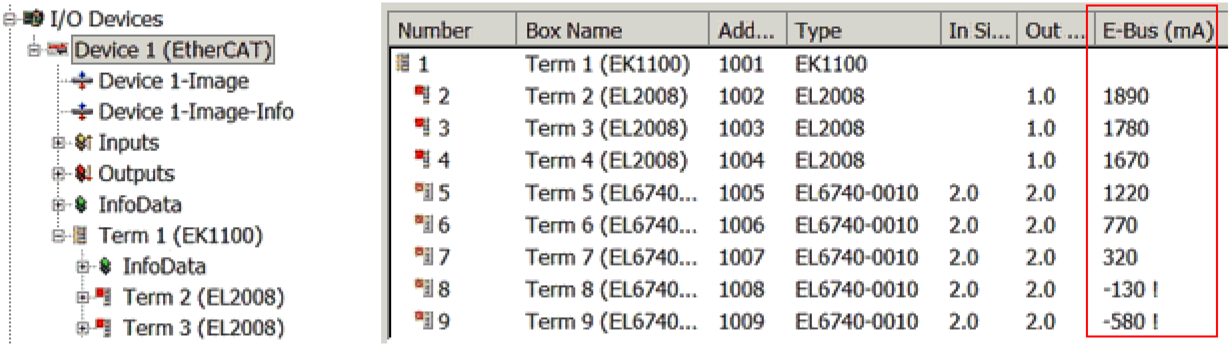
It is recommended to use the appropriate Beckhoff components e.g.
- cable sets ZK1090-9191-xxxx respectively
- RJ45 connector, field assembly ZS1090-0005
- EtherCAT cable, field assembly ZB9010, ZB9020

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

E-Bus supply

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

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



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

Fig. 20: 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!

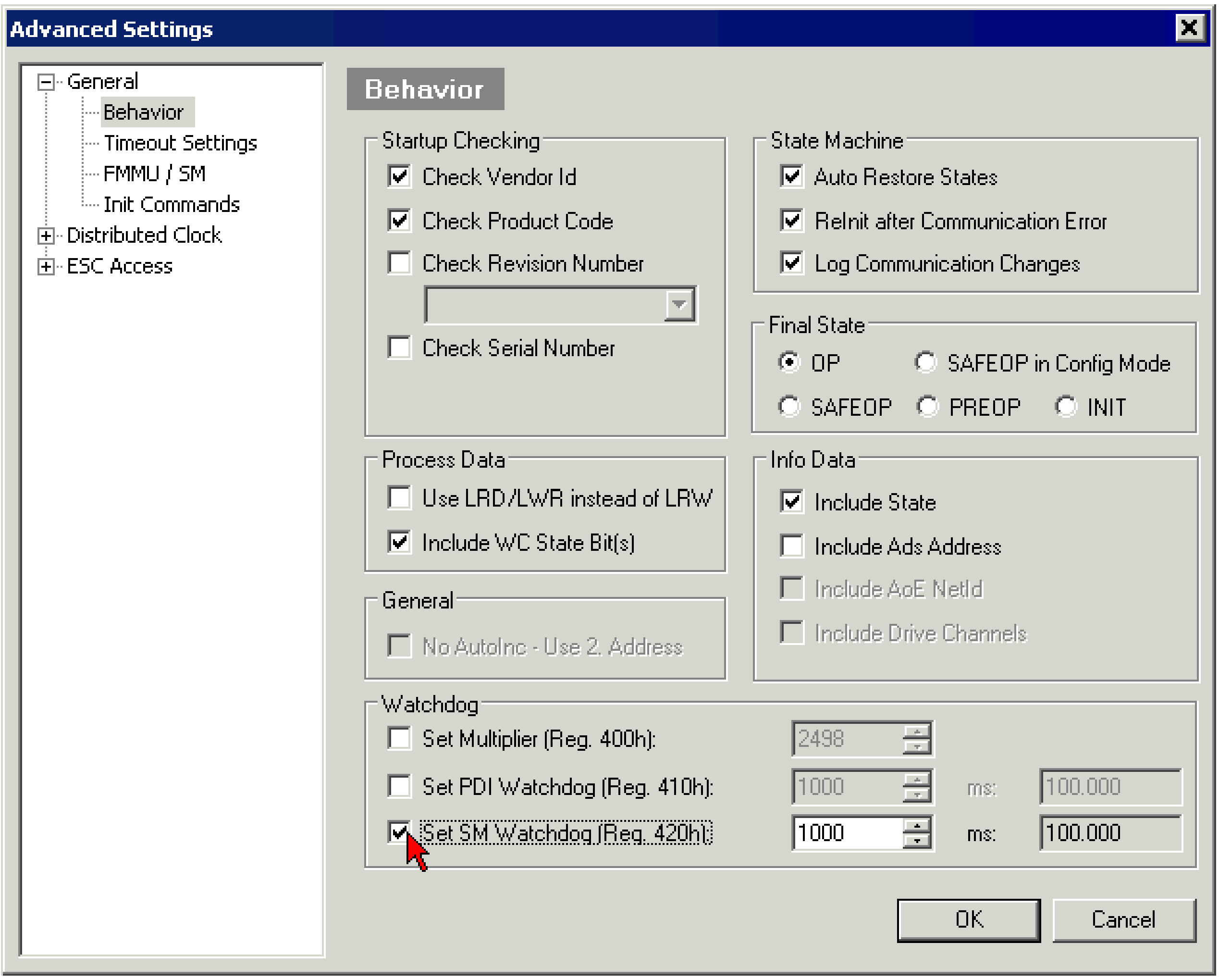
3.3 General notes for setting the watchdog

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

The EtherCAT slave controller (ESC) features two watchdogs:

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

Their times are individually parameterized in TwinCAT as follows:



Advanced Settings

Behavior

Startup Checking
- ☑ Check Vendor Id
- ☑ Check Product Code
- □ Check Revision Number
- □ Check Serial Number

State Machine
- ☑ Auto Restore States
- ☑ ReInit after Communication Error
- ☑ Log Communication Changes

Final State
- ● OP ○ SAFEOP in Config Mode
- ○ SAFEOP ○ PREOP ○ INIT

Process Data
- □ Use LRD/LWR instead of LRW
- ☑ Include WC State Bit(s)

General
- □ No AutoInc - Use 2. Address

Info Data
- ☑ Include State
- □ Include Ads Address
- □ Include AoE NetId
- □ Include Drive Channels

Watchdog
- □ Set Multiplier (Reg. 400h): 2498
- □ Set PDI Watchdog (Reg. 410h): 1000 ms: 100.000
- ☑ Set SM Watchdog (Reg. 420h): 1000 ms: 100.000

Fig. 21: eEtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

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

SM watchdog (SyncManager Watchdog)

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

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

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

PDI watchdog (Process Data Watchdog)

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

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

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

Calculation

Watchdog time = [1/25 MHz * (Watchdog multiplier + 2)] * PDI/SM watchdog

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

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

⚠ CAUTION

Undefined state possible!

The function for switching off 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.

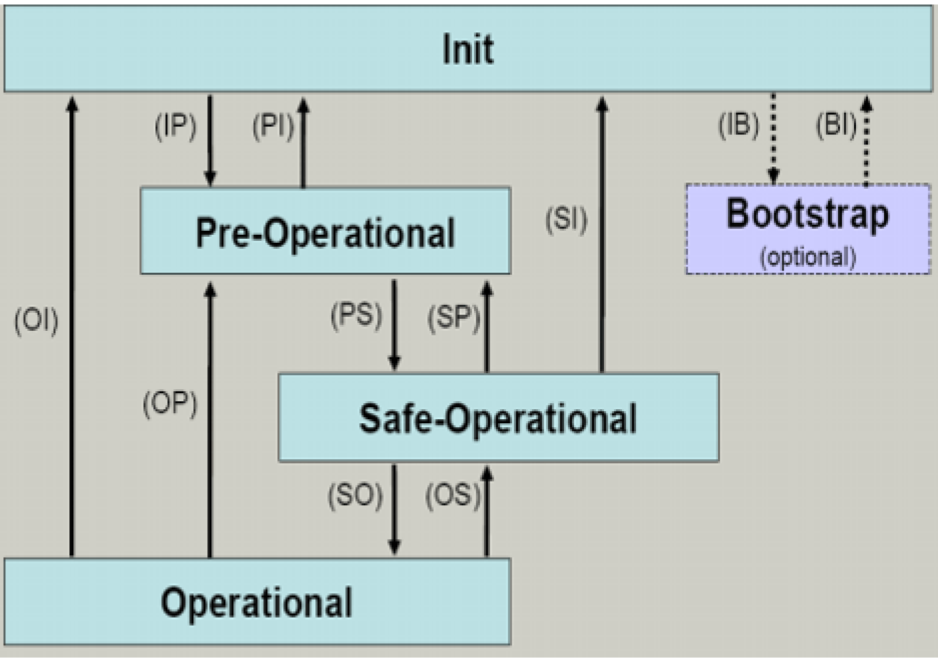


Fig. 22: States of the EtherCAT State Machine

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

Operational (Op)

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

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

Boot

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

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

3.5 CoE Interface

General description

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

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

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

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

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

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

The relevant ranges for EtherCAT fieldbus users are:

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

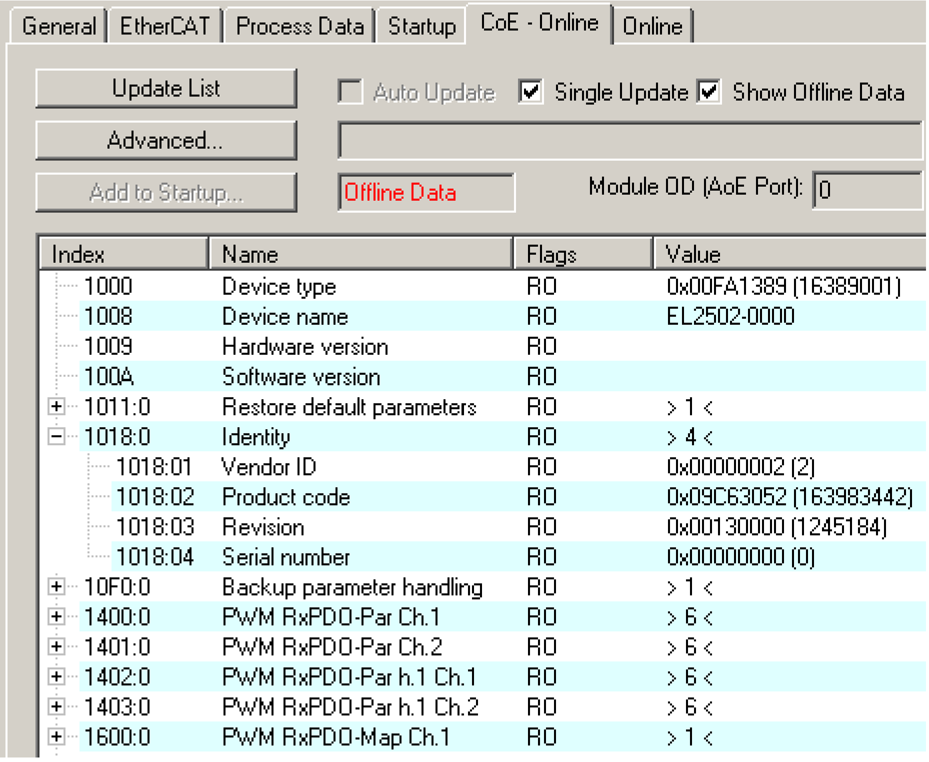
Other important ranges are:

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

Availability

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

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



☐ Auto Update ☑ Single Update ☑ Show Offline Data

Module OD (AoE Port): 0

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

Fig. 23: “CoE Online” tab

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

NOTE

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

When using/manipulating the CoE parameters observe the general CoE notes in chapter "CoE interface" of the EtherCAT system documentation:

- Keep a startup list if components have to be replaced,
- Distinction between online/offline dictionary,
- Existence of current XML description (download from the Beckhoff website),
- "CoE-Reload" for resetting the changes
- Program access during operation via PLC (see TwinCAT3 | PLC Library: Tc2_EtherCAT and Example program R/W CoE)

Data management and function “NoCoeStorage”

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

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

Data management

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

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

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

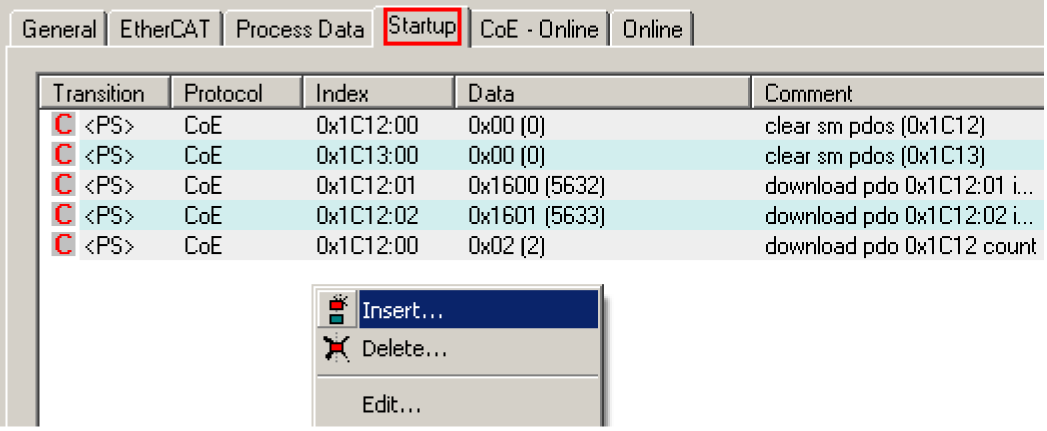
Startup list

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

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

Recommended approach for manual modification of CoE parameters

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



Transition	Protocol	Index	Data	Comment
C <PS>	CoE	0x1C12:00	0x00 (0)	clear sm pdos (0x1C12)
C <PS>	CoE	0x1C13:00	0x00 (0)	clear sm pdos (0x1C13)
C <PS>	CoE	0x1C12:01	0x1600 (5632)	download pdo 0x1C12:01 i...
C <PS>	CoE	0x1C12:02	0x1601 (5633)	download pdo 0x1C12:02 i...
C <PS>	CoE	0x1C12:00	0x02 (2)	download pdo 0x1C12 count

Fig. 24: Startup list in the TwinCAT System Manager

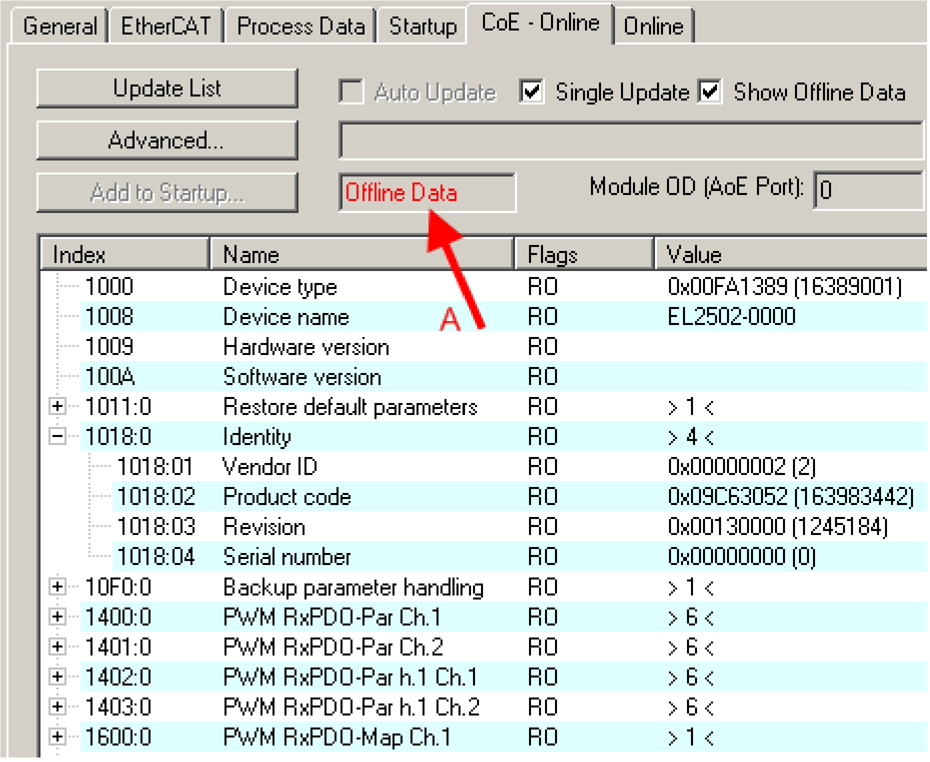
The Startup list may already contain values that were configured by the System Manager based on the ESI specifications. Additional application-specific entries can be created.

Online/offline list

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

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

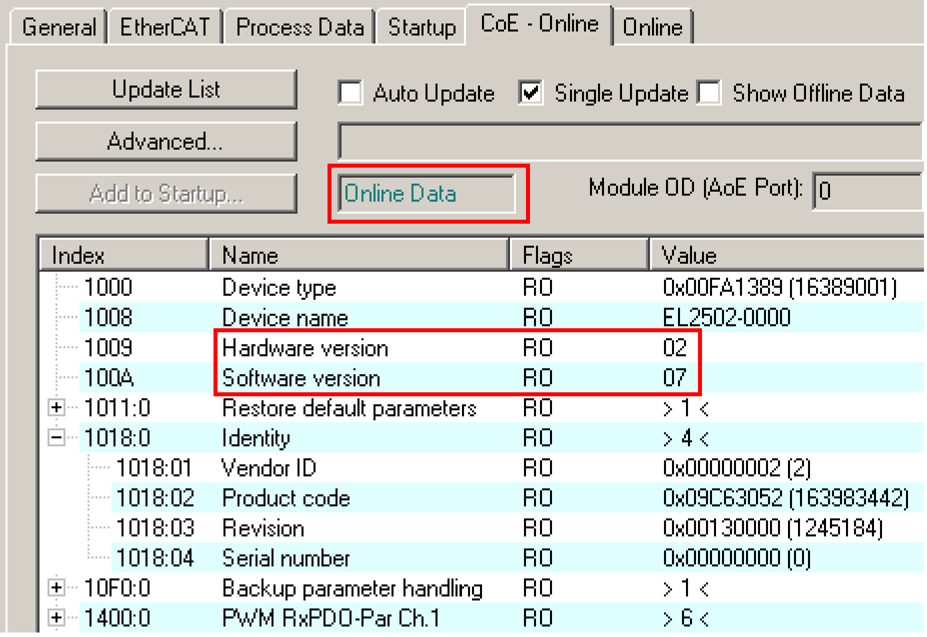
- If the slave is offline
    - The offline list from the ESI file is displayed. In this case modifications are not meaningful or possible.
    - The configured status is shown under Identity.
    - No firmware or hardware version is displayed, since these are features of the physical device.
    - **Offline**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



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

Fig. 25: Offline list

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



☐ Auto Update ☑ Single Update ☐ Show Offline Data

Online Data

Module OD (AoE Port): 0

Index	Name	Flags	Value
1000	Device type	RO	0x00FA1389 (16389001)
1008	Device name	RO	EL2502-0000
1009	Hardware version	RO	02
100A	Software version	RO	07
+ 1011:0	Restore default parameters	RO	> 1 <
− 1018:0	Identity	RO	> 4 <
1018:01	Vendor ID	RO	0x00000002 (2)
1018:02	Product code	RO	0x09C63052 (163983442)
1018:03	Revision	RO	0x00130000 (1245184)
1018:04	Serial number	RO	0x00000000 (0)
+ 10F0:0	Backup parameter handling	RO	> 1 <
+ 1400:0	PWM RxPDO-Par Ch.1	RO	> 6 <

Fig. 26: Online list

Channel-based order

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

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

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

This is generally written as 0x80n0.

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

3.6 Distributed Clock

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

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

For detailed information please refer to the EtherCAT system description.

4 Mounting and wiring

4.1 Instructions for ESD protection

NOTE

Destruction of the devices by electrostatic discharge possible!

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

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

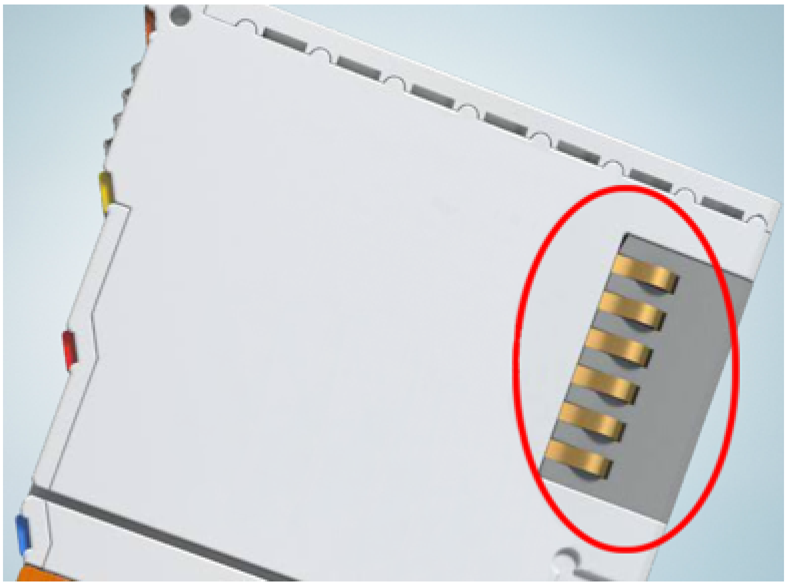


Fig. 27: Spring contacts of the Beckhoff I/O components

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!

The Bus Terminal system and is designed for mounting in a control cabinet or terminal box.

Assembly

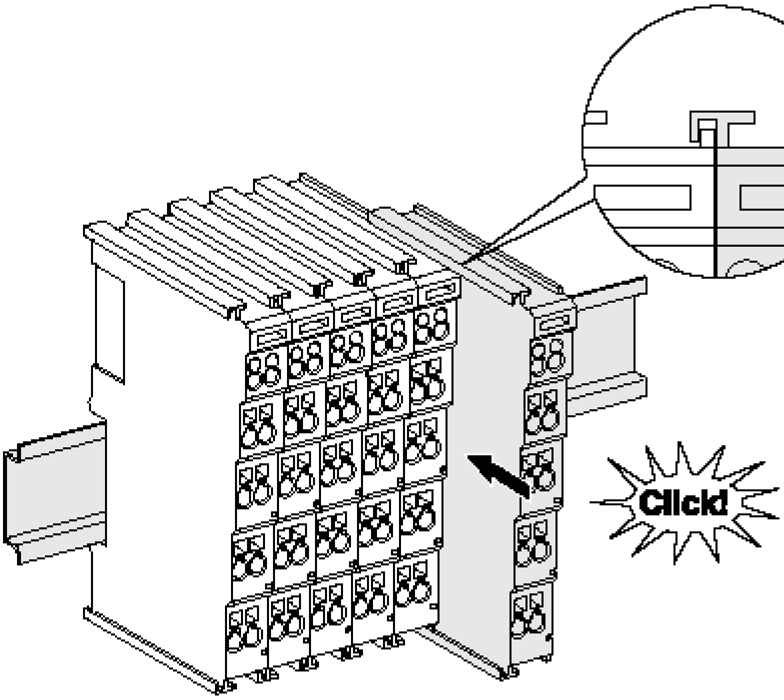


Fig. 28: Attaching on mounting rail

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

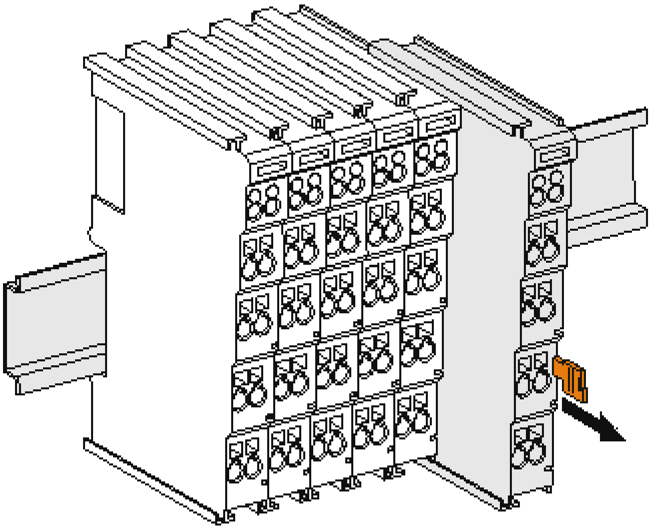


Fig. 29: Disassembling of terminal

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.

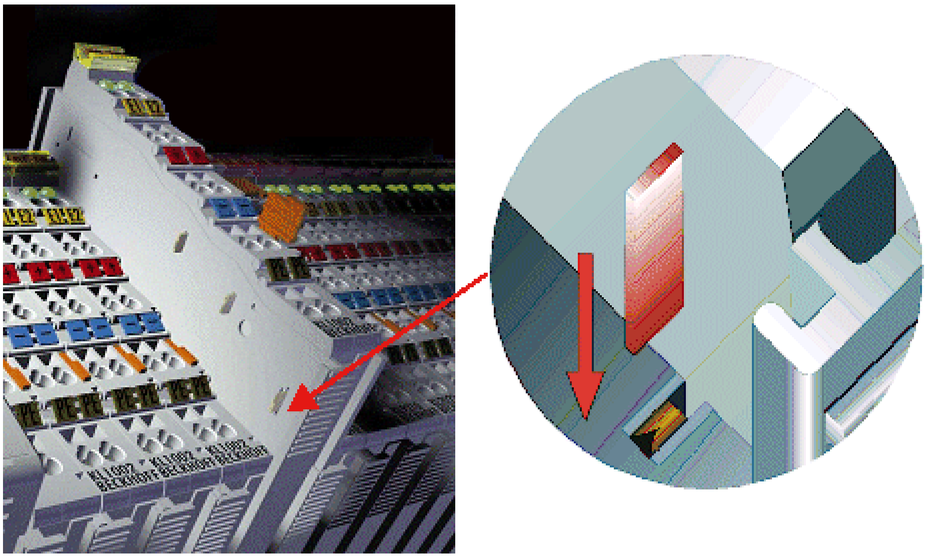


Fig. 30: Power contact on left side

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 Prescribed installation position

NOTE

Constraints regarding installation position and operating temperature range

When installing the terminals ensure that an adequate spacing is maintained between other components above and below the terminal in order to guarantee adequate ventilation!

Prescribed installation position

The prescribed 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.

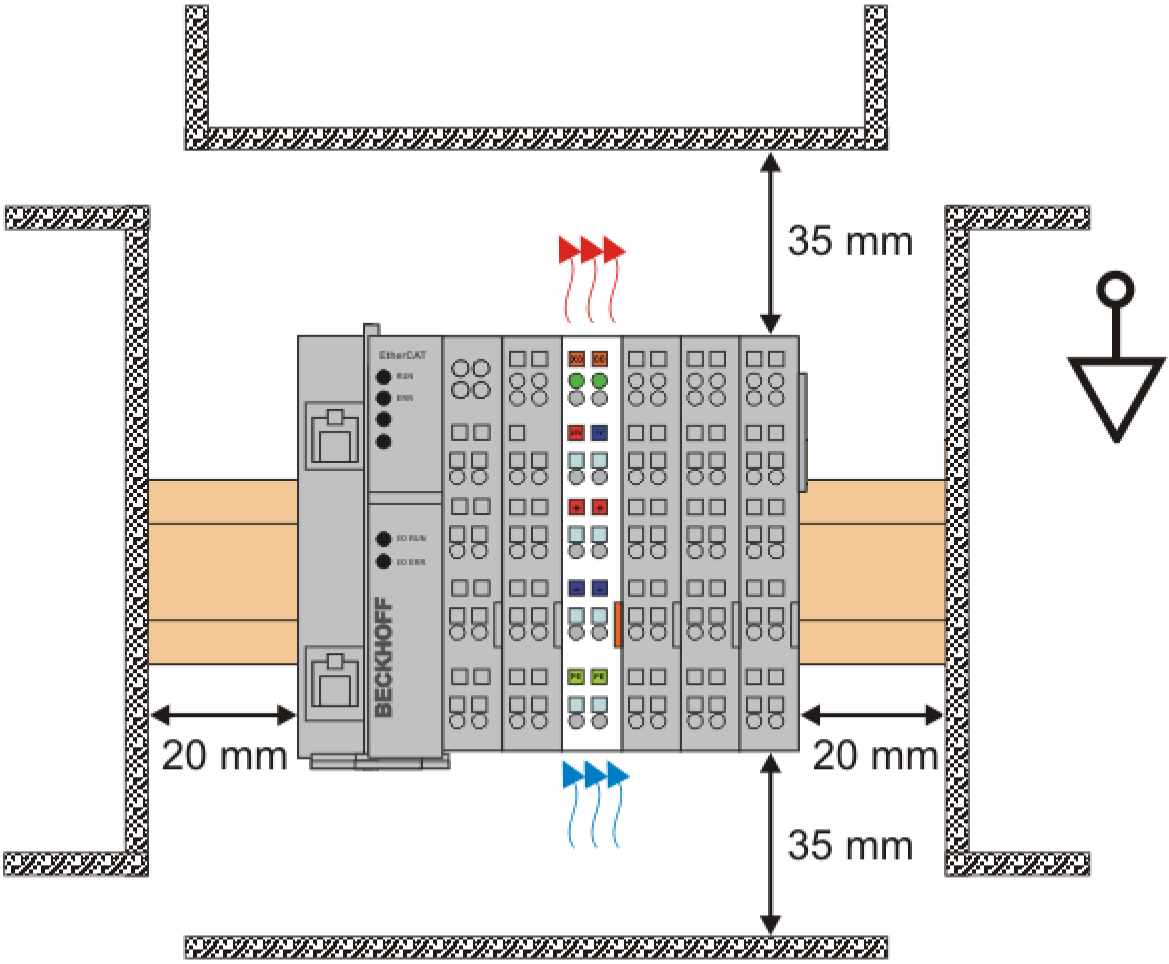


Fig. 31: Recommended minimum distances for standard installation position

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

4.4 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)

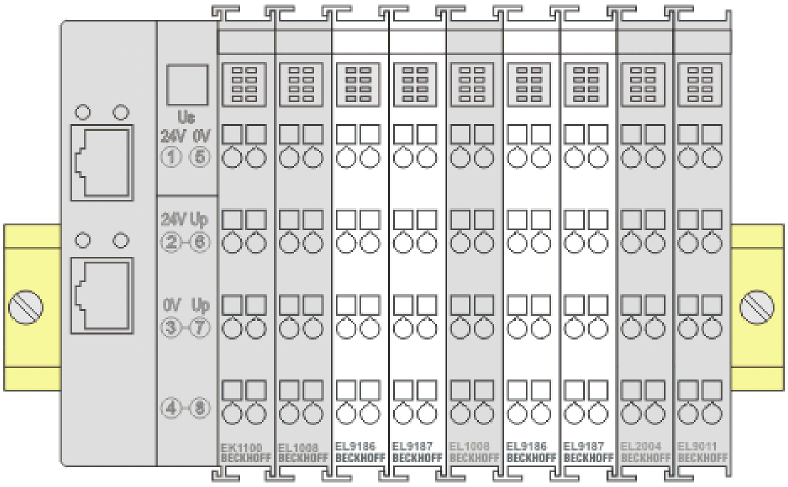


Fig. 32: Correct positioning

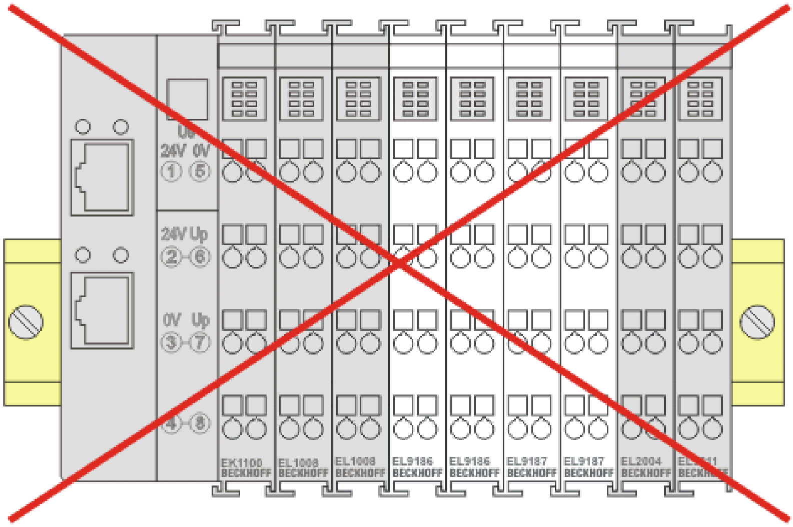


Fig. 33: Incorrect positioning

4.5 Connection

4.5.1 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)

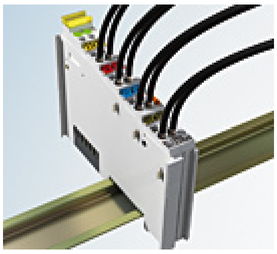


Fig. 34: Standard wiring

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)

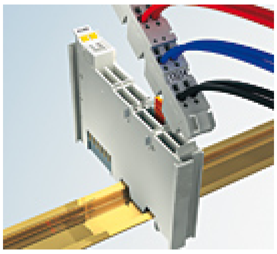


Fig. 35: Pluggable wiring

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)

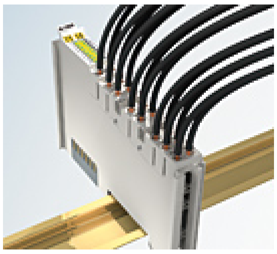


Fig. 36: High Density Terminals

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

Wiring HD Terminals

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

Ultrasonically “bonded” (ultrasonically welded) conductors

Ultrasonically “bonded” conductors

It is also possible to connect the Standard and High Density Terminals with ultrasonically “bonded” (ultrasonically welded) conductors. In this case, please note the tables concerning the wire-size width [▶ 57]!

4.5.2 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

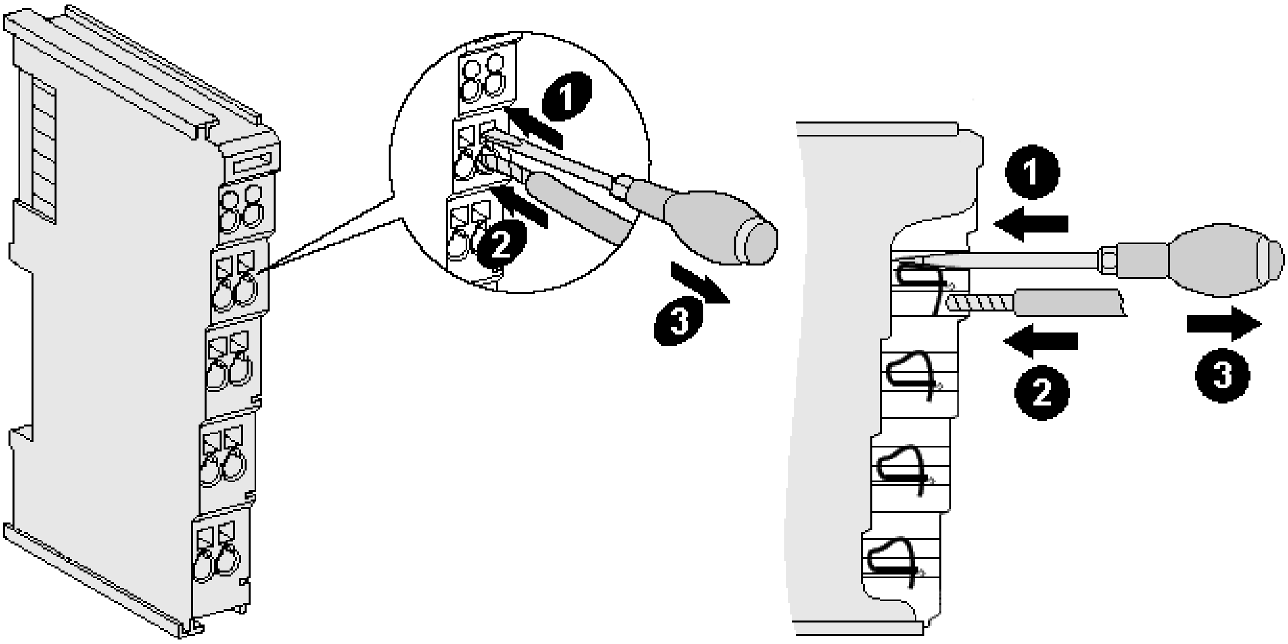


Fig. 37: Connecting a cable on a terminal point

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 [▶ 55]) 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 (see notice [▶ 56])
Wire stripping length	8 ... 9 mm

4.5.3 Shielding

Shielding

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

4.6 Note - Power supply

⚠ WARNING

Power supply from SELV/PELV power supply unit!

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

Notes:

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

4.7 Disposal

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

5 Commissioning

5.1 TwinCAT basics

5.1.1 TwinCAT Development Environment

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

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

Details:

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

Additional features:

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

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

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

5.1.1.1 Installation of the TwinCAT real-time driver

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

This can be done in several ways.

A: Via the TwinCAT Adapter dialog

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

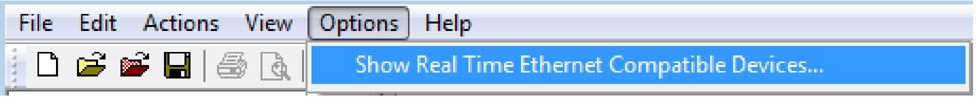


Fig. 38: System Manager “Options” (TwinCAT 2)

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

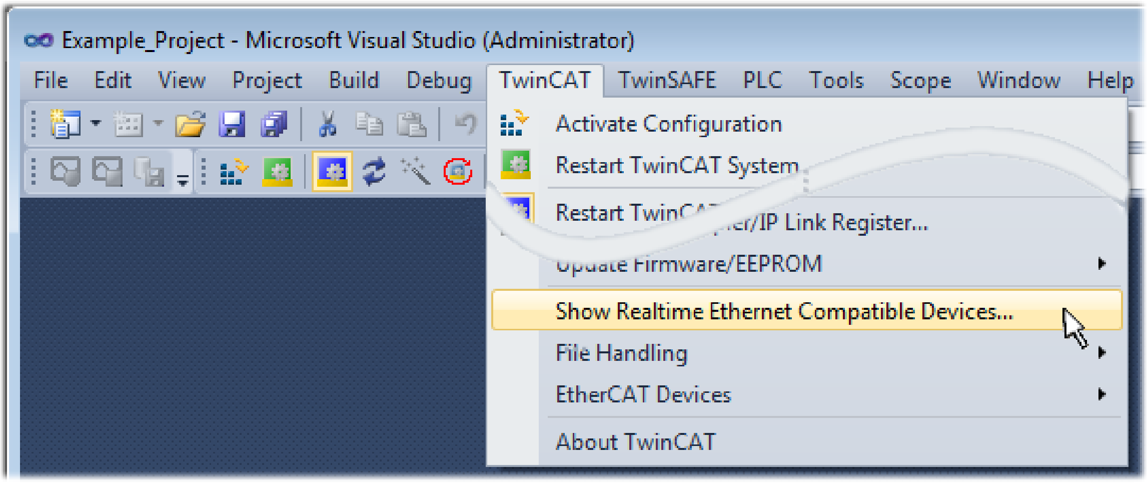


Fig. 39: Call up under VS Shell (TwinCAT 3)

B: Via TcRteInstall.exe in the TwinCAT directory

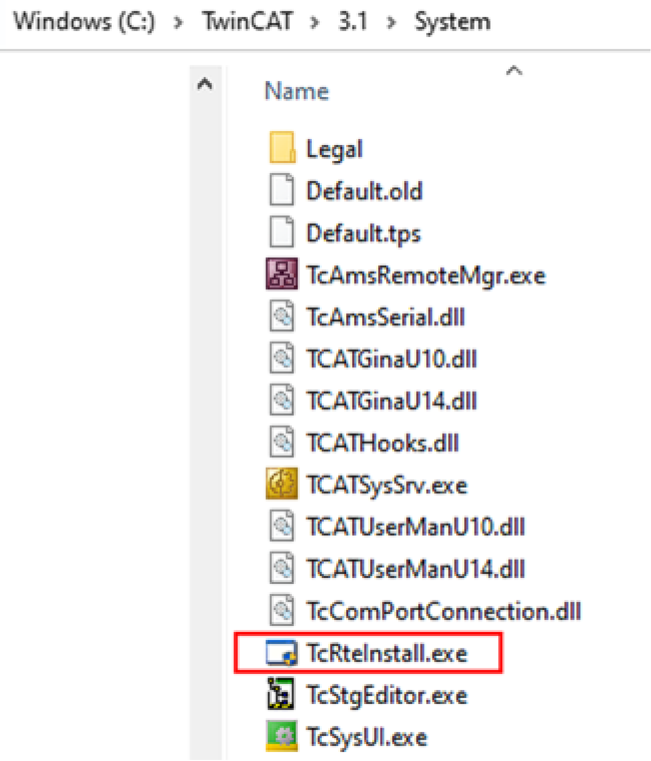


Fig. 40: TcRteInstall in the TwinCAT directory

In both cases, the following dialog appears:

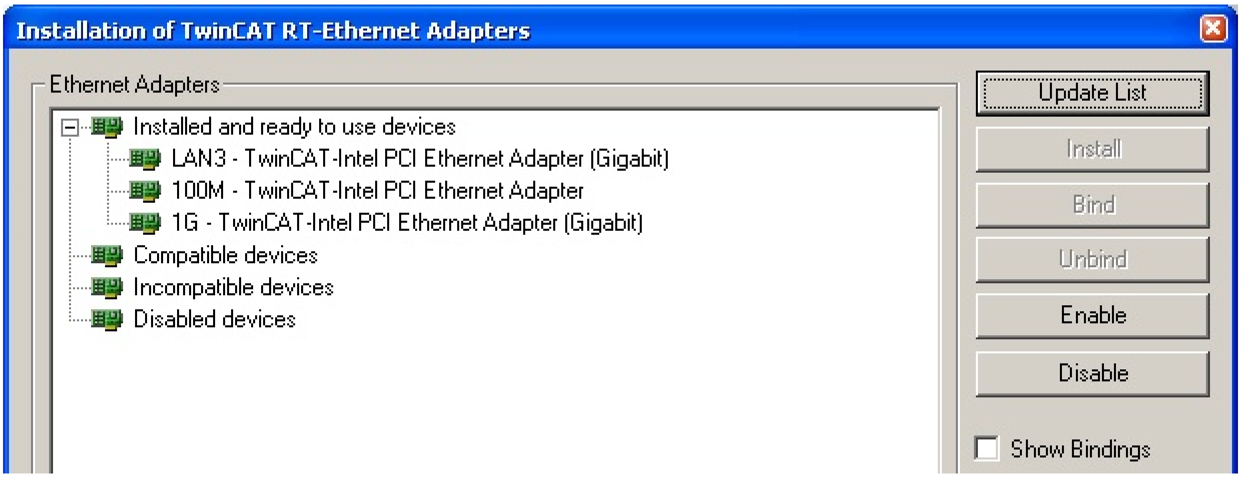
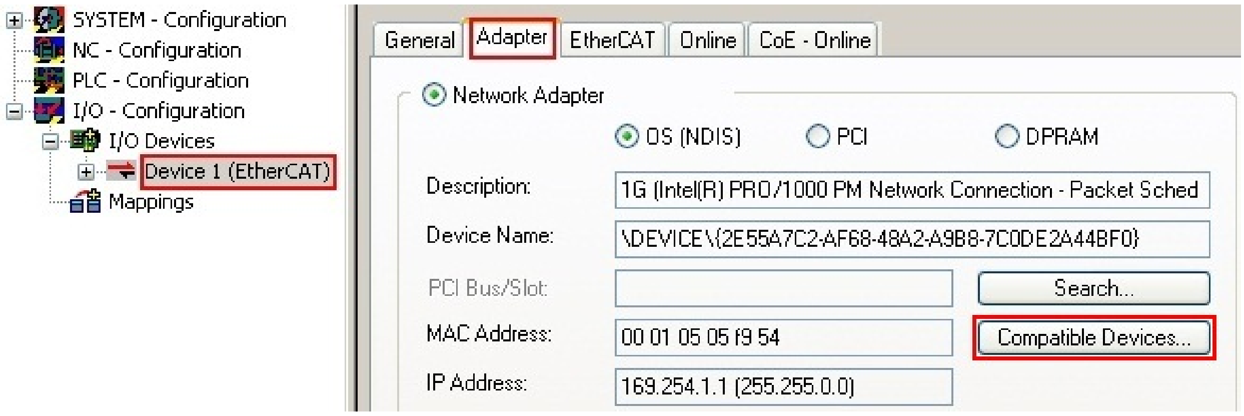

Installation of TwinCAT RT-Ethernet Adapters
Ethernet Adapters
Installed and ready to use devices
LAN3 - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
100M - TwinCAT-Intel PCI Ethernet Adapter
1G - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
Compatible devices
Incompatible devices
Disabled devices
Update List
Install
Bind
Unbind
Enable
Disable
□ Show Bindings

Fig. 41: 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” [▶ 70] in order to view the compatible ethernet ports via its EtherCAT properties (tab “Adapter”, button “Compatible Devices…”):



● Network Adapter

● OS (NDIS) ○ PCI ○ DPRAM

Description: 1G (Intel(R) PRO/1000 PM Network Connection - Packet Sched

Device Name: \DEVICE\{2E55A7C2-AF68-48A2-A9B8-7C0DE2A44BF0}

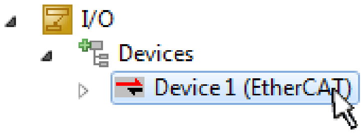
PCI Bus/Slot:

MAC Address: 00 01 05 05 f9 54

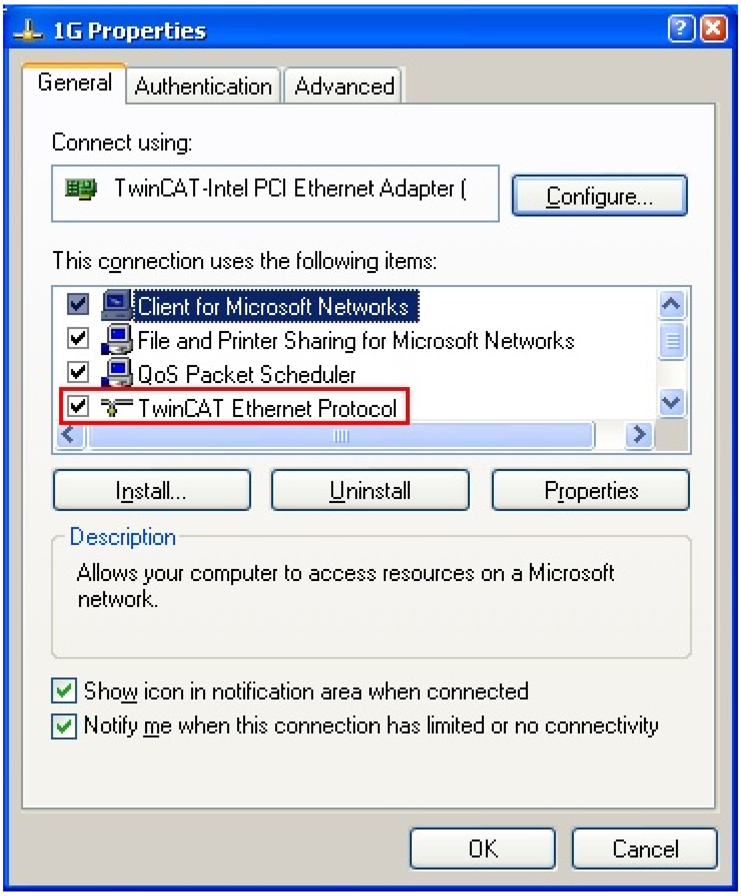
IP Address: 169.254.1.1 (255.255.0.0)

Fig. 42: EtherCAT device properties (TwinCAT 2): click on “Compatible Devices…” of tab “Adapter”

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



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



Connect using:

TwinCAT-Intel PCI Ethernet Adapter (

This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ TwinCAT Ethernet Protocol

Description

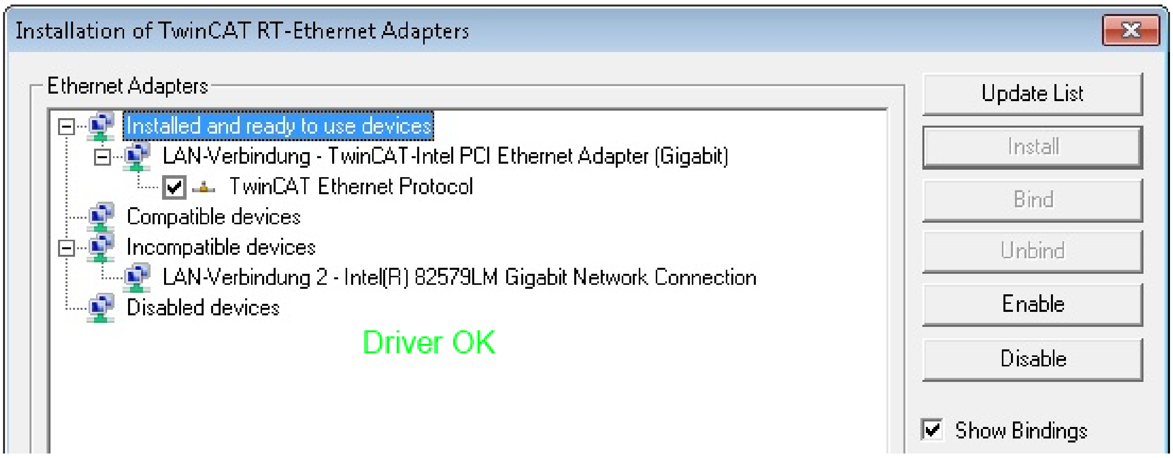
Allows your computer to access resources on a Microsoft network.

☑ Show icon in notification area when connected

☑ Notify me when this connection has limited or no connectivity

Fig. 43: Windows properties of the network interface

A correct setting of the driver could be:

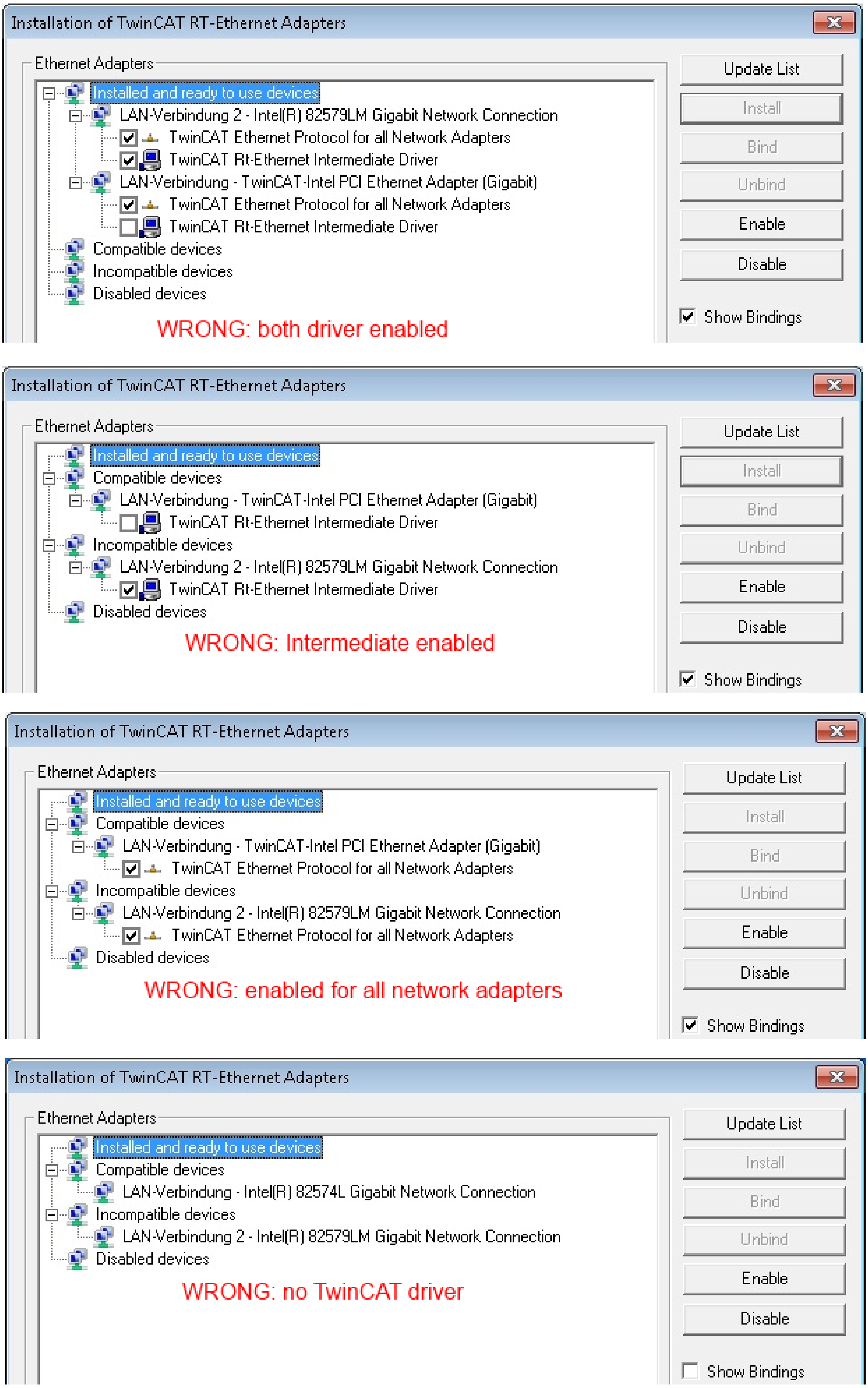


- Installed and ready to use devices
  - LAN-Verbindung - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
    - ☑ TwinCAT Ethernet Protocol
- Compatible devices
- Incompatible devices
  - LAN-Verbindung 2 - Intel(R) 82579LM Gigabit Network Connection
- Disabled devices

☑ Show Bindings

Fig. 44: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:


Installation of TwinCAT RT-Ethernet Adapters

Ethernet Adapters
- Installed and ready to use devices
  - LAN-Verbindung 2 - Intel(R) 82579LM Gigabit Network Connection
    - ☑ TwinCAT Ethernet Protocol for all Network Adapters
    - ☑ TwinCAT RT-Ethernet Intermediate Driver
  - LAN-Verbindung - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
    - ☑ TwinCAT Ethernet Protocol for all Network Adapters
    - ☐ TwinCAT RT-Ethernet Intermediate Driver
- Compatible devices
- Incompatible devices
- Disabled devices

☑ Show Bindings

WRONG: both driver enabled

Installation of TwinCAT RT-Ethernet Adapters

Ethernet Adapters
- Installed and ready to use devices
- Compatible devices
  - LAN-Verbindung - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
    - ☐ TwinCAT RT-Ethernet Intermediate Driver
- Incompatible devices
  - LAN-Verbindung 2 - Intel(R) 82579LM Gigabit Network Connection
    - ☑ TwinCAT RT-Ethernet Intermediate Driver
- Disabled devices

☑ Show Bindings

WRONG: Intermediate enabled

Installation of TwinCAT RT-Ethernet Adapters

Ethernet Adapters
- Installed and ready to use devices
- Compatible devices
  - LAN-Verbindung - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
    - ☑ TwinCAT Ethernet Protocol for all Network Adapters
- Incompatible devices
  - LAN-Verbindung 2 - Intel(R) 82579LM Gigabit Network Connection
    - ☑ TwinCAT Ethernet Protocol for all Network Adapters
- Disabled devices

☑ Show Bindings

WRONG: enabled for all network adapters

Installation of TwinCAT RT-Ethernet Adapters

Ethernet Adapters
- Installed and ready to use devices
- Compatible devices
  - LAN-Verbindung - Intel(R) 82574L Gigabit Network Connection
- Incompatible devices
  - LAN-Verbindung 2 - Intel(R) 82579LM Gigabit Network Connection
- Disabled devices

☐ Show Bindings

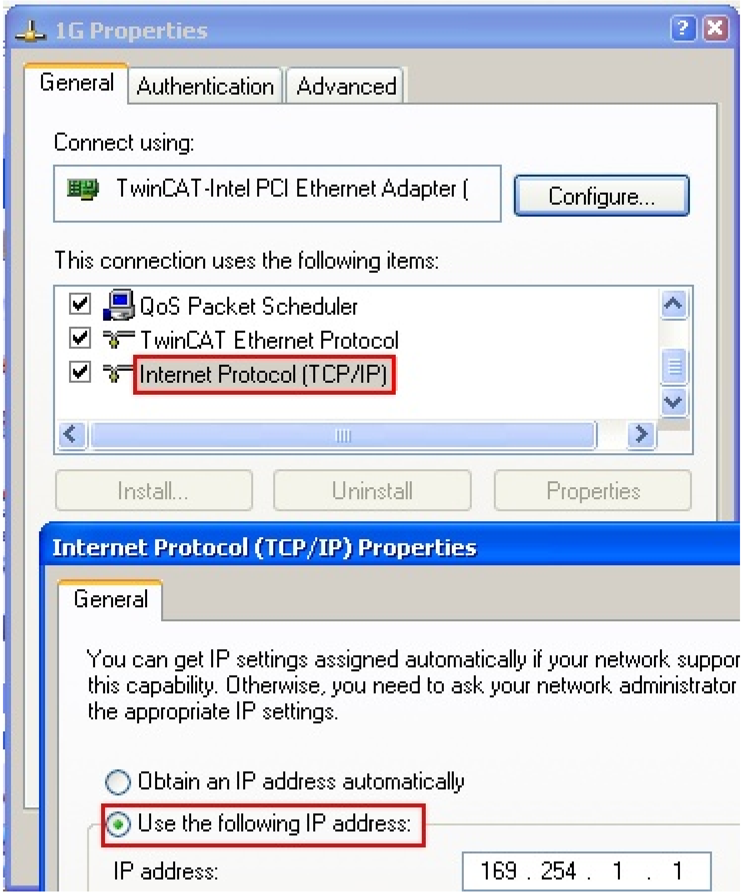
WRONG: no TwinCAT driver

Fig. 45: 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.



1G Properties

Connect using:

TwinCAT-Intel PCI Ethernet Adapter (

This connection uses the following items:

☑ QoS Packet Scheduler
☑ TwinCAT Ethernet Protocol
☑ Internet Protocol (TCP/IP)

Internet Protocol (TCP/IP) Properties

You can get IP settings assigned automatically if your network suppor this capability. Otherwise, you need to ask your network administrator the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:

IP address: 169 . 254 . 1 . 1

Fig. 46: TCP/IP setting for the Ethernet port

5.1.1.2 Notes regarding ESI device description

Installation of the latest ESI device description

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

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

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

Default settings:

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

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

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

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

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

The TwinCAT ESI Updater [▶ 69] 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”

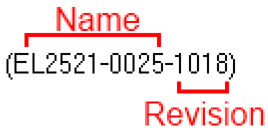
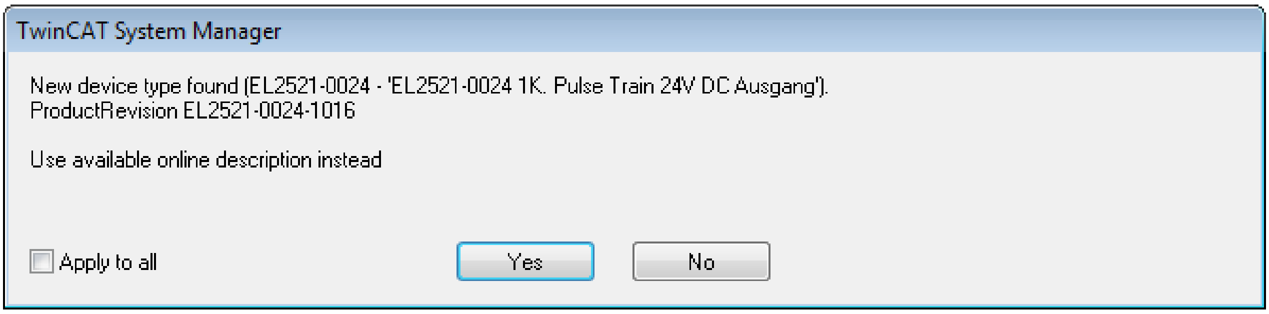


Fig. 47: Identifier structure

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

Online description

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



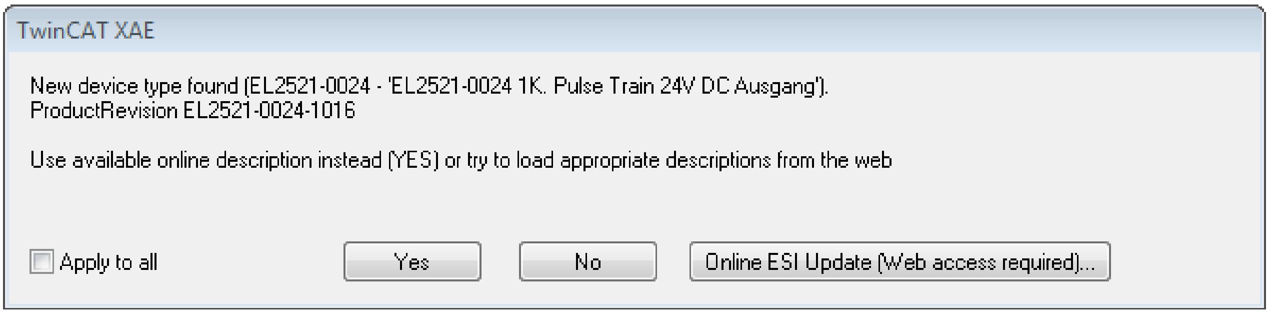
TwinCAT System Manager

New device type found [EL2521-0024 - 'EL2521-0024 1K. Pulse Train 24V DC Ausgang'].
ProductRevision EL2521-0024-1016

Use available online description instead

Fig. 48: OnlineDescription information window (TwinCAT 2)

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



TwinCAT XAE

New device type found [EL2521-0024 - 'EL2521-0024 1K. Pulse Train 24V DC Ausgang'].
ProductRevision EL2521-0024-1016

Use available online description instead (YES) or try to load appropriate descriptions from the web

Fig. 49: 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 [▶ 70]”.

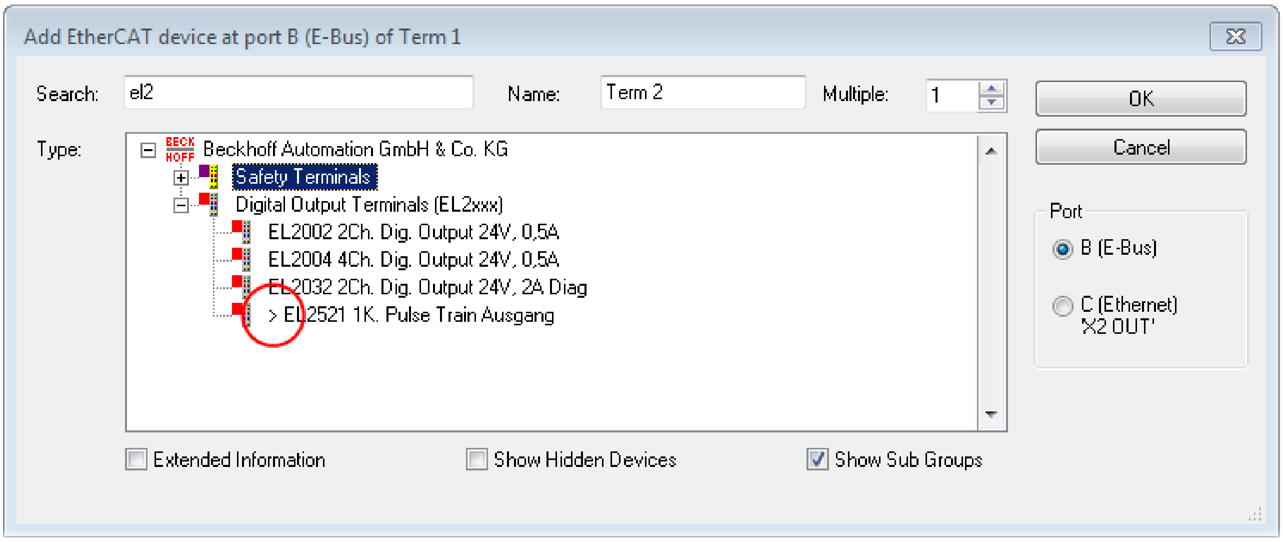
If the OnlineDescription is used regardless, the System Manager reads a copy of the device description from the EEPROM in the EtherCAT slave. In complex slaves the size of the EEPROM may not be sufficient for the complete ESI, in which case the ESI would be *incomplete* in the configurator. Therefore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

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

OnlineDescriptionCache00000002.xml

Fig. 50: 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)*.



Add EtherCAT device at port B (E-Bus) of Term 1

Search: el2
Name: Term 2
Multiple: 1

Type:
- Beckhoff Automation GmbH & Co. KG
  - Safety Terminals
  - Digital Output Terminals (EL2xxx)
    - EL2002 2Ch. Dig. Output 24V, 0,5A
    - EL2004 4Ch. Dig. Output 24V, 0,5A
    - EL2032 2Ch. Dig. Output 24V, 2A Diag
    - > EL2521 1K. Pulse Train Ausgang

Port
● B (E-Bus)
○ C (Ethernet) 'X2 OUT'

□ Extended Information □ Show Hidden Devices ☑ Show Sub Groups

Fig. 51: 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

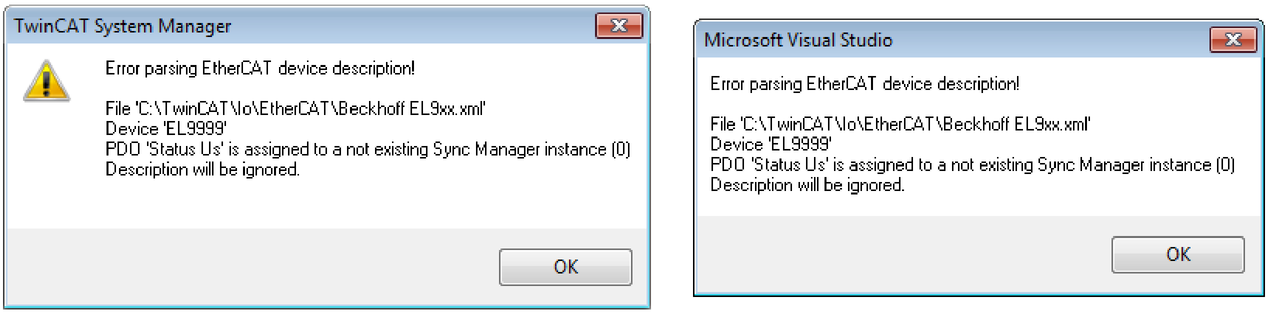
In addition to the file described above “OnlineDescription0000...xml”, a so called EtherCAT cache with new discovered devices is created by TwinCAT 3.x, e.g. under Windows 7:

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.



TwinCAT System Manager

Error parsing EtherCAT device description!

File 'C:\TwinCAT\Io\EtherCAT\Beckhoff EL9xx.xml'
Device 'EL9999'
PDO 'Status Us' is assigned to a not existing Sync Manager instance (0)
Description will be ignored.

Microsoft Visual Studio

Error parsing EtherCAT device description!

File 'C:\TwinCAT\Io\EtherCAT\Beckhoff EL9xx.xml'
Device 'EL9999'
PDO 'Status Us' is assigned to a not existing Sync Manager instance (0)
Description will be ignored.

Fig. 52: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

Reasons may include:

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

5.1.1.3 TwinCAT ESI Updater

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

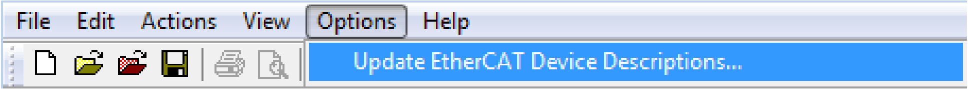
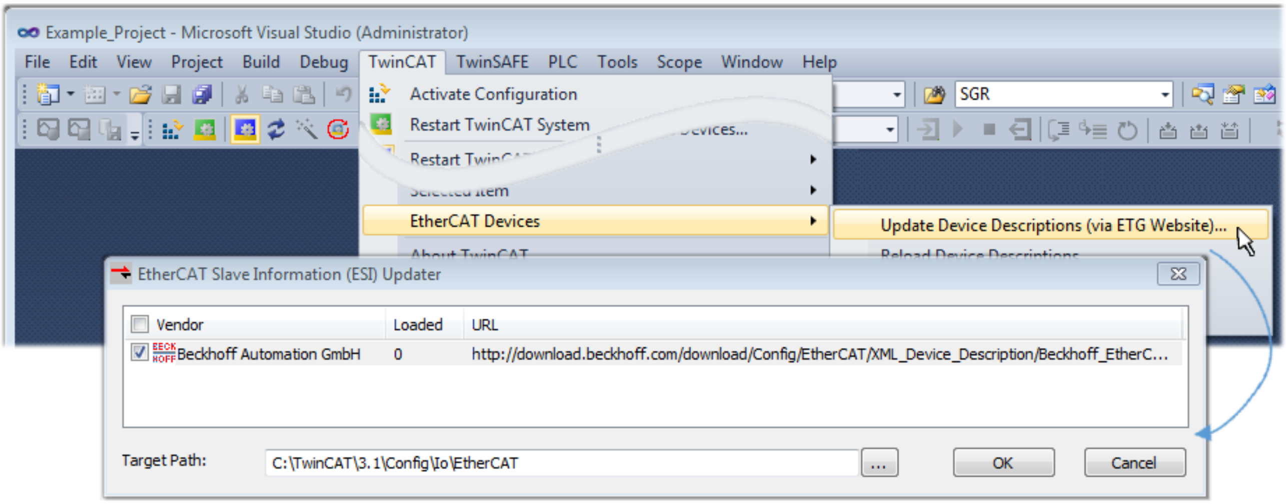


Fig. 53: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:



Vendor	Loaded	URL
☑ Beckhoff Automation GmbH	0	http://download.beckhoff.com/download/Config/EtherCAT/XML_Device_Description/Beckhoff_EtherC...

Target Path: C:\TwinCAT\3.1\Config\Io\EtherCAT

Fig. 54: Using the ESI Updater (TwinCAT 3)

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

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

5.1.1.4 Distinction between Online and Offline

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

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

In any case, during each startup the EtherCAT master checks whether the slaves it finds match the configuration. This test can be parameterised in the extended slave settings. Refer to note “Installation of the latest ESI-XML device description” [▶ 65].

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 [▸ 75] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▸ 76]. This step can be carried out independent of the preceding step
- troubleshooting [▸ 79]

The scan with existing configuration [▸ 80] can also be carried out for comparison.

5.1.1.5 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

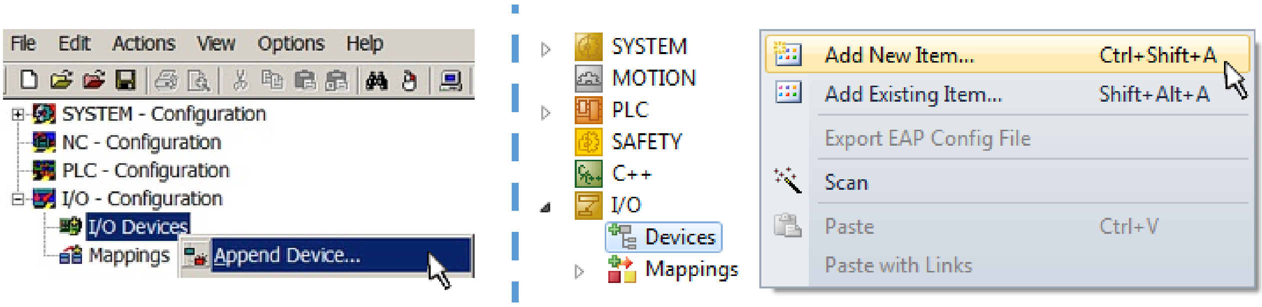


Fig. 55: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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”.

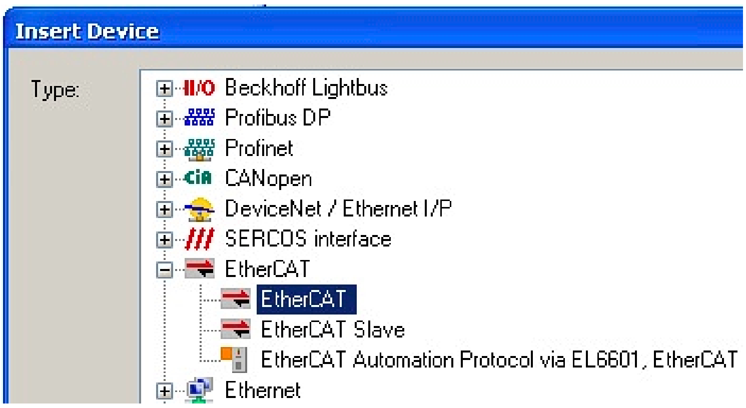


Fig. 56: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

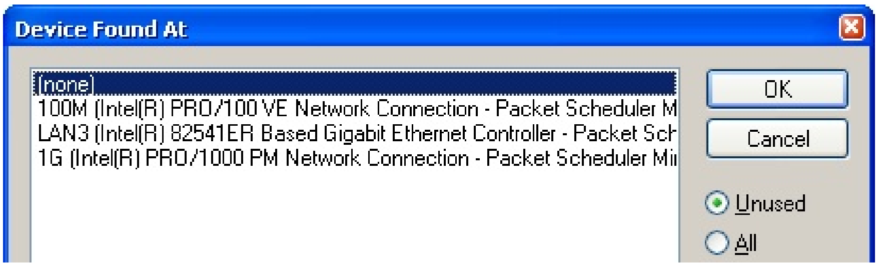
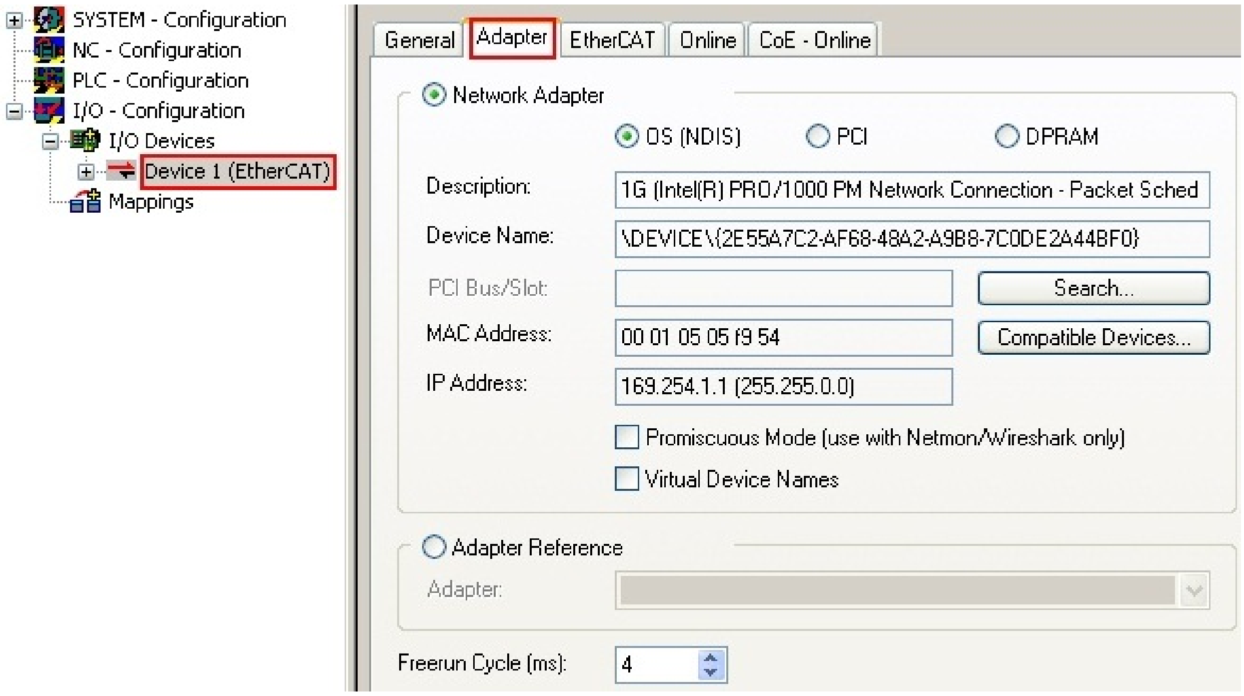

Device Found At
(none)
100M (Intel(R) PRO/100 VE Network Connection - Packet Scheduler M
LAN3 (Intel(R) 82541ER Based Gigabit Ethernet Controller - Packet Sch
1G (Intel(R) PRO/1000 PM Network Connection - Packet Scheduler Mi
OK
Cancel
● Unused
○ All

Fig. 57: 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)”.



● Network Adapter

● OS (NDIS) ○ PCI ○ DPRAM

Description: 1G (Intel(R) PRO/1000 PM Network Connection - Packet Sched

Device Name: \DEVICE\{2E55A7C2-AF68-48A2-A9B8-7C0DE2A44BF0}

PCI Bus/Slot:

MAC Address: 00 01 05 05 f9 54

IP Address: 169.254.1.1 (255.255.0.0)

□ Promiscuous Mode (use with Netmon/Wireshark only)

□ Virtual Device Names

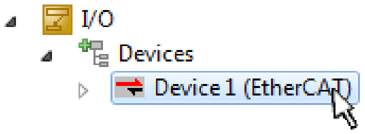
○ Adapter Reference

Adapter:

Freerun Cycle (ms): 4

Fig. 58: 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 [▶ 59].

Defining EtherCAT slaves

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

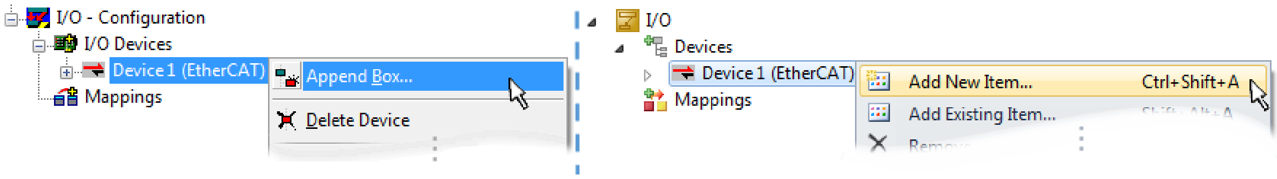


Fig. 59: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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

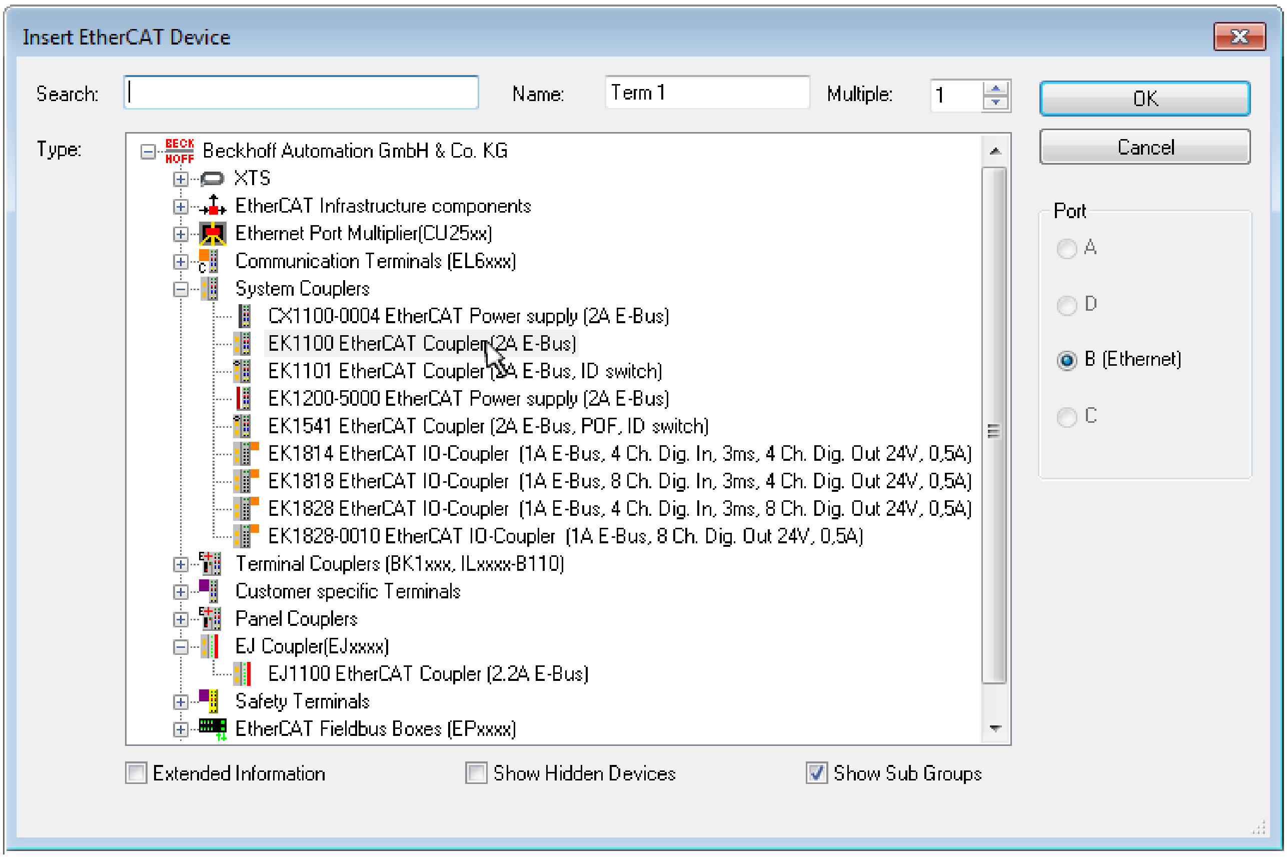
Only devices are offered for selection that can be appended to the previously selected device. Therefore, the physical layer available for this port is also displayed (Fig. “Selection dialog for new EtherCAT device”, A). In the case of cable-based Fast-Ethernet physical layer with PHY transfer, then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

Overview of physical layer

- “Ethernet”: cable-based 100BASE-TX: couplers, box modules, devices with RJ45/M8/M12 connector

- "E-Bus": LVDS "terminal bus", EtherCAT plug-in modules (EJ), EtherCAT terminals (EL/ES), various modular modules

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



Insert EtherCAT Device

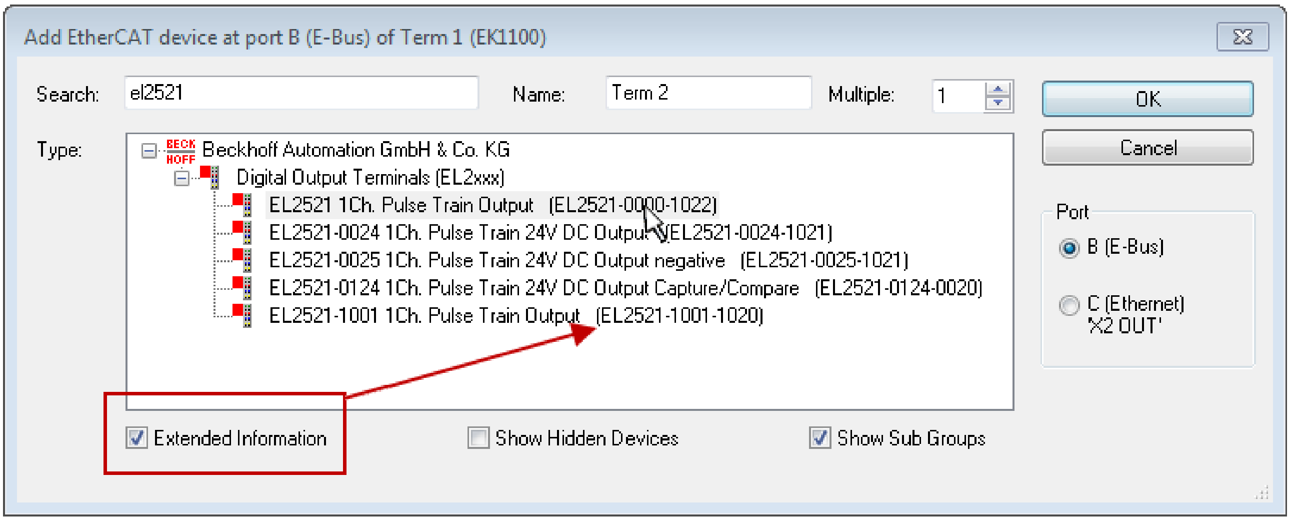
Search:
Name: Term 1
Multiple: 1

Type:
- Beckhoff Automation GmbH & Co. KG
  - XTS
  - EtherCAT Infrastructure components
  - Ethernet Port Multiplier(CU25xx)
  - Communication Terminals (EL6xxx)
  - System Couplers
    - CX1100-0004 EtherCAT Power supply (2A E-Bus)
    - EK1100 EtherCAT Coupler (2A E-Bus)
    - EK1101 EtherCAT Coupler (2A E-Bus, ID switch)
    - EK1200-5000 EtherCAT Power supply (2A E-Bus)
    - EK1541 EtherCAT Coupler (2A E-Bus, POF, ID switch)
    - EK1814 EtherCAT IO-Coupler (1A E-Bus, 4 Ch. Dig. In, 3ms, 4 Ch. Dig. Out 24V, 0,5A)
    - EK1818 EtherCAT IO-Coupler (1A E-Bus, 8 Ch. Dig. In, 3ms, 4 Ch. Dig. Out 24V, 0,5A)
    - EK1828 EtherCAT IO-Coupler (1A E-Bus, 4 Ch. Dig. In, 3ms, 8 Ch. Dig. Out 24V, 0,5A)
    - EK1828-0010 EtherCAT IO-Coupler (1A E-Bus, 8 Ch. Dig. Out 24V, 0,5A)
  - Terminal Couplers (BK1xxx, ILxxxx-B110)
  - Customer specific Terminals
  - Panel Couplers
  - EJ Coupler(EJxxxx)
    - EJ1100 EtherCAT Coupler (2.2A E-Bus)
  - Safety Terminals
  - EtherCAT Fieldbus Boxes (EPxxxx)

□ Extended Information □ Show Hidden Devices ☑ Show Sub Groups

Fig. 60: 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".



Add EtherCAT device at port B (E-Bus) of Term 1 (EK1100)

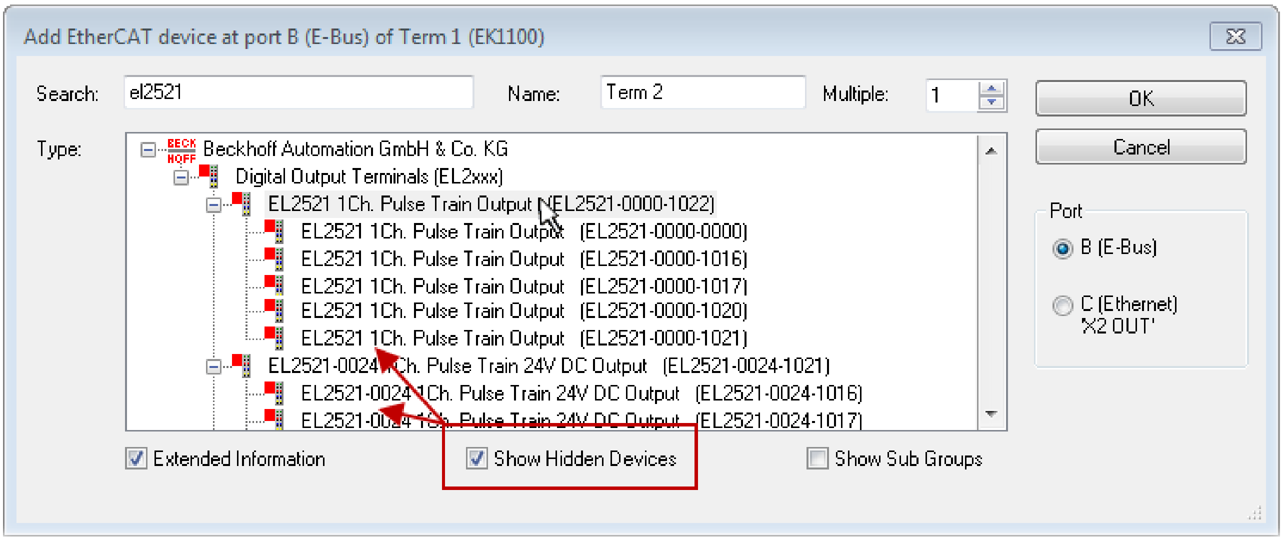
Search: el2521
Name: Term 2
Multiple: 1

Type:
- Beckhoff Automation GmbH & Co. KG
  - Digital Output Terminals (EL2xxx)
    - EL2521 1Ch. Pulse Train Output (EL2521-0000-1022)
    - EL2521-0024 1Ch. Pulse Train 24V DC Output (EL2521-0024-1021)
    - EL2521-0025 1Ch. Pulse Train 24V DC Output negative (EL2521-0025-1021)
    - EL2521-0124 1Ch. Pulse Train 24V DC Output Capture/Compare (EL2521-0124-0020)
    - EL2521-1001 1Ch. Pulse Train Output (EL2521-1001-1020)

☑ Extended Information □ Show Hidden Devices ☑ Show Sub Groups

Fig. 61: 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".



Add EtherCAT device at port B (E-Bus) of Term 1 (EK1100)

Search: el2521

Name: Term 2

Multiple: 1

Type:

- Beckhoff Automation GmbH & Co. KG
  - Digital Output Terminals (EL2xxx)
    - EL2521 1Ch. Pulse Train Output (EL2521-0000-1022)
      - EL2521 1Ch. Pulse Train Output (EL2521-0000-0000)
      - EL2521 1Ch. Pulse Train Output (EL2521-0000-1016)
      - EL2521 1Ch. Pulse Train Output (EL2521-0000-1017)
      - EL2521 1Ch. Pulse Train Output (EL2521-0000-1020)
      - EL2521 1Ch. Pulse Train Output (EL2521-0000-1021)
    - EL2521-0024 1Ch. Pulse Train 24V DC Output (EL2521-0024-1021)
      - EL2521-0024 1Ch. Pulse Train 24V DC Output (EL2521-0024-1016)
      - EL2521-0024 1Ch. Pulse Train 24V DC Output (EL2521-0024-1017)

Port

● B (E-Bus)

○ C (Ethernet) 'X2 OUT'

☑ Extended Information ☑ Show Hidden Devices □ Show Sub Groups

Fig. 62: 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.

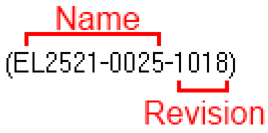


Fig. 63: Name/revision of the terminal

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, ...

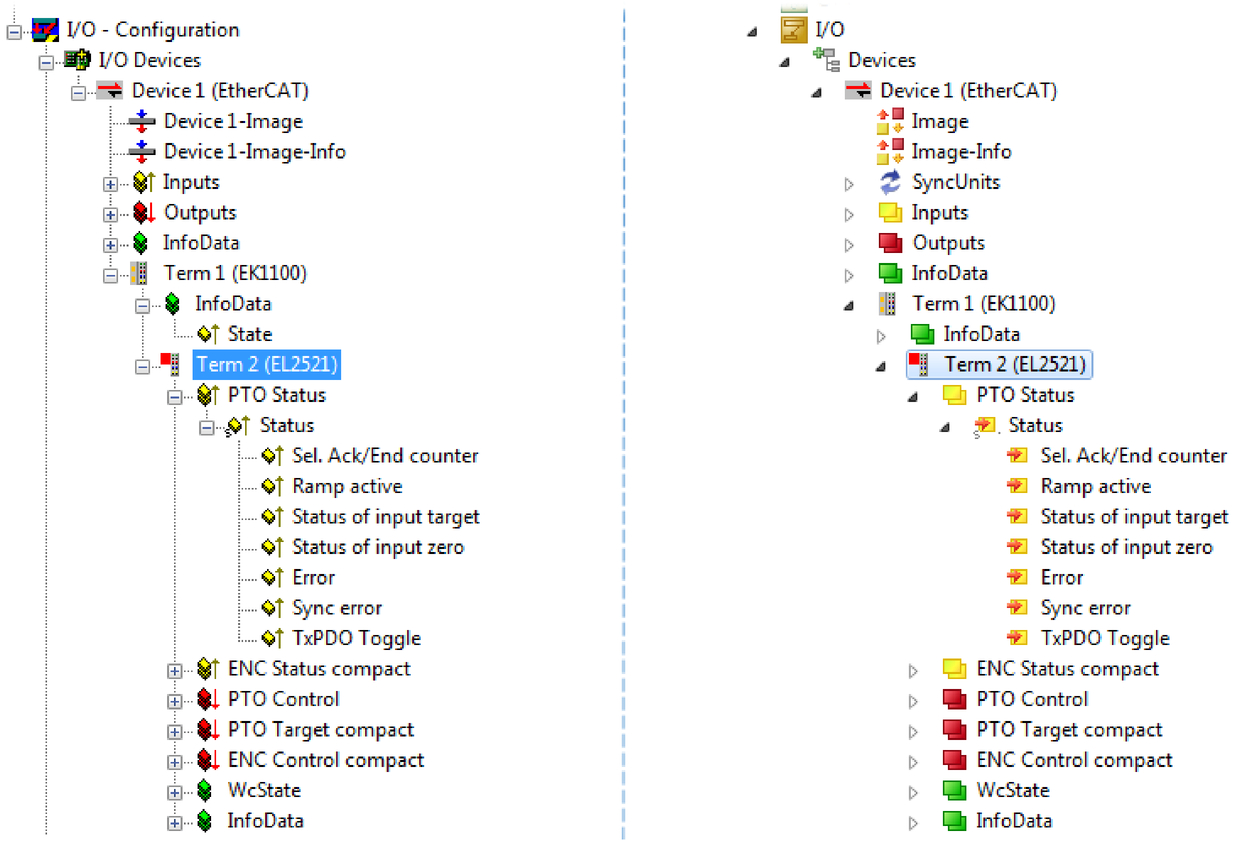


Fig. 64: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)

5.1.1.6 ONLINE configuration creation

Detecting/scanning of the EtherCAT device

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

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

TwinCAT can be set into this mode:

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

Online scanning in Config mode

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

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

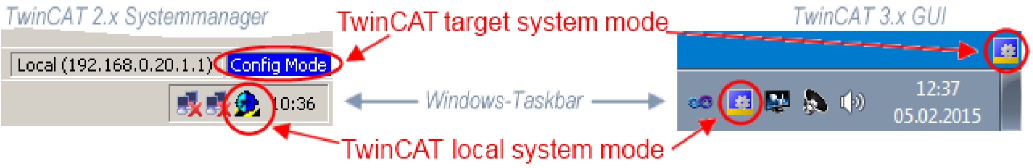


Fig. 65: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

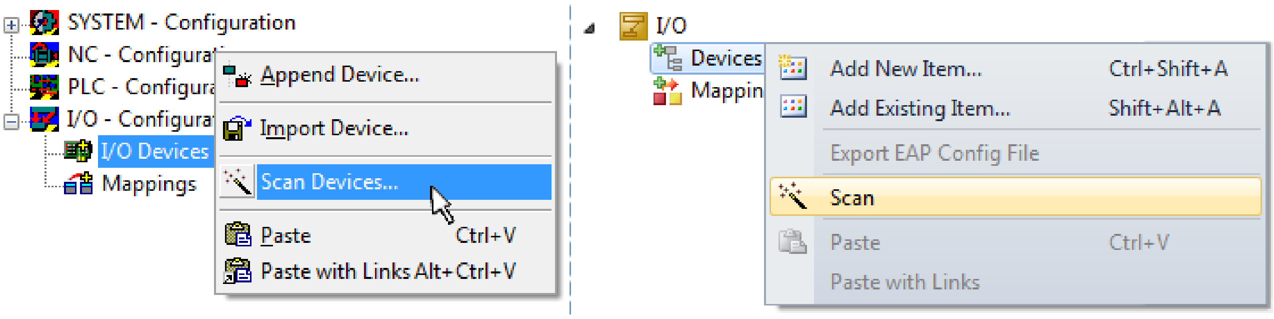
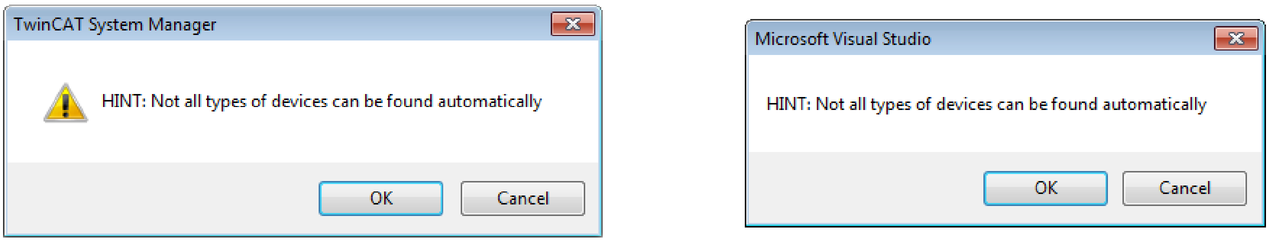


Fig. 66: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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.


TwinCAT System Manager

HINT: Not all types of devices can be found automatically

OK Cancel

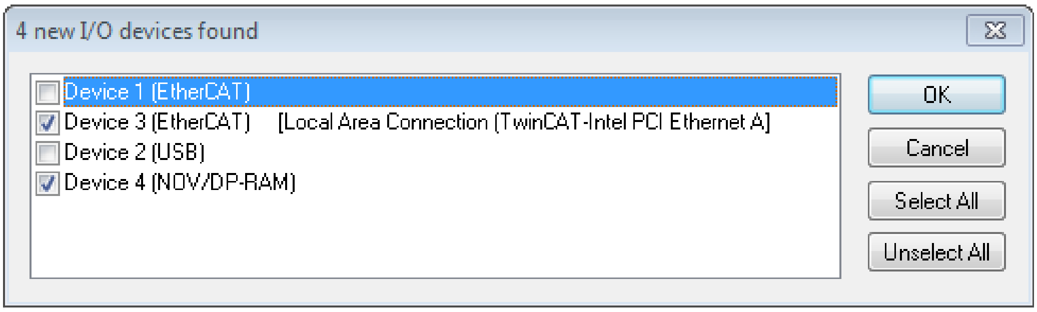
Microsoft Visual Studio

HINT: Not all types of devices can be found automatically

OK Cancel

Fig. 67: 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” .


4 new I/O devices found

□ Device 1 (EtherCAT)
☑ Device 3 (EtherCAT) [Local Area Connection (TwinCAT-Intel PCI Ethernet A]
□ Device 2 (USB)
☑ Device 4 (NOV/DP-RAM)

Fig. 68: 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 [▶ 59].

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.

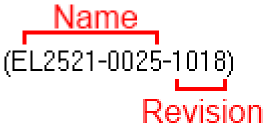


Fig. 69: Example default state

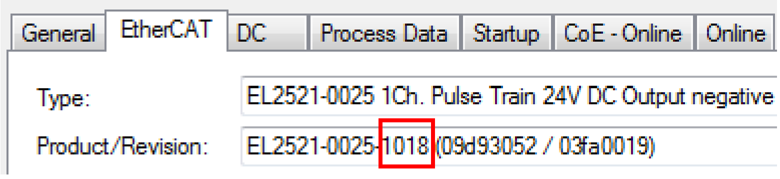
NOTE

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 [▶ 80] 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:



Type: EL2521-0025 1Ch. Pulse Train 24V DC Output negative

Product/Revision: EL2521-0025-1018 (09d93052 / 03fa0019)

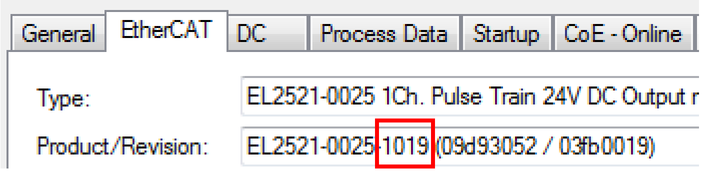
Fig. 70: Installing EthetCAT terminal with revision -1018

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 [▶ 80] 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:



Type: EL2521-0025 1Ch. Pulse Train 24V DC Output r

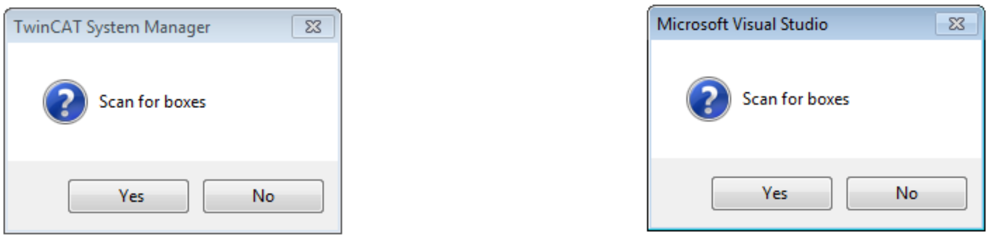
Product/Revision: EL2521-0025-1019 (09d93052 / 03fb0019)

Fig. 71: 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.



Scan for boxes

Scan for boxes

Fig. 72: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

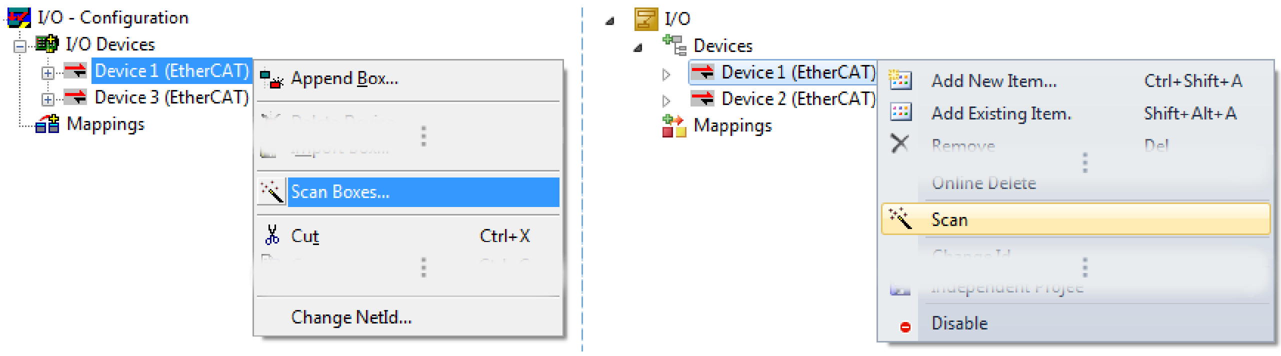


Fig. 73: Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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.

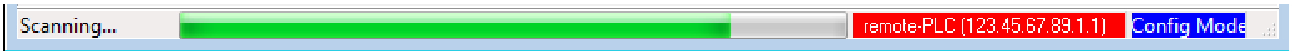
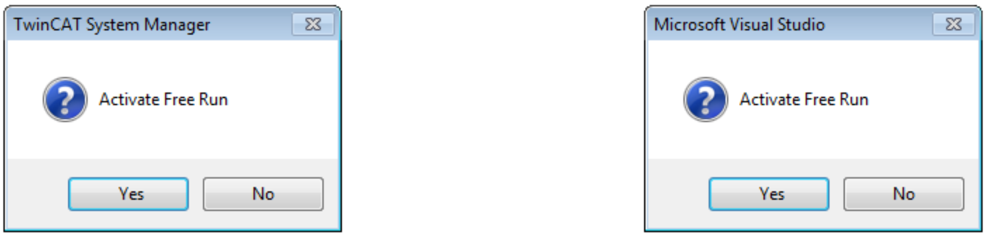


Fig. 74: *Scan progressexemplary by TwinCAT 2*

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



Activate Free Run

Activate Free Run

Fig. 75: 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).

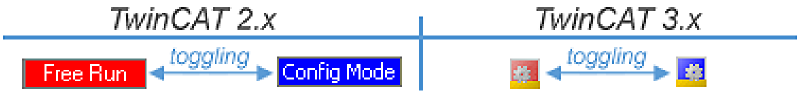


Fig. 76: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar

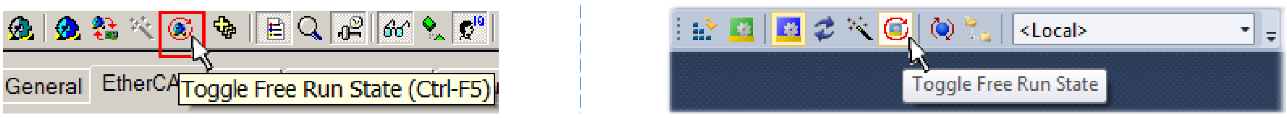
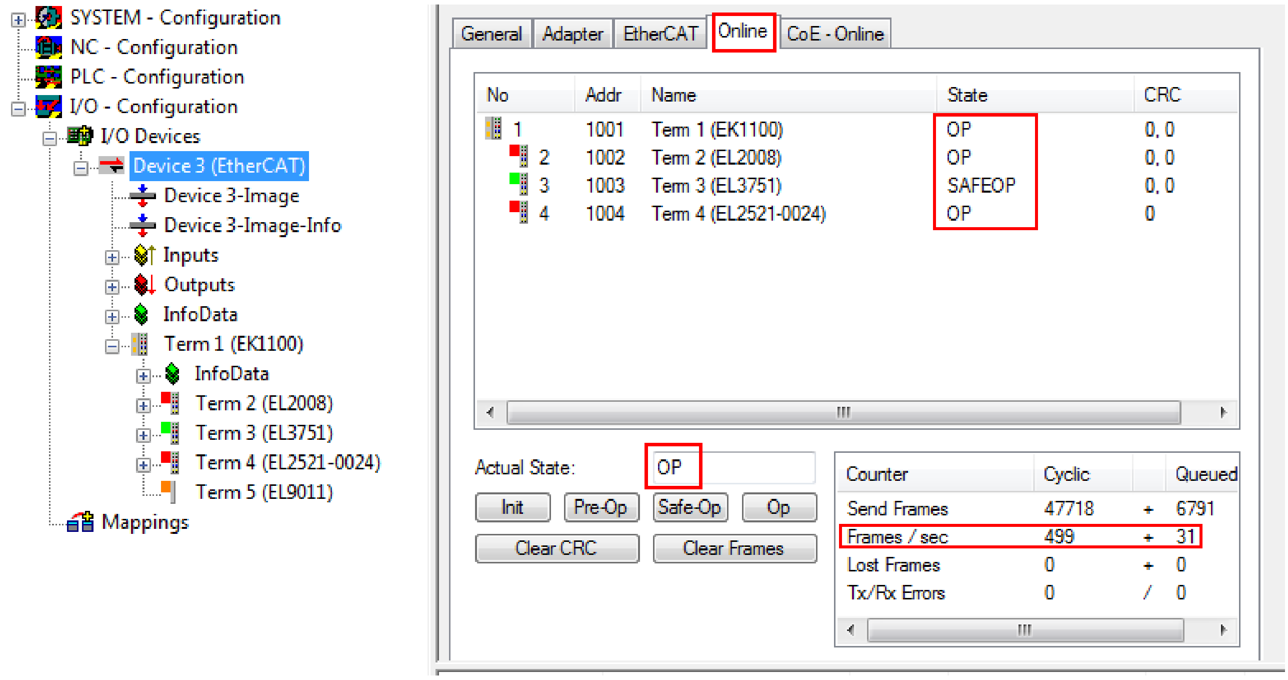


Fig. 77: TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3)

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



No	Addr	Name	State	CRC
1	1001	Term 1 (EK1100)	OP	0, 0
2	1002	Term 2 (EL2008)	OP	0, 0
3	1003	Term 3 (EL3751)	SAFEOP	0, 0
4	1004	Term 4 (EL2521-0024)	OP	0

Actual State: OP

Counter	Cyclic		Queued
Send Frames	47718	+	6791
Frames / sec	499	+	31
Lost Frames	0	+	0
Tx/Rx Errors	0	/	0

Fig. 78: 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 [▶ 70].

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.

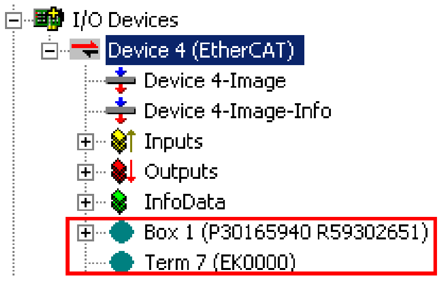


Fig. 79: Faulty identification

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.

TwinCAT System Manager

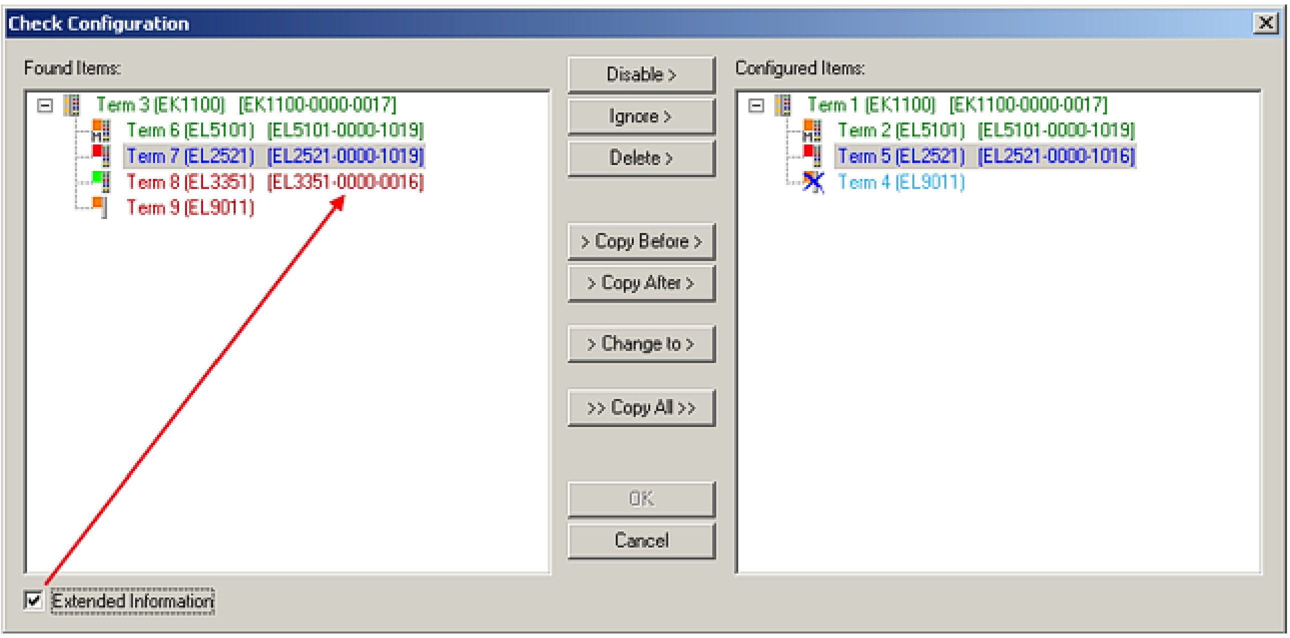
Configuration is identical

Microsoft Visual Studio

Configuration is identical

Fig. 80: 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.



Check Configuration

Found Items:

- Term 3 (EK1100) [EK1100-0000-0017]
  - Term 6 (EL5101) [EL5101-0000-1019]
  - Term 7 (EL2521) [EL2521-0000-1019]
  - Term 8 (EL3351) [EL3351-0000-0016]
  - Term 9 (EL9011)

Configured Items:

- Term 1 (EK1100) [EK1100-0000-0017]
  - Term 2 (EL5101) [EL5101-0000-1019]
  - Term 5 (EL2521) [EL2521-0000-1016]
  - Term 4 (EL9011)

☑ Extended Information

Fig. 81: 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.

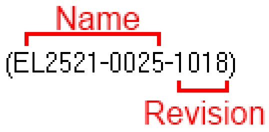


Fig. 82: Name/revision of the terminal

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, ...

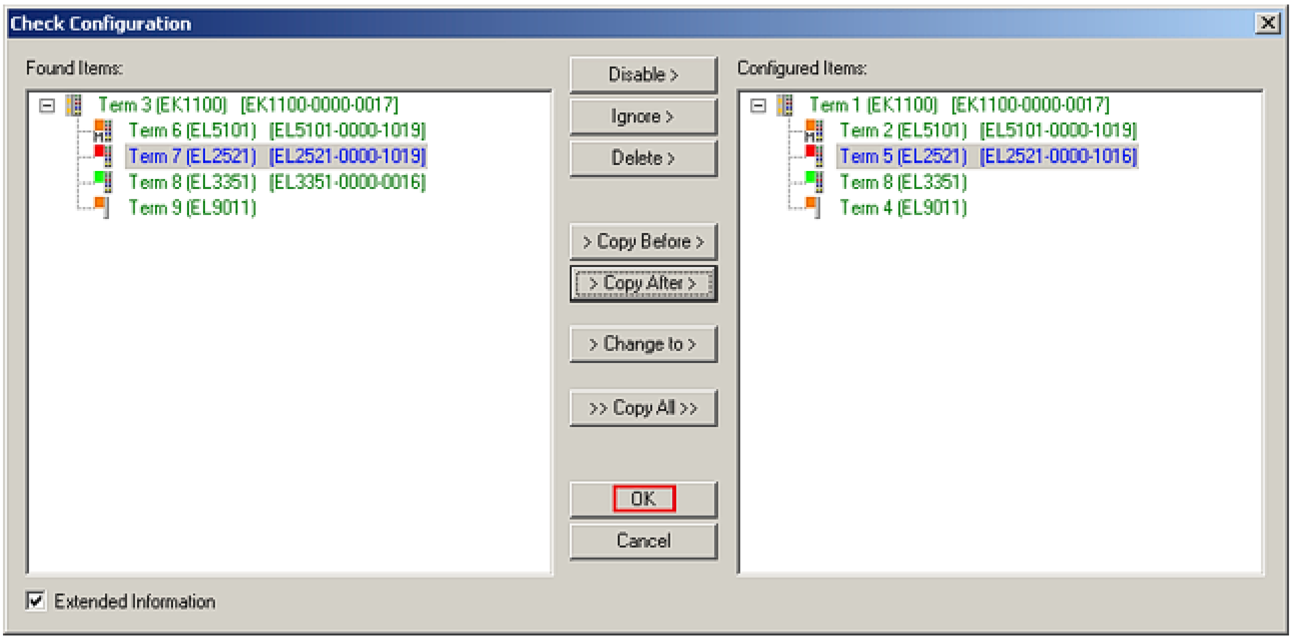

Check Configuration
Found Items:
Term 3 (EK1100) [EK1100-0000-0017]
Term 6 (EL5101) [EL5101-0000-1019]
Term 7 (EL2521) [EL2521-0000-1019]
Term 8 (EL3351) [EL3351-0000-0016]
Term 9 (EL9011)
Disable >
Ignore >
Delete >
> Copy Before >
> Copy After >
> Change to >
>> Copy All >>
OK
Cancel
Configured Items:
Term 1 (EK1100) [EK1100-0000-0017]
Term 2 (EL5101) [EL5101-0000-1019]
Term 5 (EL2521) [EL2521-0000-1016]
Term 8 (EL3351)
Term 4 (EL9011)
☑ Extended Information

Fig. 83: 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.

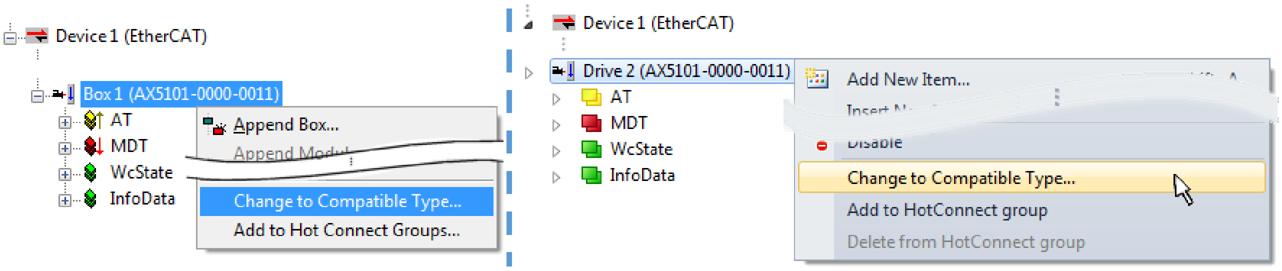


Fig. 84: Dialog “Change to Compatible Type…” (left: TwinCAT 2; right: TwinCAT 3)

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

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

- FMMU (additional ones are allowed)

- SyncManager (SM, additional ones are allowed)

- EoE (attributes MAC, IP)

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

- FoE

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

This function is preferably to be used on AX5000 devices.

Change to Alternative Type

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

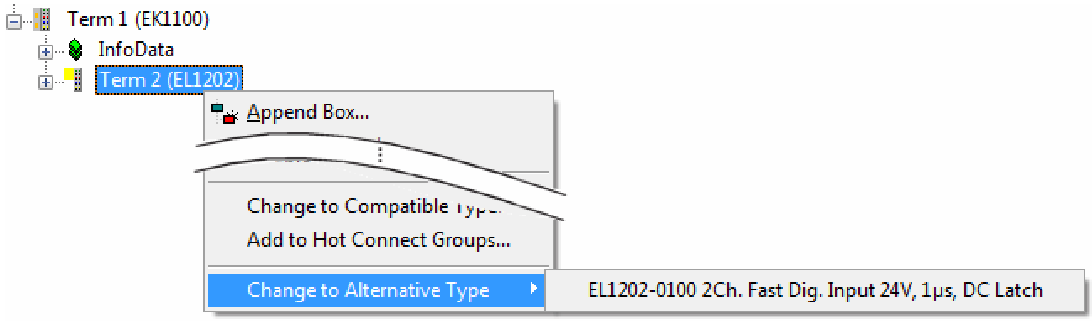

Term 1 (EK1100)
InfoData
Term 2 (EL1202)
Append Box...
Change to Compatible Type...
Add to Hot Connect Groups...
Change to Alternative Type
EL1202-0100 2Ch. Fast Dig. Input 24V, 1µs, DC Latch

Fig. 85: TwinCAT 2 Dialog Change to Alternative Type

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

5.1.1.7 EtherCAT subscriber configuration

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

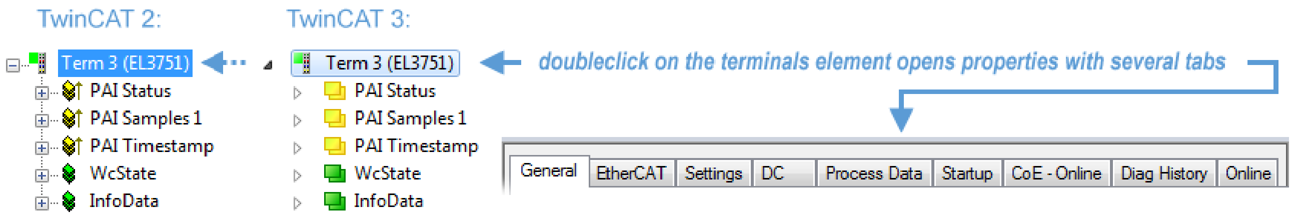
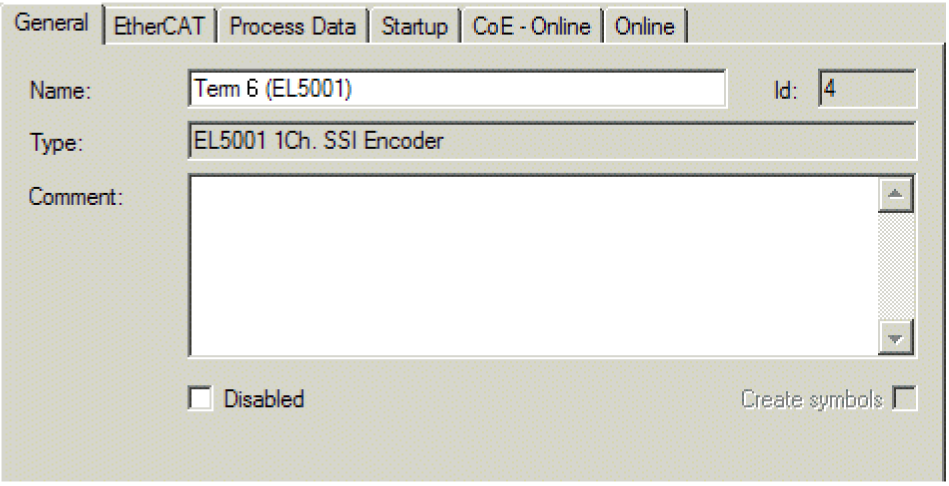


Fig. 86: Branch element as terminal EL3751

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



Name: Term 6 (EL5001) Id: 4

Type: EL5001 1Ch. SSI Encoder

Comment:

☐ Disabled Create symbols ☐

Fig. 87: “General” tab

Name Name of the EtherCAT device

Id Number of the EtherCAT device

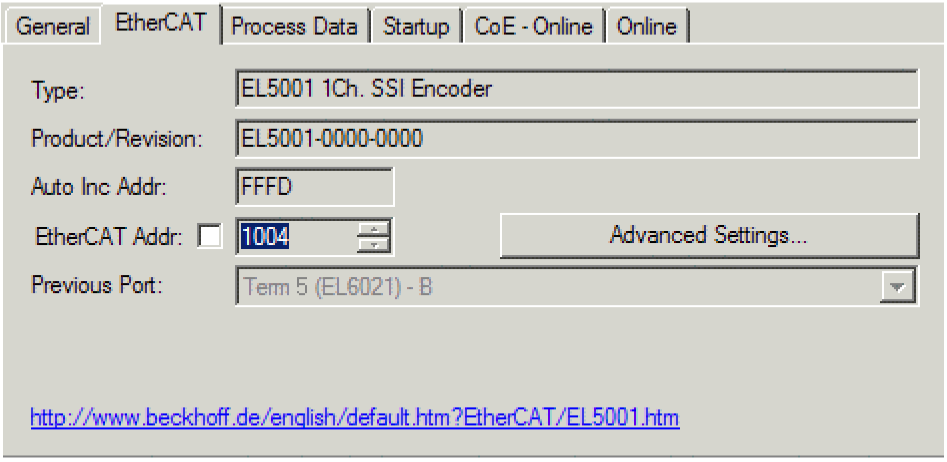
Type EtherCAT device type

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



Type: EL5001 1Ch. SSI Encoder

Product/Revision: EL5001-0000-0000

Auto Inc Addr: FFFD

EtherCAT Addr: □ 1004

Previous Port: Term 5 (EL6021) - B

http://www.beckhoff.de/english/default.htm?EtherCAT/EL5001.htm

Fig. 88: "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.

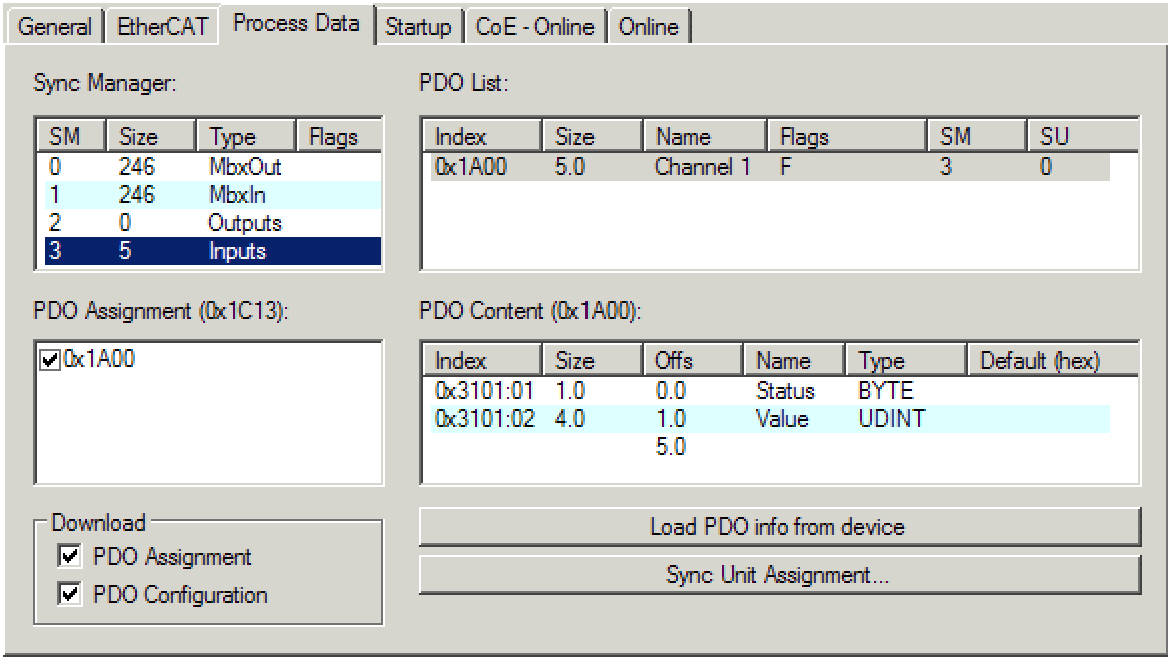
Previous Port Name and port of the EtherCAT device to which this device is connected. If it is possible to connect this device with another one without changing the order of the EtherCAT devices in the communication ring, then this combination field is activated and the EtherCAT device to which this 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.



Sync Manager:

SM	Size	Type	Flags
0	246	MbxOut	
1	246	MbxIn	
2	0	Outputs	
3	5	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	5.0	Channel 1	F	3	0

PDO Assignment (0x1C13):

☑ 0x1A00

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (hex)
0x3101:01	1.0	0.0	Status	BYTE	
0x3101:02	4.0	1.0	Value	UDINT	
		5.0			

Download

☑ PDO Assignment

☑ PDO Configuration

Fig. 89: “Process Data” tab

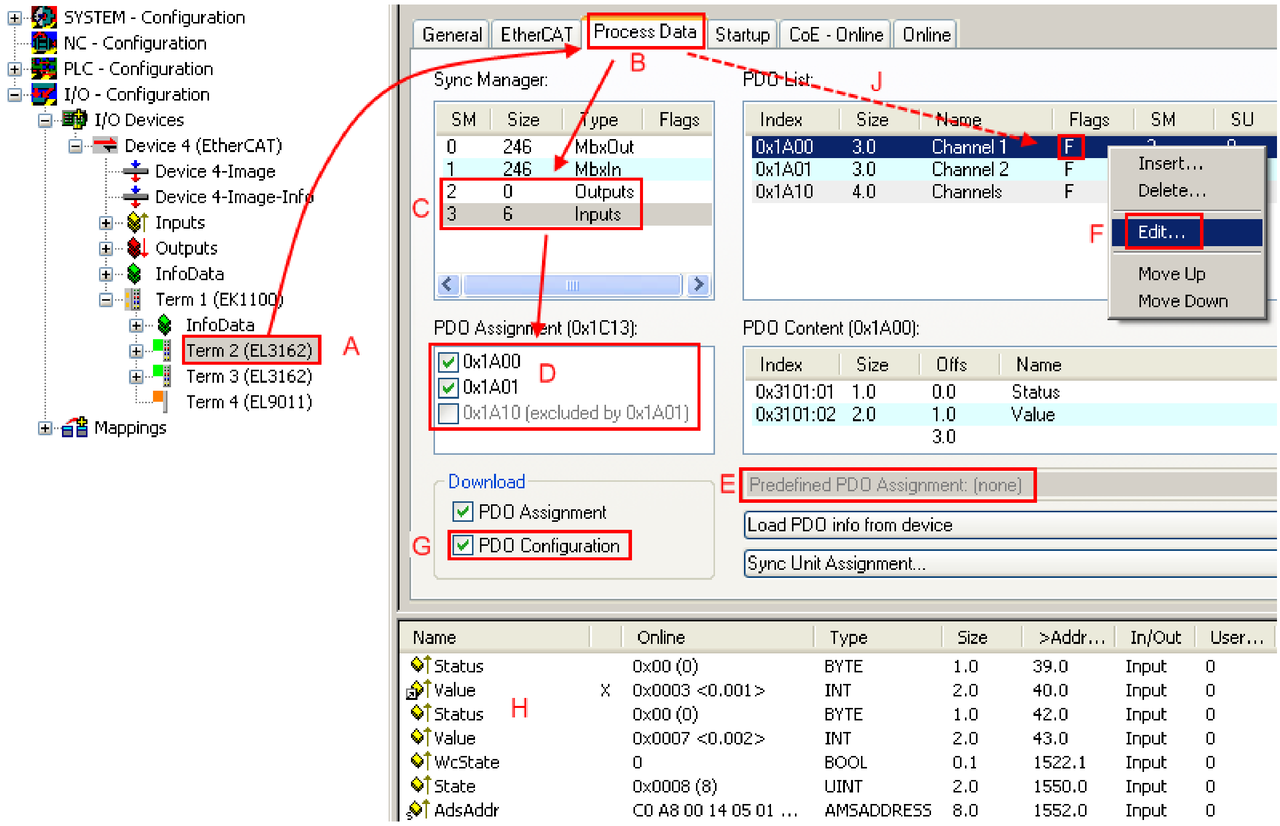
The process data (PDOs) transferred by an EtherCAT slave during each cycle are user data which the application expects to be updated cyclically or which are sent to the slave. To this end the EtherCAT master (Beckhoff TwinCAT) parameterizes each EtherCAT slave during the start-up phase to define which 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”).



Sync Manager:

SM	Size	Type	Flags
0	246	MbxOut	
1	246	MbxIn	
2	0	Outputs	
3	6	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	3.0	Channel 1	F		
0x1A01	3.0	Channel 2	F		
0x1A10	4.0	Channels	F		

PDO Assignment (0x1C13):

☑ 0x1A00
☑ 0x1A01
☐ 0x1A10 (excluded by 0x1A01)

PDO Content (0x1A00):

Index	Size	Offs	Name
0x3101:01	1.0	0.0	Status
0x3101:02	2.0	1.0	Value
		3.0	

Download
☑ PDO Assignment
☑ PDO Configuration

Predefined PDO Assignment: (none)
Load PDO info from device
Sync Unit Assignment...

Name		Online	Type	Size	>Addr...	In/Out	User...
Status		0x00 (0)	BYTE	1.0	39.0	Input	0
Value	X	0x0003 <0.001>	INT	2.0	40.0	Input	0
Status		0x00 (0)	BYTE	1.0	42.0	Input	0
Value		0x0007 <0.002>	INT	2.0	43.0	Input	0
WcState		0	BOOL	0.1	1522.1	Input	0
State		0x0008 (8)	UINT	2.0	1550.0	Input	0
AdsAddr		C0 A8 00 14 05 01 ...	AMSADDRESS	8.0	1552.0	Input	0

Fig. 90: Configuring the process data

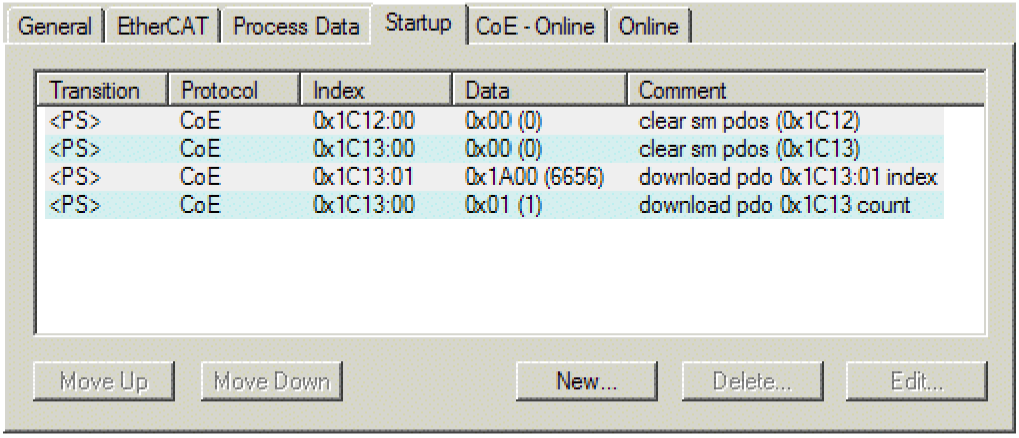
Manual modification of the process data

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 [▶ 91] 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.



Transition	Protocol	Index	Data	Comment
<PS>	CoE	0x1C12:00	0x00 (0)	clear sm pdos (0x1C12)
<PS>	CoE	0x1C13:00	0x00 (0)	clear sm pdos (0x1C13)
<PS>	CoE	0x1C13:01	0x1A00 (6656)	download pdo 0x1C13:01 index
<PS>	CoE	0x1C13:00	0x01 (1)	download pdo 0x1C13 count

Fig. 91: “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.

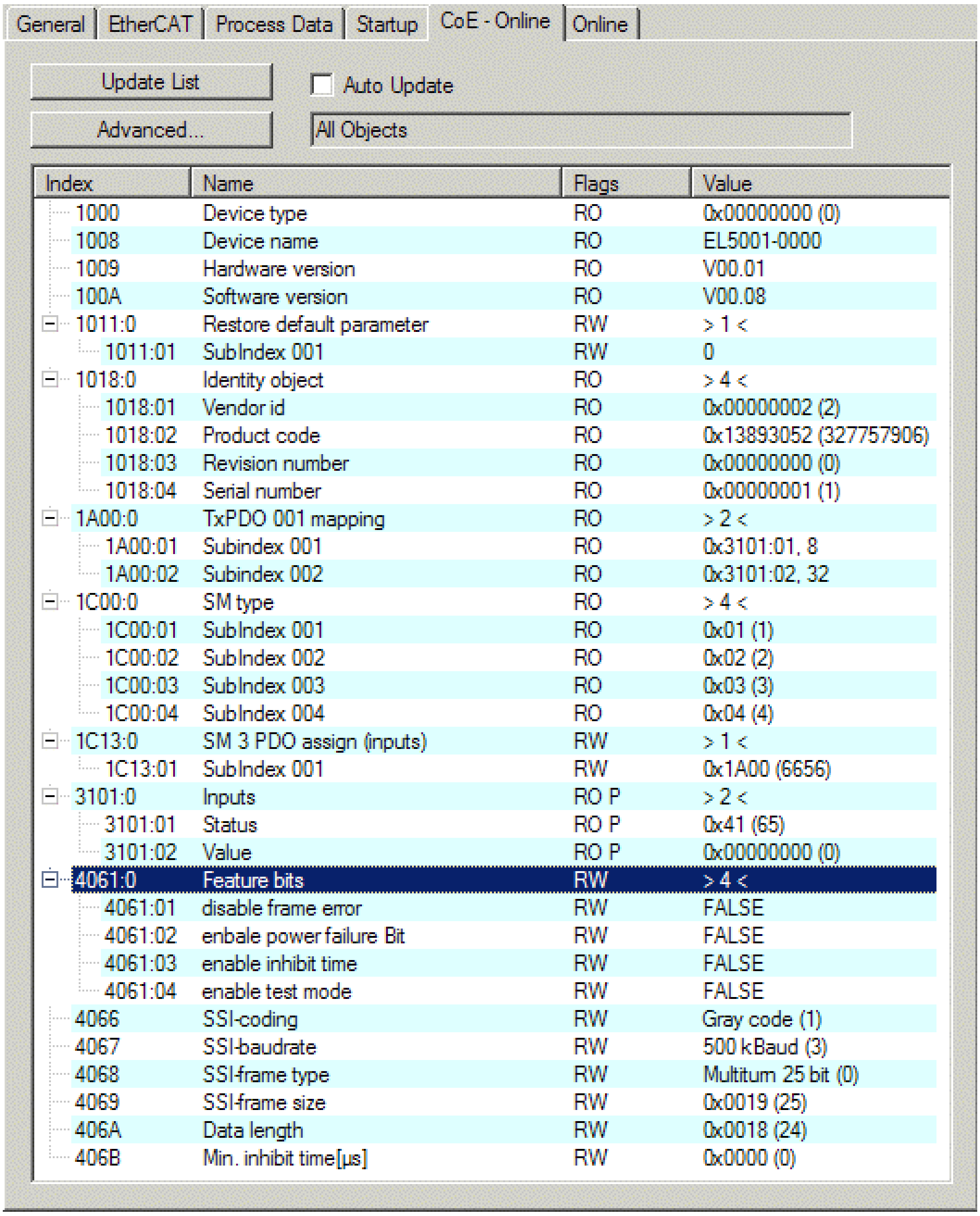
New This button adds a new mailbox download request to be sent during startup.

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.



Index	Name	Flags	Value
1000	Device type	RO	0x00000000 (0)
1008	Device name	RO	EL5001-0000
1009	Hardware version	RO	V00.01
100A	Software version	RO	V00.08
1011:0	Restore default parameter	RW	> 1 <
1011:01	SubIndex 001	RW	0
1018:0	Identity object	RO	> 4 <
1018:01	Vendor id	RO	0x00000002 (2)
1018:02	Product code	RO	0x13893052 (327757906)
1018:03	Revision number	RO	0x00000000 (0)
1018:04	Serial number	RO	0x00000001 (1)
1A00:0	TxPDO 001 mapping	RO	> 2 <
1A00:01	Subindex 001	RO	0x3101:01, 8
1A00:02	Subindex 002	RO	0x3101:02, 32
1C00:0	SM type	RO	> 4 <
1C00:01	SubIndex 001	RO	0x01 (1)
1C00:02	SubIndex 002	RO	0x02 (2)
1C00:03	SubIndex 003	RO	0x03 (3)
1C00:04	SubIndex 004	RO	0x04 (4)
1C13:0	SM 3 PDO assign (inputs)	RW	> 1 <
1C13:01	SubIndex 001	RW	0x1A00 (6656)
3101:0	Inputs	RO P	> 2 <
3101:01	Status	RO P	0x41 (65)
3101:02	Value	RO P	0x00000000 (0)
4061:0	Feature bits	RW	> 4 <
4061:01	disable frame error	RW	FALSE
4061:02	enbale power failure Bit	RW	FALSE
4061:03	enable inhibit time	RW	FALSE
4061:04	enable test mode	RW	FALSE
4066	SSI-coding	RW	Gray code (1)
4067	SSI-baudrate	RW	500 kBaud (3)
4068	SSI-frame type	RW	Multitum 25 bit (0)
4069	SSI-frame size	RW	0x0019 (25)
406A	Data length	RW	0x0018 (24)
406B	Min. inhibit time[µs]	RW	0x0000 (0)

Fig. 92: “CoE - Online” tab

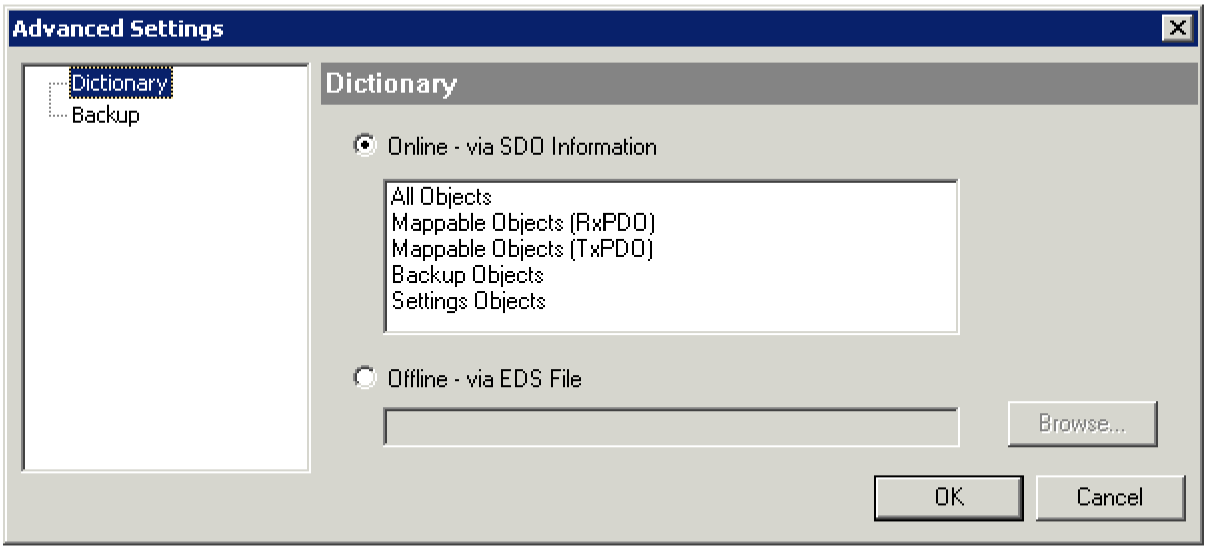
Object list display

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.



Advanced Settings

Dictionary
Backup

Dictionary

● Online - via SDO Information

All Objects
Mappable Objects (RxPDO)
Mappable Objects (TxPDO)
Backup Objects
Settings Objects

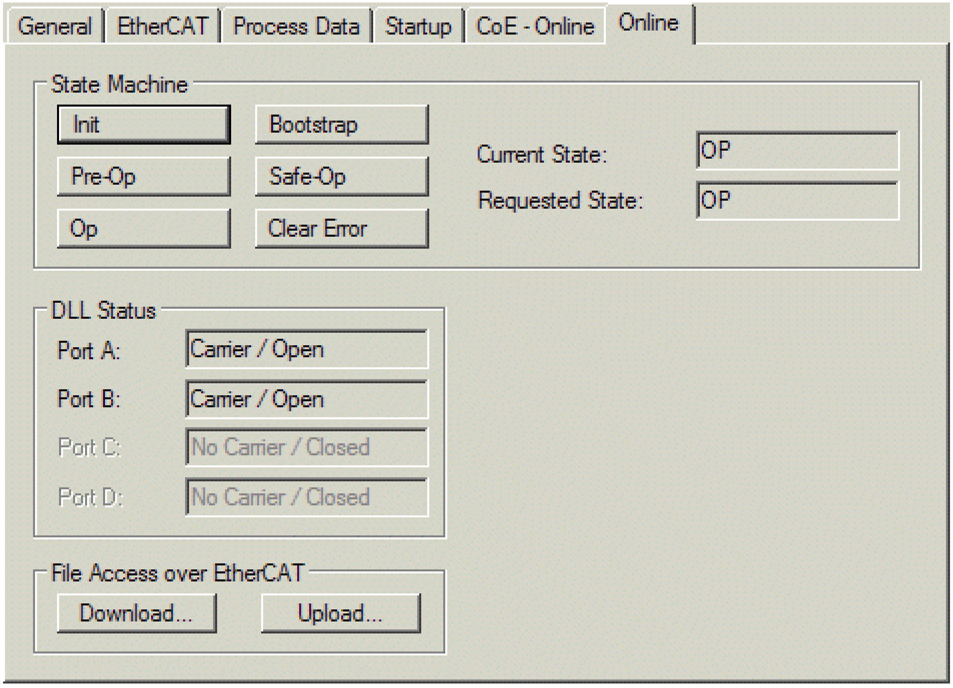
○ Offline - via EDS File

Fig. 93: 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



General | EtherCAT | Process Data | Startup | CoE - Online | Online

State Machine

Current State: OP

Requested State: OP

DLL Status

Port A: Carrier / Open

Port B: Carrier / Open

Port C: No Carrier / Closed

Port D: No Carrier / Closed

File Access over EtherCAT

Fig. 94: “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:

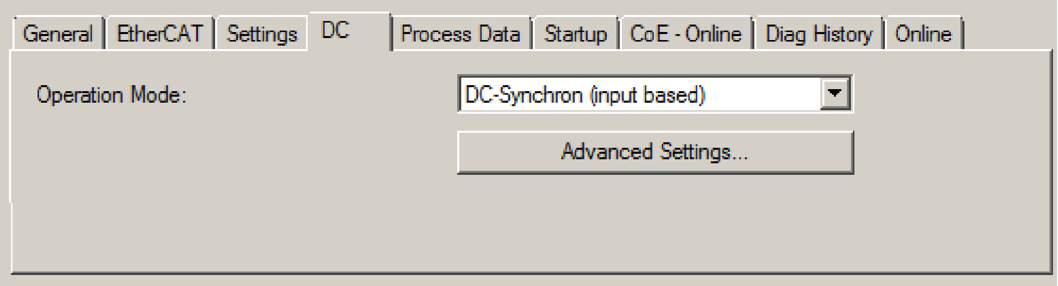
Status	Description
No Carrier / Open	No carrier signal is available at the port, but the port is open.
No Carrier / Closed	No carrier signal is available at the port, and the port is closed.
Carrier / Open	A carrier signal is available at the port, and the port is open.
Carrier / Closed	A carrier signal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



Operation Mode: DC-Synchron (input based)

Fig. 95: “DC” tab (Distributed Clocks)

Operation Mode Options (optional):

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

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

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

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

5.1.1.7.1 Detailed description of Process Data tab

Sync Manager

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

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

PDO Assignment

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

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

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

Activation of PDO assignment

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

a) the EtherCAT slave has to run through the PS status transition cycle (from pre-operational to safe-operational) once (see Online tab [▶ 89]),

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 [▸ 86] tab.

PDO Configuration

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

5.1.1.8 Import/Export of EtherCAT devices with SCI and XTI

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

5.1.1.8.1 Basic principles

An EtherCAT slave is basically parameterized through the following elements:

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

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

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

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

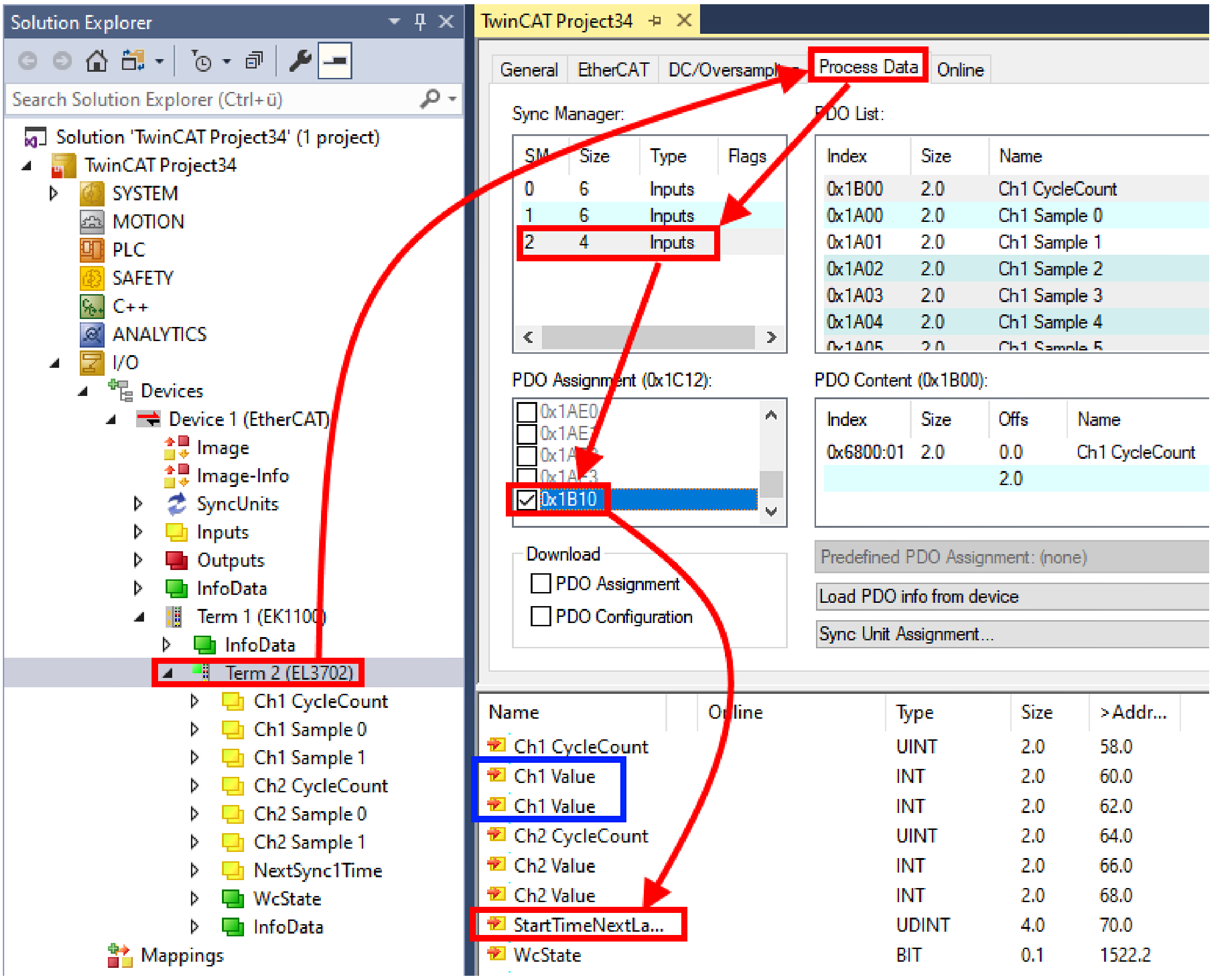
It may become necessary to reuse the modified slave in other projects in this way, without having to make equivalent configuration changes to the slave again. To accomplish this, proceed as follows:

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

TwinCAT offers two methods for this purpose:

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

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



Sync Manager:

SM	Size	Type	Flags
0	6	Inputs	
1	6	Inputs	
2	4	Inputs	

PDO List:

Index	Size	Name
0x1B00	2.0	Ch1 CycleCount
0x1A00	2.0	Ch1 Sample 0
0x1A01	2.0	Ch1 Sample 1
0x1A02	2.0	Ch1 Sample 2
0x1A03	2.0	Ch1 Sample 3
0x1A04	2.0	Ch1 Sample 4
0x1A05	2.0	Ch1 Sample 5

PDO Assignment (0x1C12):

PDO Content (0x1B00):

Index	Size	Offs	Name
0x6800:01	2.0	0.0	Ch1 CycleCount
		2.0	

Download

□ PDO Assignment

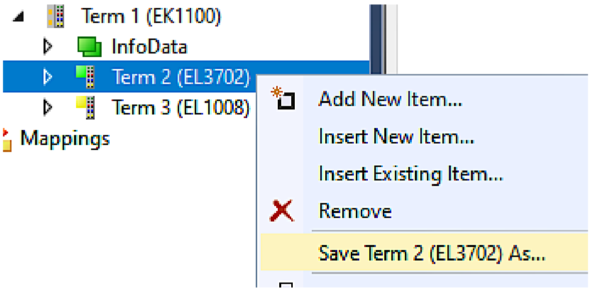
□ PDO Configuration

Name	Online	Type	Size	>Addr...
Ch1 CycleCount		UINT	2.0	58.0
Ch1 Value		INT	2.0	60.0
Ch1 Value		INT	2.0	62.0
Ch2 CycleCount		UINT	2.0	64.0
Ch2 Value		INT	2.0	66.0
Ch2 Value		INT	2.0	68.0
StartTimeNextLa...		UDINT	4.0	70.0
WcState		BIT	0.1	1522.2

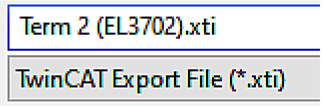
The two methods for exporting and importing the modified terminal referred to above are demonstrated below.

5.1.1.8.2 Procedure within TwinCAT with xti files

Each IO device can be exported/saved individually:

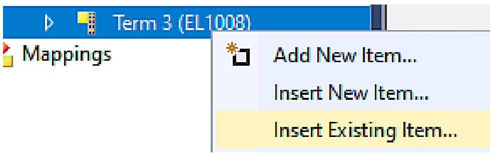


The xti file can be stored:


Term 2 (EL3702).xti

TwinCAT Export File (*.xti)

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



5.1.1.8.3 Procedure within and outside TwinCAT with sci file

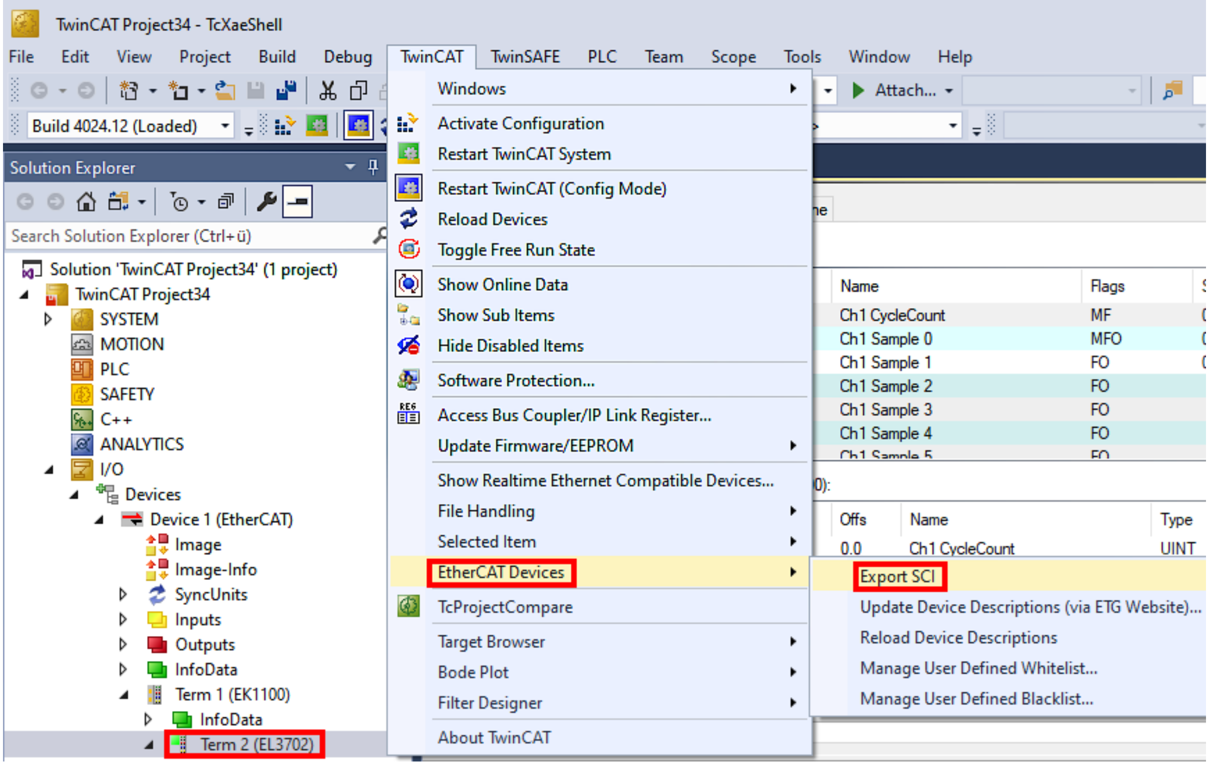
Note regarding availability (2021/01)

The SCI method is available from TwinCAT 3.1 build 4024.14.

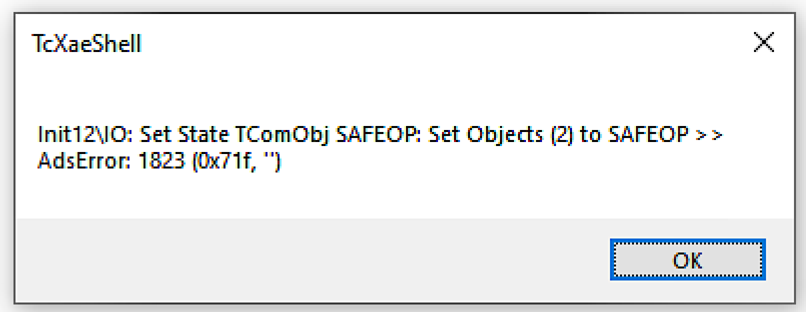
The Slave Configuration Information (SCI) describes a specific complete configuration for an EtherCAT slave (terminal, box, drive...) based on the setting options of the device description file (ESI, EtherCAT Slave Information). That is, it includes PDO, CoE, synchronization.

Export:

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



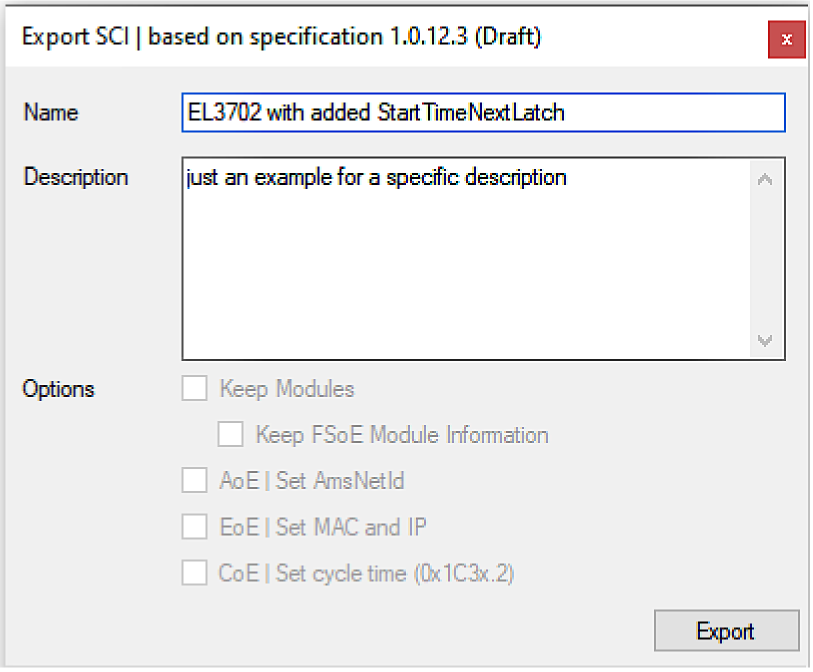
- If TwinCAT is offline (i.e. if there is no connection to an actual running controller) a warning message may appear, because after executing the function the system attempts to reload the EtherCAT segment. However, in this case this is not relevant for the result and can be acknowledged by clicking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



TcXaeShell

Init12\IO: Set State TComObj SAFEOP: Set Objects (2) to SAFEOP > >
AdsError: 1823 (0x71f, '')

- A description may also be provided:



Export SCI | based on specification 1.0.12.3 (Draft)

Name: EL3702 with added StartTimeNextLatch

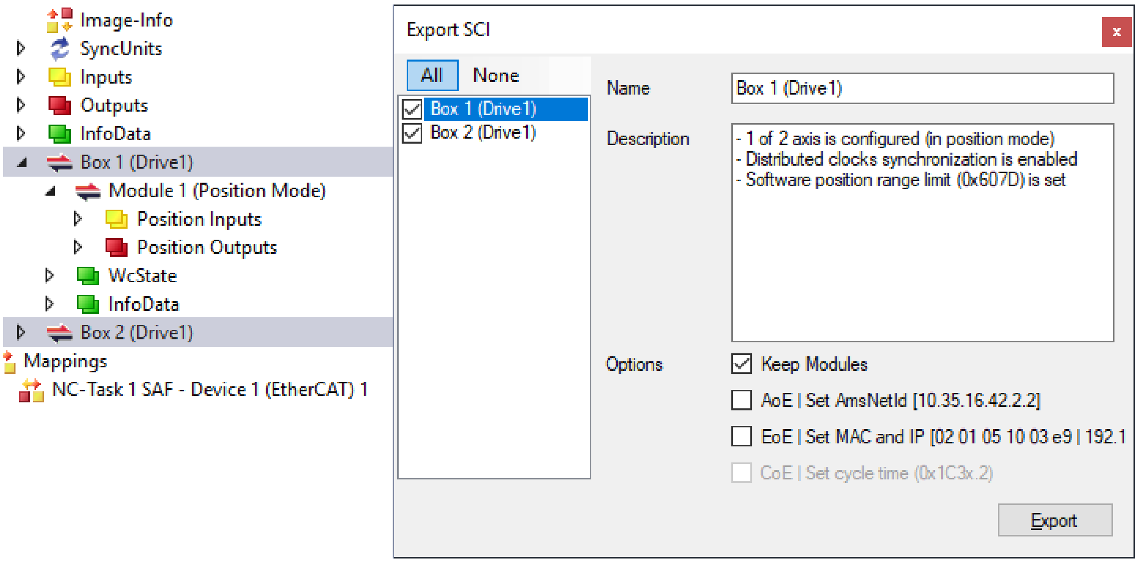
Description: just an example for a specific description

Options:
- □ Keep Modules
  - □ Keep FSoE Module Information
- □ AoE | Set AmsNetId
- □ EoE | Set MAC and IP
- □ CoE | Set cycle time (0x1C3x.2)

- Explanation of the dialog box:

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

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



☑ Box 1 (Drive1)
☑ Box 2 (Drive1)

Name: Box 1 (Drive1)

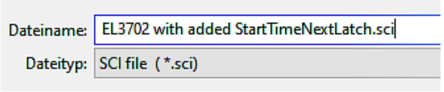
Description:
- 1 of 2 axis is configured (in position mode)
- Distributed clocks synchronization is enabled
- Software position range limit (0x607D) is set

Options:
- ☑ Keep Modules
- □ AoE | Set AmsNetId [10.35.16.42.2.2]
- □ EoE | Set MAC and IP [02 01 05 10 03 e9 | 192.1
- □ CoE | Set cycle time (0x1C3x.2)

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

- None:
  All slaves are deselected.

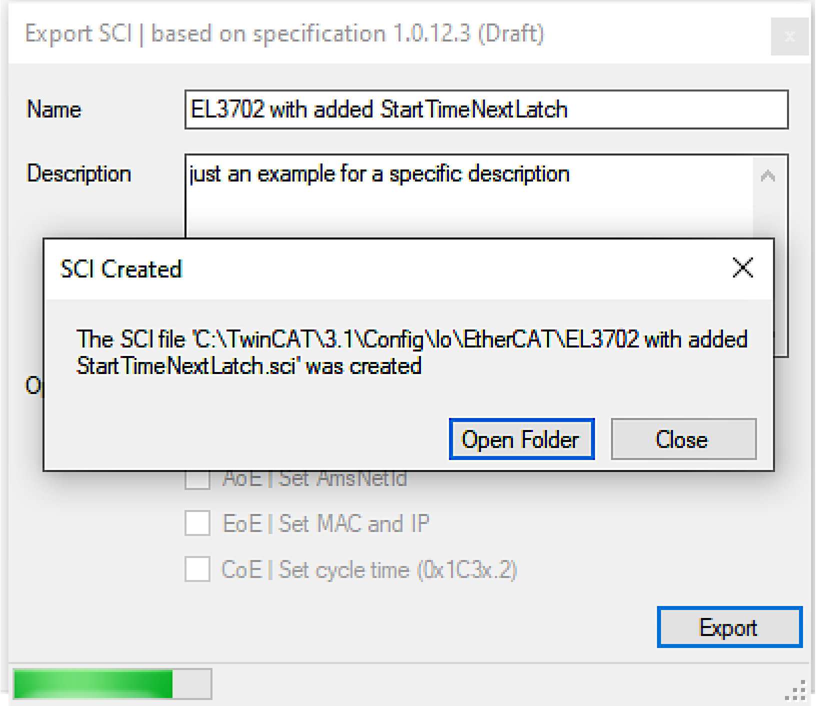
- The sci file can be saved locally:



Dateiname: EL3702 with added StartTimeNextLatch.sci

Dateityp: SCI file (*.sci)

- The export takes place:



Export SCI | based on specification 1.0.12.3 (Draft)

Name: EL3702 with added StartTimeNextLatch

Description: just an example for a specific description

SCI Created

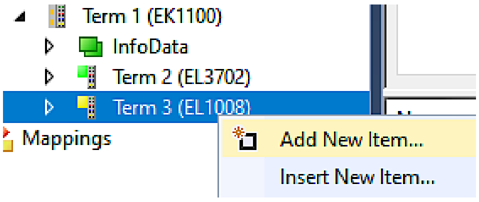
The SCI file 'C:\TwinCAT\3.1\Config\Io\EtherCAT\EL3702 with added StartTimeNextLatch.sci' was created

Import

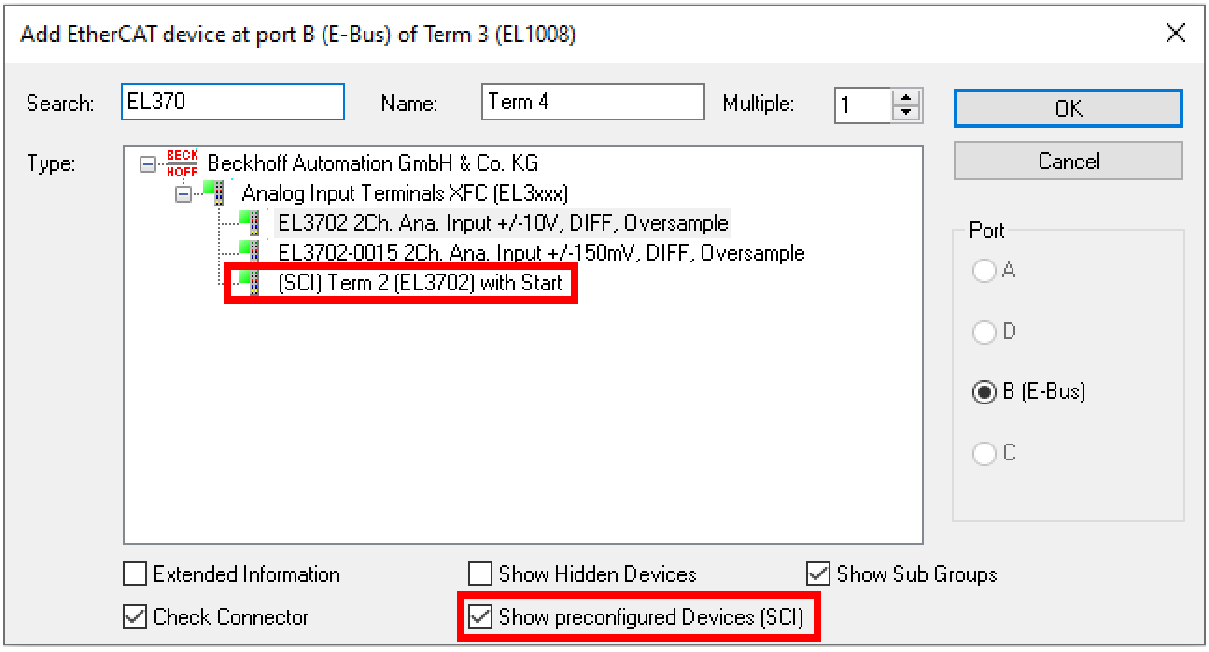
- An sci description can be inserted manually into the TwinCAT configuration like any normal Beckhoff device description.
- The sci file must be located in the TwinCAT ESI path, usually under: C:\TwinCAT\3.1\Config\Io\EtherCAT

EL3702 with added StartTimeNextLatch.sci 11.01.2021 13:29 SCI-Datei 6 KB

- Open the selection dialog:



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



Add EtherCAT device at port B (E-Bus) of Term 3 (EL1008)

Search: EL370 Name: Term 4 Multiple: 1

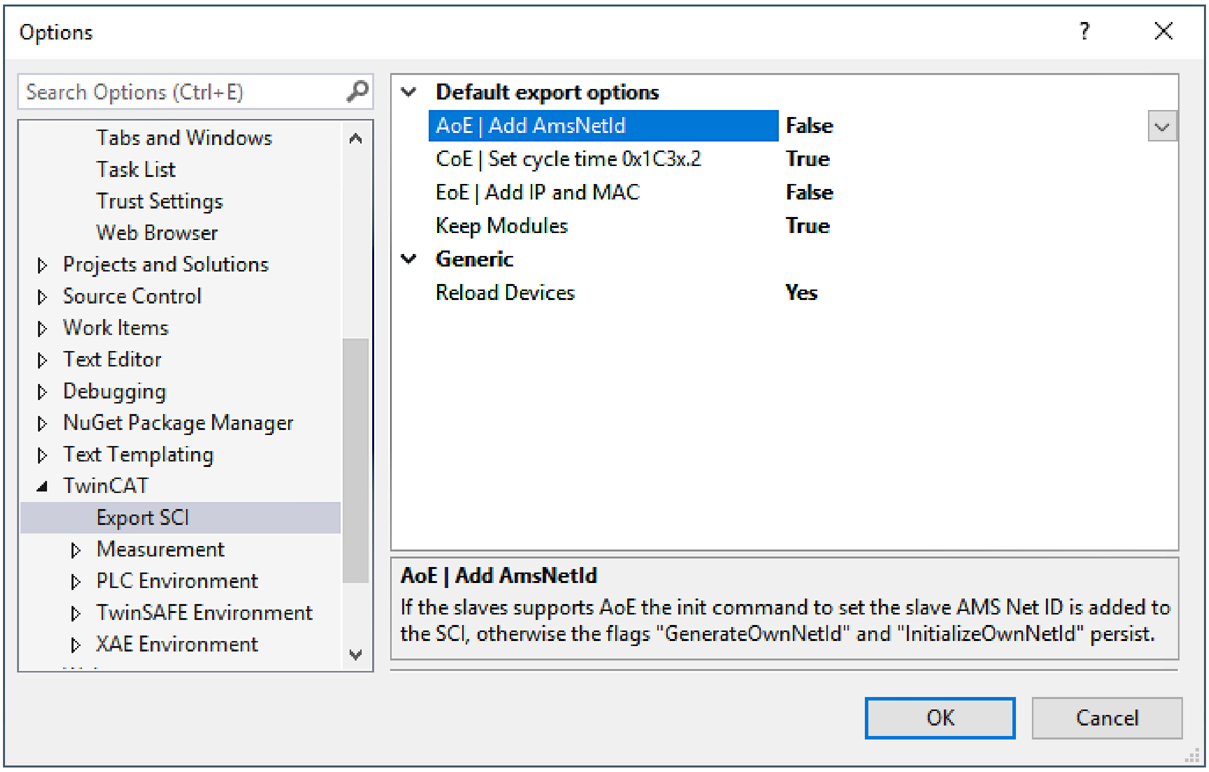
Type:
- Beckhoff Automation GmbH & Co. KG
  - Analog Input Terminals XFC (EL3xxx)
    - EL3702 2Ch. Ana. Input +/-10V, DIFF, Oversample
    - EL3702-0015 2Ch. Ana. Input +/-150mV, DIFF, Oversample
    - (SCI) Term 2 (EL3702) with Start

Port: ○ A ○ D ● B (E-Bus) ○ C

□ Extended Information □ Show Hidden Devices ☑ Show Sub Groups
☑ Check Connector ☑ Show preconfigured Devices (SCI)

Additional Notes

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



Options

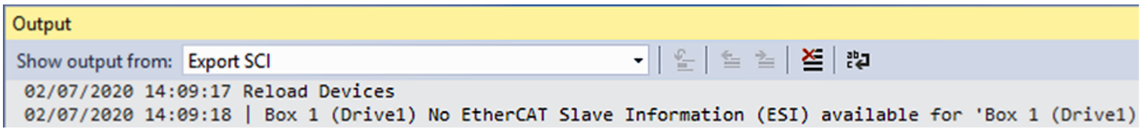
Default export options	
AoE | Add AmsNetId	False
CoE | Set cycle time 0x1C3x.2	True
EoE | Add IP and MAC	False
Keep Modules	True
Generic	
Reload Devices	Yes

AoE | Add AmsNetId
If the slaves supports AoE the init command to set the slave AMS Net ID is added to the SCI, otherwise the flags "GenerateOwnNetId" and "InitializeOwnNetId" persist.

Explanation of the settings:

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

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



```
02/07/2020 14:09:17 Reload Devices
02/07/2020 14:09:18 | Box 1 (Drive1) No EtherCAT Slave Information (ESI) available for 'Box 1 (Drive1)
```

5.1.2 TwinCAT Quick Start

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

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

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

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

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

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

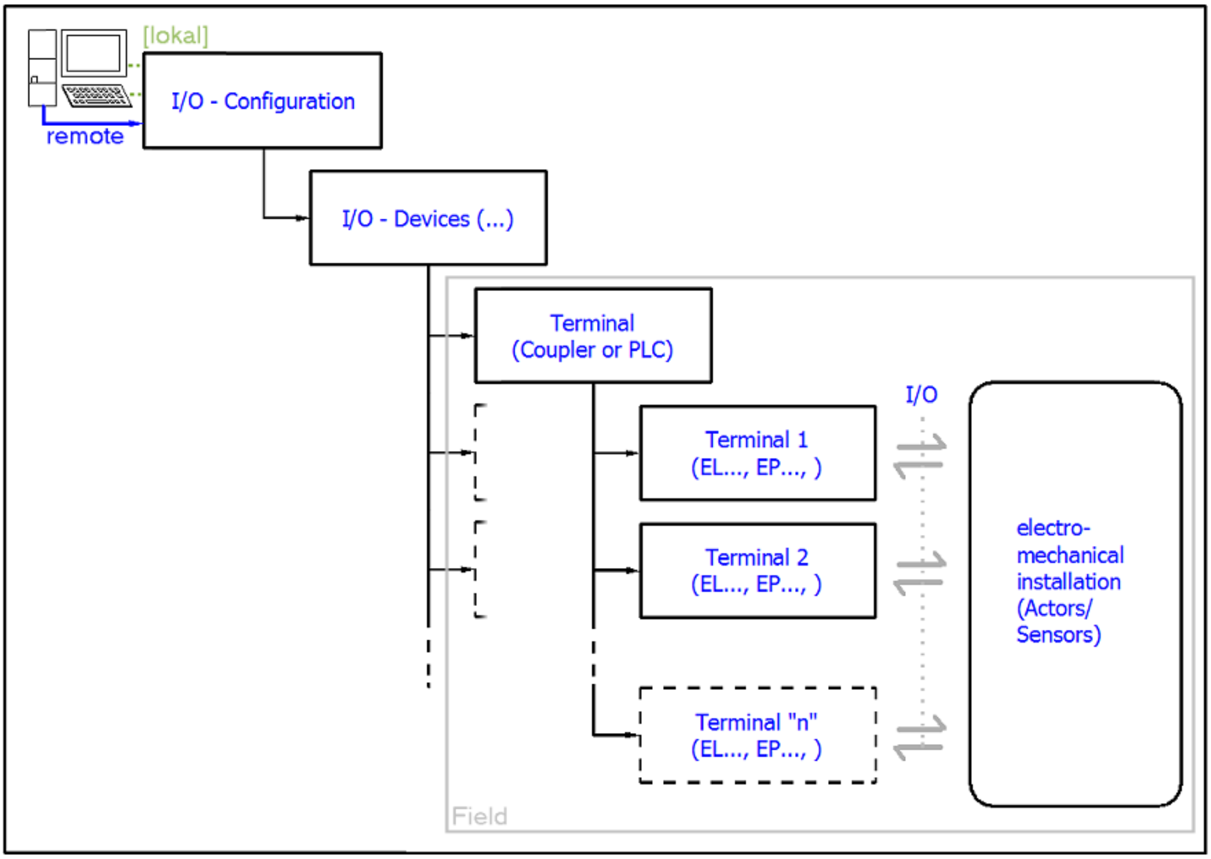


Fig. 96: Relationship between user side (commissioning) and installation

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

Example configuration (actual configuration)

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

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

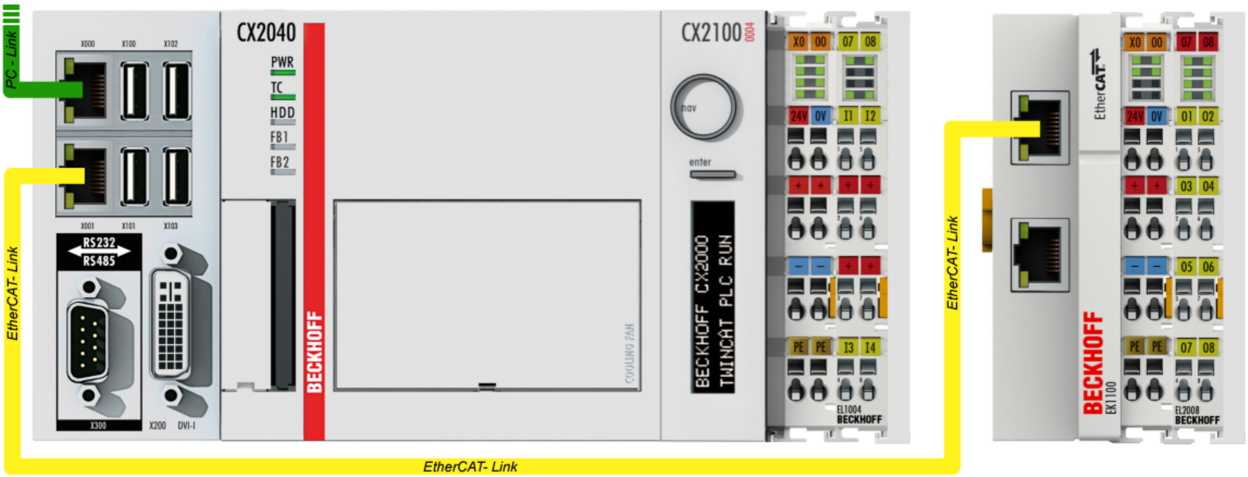


Fig. 97: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

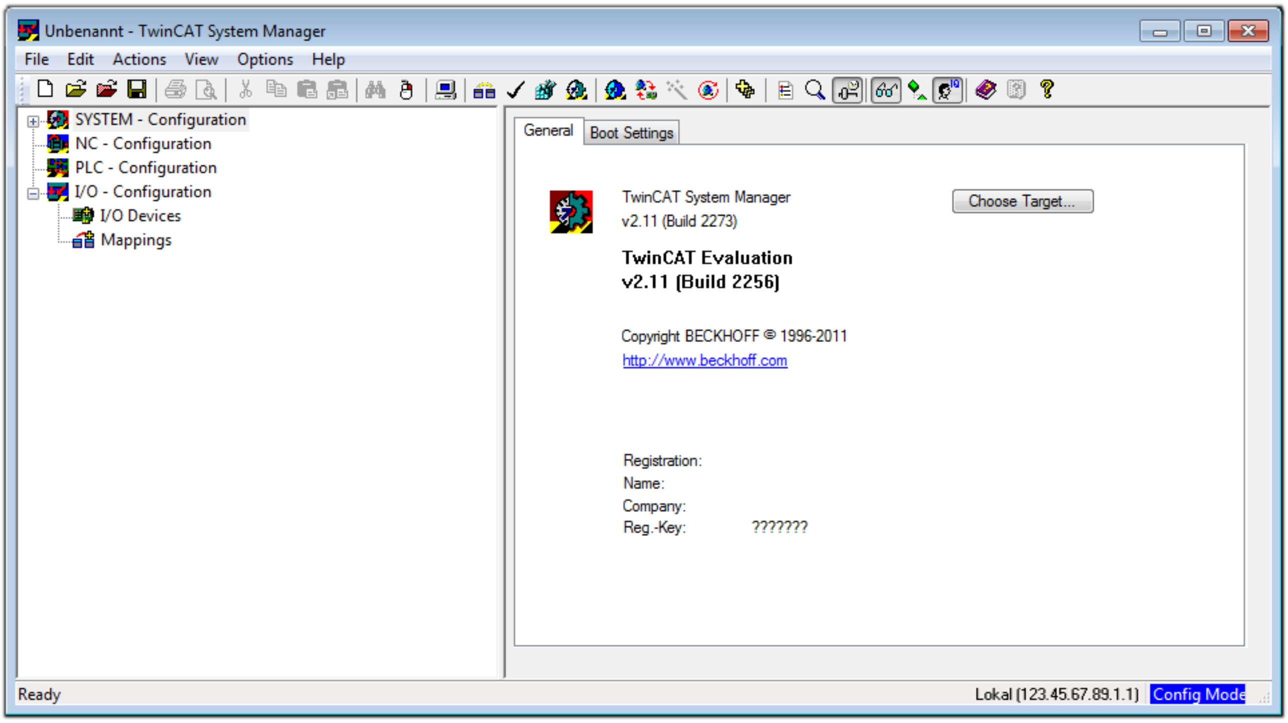
Note that all combinations of a configuration are possible; for example, the EL1004 terminal could also be connected after the coupler, or the EL2008 terminal could additionally be connected to the CX2040 on the right, in which case the EK1100 coupler wouldn’t be necessary.

5.1.2.1 TwinCAT 2

Startup

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

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



TwinCAT System Manager
v2.11 (Build 2273)

TwinCAT Evaluation
v2.11 (Build 2256)

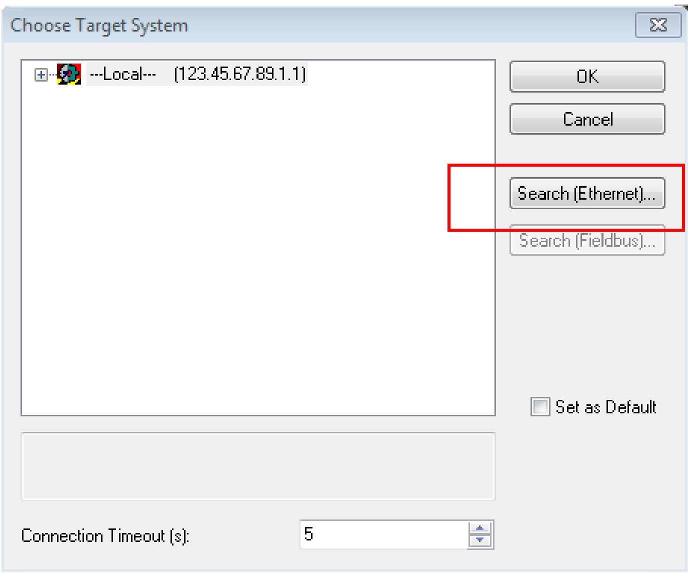
Copyright BECKHOFF © 1996-2011
http://www.beckhoff.com

Registration:
Name:
Company:
Reg.-Key: ???????

Fig. 98: Initial TwinCAT 2 user interface

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system, including the user interface (standard) is installed on the respective PLC, TwinCAT can be used in local mode and thus the next step is "Insert Device [▸ 103]".

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



---Local--- (123.45.67.89.1.1)

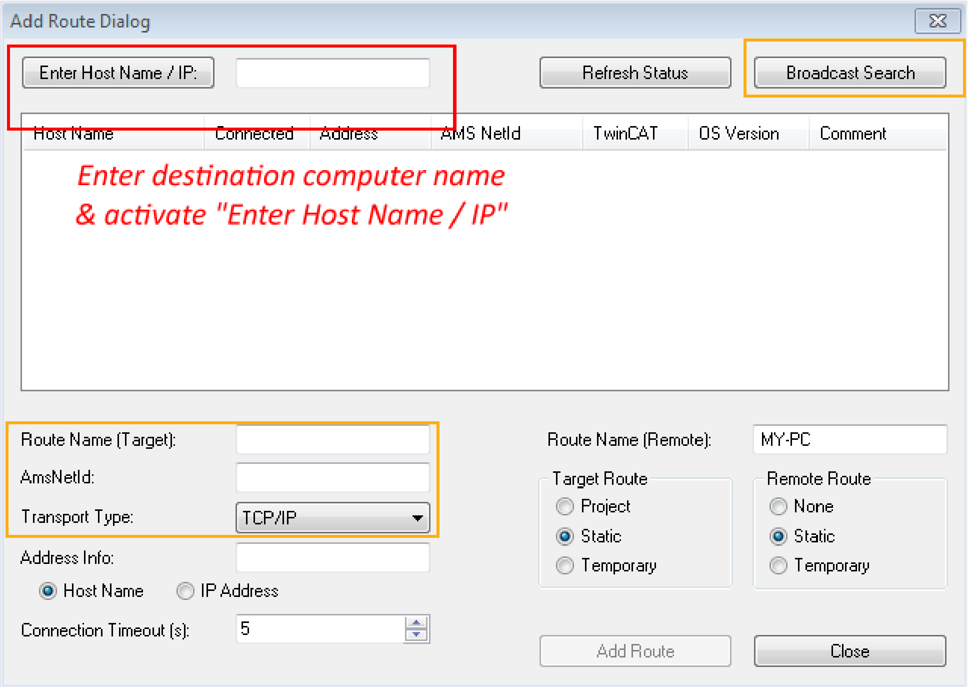
□ Set as Default

Connection Timeout (s): 5

Fig. 99: Selection of the target system

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

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



Enter destination computer name & activate "Enter Host Name / IP"

Route Name (Target):

AmsNetId:

Transport Type: TCP/IP

Address Info:

● Host Name ○ IP Address

Connection Timeout (s): 5

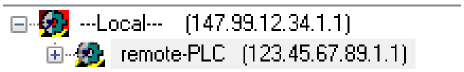
Route Name (Remote): MY-PC

Target Route: ○ Project ● Static ○ Temporary

Remote Route: ○ None ● Static ○ Temporary

Fig. 100: specify the PLC for access by the TwinCAT System Manager: selection of the target system

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



---Local--- (147.99.12.34.1.1)

remote-PLC (123.45.67.89.1.1)

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

Adding devices

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

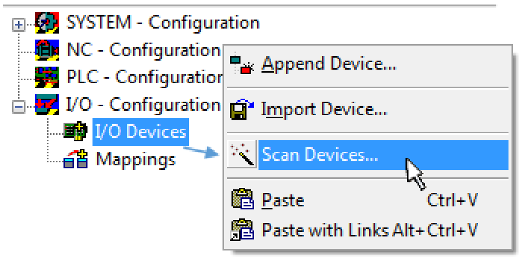
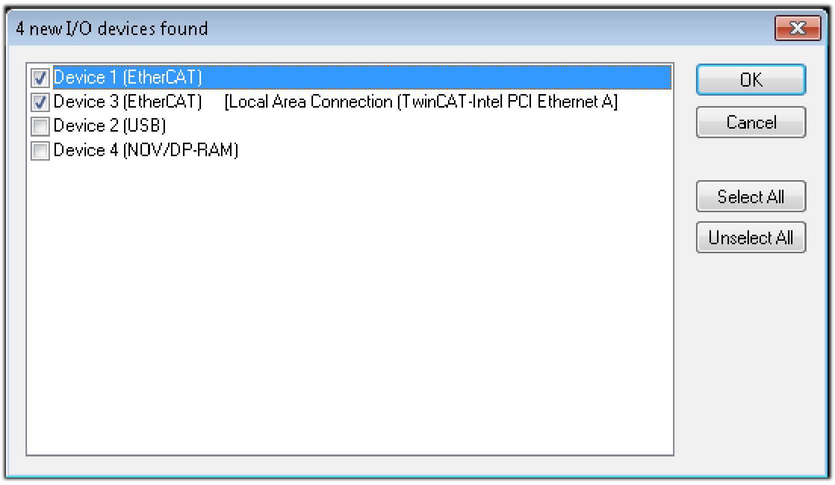


Fig. 101: Select “Scan Devices...”

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


4 new I/O devices found

☑ Device 1 (EtherCAT)
☑ Device 3 (EtherCAT) [Local Area Connection (TwinCAT-Intel PCI Ethernet A)]
□ Device 2 (USB)
□ Device 4 (NOV/DP-RAM)

Fig. 102: Automatic detection of I/O devices: selection of the devices to be integrated

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

Based on the example configuration [▶ 99] described at the beginning of this section, the result is as follows:

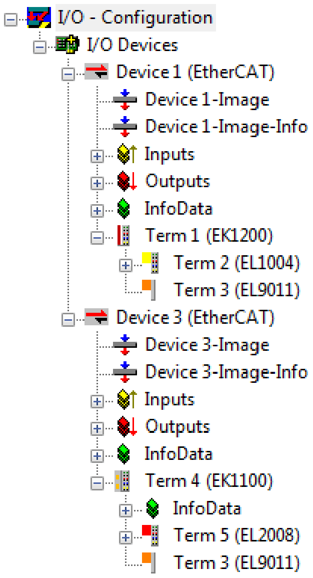


Fig. 103: Mapping of the configuration in the TwinCAT 2 System Manager

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

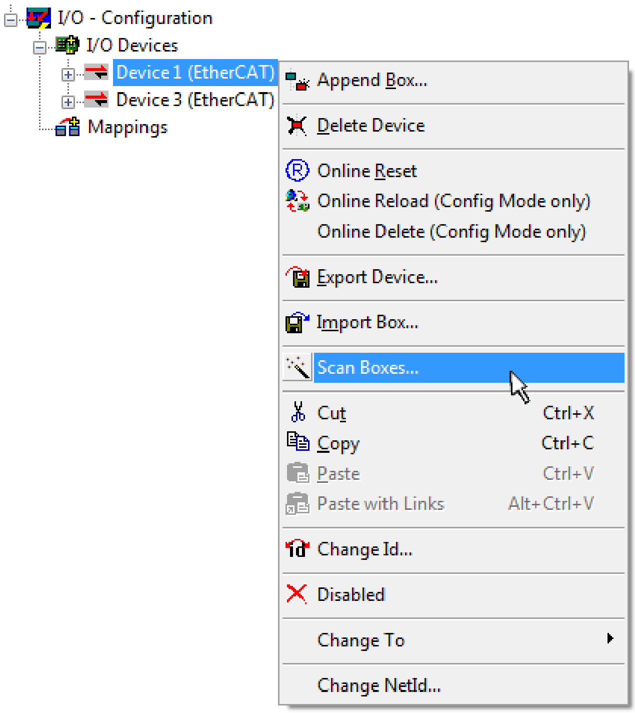


Fig. 104: Reading of individual terminals connected to a device

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

Programming and integrating the PLC

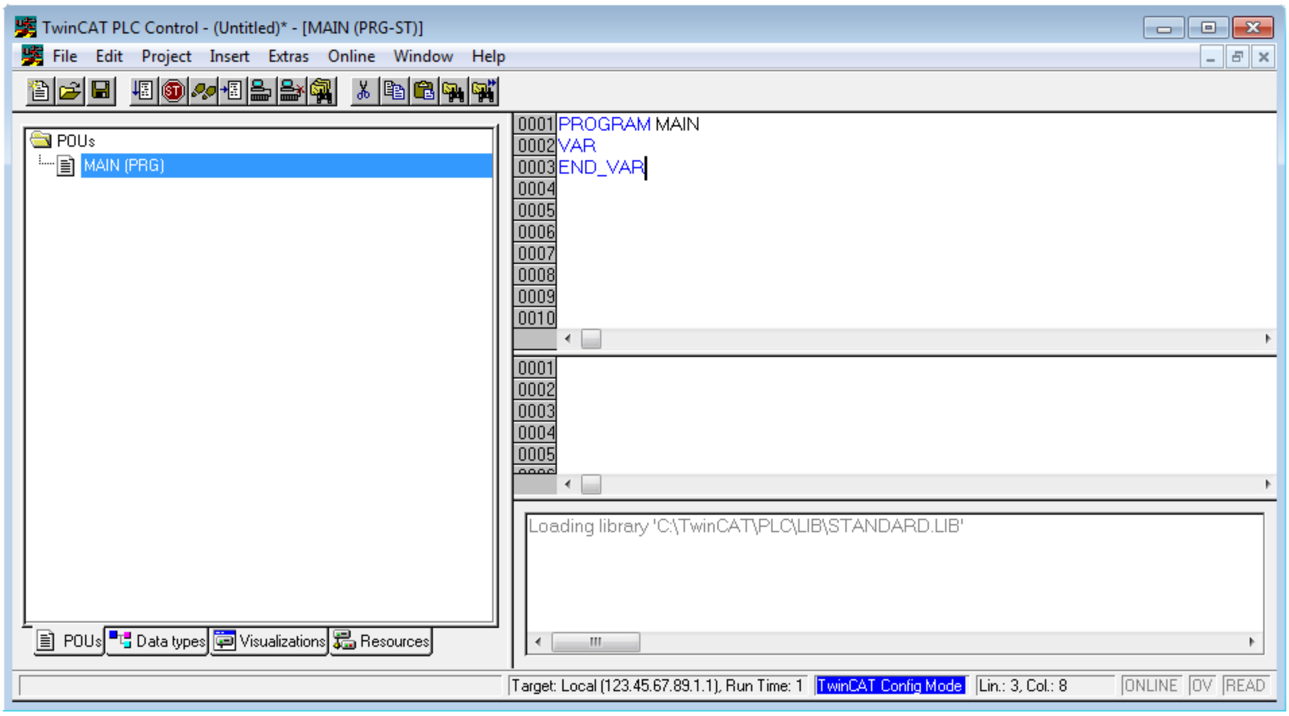
TwinCAT PLC Control is the development environment for generating the controller in different program environments: TwinCAT PLC Control supports all languages described in IEC 61131-3. There are two text-based languages and three graphical languages.

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

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

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

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



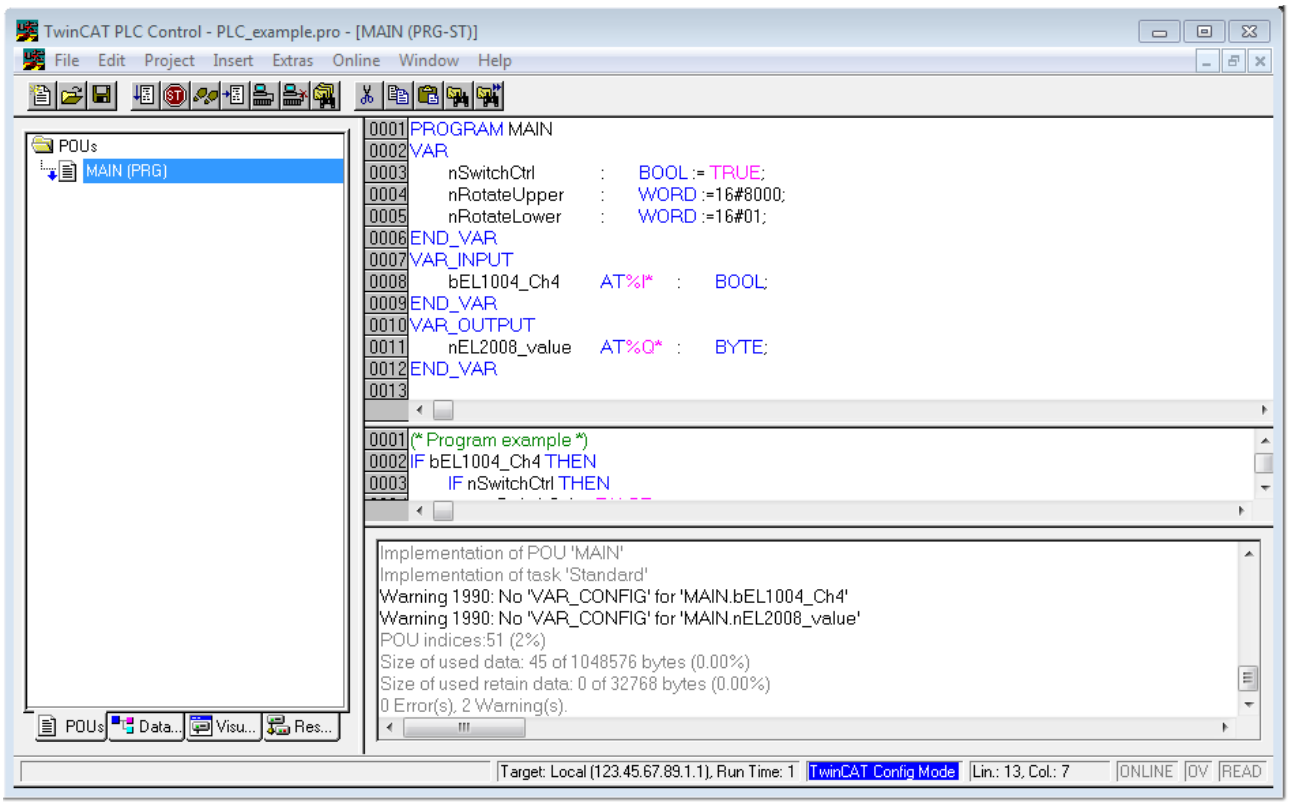
```
PROGRAM MAIN
VAR
END_VAR
```

```
Loading library 'C:\TwinCAT\PLC\LIB\STANDARD.LIB'
```

Target: Local (123.45.67.89.1.1), Run Time: 1 | TwinCAT Config Mode | Lin.: 3, Col.: 8

Fig. 105: TwinCAT PLC Control after startup

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


TwinCAT PLC Control - PLC_example.pro - [MAIN (PRG-ST)]
File Edit Project Insert Extras Online Window Help
POUs
MAIN (PRG)
POUs
Data...
Visu...
Res...

```
PROGRAM MAIN
VAR
    nSwitchCtrl      :    BOOL := TRUE;
    nRotateUpper     :    WORD :=16#8000;
    nRotateLower     :    WORD :=16#01;
END_VAR
VAR_INPUT
    bEL1004_Ch4      AT%I*   :   BOOL;
END_VAR
VAR_OUTPUT
    nEL2008_value    AT%Q*   :   BYTE;
END_VAR
```

```
(* Program example *)
IF bEL1004_Ch4 THEN
    IF nSwitchCtrl THEN
```

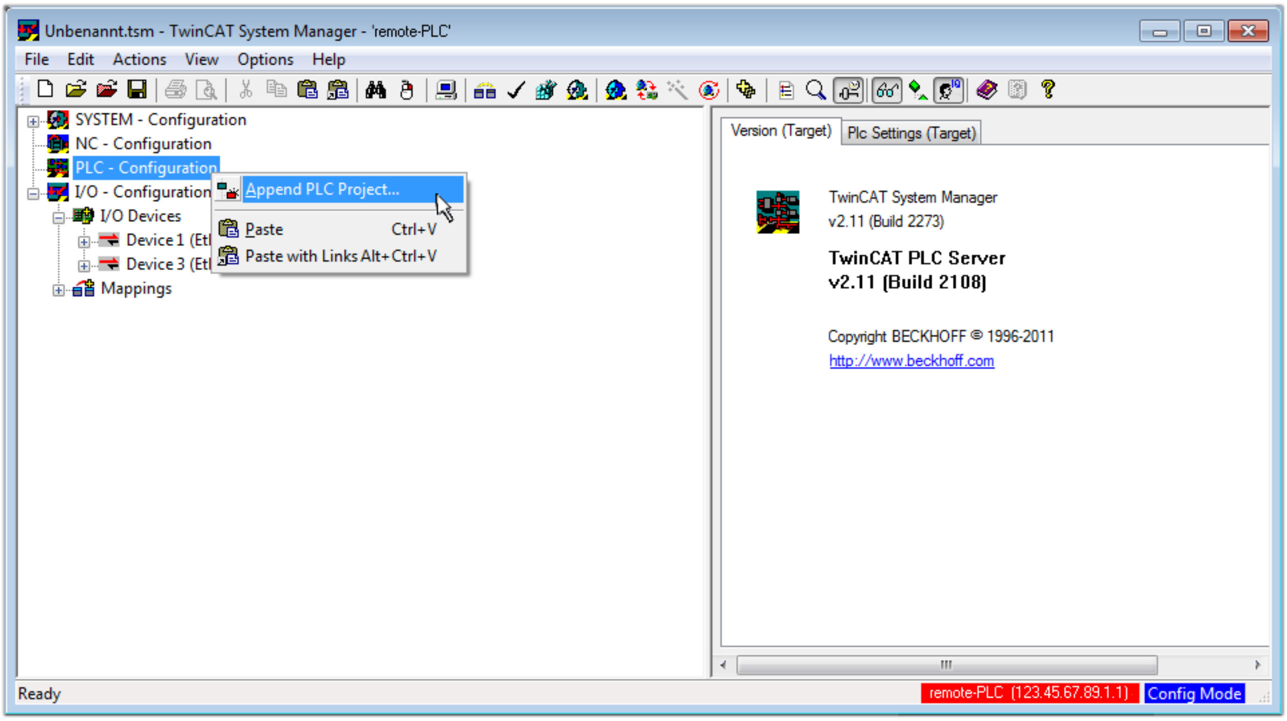
```
Implementation of POU 'MAIN'
Implementation of task 'Standard'
Warning 1990: No 'VAR_CONFIG' for 'MAIN.bEL1004_Ch4'
Warning 1990: No 'VAR_CONFIG' for 'MAIN.nEL2008_value'
POU indices:51 (2%)
Size of used data: 45 of 1048576 bytes (0.00%)
Size of used retain data: 0 of 32768 bytes (0.00%)
0 Error(s), 2 Warning(s).
```

Target: Local (123.45.67.89.1.1), Run Time: 1 | TwinCAT Config Mode | Lin.: 13, Col.: 7 | ONLINE | OV | READ

Fig. 106: Example program with variables after a compile process (without variable integration)

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

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



TwinCAT System Manager
v2.11 (Build 2273)

TwinCAT PLC Server
v2.11 (Build 2108)

Copyright BECKHOFF © 1996-2011
http://www.beckhoff.com

Ready | remote-PLC (123.45.67.89.1.1) | Config Mode

Fig. 107: 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:

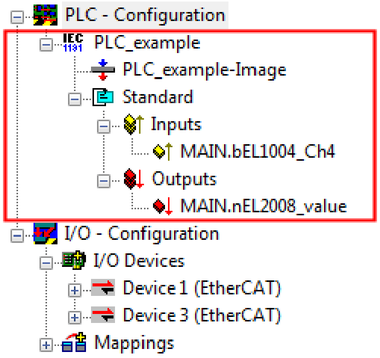
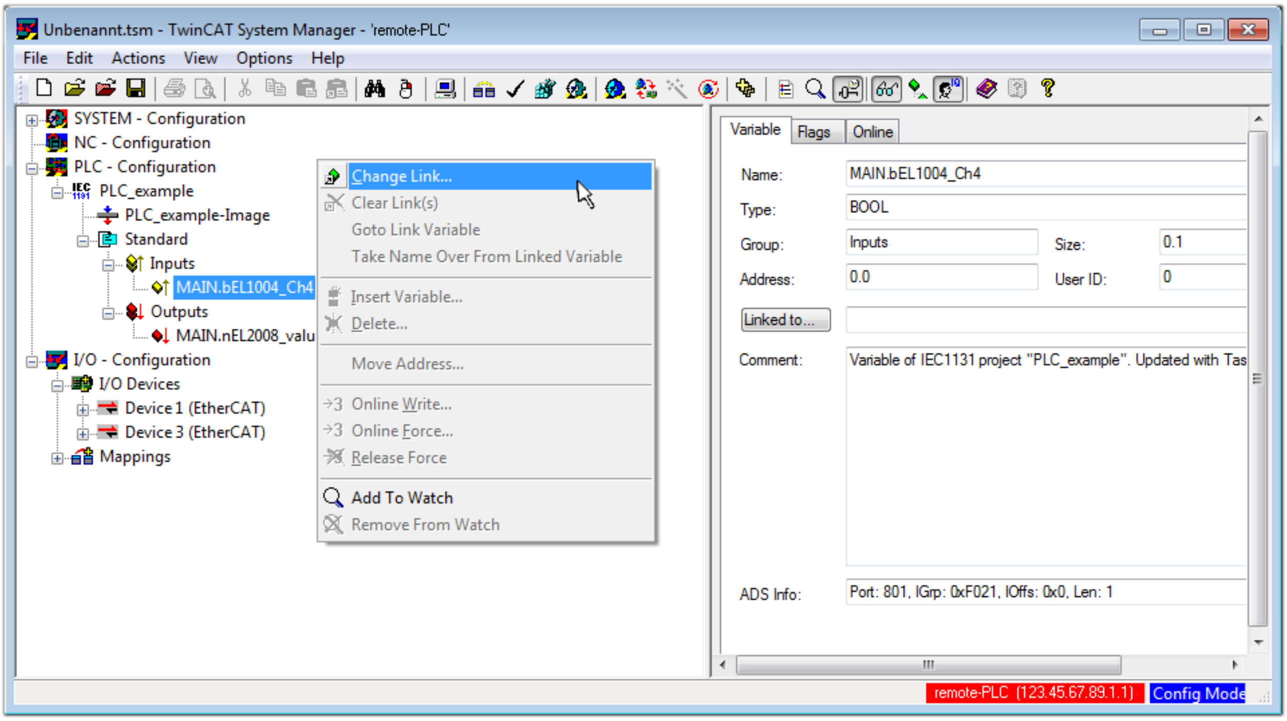


Fig. 108: PLC project integrated in the PLC configuration of the System Manager

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”:



Name: MAIN.bEL1004_Ch4

Type: BOOL

Group: Inputs Size: 0.1

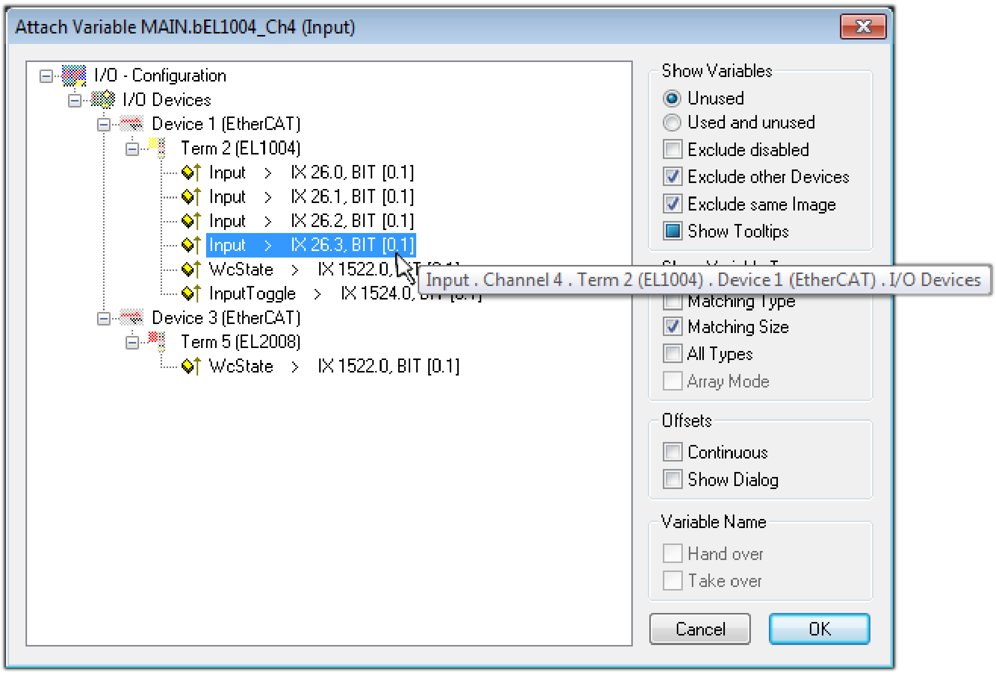
Address: 0.0 User ID: 0

Comment: Variable of IEC1131 project "PLC_example". Updated with Tas

ADS Info: Port: 801, IGrp: 0xF021, IOffs: 0x0, Len: 1

Fig. 109: Creating the links between PLC variables and process objects

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



Attach Variable MAIN.bEL1004_Ch4 (Input)

```
I/O - Configuration
  I/O Devices
    Device 1 (EtherCAT)
      Term 2 (EL1004)
        Input       >  IX 26.0, BIT [0.1]
        Input       >  IX 26.1, BIT [0.1]
        Input       >  IX 26.2, BIT [0.1]
        Input       >  IX 26.3, BIT [0.1]
        WcState     >  IX 1522.0, BIT [0.1]
        InputToggle >  IX 1524.0, BIT [0.1]
    Device 3 (EtherCAT)
      Term 5 (EL2008)
        WcState     >  IX 1522.0, BIT [0.1]
```

Input . Channel 4 . Term 2 (EL1004) . Device 1 (EtherCAT) . I/O Devices

Show Variables
- ● Unused
- ○ Used and unused
- □ Exclude disabled
- ☑ Exclude other Devices
- ☑ Exclude same Image
- ■ Show Tooltips

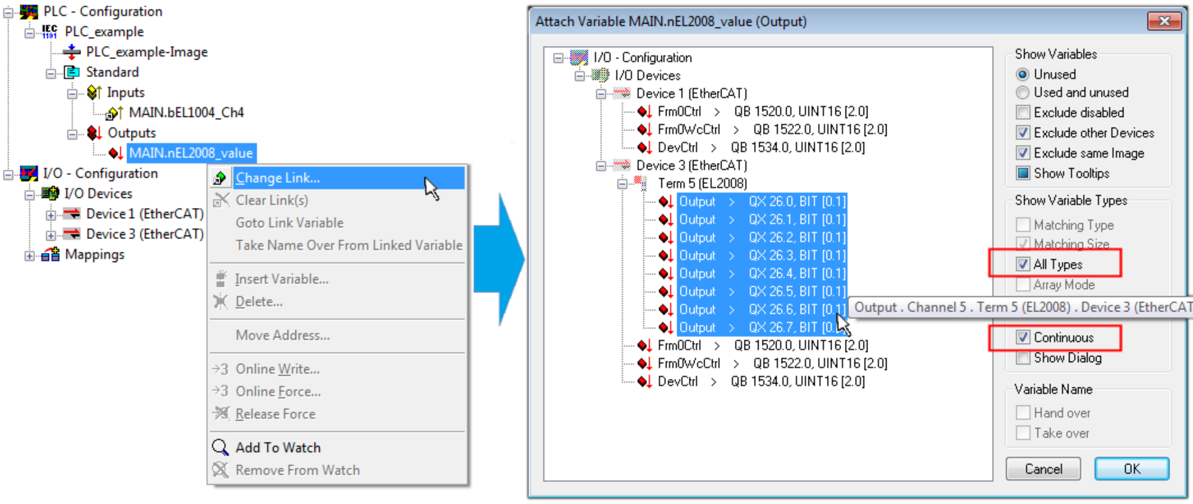
Show Variable T
- □ Matching Type
- ☑ Matching Size
- □ All Types
- □ Array Mode

Offsets
- □ Continuous
- □ Show Dialog

Variable Name
- □ Hand over
- □ Take over

Fig. 110: Selecting BOOL-type PDO

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



```
PLC - Configuration
  PLC_example
    PLC_example-Image
    Standard
      Inputs
        MAIN.bEL1004_Ch4
      Outputs
        MAIN.nEL2008_value
I/O - Configuration
  I/O Devices
    Device 1 (EtherCAT)
    Device 3 (EtherCAT)
  Mappings
```

Attach Variable MAIN.nEL2008_value (Output)

```
I/O - Configuration
  I/O Devices
    Device 1 (EtherCAT)
      Frm0Ctrl    >  QB 1520.0, UINT16 [2.0]
      Frm0WcCtrl  >  QB 1522.0, UINT16 [2.0]
      DevCtrl     >  QB 1534.0, UINT16 [2.0]
    Device 3 (EtherCAT)
      Term 5 (EL2008)
        Output    >  QX 26.0, BIT [0.1]
        Output    >  QX 26.1, BIT [0.1]
        Output    >  QX 26.2, BIT [0.1]
        Output    >  QX 26.3, BIT [0.1]
        Output    >  QX 26.4, BIT [0.1]
        Output    >  QX 26.5, BIT [0.1]
        Output    >  QX 26.6, BIT [0.1]
        Output    >  QX 26.7, BIT [0.1]
      Frm0Ctrl    >  QB 1520.0, UINT16 [2.0]
      Frm0WcCtrl  >  QB 1522.0, UINT16 [2.0]
      DevCtrl     >  QB 1534.0, UINT16 [2.0]
```

Output . Channel 5 . Term 5 (EL2008) . Device 3 (EtherCAT) . I/O Devices

Show Variables
- ● Unused
- ○ Used and unused
- □ Exclude disabled
- ☑ Exclude other Devices
- ☑ Exclude same Image
- ■ Show Tooltips

Show Variable Types
- □ Matching Type
- ☑ Matching Size
- ☑ All Types
- □ Array Mode
- ☑ Continuous
- □ Show Dialog

Variable Name
- □ Hand over
- □ Take over

Fig. 111: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

Note that the “Continuous” checkbox was also activated. This is designed to allocate the bits contained in the byte of the “nEL2008_value” variable sequentially to all eight selected output bits of the EL2008 Terminal. It is thus possible to subsequently address all eight outputs of the terminal in the program with a byte

corresponding to bit 0 for channel 1 to bit 7 for channel 8 of the PLC. A special symbol (⧄) on the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting “Goto Link Variable” from the context menu of a variable. The opposite linked object, in this case the PDO, is automatically selected:

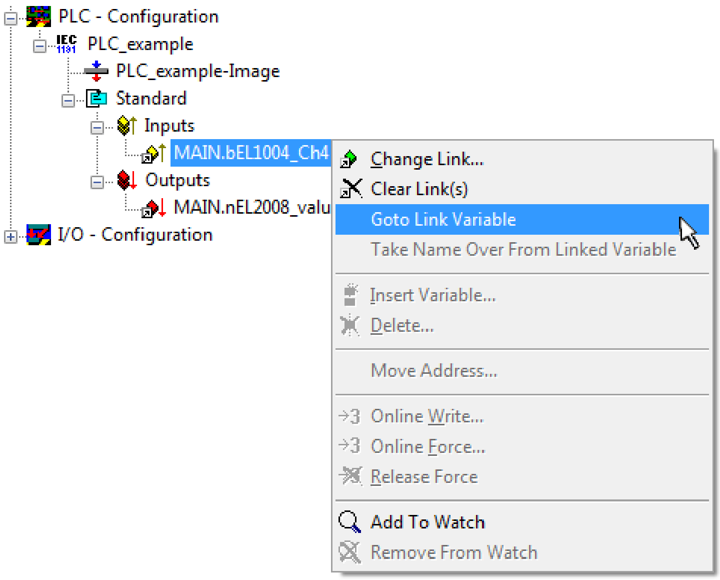
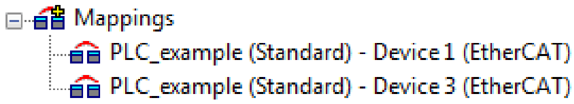


Fig. 112: Application of a “Goto Link Variable”, using “MAIN.bEL1004_Ch4” as an example

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

This can be visualized in the configuration:


Mappings
PLC_example (Standard) - Device 1 (EtherCAT)
PLC_example (Standard) - Device 3 (EtherCAT)

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

Activation of the configuration

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

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

Starting the controller

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

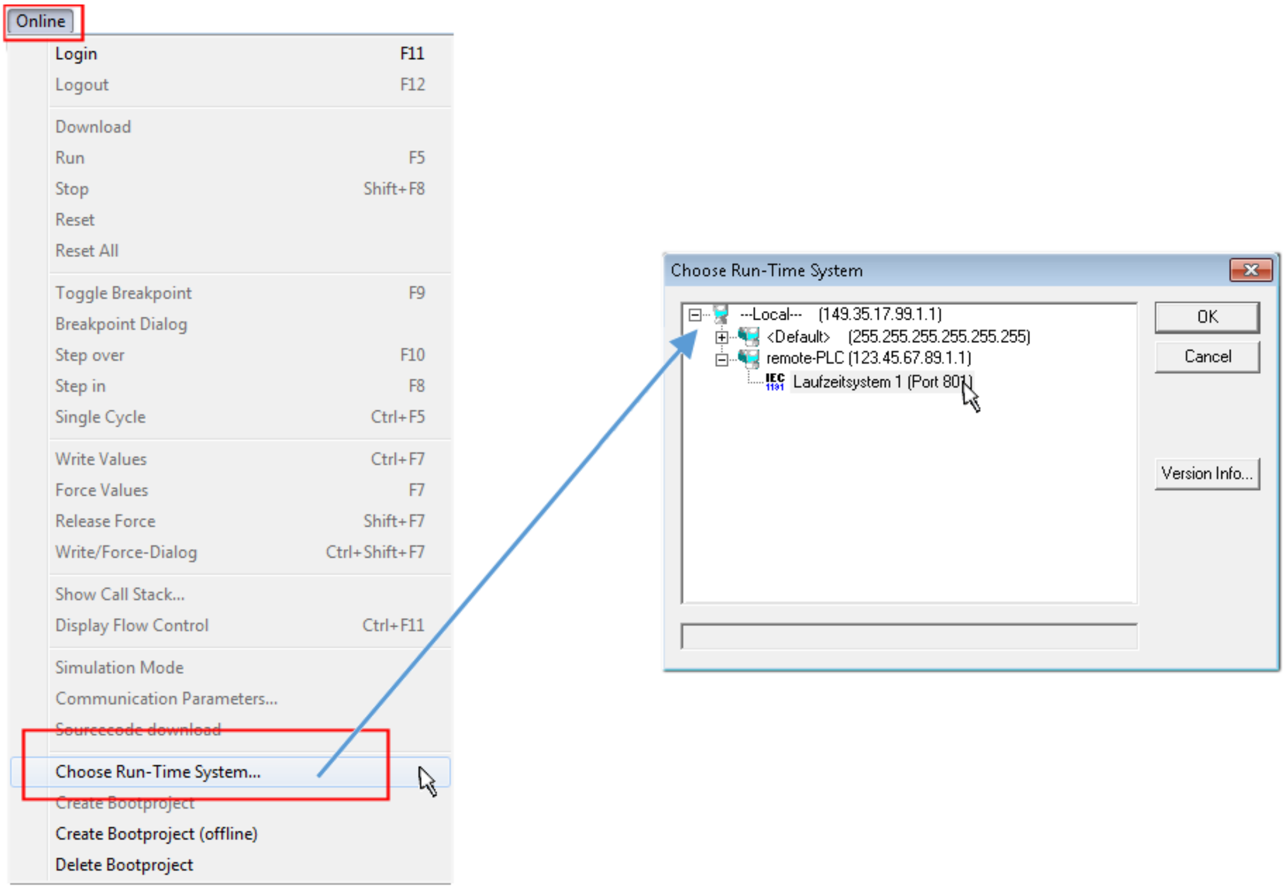
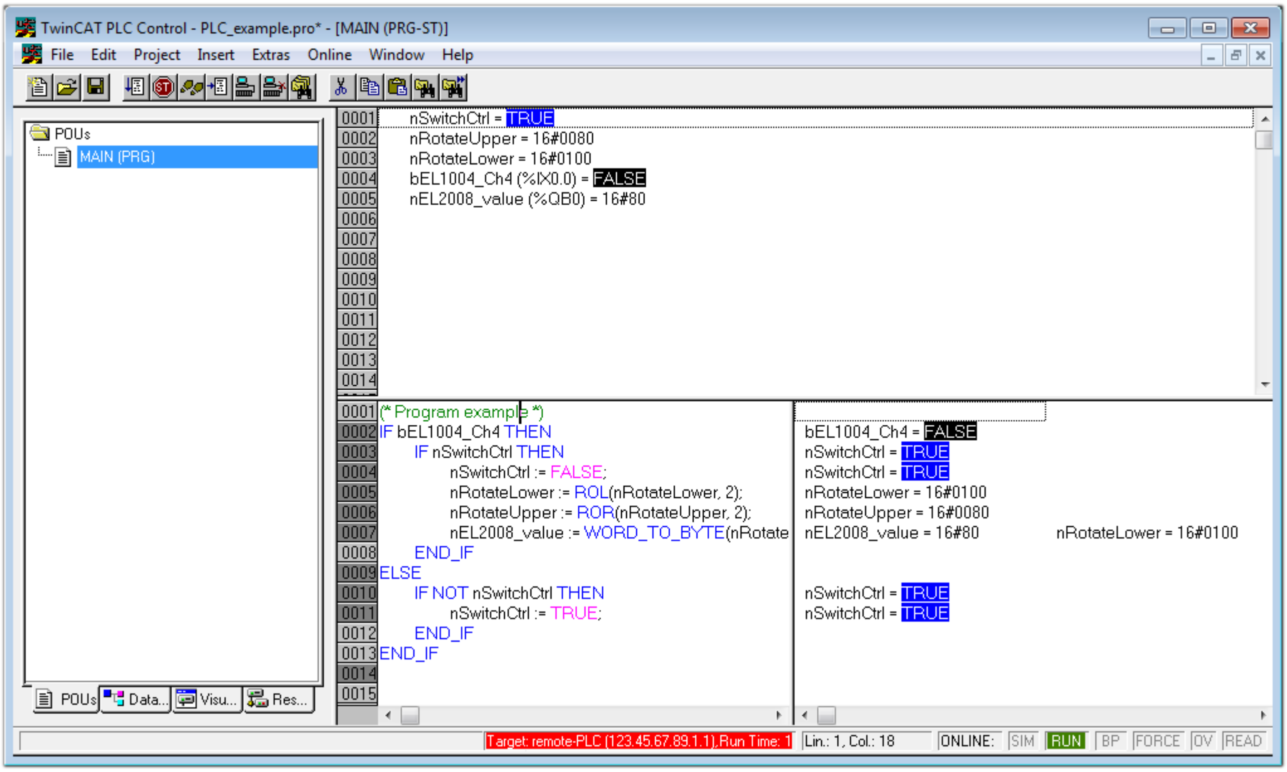


Fig. 113: Choose target system (remote)

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


TwinCAT PLC Control - PLC_example.pro* - [MAIN (PRG-ST)]
File Edit Project Insert Extras Online Window Help
POUs
MAIN (PRG)
POUs
Data...
Visu...
Res...

```
nSwitchCtrl = TRUE
nRotateUpper = 16#0080
nRotateLower = 16#0100
bEL1004_Ch4 (%IX0.0) = FALSE
nEL2008_value (%QB0) = 16#80
```

```
(* Program example *)
IF bEL1004_Ch4 THEN
    IF nSwitchCtrl THEN
        nSwitchCtrl := FALSE;
        nRotateLower := ROL(nRotateLower, 2);
        nRotateUpper := ROR(nRotateUpper, 2);
        nEL2008_value := WORD_TO_BYTE(nRotate
    END_IF
ELSE
    IF NOT nSwitchCtrl THEN
        nSwitchCtrl := TRUE;
    END_IF
END_IF
```

```
bEL1004_Ch4 = FALSE
nSwitchCtrl = TRUE
nSwitchCtrl = TRUE
nRotateLower = 16#0100
nRotateUpper = 16#0080
nEL2008_value = 16#80          nRotateLower = 16#0100

nSwitchCtrl = TRUE
nSwitchCtrl = TRUE
```

Target: remote-PLC (123.45.67.89.1.1),Run Time: 1 | Lin.: 1, Col.: 18 | ONLINE: SIM RUN BP FORCE OV READ

Fig. 114: PLC Control logged in, ready for program startup

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

5.1.2.2 TwinCAT 3

Startup

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

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

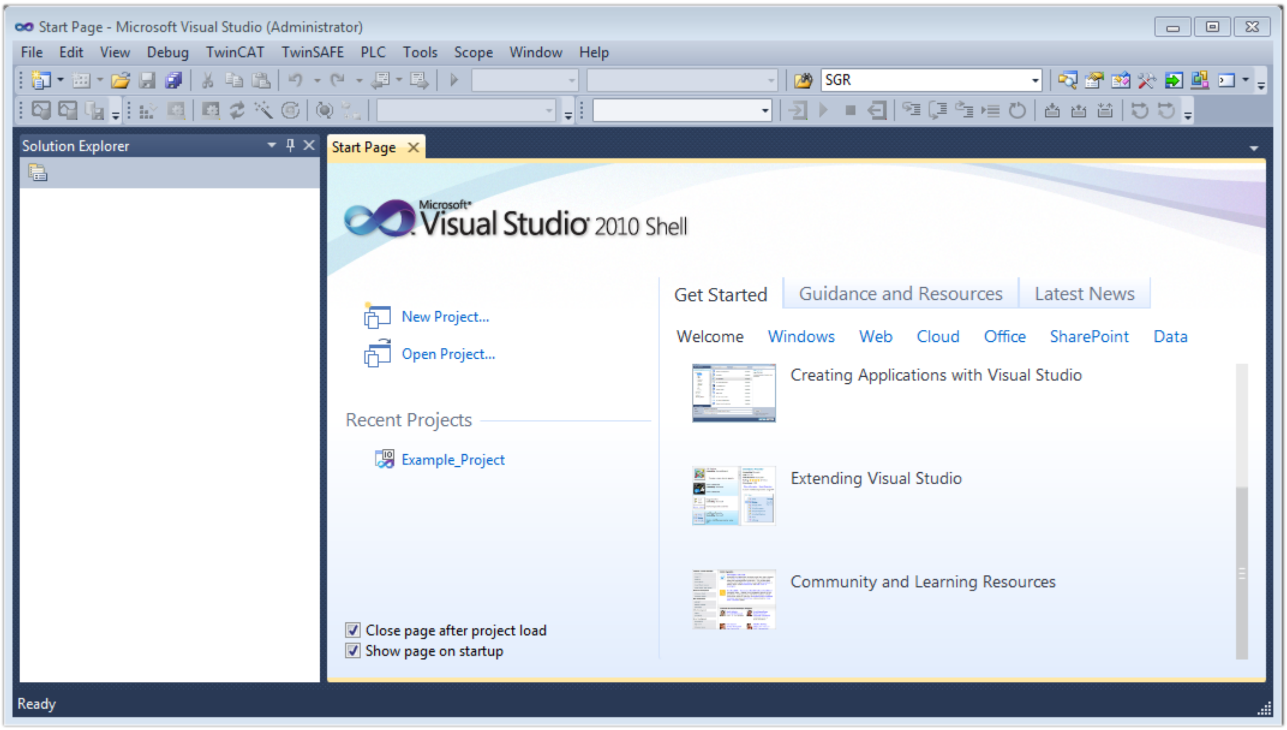
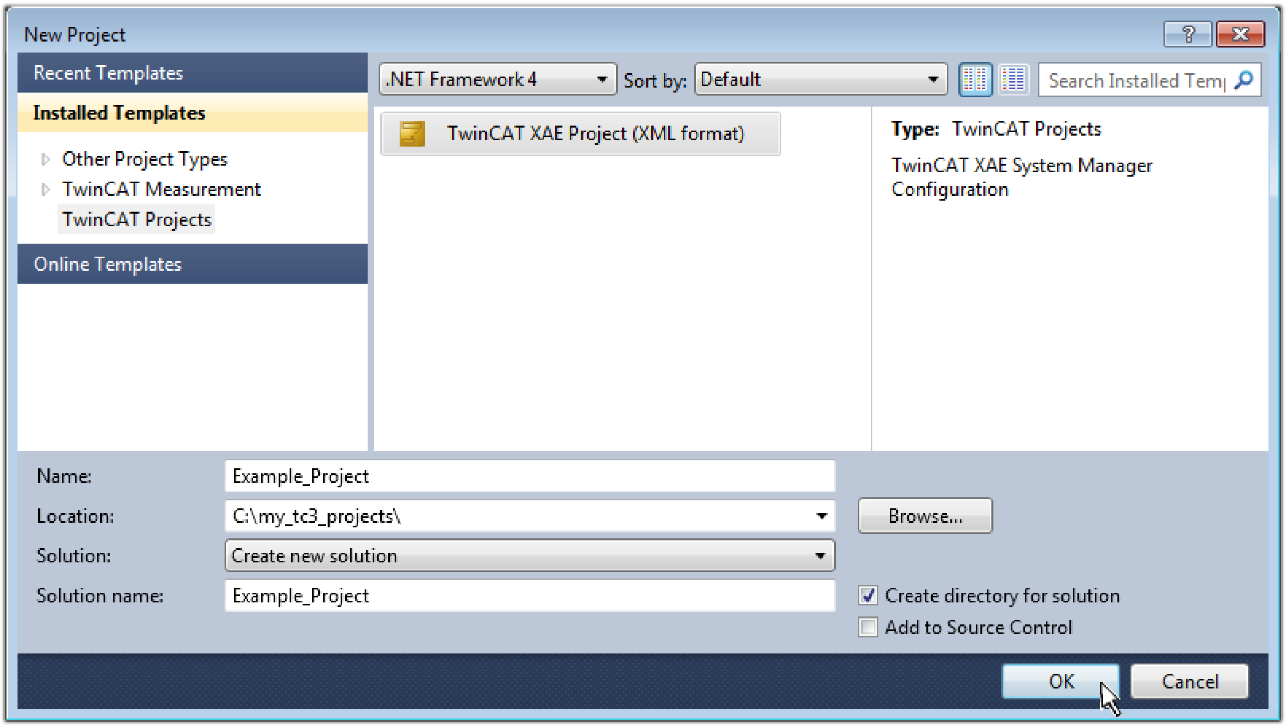

Start Page - Microsoft Visual Studio (Administrator)
File Edit View Debug TwinCAT TwinSAFE PLC Tools Scope Window Help
SGR
Solution Explorer
Start Page
Microsoft Visual Studio 2010 Shell
New Project...
Open Project...
Recent Projects
Example_Project
Get Started
Guidance and Resources
Latest News
Welcome Windows Web Cloud Office SharePoint Data
Creating Applications with Visual Studio
Extending Visual Studio
Community and Learning Resources
☑ Close page after project load
☑ Show page on startup
Ready

Fig. 115: Initial TwinCAT 3 user interface

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



Type: TwinCAT Projects

TwinCAT XAE System Manager Configuration

Name: Example_Project

Location: C:\my_tc3_projects\

Solution: Create new solution

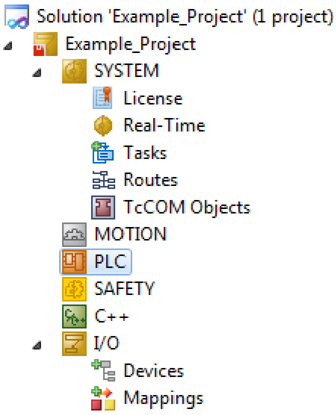
Solution name: Example_Project

☑ Create directory for solution

□ Add to Source Control

Fig. 116: Create new TwinCAT 3 project

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

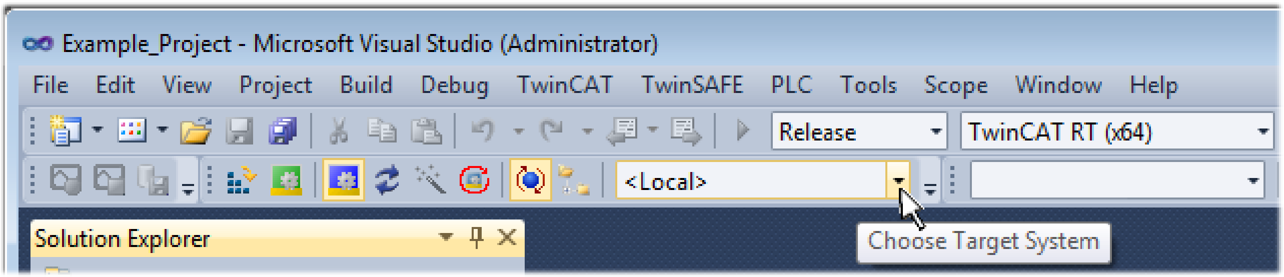


- Solution 'Example_Project' (1 project)
  - Example_Project
    - SYSTEM
      - License
      - Real-Time
      - Tasks
      - Routes
      - TcCOM Objects
    - MOTION
    - PLC
    - SAFETY
    - C++
    - I/O
      - Devices
      - Mappings

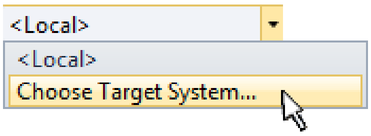
Fig. 117: New TwinCAT 3 project in the project folder explorer

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC (locally), TwinCAT can be used in local mode and the process can be continued with the next step, "Insert Device [▶ 114]".

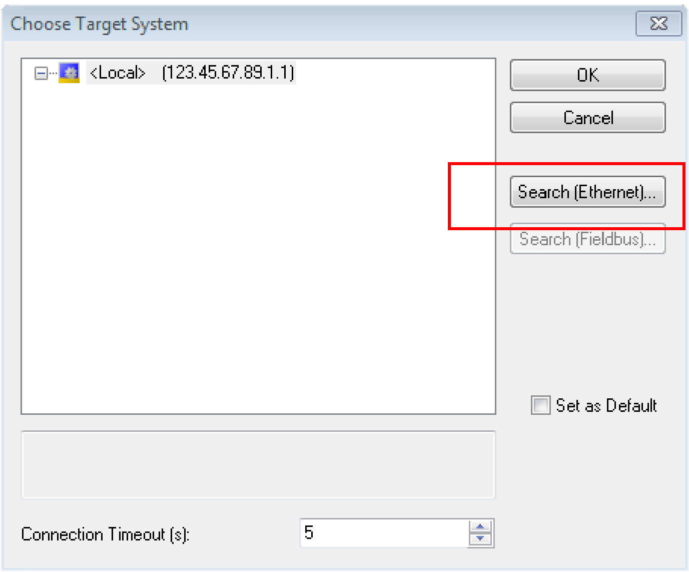
If the intention is to address the TwinCAT runtime environment installed on a PLC remotely from another system used as a development environment, the target system must be made known first. Via the symbol in the menu bar:



expand the pull-down menu:



and open the following window:

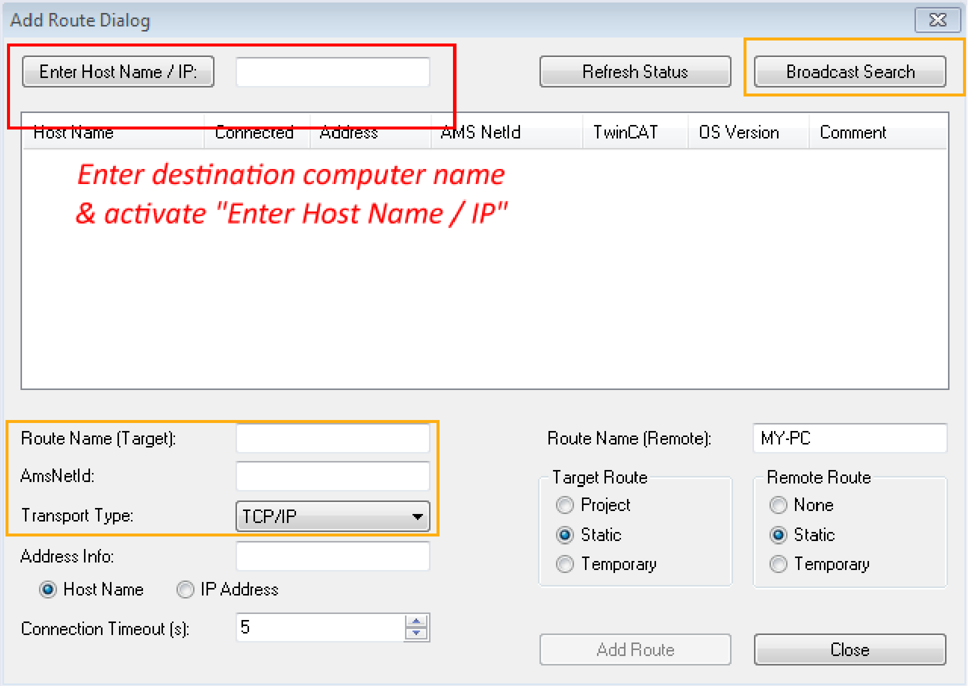


Connection Timeout (s): 5

Fig. 118: Selection dialog: Choose the target system

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

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



Enter destination computer name & activate "Enter Host Name / IP"

Route Name (Target):

AmsNetId:

Transport Type: TCP/IP

Address Info:

● Host Name ○ IP Address

Connection Timeout (s): 5

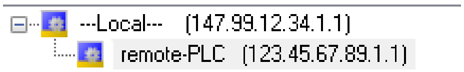
Route Name (Remote): MY-PC

Target Route: ○ Project ● Static ○ Temporary

Remote Route: ○ None ● Static ○ Temporary

Fig. 119: specify the PLC for access by the TwinCAT System Manager: selection of the target system

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



---Local--- (147.99.12.34.1.1)

remote-PLC (123.45.67.89.1.1)

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

Adding devices

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

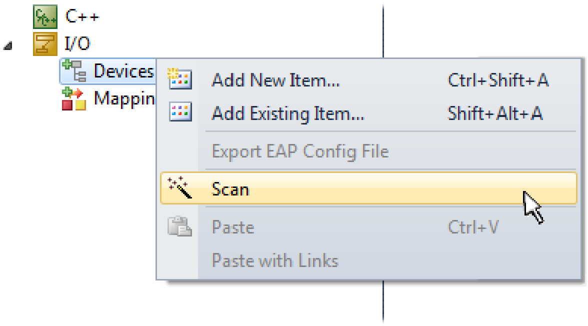
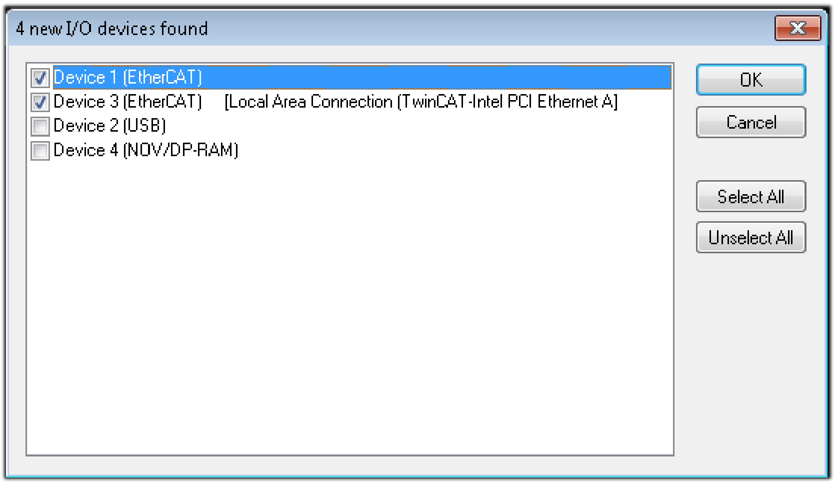


Fig. 120: Select “Scan”

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


4 new I/O devices found

- ☑ Device 1 (EtherCAT)
- ☑ Device 3 (EtherCAT) [Local Area Connection (TwinCAT-Intel PCI Ethernet A)]
- □ Device 2 (USB)
- □ Device 4 (NOV/DP-RAM)

Fig. 121: Automatic detection of I/O devices: selection of the devices to be integrated

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

Based on the example configuration [▶ 99] described at the beginning of this section, the result is as follows:

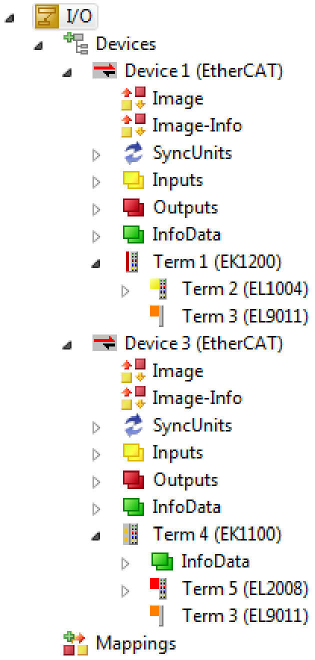

- I/O
  - Devices
    - Device 1 (EtherCAT)
      - Image
      - Image-Info
      - SyncUnits
      - Inputs
      - Outputs
      - InfoData
      - Term 1 (EK1200)
        - Term 2 (EL1004)
        - Term 3 (EL9011)
    - Device 3 (EtherCAT)
      - Image
      - Image-Info
      - SyncUnits
      - Inputs
      - Outputs
      - InfoData
      - Term 4 (EK1100)
        - InfoData
        - Term 5 (EL2008)
        - Term 3 (EL9011)
  - Mappings

Fig. 122: Mapping of the configuration in VS shell of the TwinCAT 3 environment

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

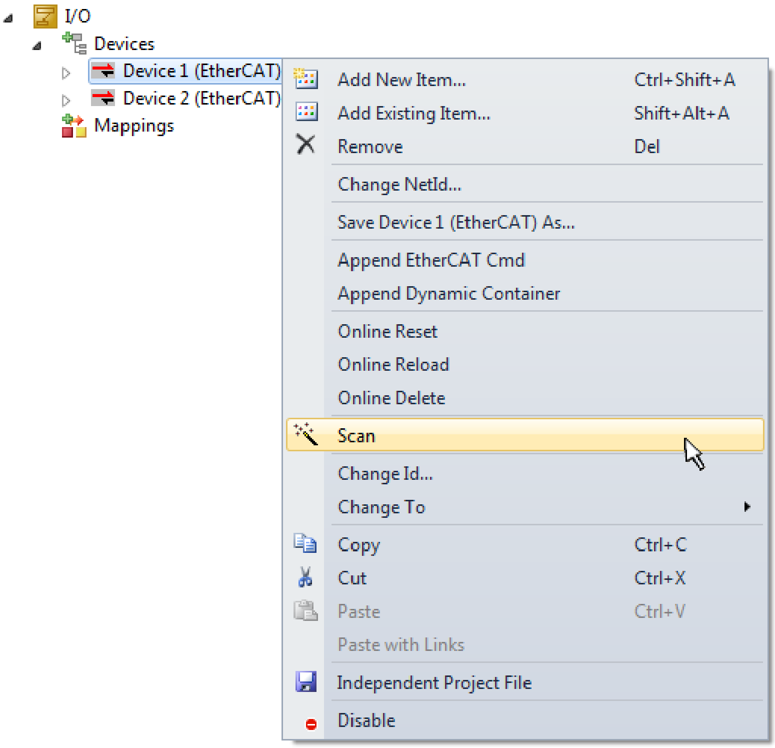


Fig. 123: Reading of individual terminals connected to a device

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

Programming the PLC

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

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

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

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

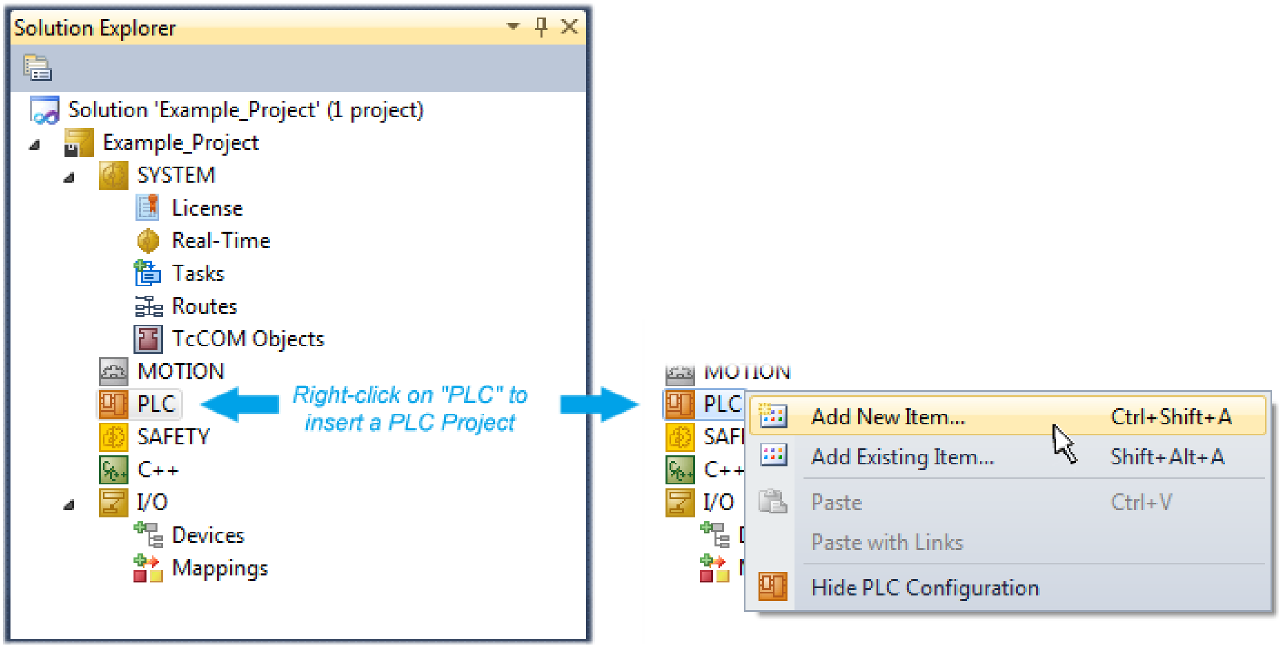
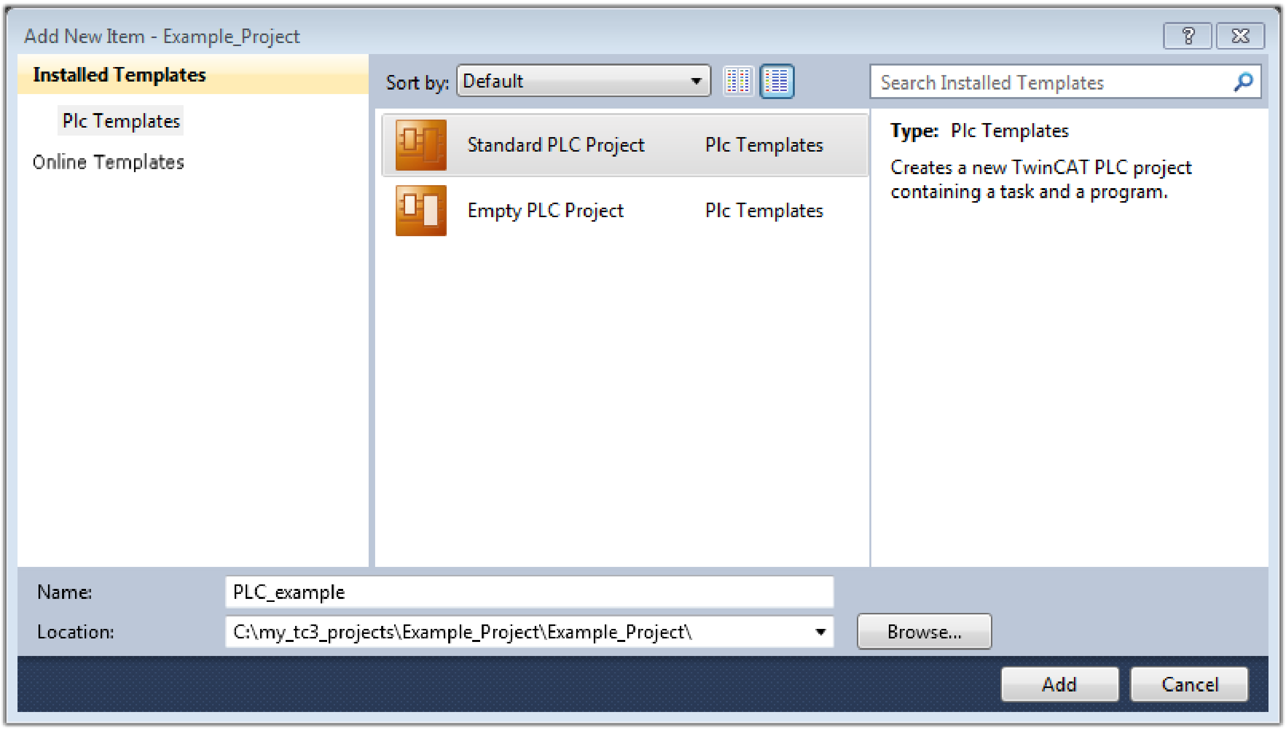


Fig. 124: Adding the programming environment in “PLC”

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


Add New Item - Example_Project
Installed Templates
Plc Templates
Online Templates
Sort by: Default
Search Installed Templates
Standard PLC Project Plc Templates
Empty PLC Project Plc Templates
Browse...
Add
Cancel

Type: Plc Templates

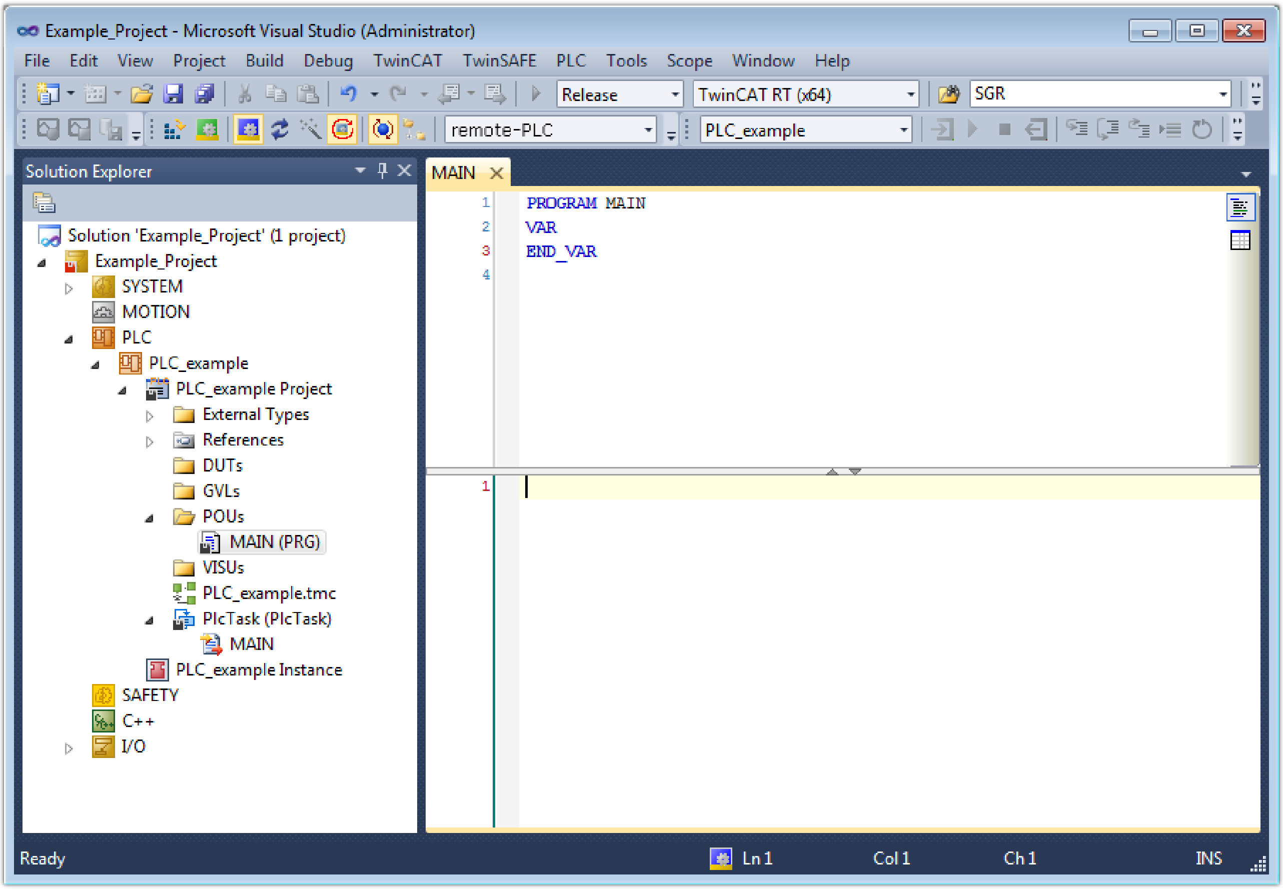
Creates a new TwinCAT PLC project containing a task and a program.

Name: PLC_example

Location: C:\my_tc3_projects\Example_Project\Example_Project\

Fig. 125: Specifying the name and directory for the PLC programming environment

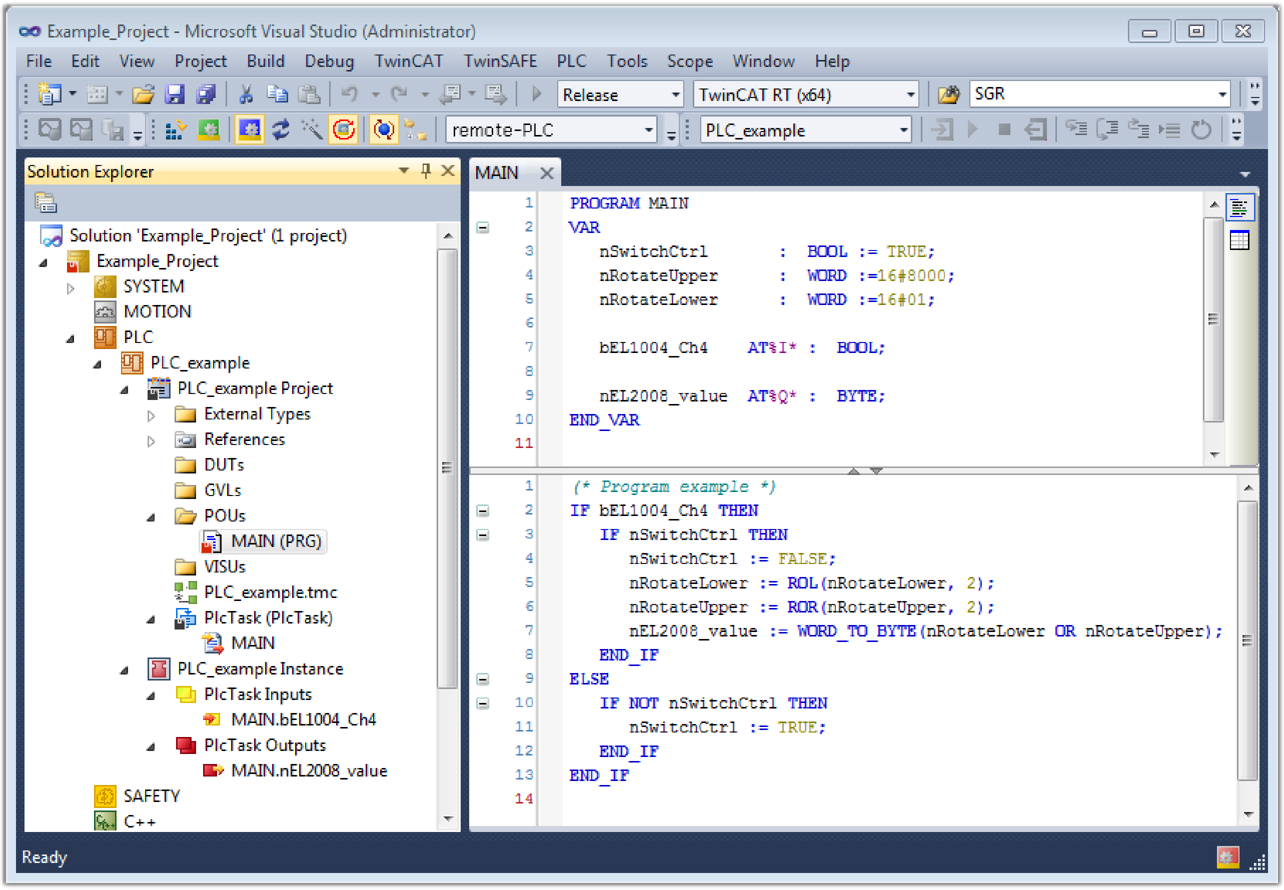
The “Main” program, which already exists due to selecting “Standard PLC project”, can be opened by double-clicking on “PLC_example_project” in “POUs”. The following user interface is shown for an initial project:



```
PROGRAM MAIN
VAR
END_VAR
```

Fig. 126: Initial “Main” program for the standard PLC project

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



```
PROGRAM MAIN
VAR
    nSwitchCtrl       :  BOOL := TRUE;
    nRotateUpper      :  WORD :=16#8000;
    nRotateLower      :  WORD :=16#01;

    bEL1004_Ch4    AT%I* :  BOOL;

    nEL2008_value  AT%Q* :  BYTE;
END_VAR
```

```
(* Program example *)
IF bEL1004_Ch4 THEN
    IF nSwitchCtrl THEN
        nSwitchCtrl  := FALSE;
        nRotateLower := ROL(nRotateLower, 2);
        nRotateUpper := ROR(nRotateUpper, 2);
        nEL2008_value := WORD_TO_BYTE(nRotateLower OR nRotateUpper);
    END_IF
ELSE
    IF NOT nSwitchCtrl THEN
        nSwitchCtrl  := TRUE;
    END_IF
END_IF
```

Fig. 127: Example program with variables after a compile process (without variable integration)

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

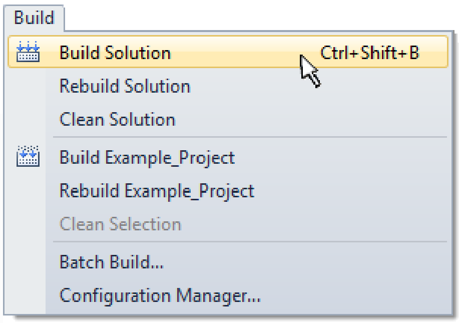
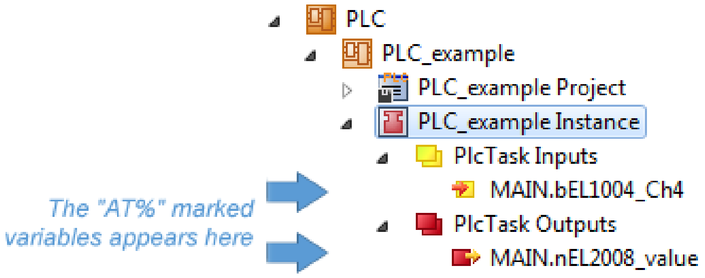


Fig. 128: Start program compilation

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



Assigning variables

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

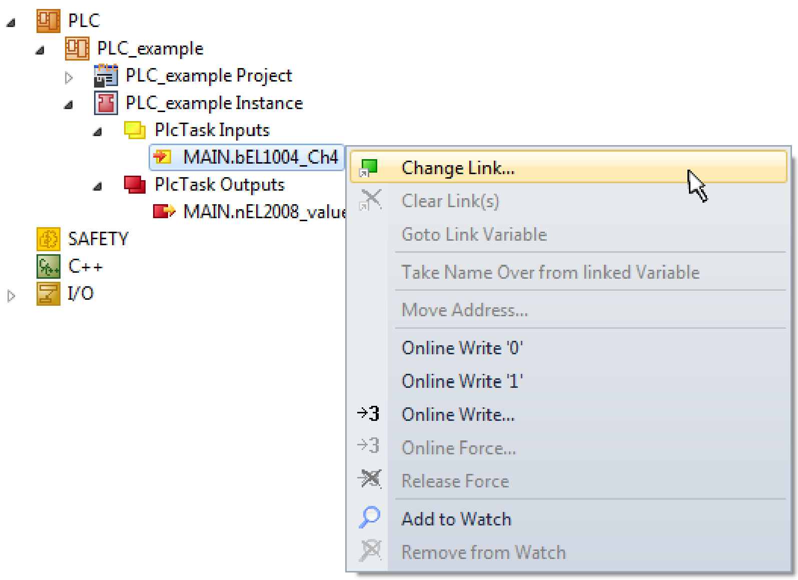
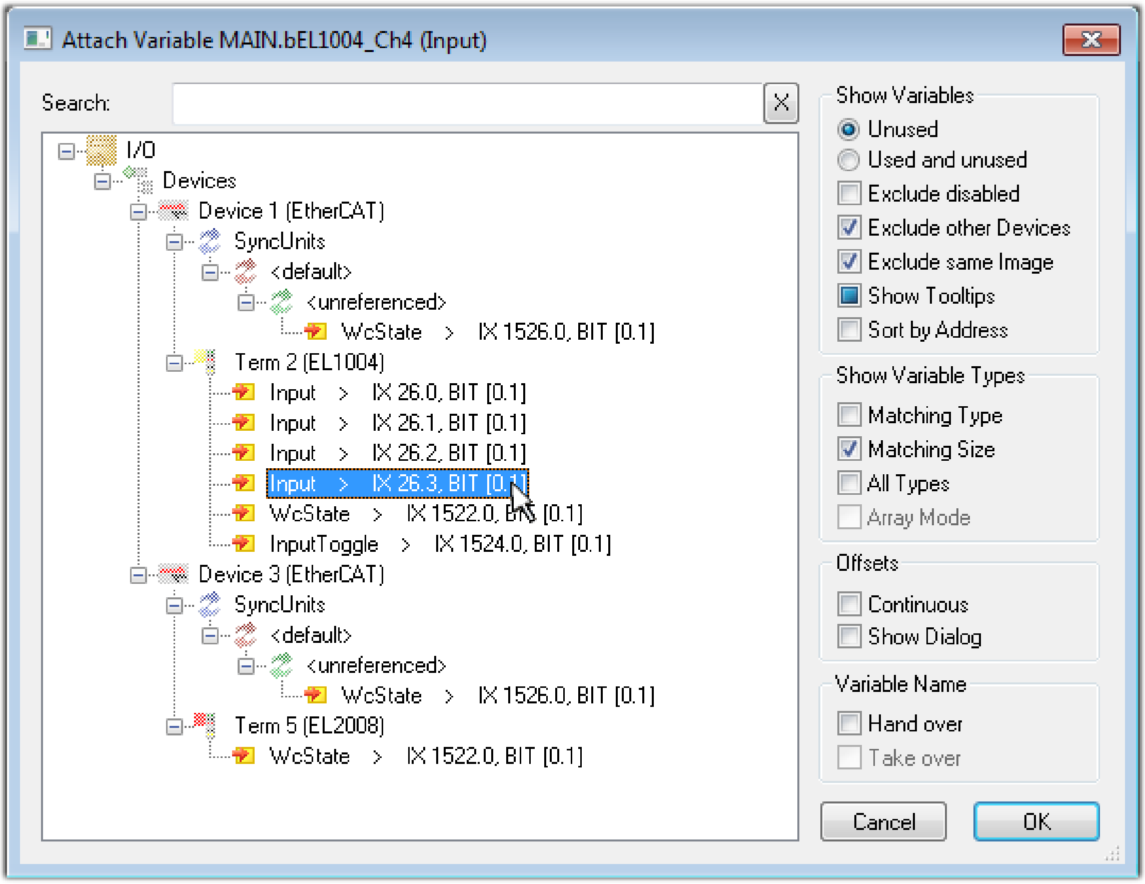


Fig. 129: Creating the links between PLC variables and process objects

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


Attach Variable MAIN.bEL1004_Ch4 (Input)

Search:

```
I/O
  Devices
    Device 1 (EtherCAT)
      SyncUnits
        <default>
          <unreferenced>
            WcState   >   IX 1526.0, BIT [0.1]
      Term 2 (EL1004)
        Input   >   IX 26.0, BIT [0.1]
        Input   >   IX 26.1, BIT [0.1]
        Input   >   IX 26.2, BIT [0.1]
        Input   >   IX 26.3, BIT [0.1]
        WcState   >   IX 1522.0, BIT [0.1]
        InputToggle   >   IX 1524.0, BIT [0.1]
    Device 3 (EtherCAT)
      SyncUnits
        <default>
          <unreferenced>
            WcState   >   IX 1526.0, BIT [0.1]
      Term 5 (EL2008)
        WcState   >   IX 1522.0, BIT [0.1]
```

Show Variables
● Unused
○ Used and unused
□ Exclude disabled
☑ Exclude other Devices
☑ Exclude same Image
■ Show Tooltips
□ Sort by Address

Show Variable Types
□ Matching Type
☑ Matching Size
□ All Types
□ Array Mode

Offsets
□ Continuous
□ Show Dialog

Variable Name
□ Hand over
□ Take over

Fig. 130: Selecting BOOL-type PDO

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

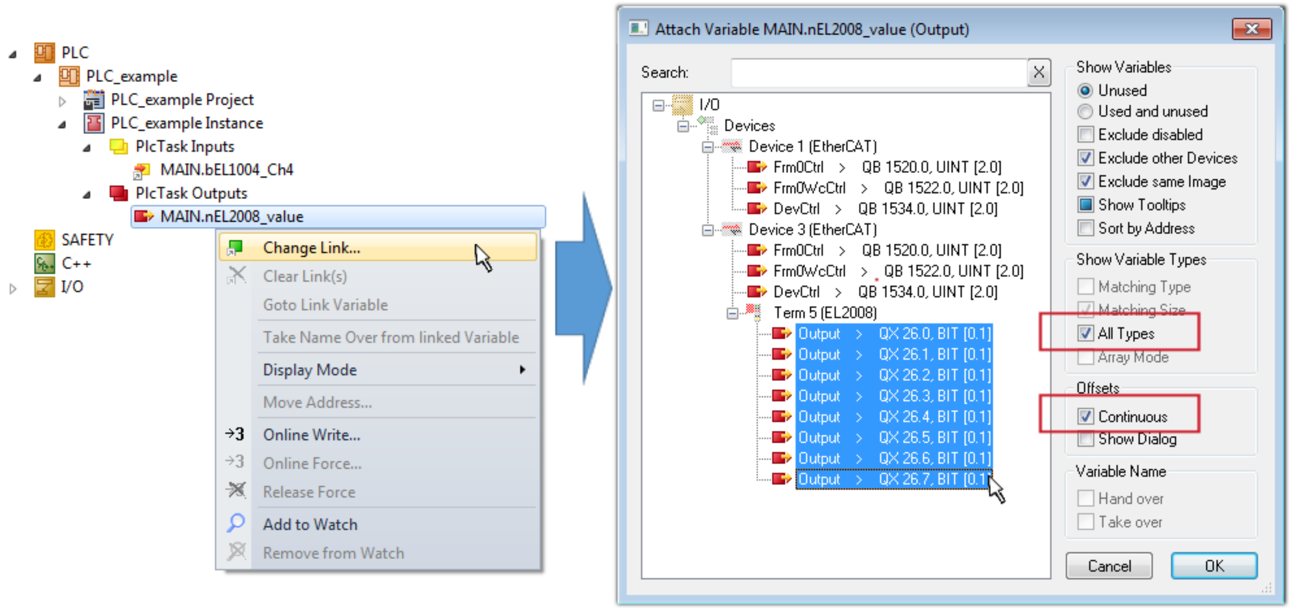


Fig. 131: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

Note that the “Continuous” checkbox was also activated. This is designed to allocate the bits contained in the byte of the “nEL2008_value” variable sequentially to all eight selected output bits of the EL2008 Terminal. It is thus possible to subsequently address all eight outputs of the terminal in the program with a byte

corresponding to bit 0 for channel 1 to bit 7 for channel 8 of the PLC. A special symbol (⊡) on the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting “Goto Link Variable” from the context menu of a variable. The opposite linked object, in this case the PDO, is automatically selected:

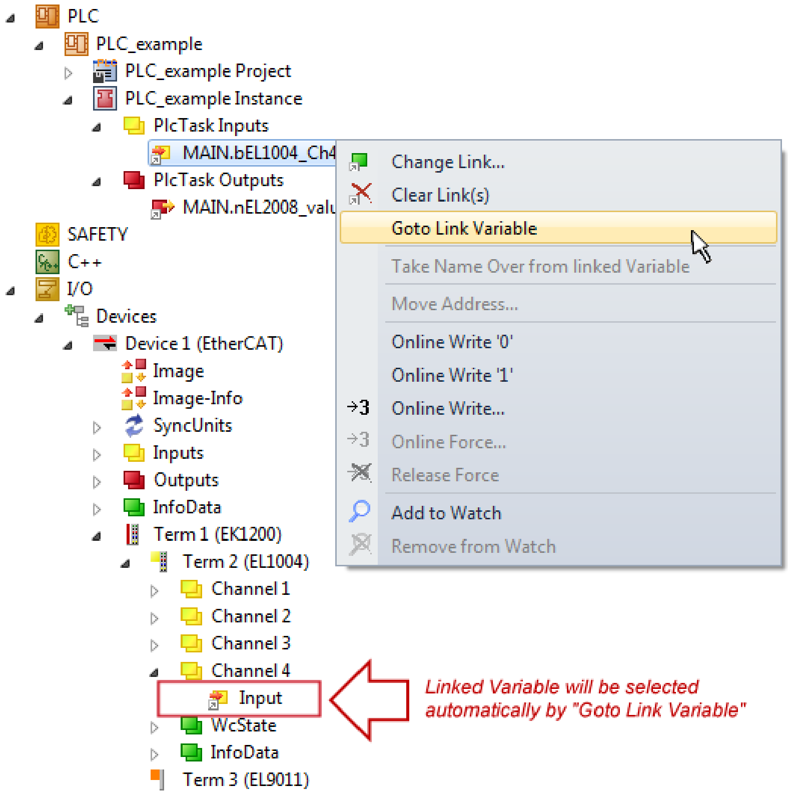


Fig. 132: Application of a “Goto Link Variable”, using “MAIN.bEL1004_Ch4” as an example

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

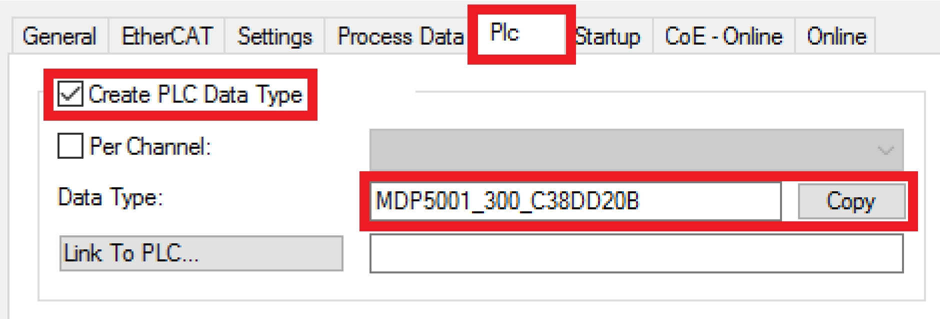
Note on type of variable assignment

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

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

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

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



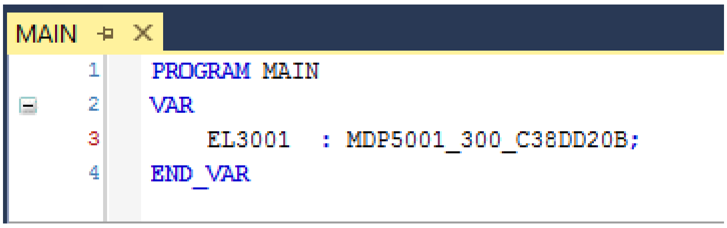
☑ Create PLC Data Type

□ Per Channel:

Data Type: MDP5001_300_C38DD20B

Fig. 133: Creating a PLC data type

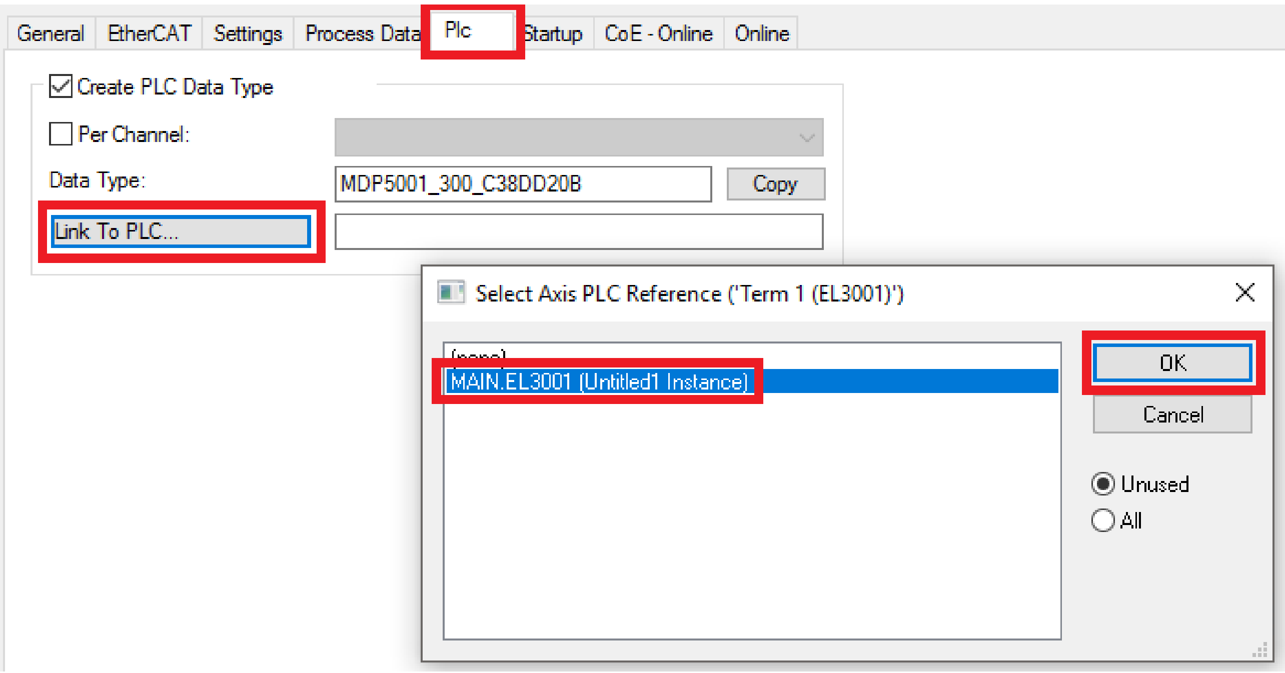
4. An instance of the data structure of the copied data type must then be created in the PLC.



```
PROGRAM MAIN
VAR
    EL3001   : MDP5001_300_C38DD20B;
END_VAR
```

Fig. 134: 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.



☑ Create PLC Data Type

□ Per Channel:

Data Type: MDP5001_300_C38DD20B

Select Axis PLC Reference ('Term 1 (EL3001)')

(none)

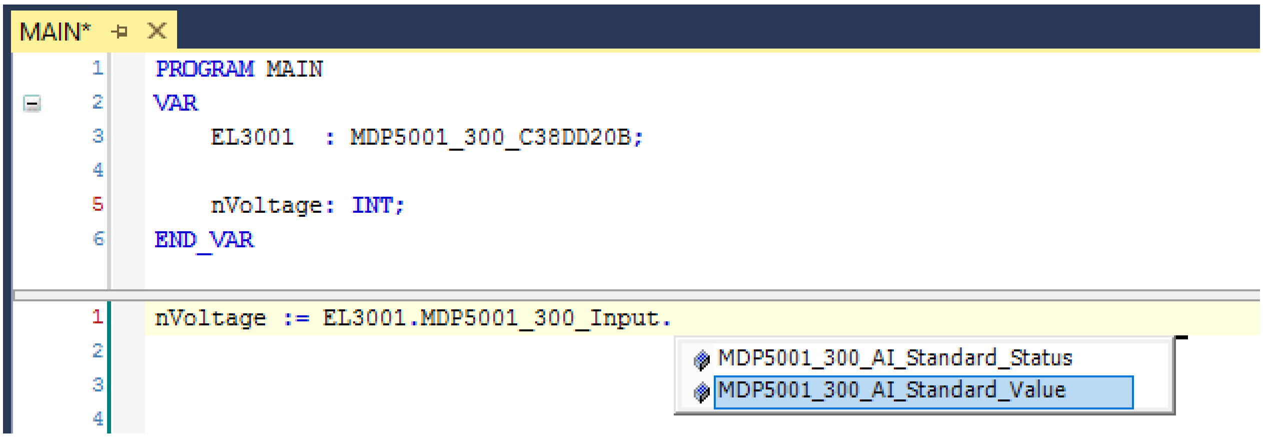
MAIN.EL3001 (Untitled1 Instance)

● Unused

○ All

Fig. 135: Linking the structure

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



```
PROGRAM MAIN
VAR
    EL3001   : MDP5001_300_C38DD20B;

    nVoltage: INT;
END_VAR
```

```
nVoltage := EL3001.MDP5001_300_Input.
```

Fig. 136: Reading a variable from the structure of the process data

Activation of the configuration

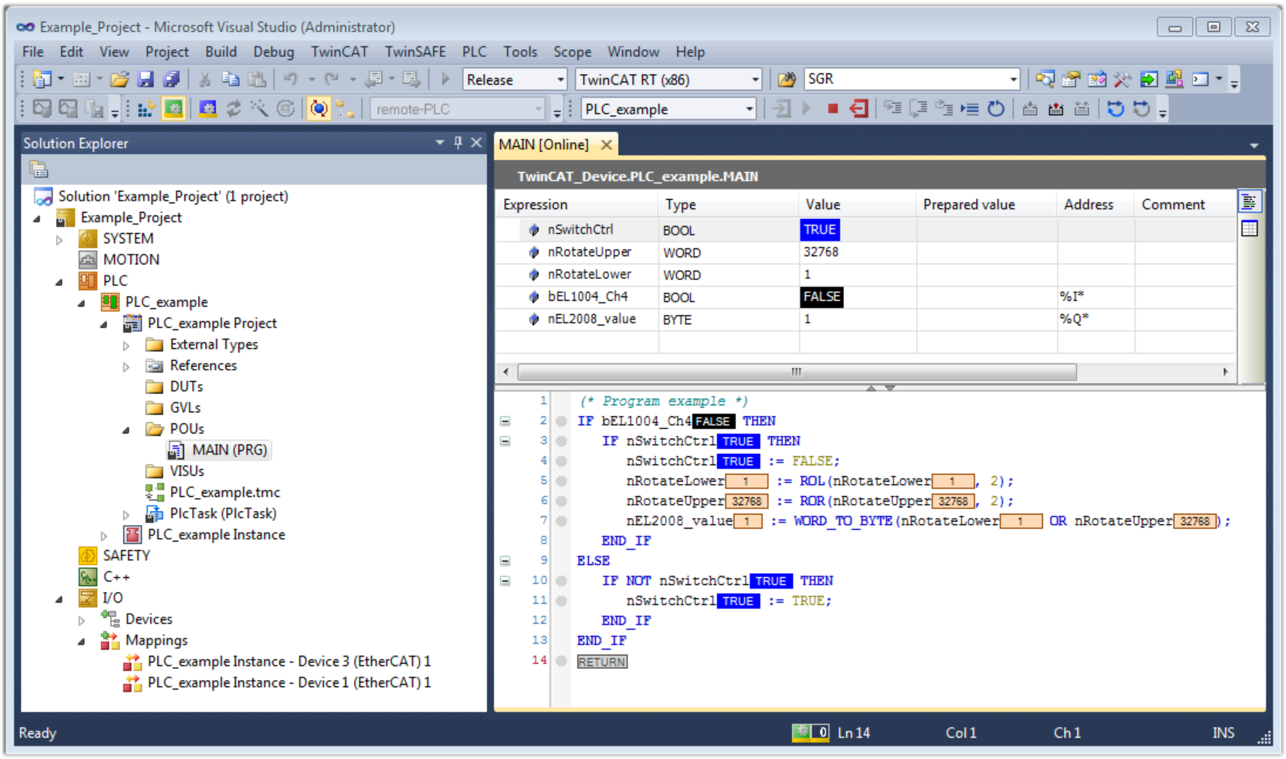
The allocation of PDO to PLC variables has now established the connection from the controller to the inputs and outputs of the terminals. The configuration can now be activated with or via the menu under “TwinCAT” in order to transfer the settings of the development environment to the runtime system.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

- Mappings
  - PLC_example Instance - Device 3 (EtherCAT) 1
  - PLC_example Instance - Device 1 (EtherCAT) 1

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

Starting the controller

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



TwinCAT_Device.PLC_example.MAIN

Expression	Type	Value	Prepared value	Address	Comment
nSwitchCtrl	BOOL	TRUE			
nRotateUpper	WORD	32768			
nRotateLower	WORD	1			
bEL1004_Ch4	BOOL	FALSE		%I*	
nEL2008_value	BYTE	1		%Q*	

```
(* Program example *)
IF bEL1004_Ch4 FALSE THEN
   IF nSwitchCtrl TRUE THEN
      nSwitchCtrl TRUE := FALSE;
      nRotateLower 1 := ROL(nRotateLower 1 , 2);
      nRotateUpper 32768 := ROR(nRotateUpper 32768 , 2);
      nEL2008_value 1 := WORD_TO_BYTE(nRotateLower 1 OR nRotateUpper 32768 );
   END_IF
ELSE
   IF NOT nSwitchCtrl TRUE THEN
      nSwitchCtrl TRUE := TRUE;
   END_IF
END_IF
RETURN
```

Fig. 137: TwinCAT 3 development environment (VS shell): logged-in, after program startup

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

5.2 Quick Start

5.2.1 Notes on commissioning

Observe the following notes before commissioning:

NOTE

Application from FW02/Revision0017

The documentation applies to the EL2596 only from firmware 02 with the revision -0017. With all older software revisions it is mandatory to perform a firmware update EL/ES/ELM/EM/EPxxxx and to update the ESI files in TwinCAT.

- After a firmware update, [▸ 249] the default parameters should **always** be reset. To do this, the password 0x64616F6C must be entered in the CoE object 0x1011:01 [▸ 223].
- If the EL2596-xxxx is operated as a power source, no check is carried out when setting the LED current as to whether or not the connected LED is designed for the set current.
- The load must always be connected potential-free to the EL2596-xxxx with a two-wire connection. A connection to the input voltage is not permitted due to the internal current measurement.
- Operation of multi-channel LEDs
  - If multi-channel/multi-color LEDs are to be operated with the single-channel EL2596-xxxx, a corresponding number of EL2596-xxxxs is required
  - Operation in a common anode circuit with a series resistor, i.e. in voltage mode, or without a series resistor in current-controlled mode with external supply is possible as described in Operation of a multi-color Common Anode LED [▸ 190]
  - Operation in a common cathode circuit is not possible.
- Short cable lengths are recommended for high-precision control of the LED lighting.
- The EL2596-xxxx must always be supplied with power via the connection contacts 6, 7 or 8 and 14, 15 or 16.
  Contacts 6, 7 and 8 and 14, 15 and 16 are each connected internally.
  The terminal is **not** supplied via the power contacts.

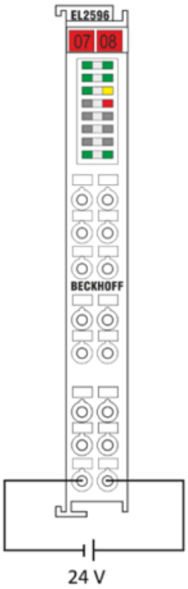


Fig. 138: Power supply

- Only regulated power supply units should be used as power supplies.
- The use of an EL9570 is recommended for a power supply with voltage peaks. The 24 V supply can be smoothed via the passive buffer capacitor terminal.
- The operating range is limited by three parameters:

- ◦ the maximum continuous output current, which is limited by internal components
- ◦ the maximum output voltage of the terminal
- ◦ the maximum output power of 14.4 W, which correlates with the internal power loss. The internal power dissipation in mW can be checked in the CoE object 0xF900:13 [▶ 221].

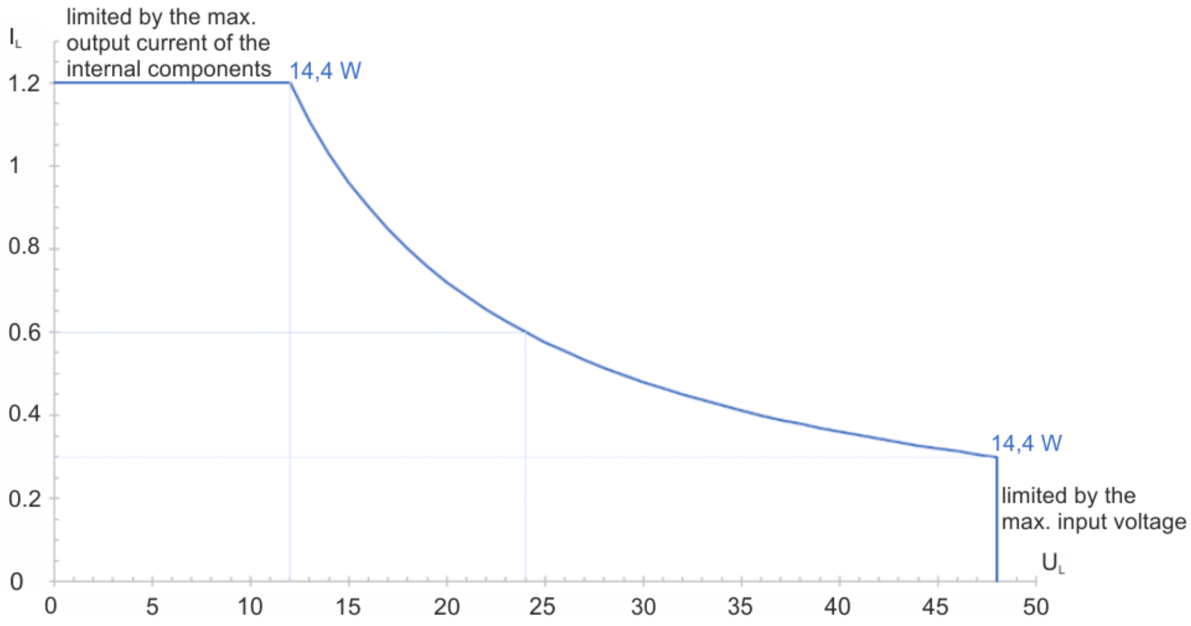


Fig. 139: EL2596 operating range in Current Control and Voltage Control Mode

- If currents are to be operated above the nominal current of the LED in a pulse mode, a higher specification is necessary for the output voltage in the CoE parameter 0x8000:04 [▶ 217] due to the non-linear behavior of LEDs and the current control capability of the circuit.
- In the PDOs, all times (pulse length, delay, etc.) are specified in the unit 1 µs. These times are specified in the unit 0.1 µs in the CoE parameters. The resolution of the times in the PDOs can be reduced to 100 ns (0.1 µs) with the CoE parameter 0x8002:31 [▶ 218] “Pulse resolution 100 ns”.
- If the predefined PDOs are selected with “info data”, internal data of the EL2596 can be mapped in order to monitor the values. The variables to be displayed can be selected in the CoE parameters 0x8002:11 [▶ 218] and 0x8002:19 [▶ 218].
- The “Ready to activate” bit must be active in order to be able to operate the LED. This is only the case if no errors or warnings are pending. (see Diagnostics [▶ 204])
- Especially in the pulsing modes leakage currents can occur which cause the connected LED to glow. To avoid this glow, the "Enable" bit must be deactivated if the LED output should be inactive.
- No wire break detection is possible in pulsing modes.
- External ventilation
  - ◦ The ZB8610 fan cartridge can be mounted under the terminal. It reduces the internal temperature by approx. 20 °C. The internal temperature of the terminal can be viewed in the CoE 0xF900 or via the PDO InfoData. It is usually approx. 30 °C above the ambient temperature without a fan module.
  - ◦ EL2596: For demanding applications with high power requirements, it may be useful to mount a fan cartridge under the terminal to support cooling.
  - ◦ EL2596-0010: For operation of the 48 V variant EL2596-0010, a ZB8610 fan cartridge is mandatory to ensure that the terminal does not exceed the maximum allowed internal temperature.
- The parallel connection of several EL2596-xxxx for the purpose of increasing the current through one or more LEDs is not permitted due to the internal circuit in the EL2596-xxxx.

Time behavior in continuous light mode

The time behavior in continuous light mode is illustrated in the following figure.

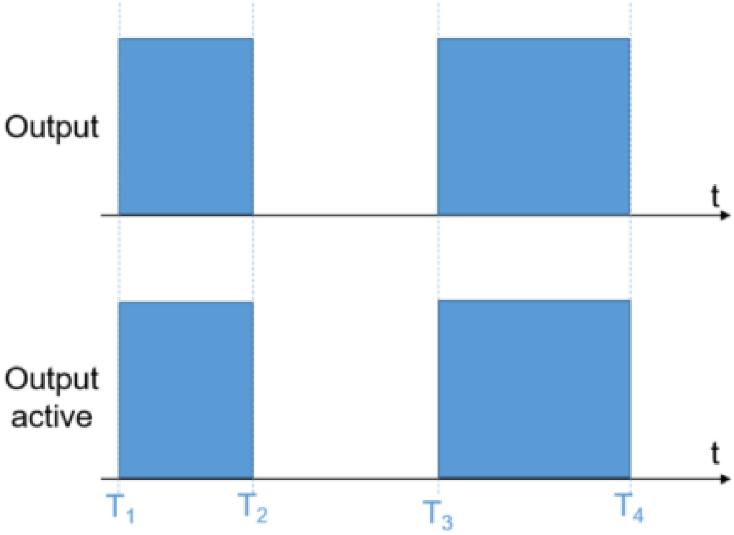


Fig. 140: Timing - Continuous light

- T1 and T3: Switch-on times
- T2 and T4: Switch-off times
- In the case of continuous light, the signal “DOX Control” → “Control” → “Output is switched directly through to the driver stage”.
- The status is returned as feedback via “DOX Status” → “Status” → “**Output active”.**

Time behavior in pulse mode

The time behavior in a pulse mode is illustrated in the following figure.

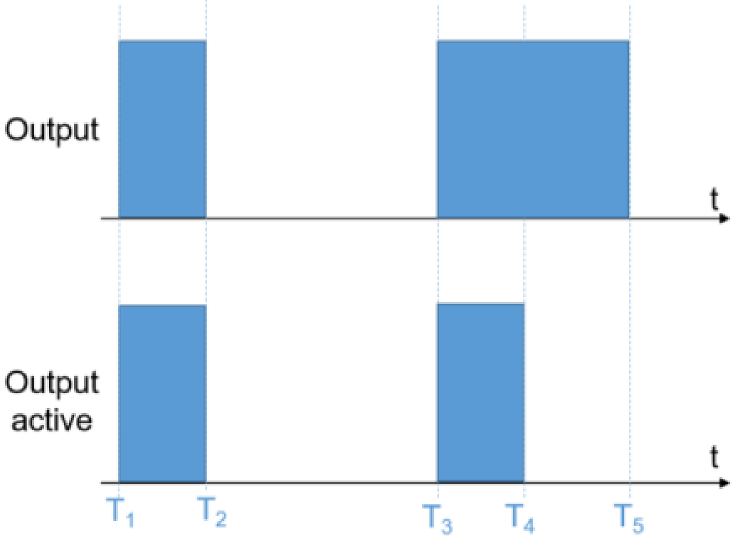


Fig. 141: Timing - Pulse

- T1 and T3: Switch-on times
- T2 and T5: Switch-off time of the output bit
- T4: Switch-off time of the LED
- The LED output is switched on as soon as the output bit is set to TRUE.
- If the output bit is set to FALSE **before** the specified pulse length expires, the LED is also switched off

- If the output bit is TRUE for longer than the specified pulse length, the LED is switched off after the expiry of the time (T4 - T3). T5 then no has no further influence on the output.

The time behavior in pulse mode with an external trigger signal is illustrated in the following figure. Two different cases of the output of a pulse by trigger input are shown.

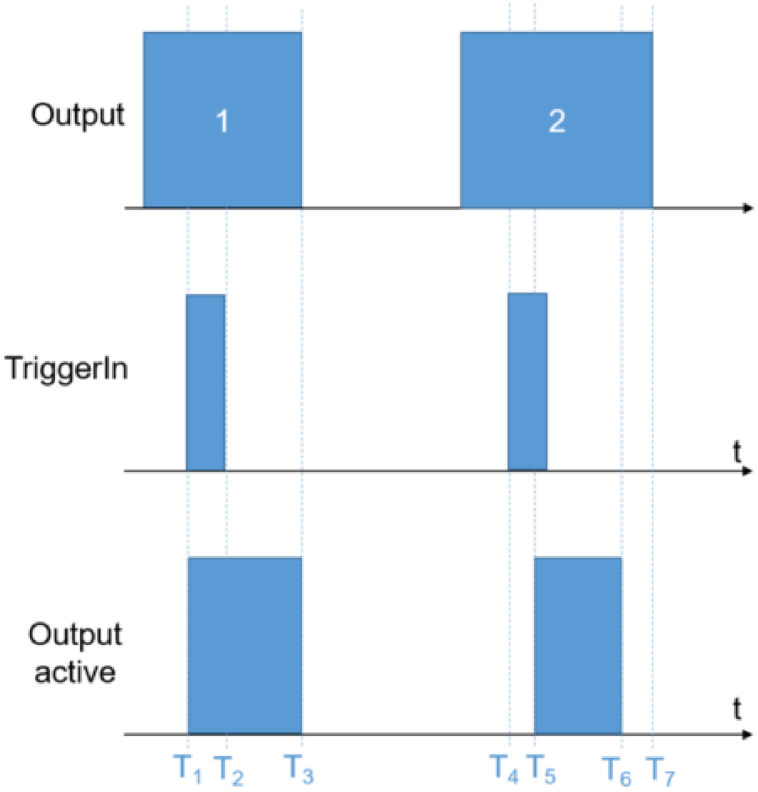


Fig. 142: Timing - External trigger

In order to set the LED output, **Input trigger enable** must be activated in addition to the **Output** bit in order to enable the trigger function.

Case 1 – no delay:

- **Output** and **Input trigger enable** are already enabled at T1 and the external trigger event is detected.
- **Output active** is set immediately, as no output delay is set.
- After the specified pulse length T3 - T1, or if the **output** bit is set to FALSE, the LED output is also reset.

Case 2 – with delay:

- **Output** and **Input trigger enable** are already enabled at T4 and the external trigger event is detected.
- **Output active** is set for the specified pulse duration T6 – T5 after expiry of the specified delay time.
- The resetting of **Output** at time T7 has no effect on the output, since the output time has already expired.

5.2.2 Commissioning

Connecting the terminal

Install the terminal as described in the chapter entitled Installation [▶ 49].

Adjusting the terminal

This documentation applies from firmware 02, Revision 0017.

Note: Input voltage and output voltage are monitored, the limits can be set in the CoE. This can look like the following, for example:

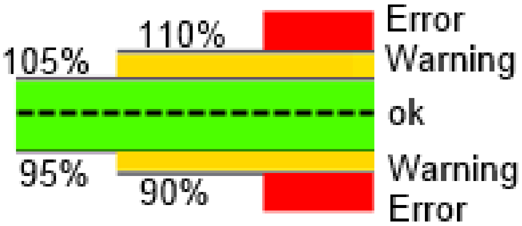
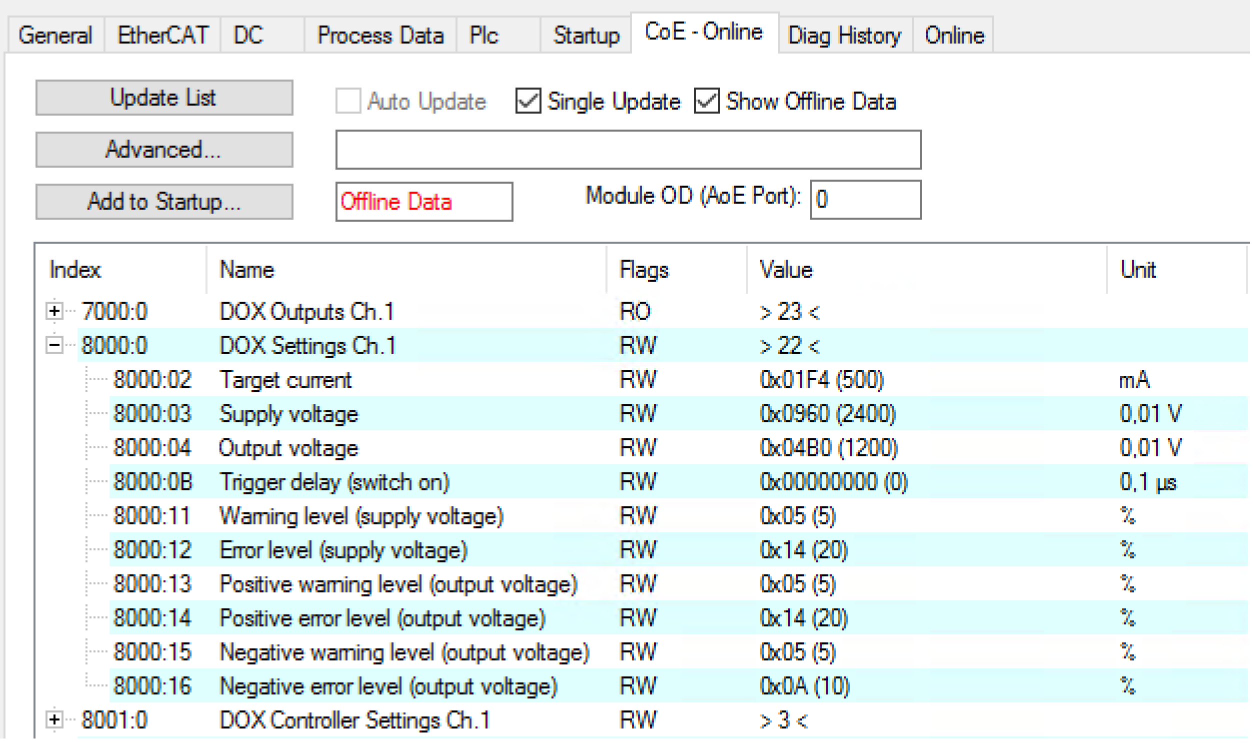


Fig. 143: Representation of warning thresholds

1. Install the current ESI (XML file) EL2596-00x0-xxxx in TwinCAT (usually under C:\TwinCAT\3.1\Config\IO\EtherCAT\).
   If necessary, download the latest ESI from the Beckhoff website.
2. If the terminal has already been used to perform tests or if you have performed a firmware update, perform a "Restore default parameter" in the CoE: enter “0x64616F6C” in CoE parameter 0x1011:01
3. Enter the key data of the supply in the terminal
   In 0x8000:03 [▶ 217] the supply voltage “Supply Voltage”
   Warning and ErrorLevel for this in 0x8000:11/12 [▶ 217] (lower limit only)
4. Determine the key data of the intended LED:
   - Nominal current of the LED in continuous light mode or maximum permissible limit current when overdriving [mA]. Enter this in CoE parameter 0x8000:02 [▶ 217] “Target current”
   - Required forward voltage for that [V]
     Reason: the current controller regulates the forward voltage itself in order to reach/maintain the desired load current.
     Self-heating effects, for example, are thus compensated. So that the load current is as accurate as possible in the first few milliseconds after switch-on, however, the controller needs to know the approximate forward voltage that it has to operate with. The forward voltage or characteristic curve of an LED with a specified forward current can also be determined automatically by the terminal (see Automatic determination of the output voltage and display of the characteristic curve [▶ 166])
     Enter the forward voltage in 0x8000:04 [▶ 217] “Output voltage”
     Warning and ErrorLevel for this in 0x8000:13/16 [▶ 217]



Index	Name	Flags	Value	Unit
7000:0	DOX Outputs Ch.1	RO	> 23 <	
8000:0	DOX Settings Ch.1	RW	> 22 <	
8000:02	Target current	RW	0x01F4 (500)	mA
8000:03	Supply voltage	RW	0x0960 (2400)	0,01 V
8000:04	Output voltage	RW	0x04B0 (1200)	0,01 V
8000:0B	Trigger delay (switch on)	RW	0x00000000 (0)	0,1 µs
8000:11	Warning level (supply voltage)	RW	0x05 (5)	%
8000:12	Error level (supply voltage)	RW	0x14 (20)	%
8000:13	Positive warning level (output voltage)	RW	0x05 (5)	%
8000:14	Positive error level (output voltage)	RW	0x14 (20)	%
8000:15	Negative warning level (output voltage)	RW	0x05 (5)	%
8000:16	Negative error level (output voltage)	RW	0x0A (10)	%
8001:0	DOX Controller Settings Ch.1	RW	> 3 <	

Fig. 144: Setting current and voltage values in the CoE directory

5. The specific settings for the various operation modes can be found in the chapter Setting the operation modes [▶ 131].

6. Further useful CoE settings

   - The information to be transmitted for the real-time diagnosis can be set in 0x8002:11 [▶ 218] and 0x8002:19 [▶ 218].
   - Diagnostics [▶ 204] over CoE
   - For safety's sake these CoE data can also be entered into the Startup list. If the EL2596 is exchanged later, the online CoE data are no longer available. The System Manager then writes the data into the new terminal.

Additional Notes

For all notes on commissioning, see Notes on commissioning [▶ 125].

5.3 Setting the operation modes

From FW04 the EL2596-xxxx provides eight modes for controlling an LED. The operation mode must be selected in the CoE parameter 0x8004:01 [▸ 219] “Led operation mode”.

1. Current Control
   - The current at the LED output is continuously controlled. This mode thus enables lighting without flickering. However, in contrast to the pulsed modes, this operation mode is sluggish and is therefore not suitable for short pulses. The output current has to be reduced in order to reduce the brightness.
2. Current Control Timestamp Pulse
   - This operation mode enables an LED to be driven via the Distributed Clocks. Pulse peak current, pulse duration and pulse start time are defined by the process data. The pulse time is specified via the Distributed Clocks. Pulses between 25 µs and 10 s can be generated.
3. Current Control Trigger Pulse
   - Pulse peak current and pulse duration are defined by the process data. The pulse can be triggered by applying a rising or falling edge to the trigger input “TrigIn”. It is possible to delay the output signal by a specified time. Pulses between 25 µs and 10 s can be generated. From FW04, an output synchronous to the trigger input is also configurable, so that the LED output is active as long as a signal is applied to the trigger input.
4. Current Control PLC Pulse
   - Pulse peak current and pulse duration are defined by the process data. The pulse is triggered by a bit in the PLC. Pulses between 25 µs and 10 s can be generated.
5. Current Control PWM
   - This operation mode offers constant current operation with PWM support. The brightness of the LED can be controlled via the PWM frequency and the PWM duration as well as the amplitude of the output current. If the set output current corresponds to the nominal current of the LED, the brightness can be specified very precisely with true color via the duty cycle. Alternatively, it is also possible to reduce the brightness by reducing the set output current.
6. Voltage Control
   - The voltage at the output terminal is kept constant.
7. Voltage Control PWM
   - This operation mode offers constant voltage operation with PWM support. The brightness of the LED can be controlled via the PWM frequency and the PWM duration as well as the voltage amplitude.
8. Current Sink PWM (from FW04)
   - In this mode, LEDs can be operated with an external power supply. This can be used, for example, when controlling multi-color Common Anode LEDs. The power can be provided by another EL2596 or an external power supply unit.

The specific settings for each operation mode are described below.

5.3.1 Current control

It is possible to use the LED output as a digital or current-controlled LED output via the selection of the Predefined PDOs.

In this mode the maximum output current is 1.2 A up to an LED voltage of 12 V. Above 12 V the user must consider derating, as otherwise the module may switch off due to overtemperature if the power exceeds 14.4 W.

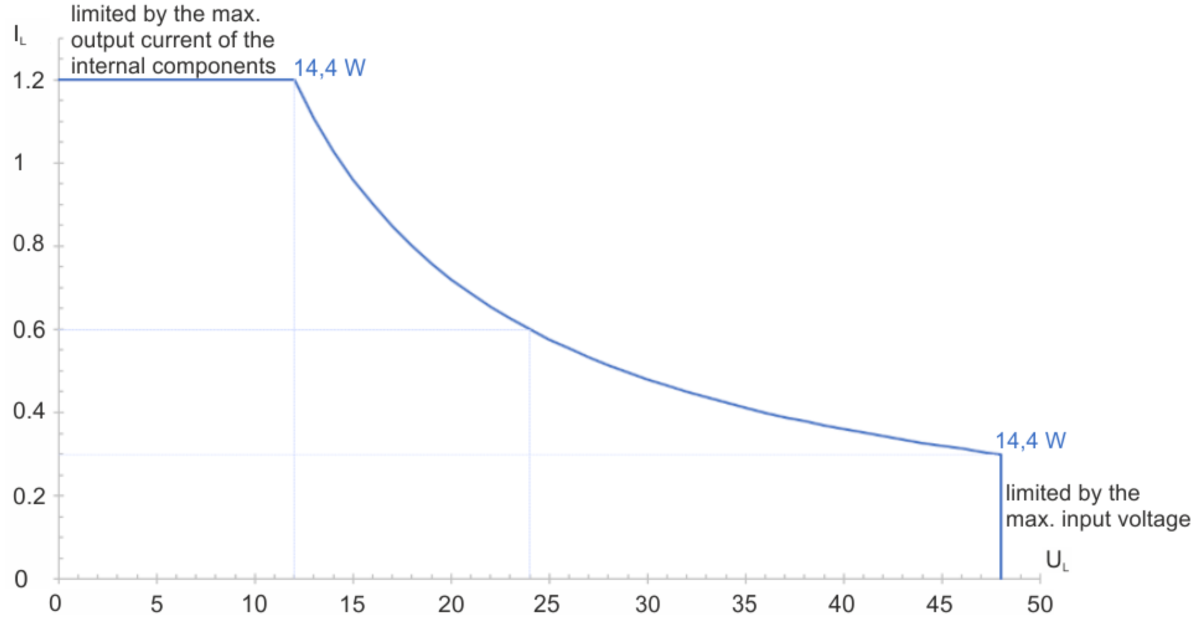


Fig. 145: Operating range of EL2596 in "Current control" mode

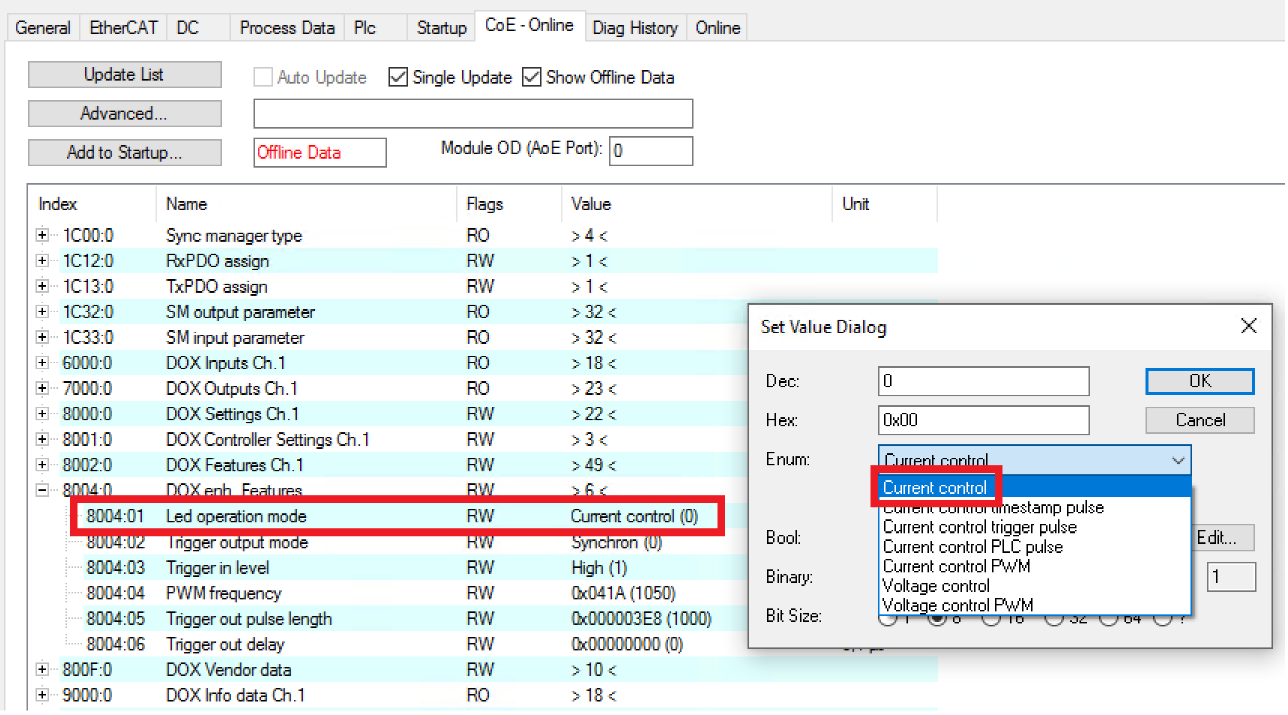
If an LED is to be operated with an nominal current $I_{nominal}$ < 10 mA, the Current Control PWM [▸ 148] mode should be used.

In this operation mode, it is possible to configure the trigger input as an enable input so that it can be used as an external switch. An output can then only be actuated with a specified signal at the trigger input. More detailed information and commissioning can be found in the chapter Hardware enable [▸ 178].

Current-controlled LED output

Make the following settings:

1. Nominal/limit current of the LED in the unit mA in the CoE parameter 0x8000:02 [▸ 217] “Target current”
2. Input voltage in the unit 0.01 V in the CoE parameter 0x8000:03 [▸ 217] “Supply voltage”
3. Maximum output voltage in the unit 0.01 V (max. U_{IN} - 0.5 V) in the CoE parameter 0x8000:04 [▸ 217] “Output voltage”. In this mode, the controller outputs only the forward voltage required for the set forward current. Nevertheless, the forward voltage should still be specified for the desired LED current in order to protect the connected LED from overvoltage. The TeachIn function [▸ 166] can also be used to determine the output voltage.
4. Set the operation mode to “Current control” in the CoE directory in the parameter 0x8004:01 [▸ 217]



□ Auto Update ☑ Single Update ☑ Show Offline Data

Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
1C00:0	Sync manager type	RO	> 4 <	
1C12:0	RxPDO assign	RW	> 1 <	
1C13:0	TxPDO assign	RW	> 1 <	
1C32:0	SM output parameter	RO	> 32 <	
1C33:0	SM input parameter	RO	> 32 <	
6000:0	DOX Inputs Ch.1	RO	> 18 <	
7000:0	DOX Outputs Ch.1	RO	> 23 <	
8000:0	DOX Settings Ch.1	RW	> 22 <	
8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
8002:0	DOX Features Ch.1	RW	> 49 <	
8004:0	DOX enh. Features	RW	> 6 <	
8004:01	Led operation mode	RW	Current control (0)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	
8004:06	Trigger out delay	RW	0x00000000 (0)	
800F:0	DOX Vendor data	RW	> 10 <	
9000:0	DOX Info data Ch.1	RO	> 18 <	

Dec: 0

Hex: 0x00

Enum: Current control

- Current control
- Current control timestamp pulse
- Current control trigger pulse
- Current control PLC pulse
- Current control PWM
- Voltage control
- Voltage control PWM

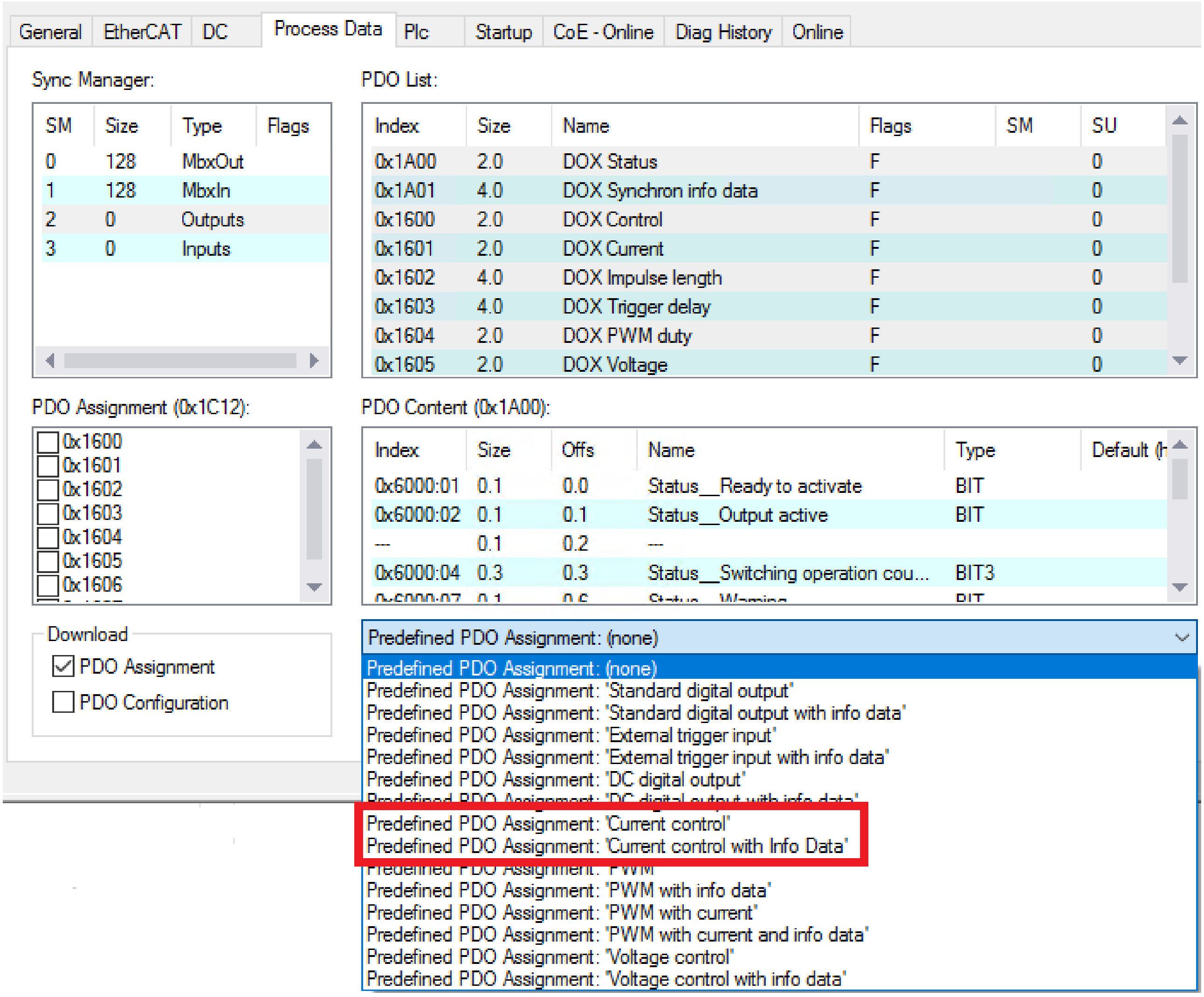
Bool:

Binary: 1

Bit Size: ○ 1 ● 8 ○ 16 ○ 32 ○ 64 ○ ?

Fig. 146: Operation mode setting “Current control”

5. Under “Predefined PDO Assignments”, set “Current control (with info data)”.



General | EtherCAT | DC | Process Data | Plc | Startup | CoE - Online | Diag History | Online

Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	0	Outputs	
3	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F		0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F		0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0

PDO Assignment (0x1C12):

□ 0x1600
□ 0x1601
□ 0x1602
□ 0x1603
□ 0x1604
□ 0x1605
□ 0x1606

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	

Download

☑ PDO Assignment

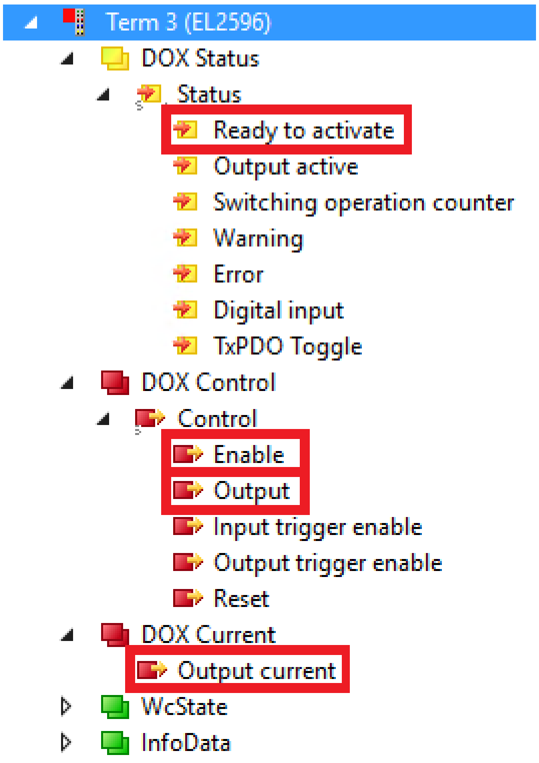
□ PDO Configuration

Predefined PDO Assignment: (none)

- Predefined PDO Assignment: (none)
- Predefined PDO Assignment: 'Standard digital output'
- Predefined PDO Assignment: 'Standard digital output with info data'
- Predefined PDO Assignment: 'External trigger input'
- Predefined PDO Assignment: 'External trigger input with info data'
- Predefined PDO Assignment: 'DC digital output'
- Predefined PDO Assignment: 'DC digital output with info data'
- Predefined PDO Assignment: 'Current control'
- Predefined PDO Assignment: 'Current control with Info Data'
- Predefined PDO Assignment: 'PWM'
- Predefined PDO Assignment: 'PWM with info data'
- Predefined PDO Assignment: 'PWM with current'
- Predefined PDO Assignment: 'PWM with current and info data'
- Predefined PDO Assignment: 'Voltage control'
- Predefined PDO Assignment: 'Voltage control with info data'

Fig. 147: PDO setting “Current Control (with info data)”

6. Specify the set current in the unit mA via “DOX Current” → “Output Current”
7. Check under “DOX Status” → “Status” whether the “Ready to activate” bit is 1.
8. Activate the control under “DOX Control” → “Control” via the “Enable” bit.
9. Switch on the LED output under “DOX Control” → “Control” by activating the “Output” bit.



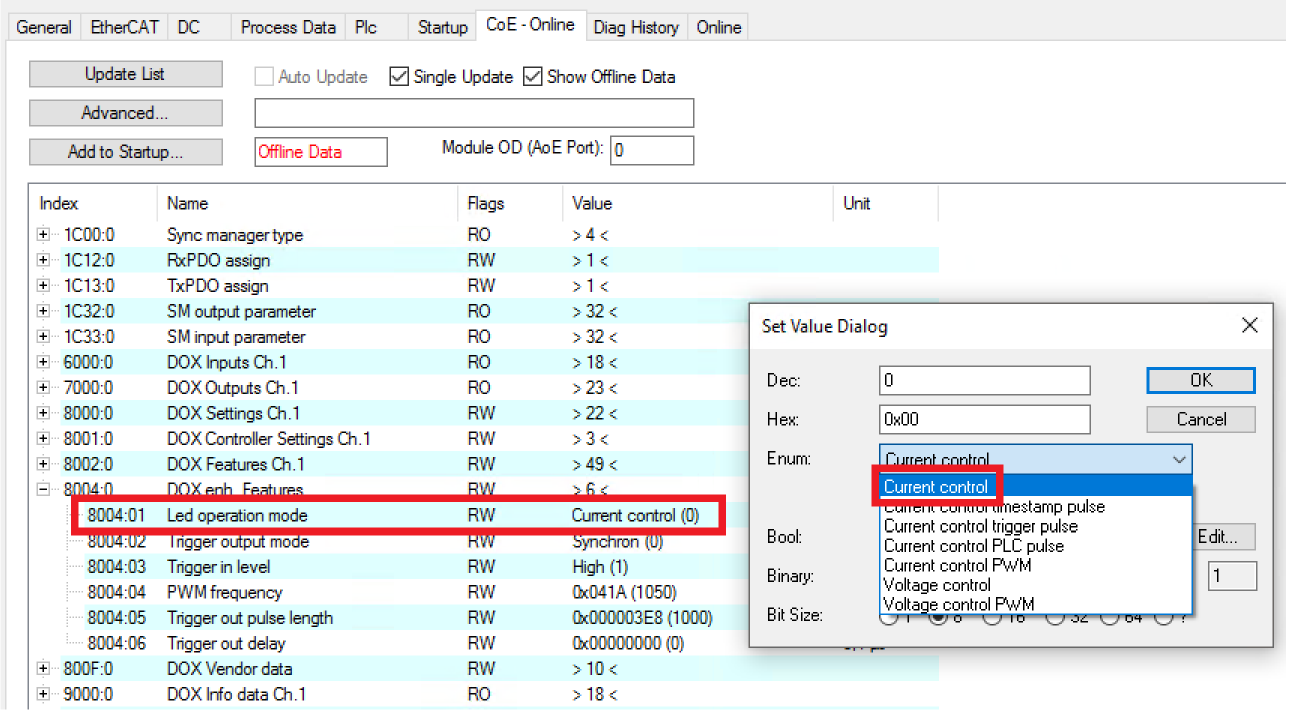
- Term 3 (EL2596)
  - DOX Status
    - Status
      - Ready to activate
      - Output active
      - Switching operation counter
      - Warning
      - Error
      - Digital input
      - TxPDO Toggle
  - DOX Control
    - Control
      - Enable
      - Output
      - Input trigger enable
      - Output trigger enable
      - Reset
  - DOX Current
    - Output current
  - WcState
  - InfoData

Fig. 148: Activate the output in the operation mode “Current control”

Digital current output

Make the following settings:

1. Output current in the unit mA in the CoE parameter 0x8000:02 [▸ 217] “Target current”
2. Input voltage in the unit 0.01 V in the CoE parameter 0x8000:03 [▸ 217] “Supply voltage”
3. Desired output voltage in the unit 0.01 V (max. U_{IN} - 0.5 V) in the CoE parameter 0x8000:04 [▸ 217] “Output voltage”
4. Set the operation mode in the CoE directory in the parameter 0x8004:01 [▸ 217] to “Current Control”



□ Auto Update ☑ Single Update ☑ Show Offline Data

Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
1C00:0	Sync manager type	RO	> 4 <	
1C12:0	RxPDO assign	RW	> 1 <	
1C13:0	TxPDO assign	RW	> 1 <	
1C32:0	SM output parameter	RO	> 32 <	
1C33:0	SM input parameter	RO	> 32 <	
6000:0	DOX Inputs Ch.1	RO	> 18 <	
7000:0	DOX Outputs Ch.1	RO	> 23 <	
8000:0	DOX Settings Ch.1	RW	> 22 <	
8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
8002:0	DOX Features Ch.1	RW	> 49 <	
8004:0	DOX enh. Features	RW	> 6 <	
8004:01	Led operation mode	RW	Current control (0)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	
8004:06	Trigger out delay	RW	0x00000000 (0)	
800F:0	DOX Vendor data	RW	> 10 <	
9000:0	DOX Info data Ch.1	RO	> 18 <	

Set Value Dialog

Dec: 0

Hex: 0x00

Enum: Current control

- Current control
- Current control timestamp pulse
- Current control trigger pulse
- Current control PLC pulse
- Current control PWM
- Voltage control
- Voltage control PWM

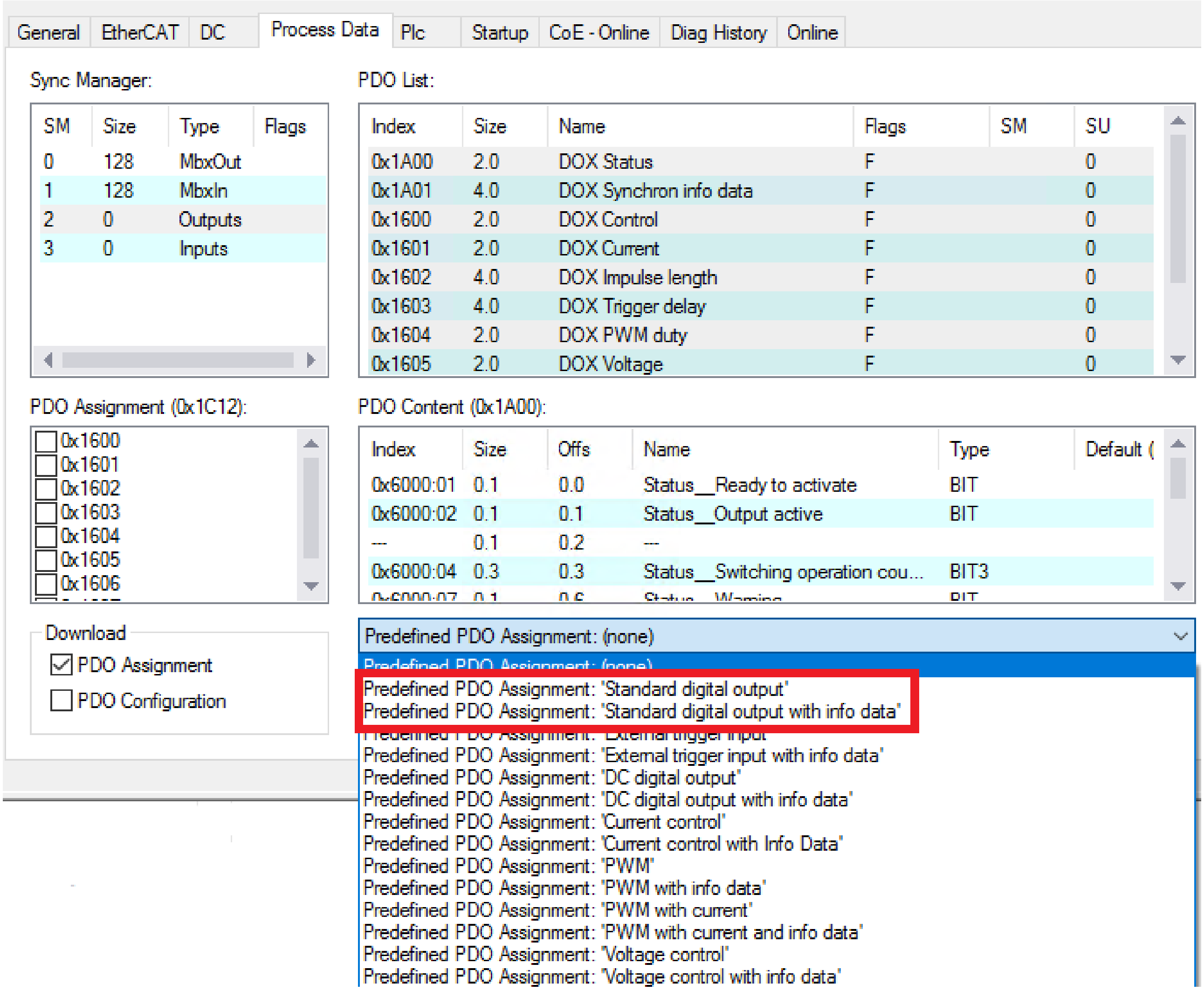
Bool:

Binary: 1

Bit Size: ○ 1 ● 8 ○ 16 ○ 32 ○ 64 ○ ?

Fig. 149: Operation mode setting “Current Control”

5. Set the “Predefined PDO Assignments” to “Standard digital output (with info data)”.



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	0	Outputs	
3	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F		0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F		0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0

PDO Assignment (0x1C12):

□ 0x1600
□ 0x1601
□ 0x1602
□ 0x1603
□ 0x1604
□ 0x1605
□ 0x1606

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	

Download
☑ PDO Assignment
□ PDO Configuration

Predefined PDO Assignment: (none)
Predefined PDO Assignment: (none)
Predefined PDO Assignment: 'Standard digital output'
Predefined PDO Assignment: 'Standard digital output with info data'
Predefined PDO Assignment: 'External trigger input'
Predefined PDO Assignment: 'External trigger input with info data'
Predefined PDO Assignment: 'DC digital output'
Predefined PDO Assignment: 'DC digital output with info data'
Predefined PDO Assignment: 'Current control'
Predefined PDO Assignment: 'Current control with Info Data'
Predefined PDO Assignment: 'PWM'
Predefined PDO Assignment: 'PWM with info data'
Predefined PDO Assignment: 'PWM with current'
Predefined PDO Assignment: 'PWM with current and info data'
Predefined PDO Assignment: 'Voltage control'
Predefined PDO Assignment: 'Voltage control with info data'

Fig. 150: PDO setting “Standard digital output (with info data)”

6. Check under “DOX Status” → “Status” whether the “Ready to activate” bit is 1.
7. Activate the control under “DOX Control” → “Control” via the “Enable” bit.
8. Switch on the LED output under “DOX Control” → “Control” by activating the “Output” bit.

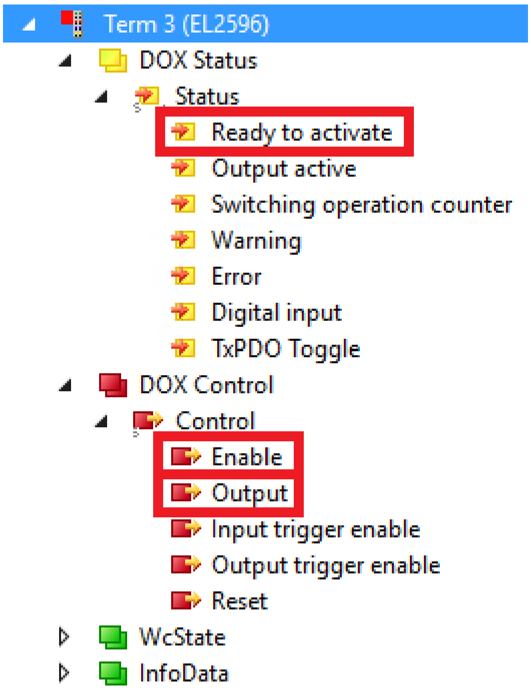


Fig. 151: Activate the output in operation mode “Current Control” as a digital output

5.3.2 Current control timestamp pulse

In pulse mode, the maximum current amplitude is 3 A with 12 V LEDs. It is possible to output a pulse in each cycle. The specified parameters for the current intensity and pulse duration are transmitted with each new specified start time. The data can be transmitted at the same time as the start time, so that a new pulse output with different parameters is possible in each cycle.

For all pulsed operating modes (current control timestamp pulse, current control trigger pulse, current control PLC pulse) the maximum constant duty cycle as a function of the output current must be considered.

- Iout = 1 A: 20% duty cycle

- Iout = 2 A: 10% Duty Cycle

- Iout = 3 A: 8% Duty Cycle

In this operation mode, it is possible to configure the trigger input as an enable input so that it can be used as an external switch. An output can then only be actuated with a specified signal at the trigger input. If the trigger input is active, a pulse with the specified parameters is output when the specified DC start time is reached. More detailed information and commissioning can be found in the chapter Hardware enable [▸ 178].

Especially in the current controlled pulse operation modes the use of the automatic determination of the output voltage [▸ 168] is recommended, because possibly for the generation of high current pulses at the output a high pre-voltage is needed to generate steep edges. These may be higher than voltages specified in the LED data sheet.

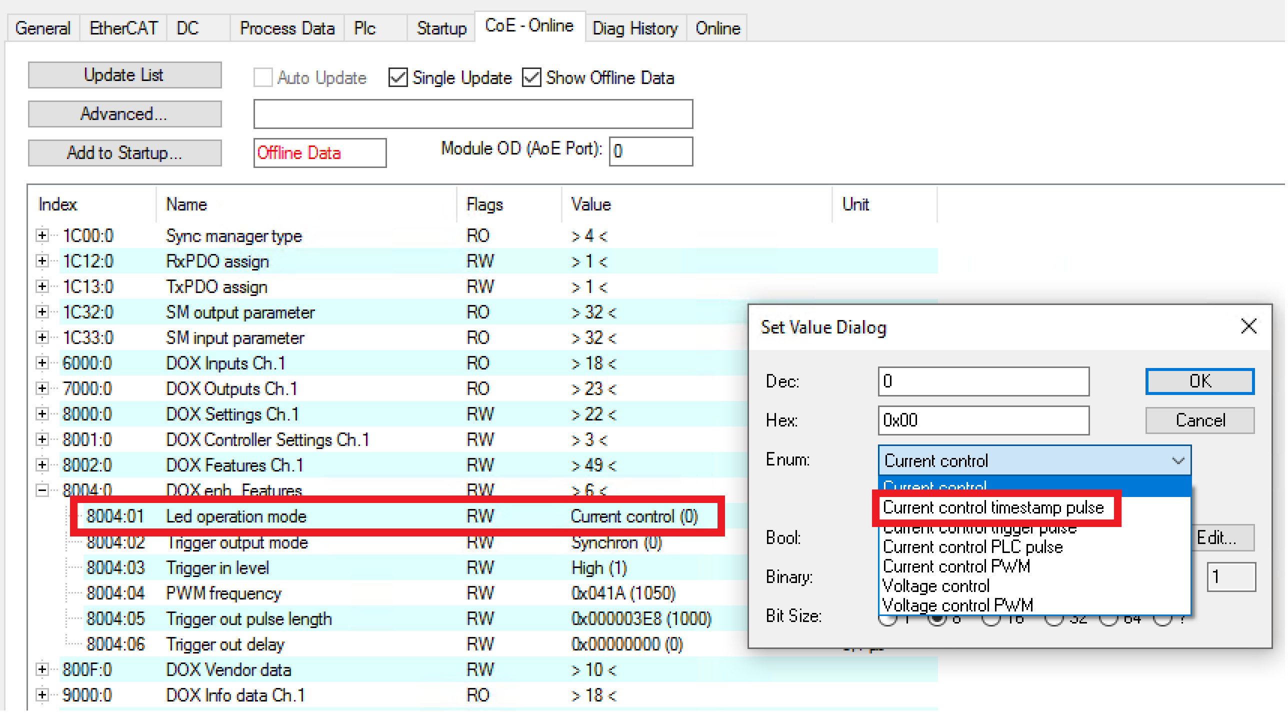
If the supply voltage is removed during operation in the pulsing modes, there will be a malfunction at the output. In addition, an overcurrent may occur at the LED output. Therefore, a stable voltage supply must be ensured. To minimize the risk of malfunction and overcurrent, the enable bit can be set together with the output bit so that the enable is not permanently set.

To set the LED output as a distributed clock controlled pulse output, make the following settings:

1. Nominal/limit current of the LED in the unit mA in the CoE parameter 0x8000:02 [▸ 217] “Target current”
2. Input voltage in the unit 0.01 V in the CoE parameter 0x8000:03 [▸ 217] “Supply voltage”
3. Desired output voltage in the unit 0.01 V (max. U_{IN} - 2 V) in the CoE parameter 0x8000:04 [▸ 217] “Output voltage”.

- ◦ If currents are to be output above the nominal current of the LED, a higher specification is necessary for the output voltage in the CoE parameter 0x8000:04 [▶ 217] due to the non-linear behavior of LEDs and the current control capability of the circuit.
- ◦ In this operation mode, the value set here corresponds to the output on the controller. If the forward voltage of the connected LED is lower than the value set here, the remaining voltage drops internally in the EL2596. In many cases, this leads to overtemperature in the terminal and thus to the switching off of the LED output. As the desired output voltage, therefore, the forward voltage at the desired output current should be set here. The TeachIn function [▶ 168] can also be used to determine the output voltage.

4. Set the operation mode in the CoE directory in the CoE parameter 0x8004:01 [▶ 219] to “Current Control timestamp pulse”



□ Auto Update ☑ Single Update ☑ Show Offline Data

Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
1C00:0	Sync manager type	RO	> 4 <	
1C12:0	RxPDO assign	RW	> 1 <	
1C13:0	TxPDO assign	RW	> 1 <	
1C32:0	SM output parameter	RO	> 32 <	
1C33:0	SM input parameter	RO	> 32 <	
6000:0	DOX Inputs Ch.1	RO	> 18 <	
7000:0	DOX Outputs Ch.1	RO	> 23 <	
8000:0	DOX Settings Ch.1	RW	> 22 <	
8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
8002:0	DOX Features Ch.1	RW	> 49 <	
8004:0	DOX enh. Features	RW	> 6 <	
8004:01	Led operation mode	RW	Current control (0)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	
8004:06	Trigger out delay	RW	0x00000000 (0)	
800F:0	DOX Vendor data	RW	> 10 <	
9000:0	DOX Info data Ch.1	RO	> 18 <	

Set Value Dialog

Dec: 0

Hex: 0x00

Enum: Current control

- Current control
- Current control timestamp pulse
- Current control trigger pulse
- Current control PLC pulse
- Current control PWM
- Voltage control
- Voltage control PWM

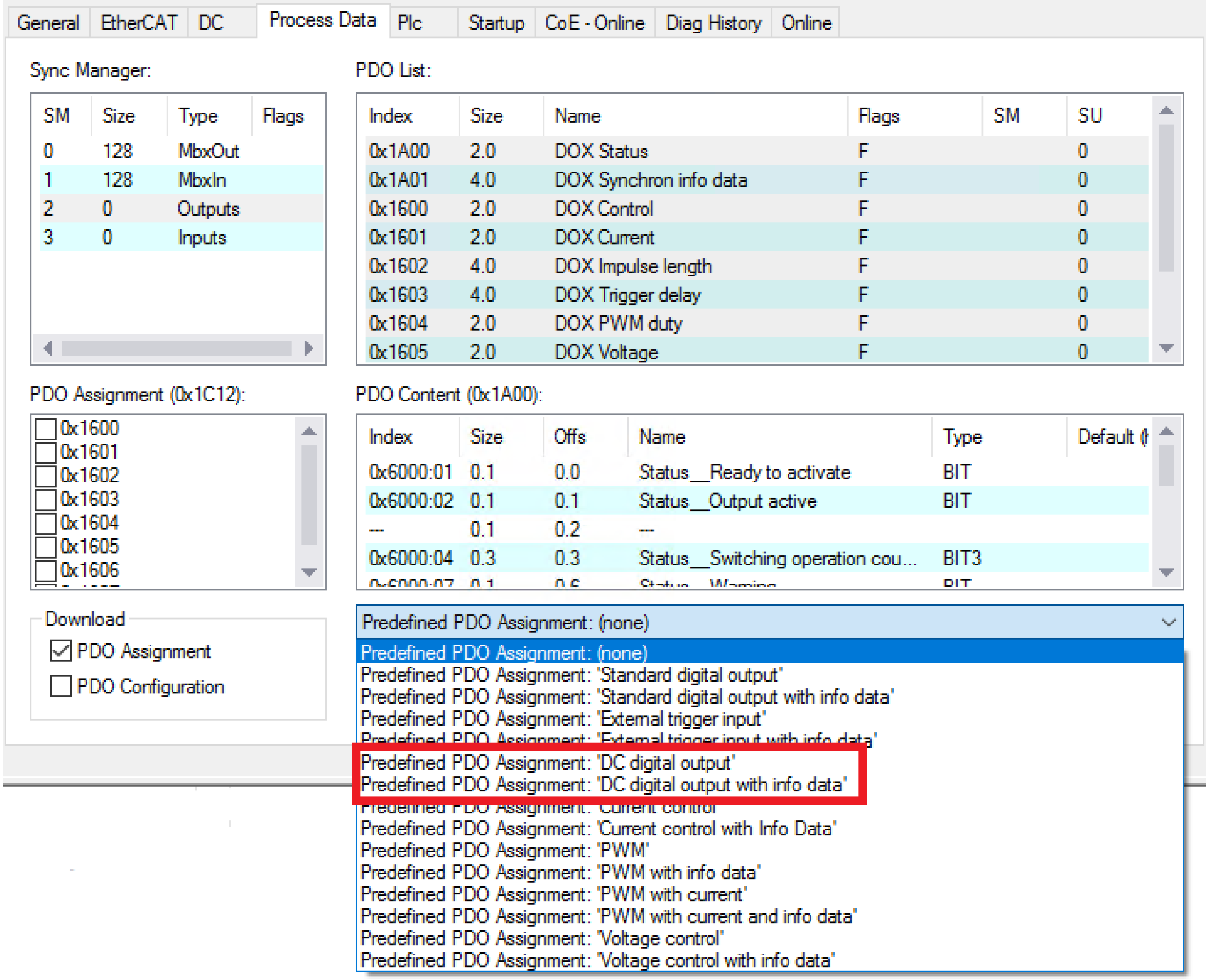
Bool:

Binary: 1

Bit Size: ○ 1 ● 8 ○ 16 ○ 32 ○ 64 ○ ?

Fig. 152: Operation mode setting “Current Control timestamp pulse”

5. In “Predefined PDO assignments”, select “DC digital output (with info data)”



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	0	Outputs	
3	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F		0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F		0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0

PDO Assignment (0x1C12):

□ 0x1600
□ 0x1601
□ 0x1602
□ 0x1603
□ 0x1604
□ 0x1605
□ 0x1606

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (!
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	

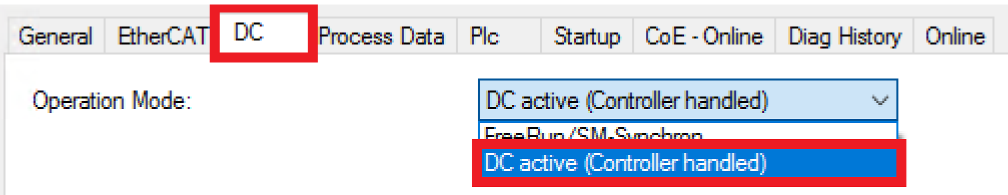
Download
☑ PDO Assignment
□ PDO Configuration

Predefined PDO Assignment: (none)

Predefined PDO Assignment: (none)
Predefined PDO Assignment: 'Standard digital output'
Predefined PDO Assignment: 'Standard digital output with info data'
Predefined PDO Assignment: 'External trigger input'
Predefined PDO Assignment: 'External trigger input with info data'
Predefined PDO Assignment: 'DC digital output'
Predefined PDO Assignment: 'DC digital output with info data'
Predefined PDO Assignment: 'Current control'
Predefined PDO Assignment: 'Current control with Info Data'
Predefined PDO Assignment: 'PWM'
Predefined PDO Assignment: 'PWM with info data'
Predefined PDO Assignment: 'PWM with current'
Predefined PDO Assignment: 'PWM with current and info data'
Predefined PDO Assignment: 'Voltage control'
Predefined PDO Assignment: 'Voltage control with info data'

Fig. 153: PDO setting “DC digital output (with info data)”

6. On the “DC” tab, select the operation mode “DC active (Controller handled)”

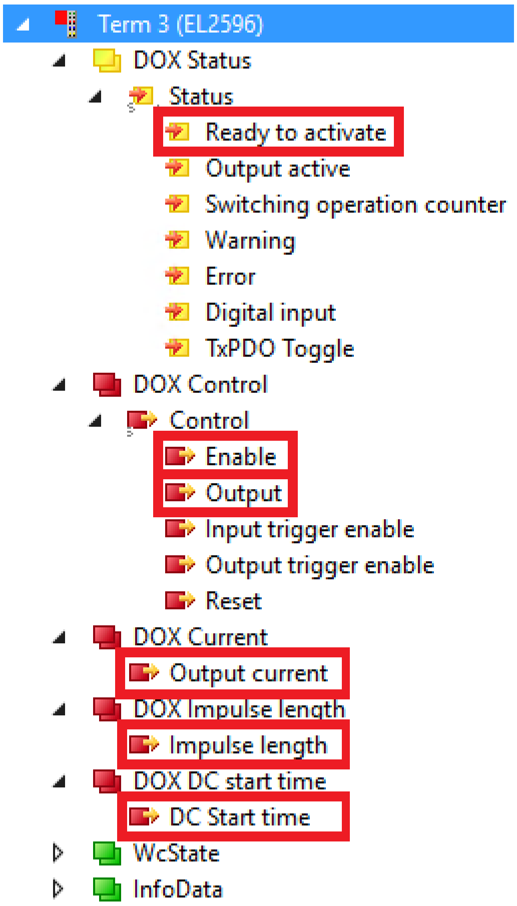


Operation Mode: DC active (Controller handled)

FreeRun/SM-Synchron
DC active (Controller handled)

Fig. 154: Activating the Distributed Clocks

7. Specify the set current in the unit mA via “DOX Current” → “Output Current”
8. Specify the pulse length in the unit µs via “DOX Impulse length” → “Impulse length”. The resolution of the time can be reduced from 1 µs to 100 ns using the CoE parameter 0x8002:31 [▶ 218] “Pulse resolution 100 ns”.
9. Specify the start time of the pulse via “DOX DC Start time” → “DC Start time”.
10. Check under “DOX Status” → “Status” whether the “Ready to activate” bit is 1.
11. Activate the control under “DOX Control” → “Control” via the “Enable” bit.
12. Switch on the LED output under “DOX Control” → “Control” by activating the “Output” bit.



- Term 3 (EL2596)
  - DOX Status
    - Status
      - Ready to activate
      - Output active
      - Switching operation counter
      - Warning
      - Error
      - Digital input
      - TxPDO Toggle
  - DOX Control
    - Control
      - Enable
      - Output
      - Input trigger enable
      - Output trigger enable
      - Reset
  - DOX Current
    - Output current
  - DOX Impulse length
    - Impulse length
  - DOX DC start time
    - DC Start time
  - WcState
  - InfoData

Fig. 155: Activate the output in the operation mode “Current control timestamp pulse”

5.3.3 Current control trigger pulse

In pulse mode, the maximum current amplitude is 3 A with 12 V LEDs.

The specific parameters for setting the LED output as a trigger-controlled pulse output are described below. In order to generate a reproducible behavior at the LED output via a signal at the trigger input, the signal at the trigger input should have a steep edge.

For all pulsed operating modes (current control timestamp pulse, current control trigger pulse, current control PLC pulse) the maximum constant duty cycle as a function of the output current must be considered.

- Iout = 1 A: 20% duty cycle

- Iout = 2 A: 10% Duty Cycle

- Iout = 3 A: 8% Duty Cycle

In this operation mode, it is possible to configure the trigger input as an enable input so that it can be used as an external switch. An output can then only be actuated with a specified signal at the trigger input. The LED output then switches synchronously with the signal at the trigger input. More detailed information and commissioning can be found in the chapter Hardware enable [▶ 178]. Alternatively, the trigger input is used as a trigger input, so that a pulse with a specified delay, duration and current is output at the LED output after a signal at the trigger input.

Especially in the current controlled pulse operation modes the use of the automatic determination of the output voltage [▶ 168] is recommended, because possibly for the generation of high current pulses at the output a high pre-voltage is needed to generate steep edges. These may be higher than voltages specified in the LED data sheet.

If the supply voltage is removed during operation in the pulsing modes, there will be a malfunction at the output. In addition, an overcurrent may occur at the LED output. Therefore, a stable voltage supply must be ensured. To minimize the risk of malfunction and overcurrent, the enable bit can be set together with the output bit so that the enable is not permanently set.

For delay-free triggering, it is recommended to trigger by PLC or internally via distributed clocks.

When using an external trigger, it should be noted that, based on internal terminal delays, the minimum delay between the trigger input and the signal output is approximately 2 µs. The offset of 2 µs must also be observed for all set delay values (regardless of whether they are set via PDO or CoE). The resulting delay is always

$$Real\ Delay = Set\ \ Delay + Offset$$

In the hardware versions 01 and 02 and firmware versions <FW04, oscillations of the voltage can occur at the trigger input due to the line inductance of the connected line. This can lead to incorrect detection of pulses, which causes pulses to be output at the LED output. This problem can be minimized by a resistance of approx. 1 kΩ between the trigger signal and pin 5 (TrigIn+).

From FW04, the CoE parameter 0x8004:09 [▶ 219] “Trigger input blind time” has an adjustable waiting time for signal detection at the trigger input. The inactive time of the trigger input from the trigger start time to the end of the pulse output plus "Trigger input blind time" is adjustable in the range of 20..65535 µs. This allows interference to be intercepted.

From HW03, asynchronous interference is also filtered by the hardware.

The function "trigger input blind time" can not only be used to eliminate disturbances, but also as a kind of "dead time" to preset a fixed duty cycle, so that there is no reaction at the output even if events occur at the trigger input to protect the terminal and the LED from overheating.

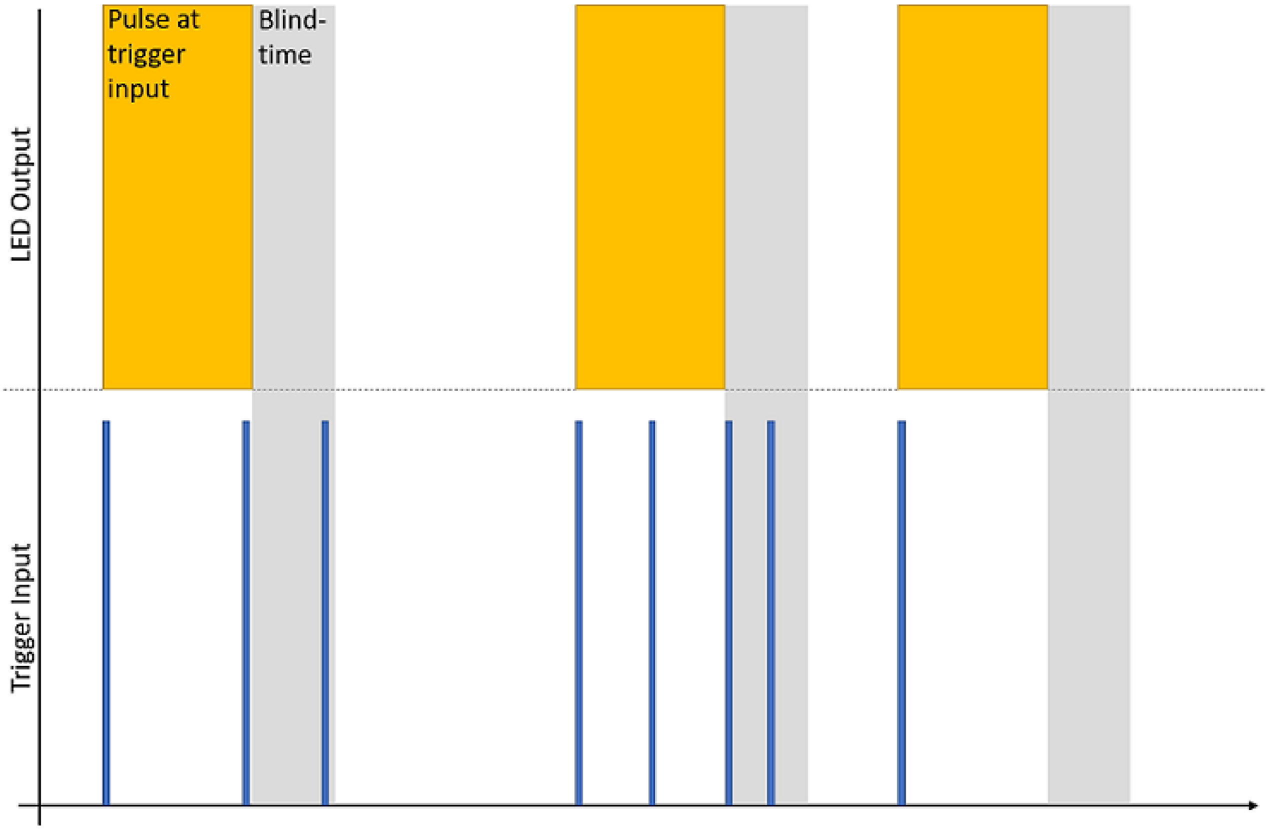
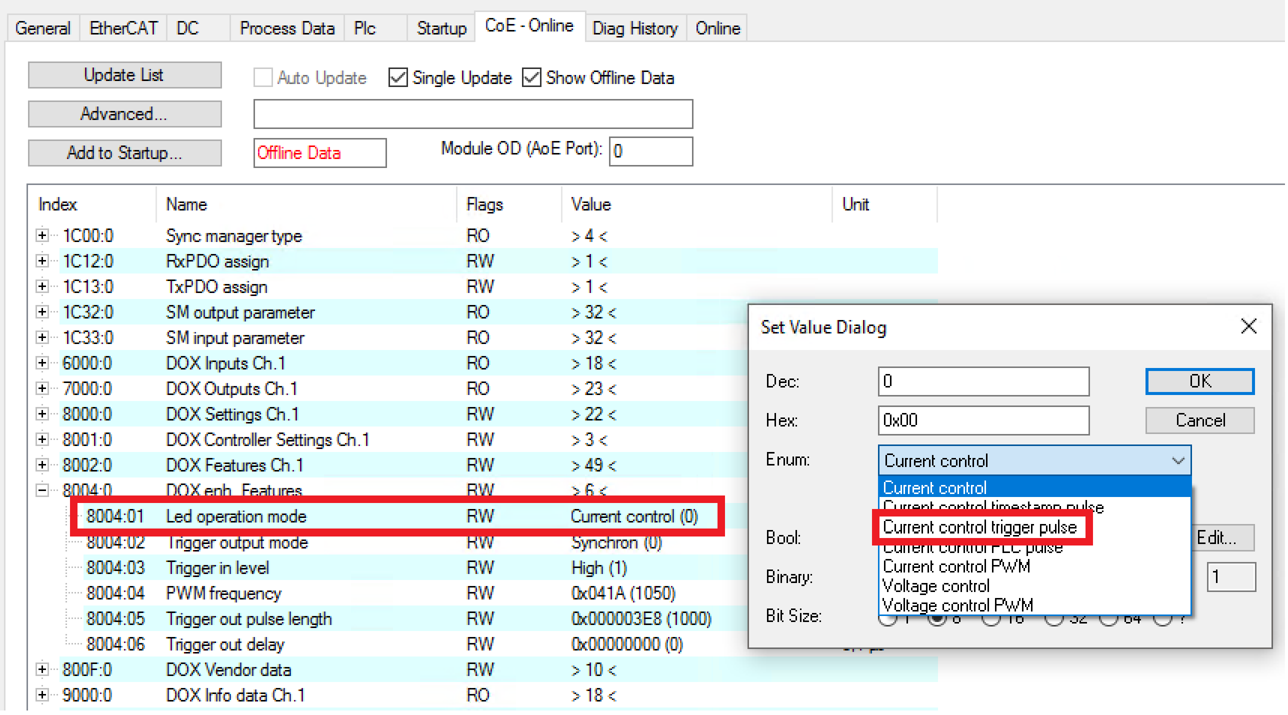


Fig. 156: Behavior with "trigger input blind time"

Example: The configuration specifies a pulse length of 500 µs, a pulse delay in relation to the trigger input of 0 µs and a blind time of 150 µs. The first pulse at the trigger input is detected, as shown in the figure, and a pulse with the specified values is triggered at the LED output. During this time possible further edges at the trigger input are "ignored". After the 500 µs pulse at the LED output, the blind time begins, which is specified with 150 µs as an example. During these 150 µs, edges at the trigger input continue to be ignored. This means that no new pulses can be generated for a total of 650 µs. Only 650 µs after the first edge that triggered the pulse at the output, the trigger input reacts again to new edges (in the example the 4th edge at the trigger input). Then the same starts from the beginning. During the entire blind time, the edges are not stored.

Make the following settings:

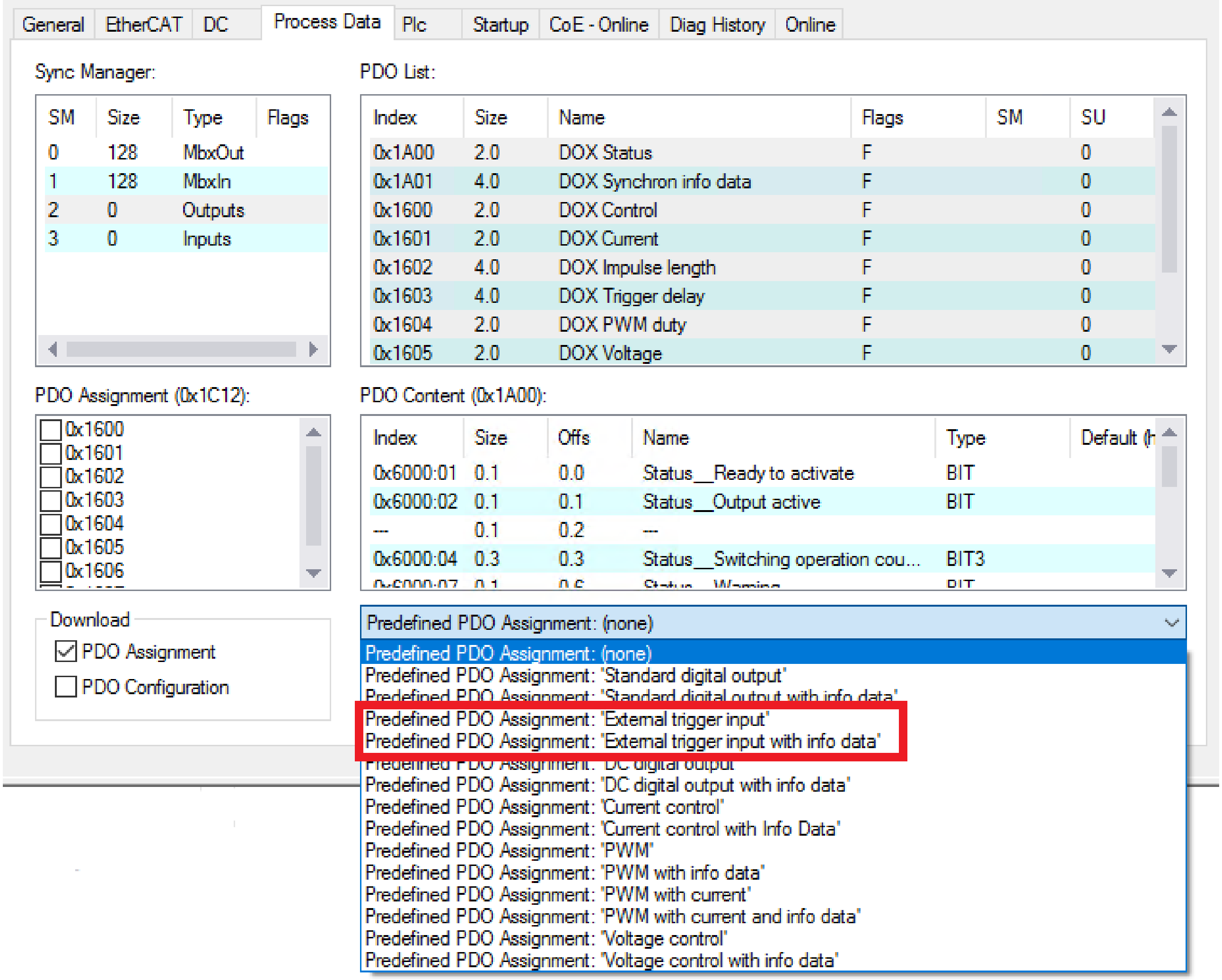
1. Nominal/limit current of the LED in the unit mA in the CoE parameter 0x8000:02 [▸ 217] “Target current”
2. Input voltage in the unit 0.01 V in the parameter 0x8000:03 [▸ 217] “Supply voltage”
3. Desired output voltage in the unit 0.01 V (max. U_{IN} - 2 V) in the parameter 0x8000:04 [▸ 217] “Output voltage”.
   - If currents are to be output above the nominal current of the LED, a higher specification is necessary for the output voltage in the CoE parameter 0x8000:04 [▸ 217] due to the non-linear behavior of LEDs and the current control capability of the circuit.
   - In this operation mode, the value set here corresponds to the output on the controller. If the forward voltage of the connected LED is lower than the value set here, the remaining voltage drops internally in the EL2596. In many cases, this leads to overtemperature in the terminal and thus to the switching off of the LED output. As the desired output voltage, therefore, the forward voltage at the desired output current should be set here. The TeachIn function [▸ 168] can also be used to determine the output voltage.
4. Set the operation mode in the CoE directory in the parameter 0x8004:01 [▸ 219] to “Current Control trigger pulse”



Index	Name	Flags	Value	Unit
1C00:0	Sync manager type	RO	> 4 <	
1C12:0	RxPDO assign	RW	> 1 <	
1C13:0	TxPDO assign	RW	> 1 <	
1C32:0	SM output parameter	RO	> 32 <	
1C33:0	SM input parameter	RO	> 32 <	
6000:0	DOX Inputs Ch.1	RO	> 18 <	
7000:0	DOX Outputs Ch.1	RO	> 23 <	
8000:0	DOX Settings Ch.1	RW	> 22 <	
8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
8002:0	DOX Features Ch.1	RW	> 49 <	
8004:0	DOX enh. Features	RW	> 6 <	
8004:01	Led operation mode	RW	Current control (0)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	
8004:06	Trigger out delay	RW	0x00000000 (0)	
800F:0	DOX Vendor data	RW	> 10 <	
9000:0	DOX Info data Ch.1	RO	> 18 <	

Fig. 157: Operation mode setting “Current control trigger pulse”

5. Set “Predefined PDO Assignments” to “External trigger input (with info data)”.



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	0	Outputs	
3	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F		0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F		0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0

PDO Assignment (0x1C12):

- □ 0x1600
- □ 0x1601
- □ 0x1602
- □ 0x1603
- □ 0x1604
- □ 0x1605
- □ 0x1606

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	

Download

☑ PDO Assignment

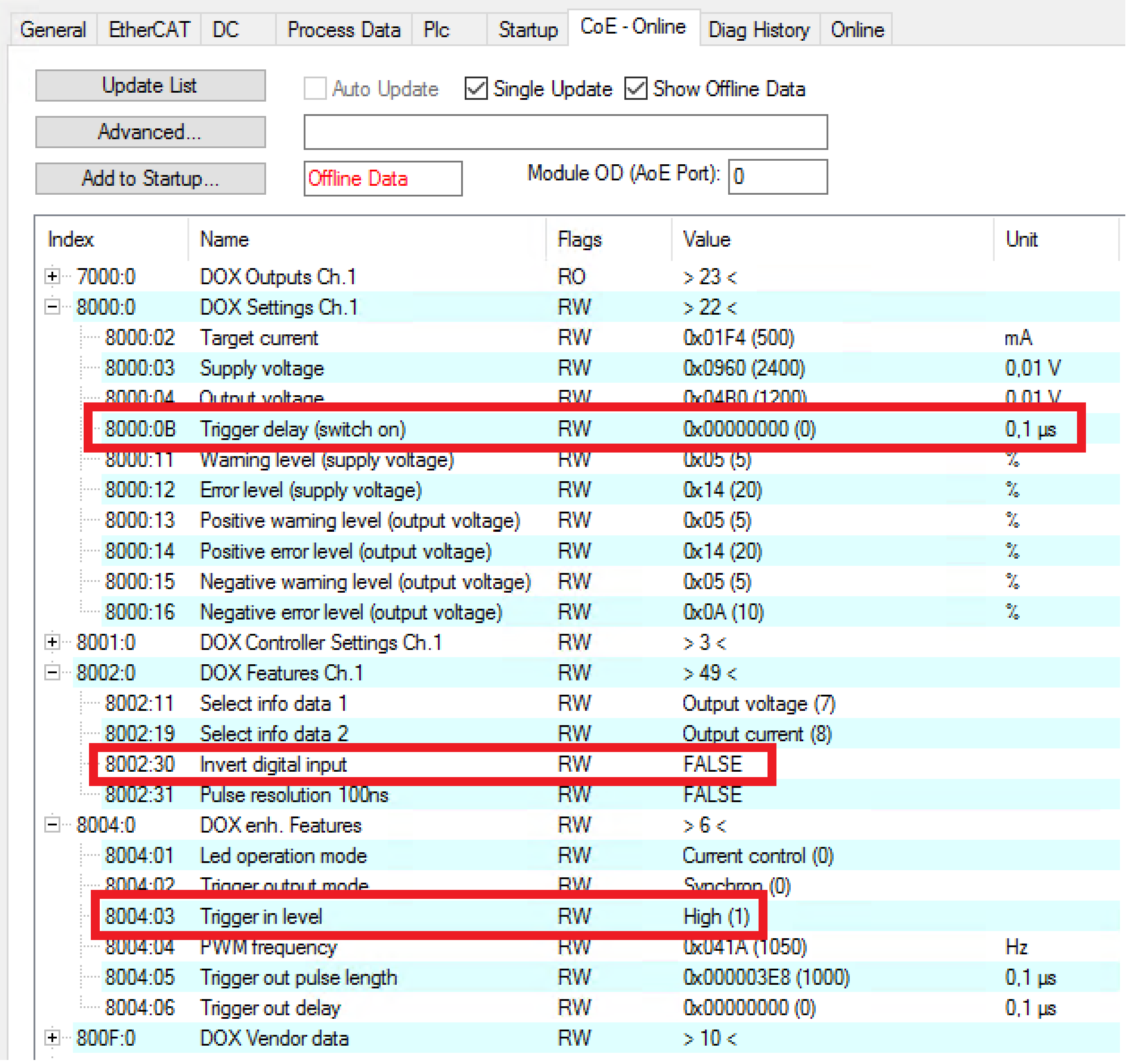
□ PDO Configuration

Predefined PDO Assignment: (none)

- Predefined PDO Assignment: (none)
- Predefined PDO Assignment: 'Standard digital output'
- Predefined PDO Assignment: 'Standard digital output with info data'
- Predefined PDO Assignment: 'External trigger input'
- Predefined PDO Assignment: 'External trigger input with info data'
- Predefined PDO Assignment: 'DC digital output'
- Predefined PDO Assignment: 'DC digital output with info data'
- Predefined PDO Assignment: 'Current control'
- Predefined PDO Assignment: 'Current control with Info Data'
- Predefined PDO Assignment: 'PWM'
- Predefined PDO Assignment: 'PWM with info data'
- Predefined PDO Assignment: 'PWM with current'
- Predefined PDO Assignment: 'PWM with current and info data'
- Predefined PDO Assignment: 'Voltage control'
- Predefined PDO Assignment: 'Voltage control with info data'

Fig. 158: PDO setting “External trigger pulse (with info data)”

6. Specify the set current in the unit mA via “DOX Current” → “Output Current”.
7. Specify the pulse length in the unit µs via “DOX Impulse length” → “Impulse length”. The resolution of the time can be reduced from 1 µs to 100 ns using the CoE parameter 0x8002:31 “Pulse resolution 100 ns”.
8. Specify the trigger signal delay in the unit µs via “DOX Trigger delay” → “Trigger delay”. The resolution of the time can be reduced from 1 µs to 100 ns using the CoE parameter 0x8002:31 [▶ 218] “Pulse resolution 100 ns”. Alternatively, the delay can be firmly specified in the unit 0.1 µs in the CoE parameter 0x8000:0B [▶ 217]. If the delay is specified both in the PDOs and in the CoE parameters, these values are added together and the delay is the sum of the two.
9. In the CoE parameter 0x8002:30 [▶ 218], choose whether the LED pulse should be generated by a rising or falling edge on the trigger input.
10. In the CoE parameter 0x8004:03 [▶ 219], specify the level at the trigger input at which the LED output should be switched (high = 24 V, low = 5 V).

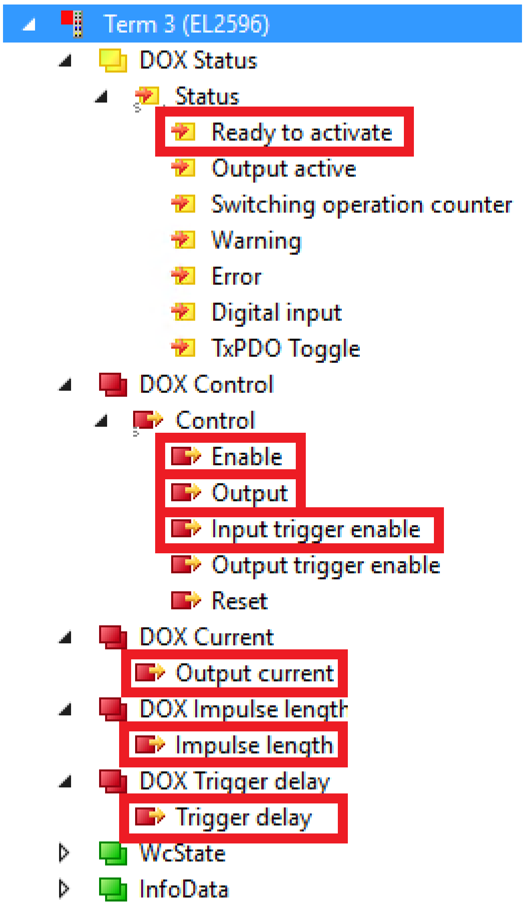


Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
+ 7000:0	DOX Outputs Ch.1	RO	> 23 <	
− 8000:0	DOX Settings Ch.1	RW	> 22 <	
8000:02	Target current	RW	0x01F4 (500)	mA
8000:03	Supply voltage	RW	0x0960 (2400)	0,01 V
8000:04	Output voltage	RW	0x04B0 (1200)	0,01 V
8000:0B	Trigger delay (switch on)	RW	0x00000000 (0)	0,1 µs
8000:11	Warning level (supply voltage)	RW	0x05 (5)	%
8000:12	Error level (supply voltage)	RW	0x14 (20)	%
8000:13	Positive warning level (output voltage)	RW	0x05 (5)	%
8000:14	Positive error level (output voltage)	RW	0x14 (20)	%
8000:15	Negative warning level (output voltage)	RW	0x05 (5)	%
8000:16	Negative error level (output voltage)	RW	0x0A (10)	%
+ 8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
− 8002:0	DOX Features Ch.1	RW	> 49 <	
8002:11	Select info data 1	RW	Output voltage (7)	
8002:19	Select info data 2	RW	Output current (8)	
8002:30	Invert digital input	RW	FALSE	
8002:31	Pulse resolution 100ns	RW	FALSE	
− 8004:0	DOX enh. Features	RW	> 6 <	
8004:01	Led operation mode	RW	Current control (0)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	Hz
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	0,1 µs
8004:06	Trigger out delay	RW	0x00000000 (0)	0,1 µs
+ 800F:0	DOX Vendor data	RW	> 10 <	

Fig. 159: Settings in the CoE objects for the operation mode “Current control trigger pulse”

11. Connect the trigger signal to the “TrigIn+” (5) and “TrigIn-“ (13) connections.
12. Activate the trigger input under “DOX Control” → “Control” → “Input Trigger Enable”
13. Check under “DOX Status” → “Status” whether the “Ready to activate” bit is 1.
14. Activate the control under “DOX Control” → “Control” via the “Enable” bit.
15. Switch on the LED output under “DOX Control” → “Control” by activating the “Output” bit.



- Term 3 (EL2596)
  - DOX Status
    - Status
      - Ready to activate
      - Output active
      - Switching operation counter
      - Warning
      - Error
      - Digital input
      - TxPDO Toggle
  - DOX Control
    - Control
      - Enable
      - Output
      - Input trigger enable
      - Output trigger enable
      - Reset
  - DOX Current
    - Output current
  - DOX Impulse length
    - Impulse length
  - DOX Trigger delay
    - Trigger delay
  - WcState
  - InfoData

Fig. 160: Activating the output in the operation mode “Current control trigger pulse”

5.3.4 Current control PLC pulse

In pulse mode, the maximum current amplitude is 3 A with 12 V LEDs. The pulses at the LED output can be triggered by a rising edge on the output bit.

For all pulsed operating modes (current control timestamp pulse, current control trigger pulse, current control PLC pulse) the maximum constant duty cycle as a function of the output current must be considered.

- Iout = 1 A: 20% duty cycle

- Iout = 2 A: 10% Duty Cycle

- Iout = 3 A: 8% Duty Cycle

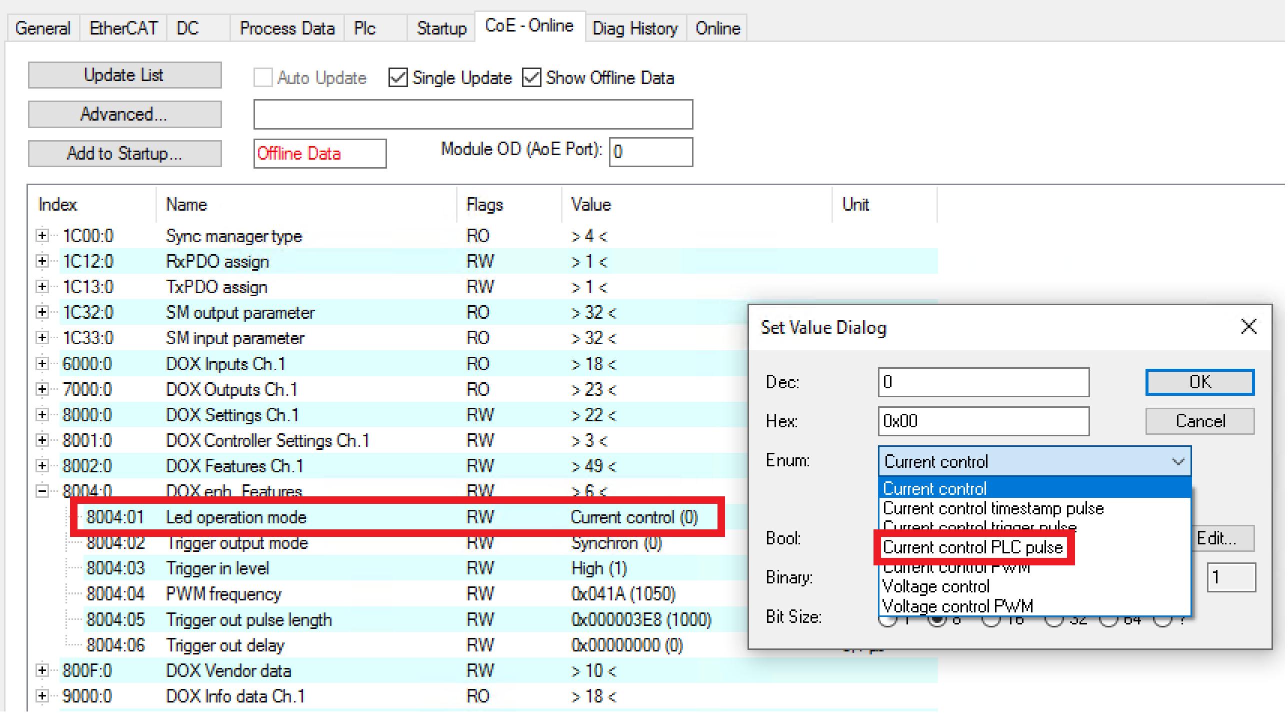
In this operation mode, it is possible to configure the trigger input as an enable input so that it can be used as an external switch. An output can then only be actuated with a specified signal at the trigger input. If the trigger input is active, a pulse with the specified parameters is output from the PLC on a rising edge on the output bit. More detailed information and commissioning can be found in the chapter Hardware enable [▸ 178].

Especially in the current controlled pulse operation modes the use of the automatic determination of the output voltage [▸ 168] is recommended, because possibly for the generation of high current pulses at the output a high pre-voltage is needed to generate steep edges. These may be higher than voltages specified in the LED data sheet.

If the supply voltage is removed during operation in the pulsing modes, there will be a malfunction at the output. In addition, an overcurrent may occur at the LED output. Therefore, a stable voltage supply must be ensured. To minimize the risk of malfunction and overcurrent, the enable bit can be set together with the output bit so that the enable is not permanently set.

The specific parameters for setting the LED output as a PLC-controlled pulse output are described below.

1. Output current in the unit mA in the CoE parameter 0x8000:02 [▶ 217] “Target current”
2. Input voltage in the unit 0.01 V in the CoE parameter 0x8000:03 [▶ 217] “Supply voltage”
3. Desired output voltage in the unit 0.01 V (max. U_{IN} - 2 V) in the parameter 0x8000:04 [▶ 217] “Output voltage”.
    - If currents are to be output above the nominal current of the LED, a higher specification is necessary for the output voltage in the CoE parameter 0x8000:04 [▶ 217] due to the non-linear behavior of LEDs and the current control capability of the circuit.
    - In this operation mode, the value set here corresponds to the output on the controller. If the forward voltage of the connected LED is lower than the value set here, the remaining voltage drops internally in the EL2596. In many cases, this leads to overtemperature in the terminal and thus to the switching off of the LED output. As the desired output voltage, therefore, the forward voltage at the desired output current should be set here. The TeachIn function [▶ 168] can also be used to determine the output voltage.
4. Set the operation mode in the CoE directory in the parameter 0x8004:01 [▶ 219] to “Current Control PLC pulse”



□ Auto Update ☑ Single Update ☑ Show Offline Data

Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
1C00:0	Sync manager type	RO	> 4 <	
1C12:0	RxPDO assign	RW	> 1 <	
1C13:0	TxPDO assign	RW	> 1 <	
1C32:0	SM output parameter	RO	> 32 <	
1C33:0	SM input parameter	RO	> 32 <	
6000:0	DOX Inputs Ch.1	RO	> 18 <	
7000:0	DOX Outputs Ch.1	RO	> 23 <	
8000:0	DOX Settings Ch.1	RW	> 22 <	
8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
8002:0	DOX Features Ch.1	RW	> 49 <	
8004:0	DOX enh. Features	RW	> 6 <	
8004:01	Led operation mode	RW	Current control (0)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	
8004:06	Trigger out delay	RW	0x00000000 (0)	
800F:0	DOX Vendor data	RW	> 10 <	
9000:0	DOX Info data Ch.1	RO	> 18 <	

Set Value Dialog

Dec: 0

Hex: 0x00

Enum: Current control

- Current control
- Current control timestamp pulse
- Current control trigger pulse
- Current control PLC pulse
- Current control PWM
- Voltage control
- Voltage control PWM

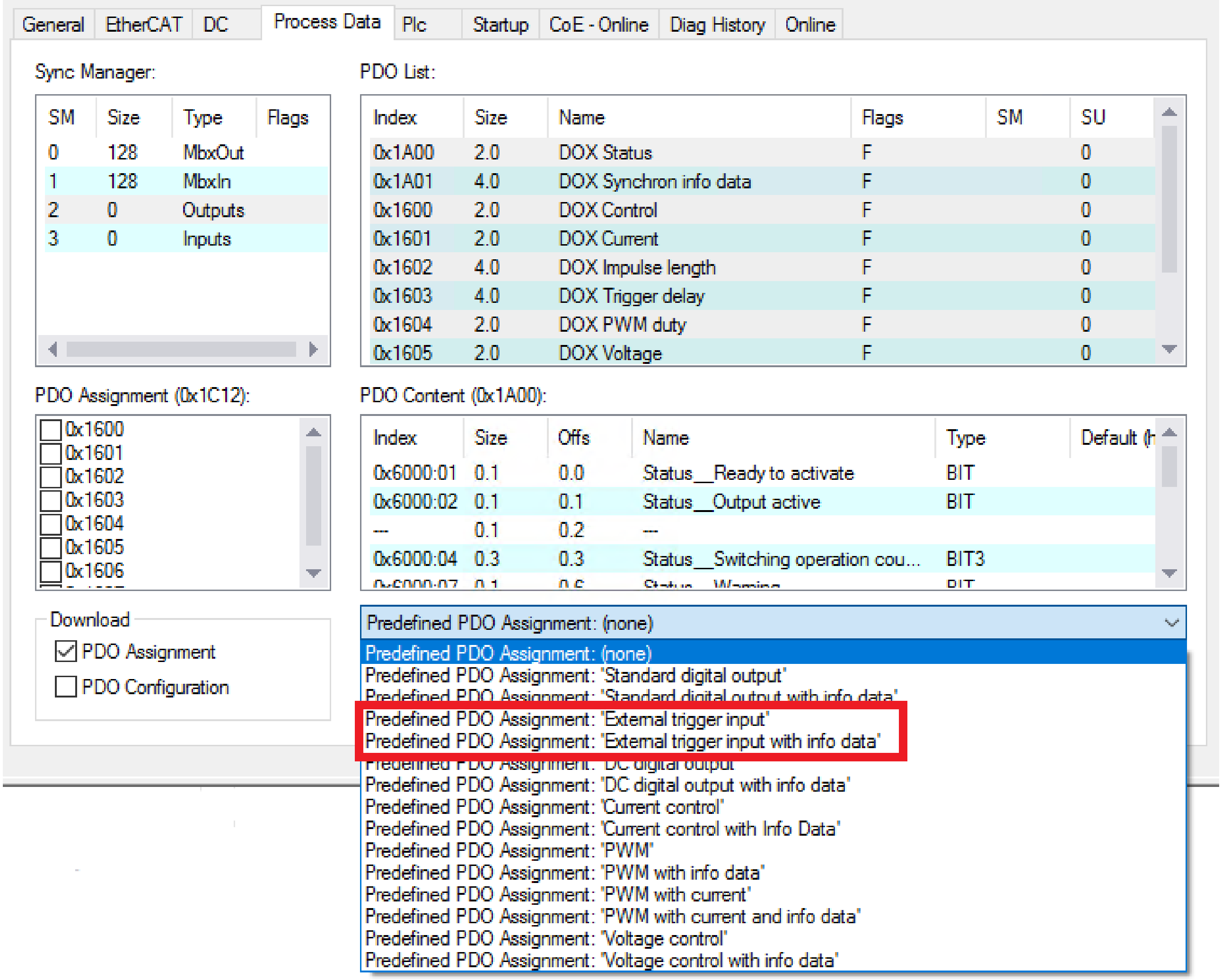
Bool:

Binary: 1

Bit Size: ○ 1 ● 8 ○ 16 ○ 32 ○ 64 ○ ?

Fig. 161: Operation mode setting “Current control PLC pulse”

5. Set “Predefined PDO Assignments” to “External trigger input (with info data)”.



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	0	Outputs	
3	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F		0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F		0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0

PDO Assignment (0x1C12):

□ 0x1600
□ 0x1601
□ 0x1602
□ 0x1603
□ 0x1604
□ 0x1605
□ 0x1606

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	

Download

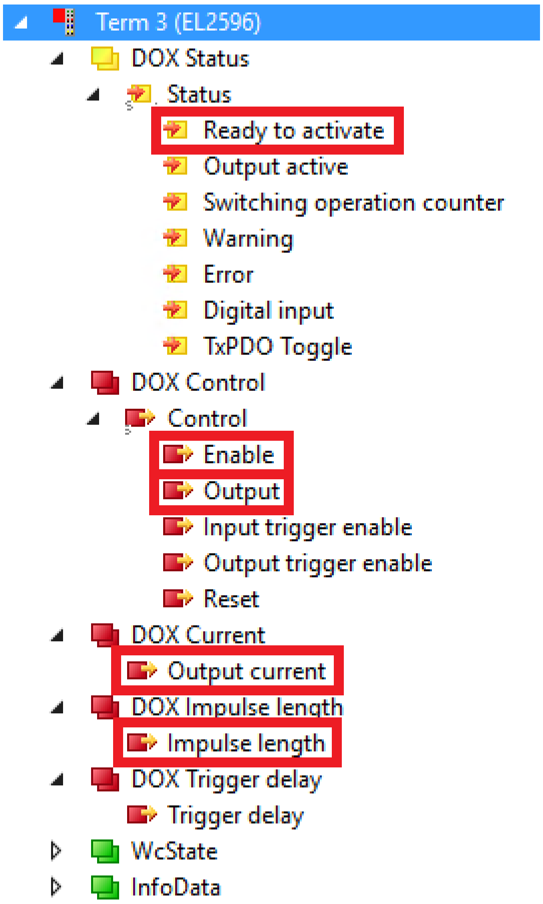
☑ PDO Assignment
□ PDO Configuration

Predefined PDO Assignment: (none)

Predefined PDO Assignment: (none)
Predefined PDO Assignment: 'Standard digital output'
Predefined PDO Assignment: 'Standard digital output with info data'
Predefined PDO Assignment: 'External trigger input'
Predefined PDO Assignment: 'External trigger input with info data'
Predefined PDO Assignment: 'DC digital output'
Predefined PDO Assignment: 'DC digital output with info data'
Predefined PDO Assignment: 'Current control'
Predefined PDO Assignment: 'Current control with Info Data'
Predefined PDO Assignment: 'PWM'
Predefined PDO Assignment: 'PWM with info data'
Predefined PDO Assignment: 'PWM with current'
Predefined PDO Assignment: 'PWM with current and info data'
Predefined PDO Assignment: 'Voltage control'
Predefined PDO Assignment: 'Voltage control with info data'

Fig. 162: PDO setting “External trigger pulse (with info data)”

6. Specify the set current in the unit mA via “DOX Current” → “Output Current”.
7. Specify the pulse length in the unit µs via “DOX Impulse length” → “Impulse length”. The resolution of the time can be reduced from 1 µs to 100 ns using the CoE parameter 0x8002:31 [▶ 218] “Pulse resolution 100 ns”.
8. Check under “DOX Status” → “Status” whether the “Ready to activate” bit is 1.
9. Activate the control under “DOX Control” → “Control” via the “Enable” bit.
10. The current-controlled LED output can then be enabled via the “Output” bit under “DOX Control” → “Control”. With each rising edge on this bit, a pulse is output at the LED output with the specified parameters for current, pulse duration and delay.



- Term 3 (EL2596)
  - DOX Status
    - Status
      - Ready to activate
      - Output active
      - Switching operation counter
      - Warning
      - Error
      - Digital input
      - TxPDO Toggle
  - DOX Control
    - Control
      - Enable
      - Output
      - Input trigger enable
      - Output trigger enable
      - Reset
  - DOX Current
    - Output current
  - DOX Impulse length
    - Impulse length
  - DOX Trigger delay
    - Trigger delay
  - WcState
  - InfoData

Fig. 163: Activating the output in the operation mode “Current control PLC pulse”

5.3.5 Current control PWM

This mode allows the control of an LED by pulse width modulation. The amplitude of the output current is controlled by an analog circuit. This leads to additional self-heating. In order to prevent damage to the module, the output power is automatically limited in this mode by reducing the maximum accepted value for the duty cycle. The maximum accepted value can be calculated by

$$Duty\ Cycle = \frac{32767 \cdot 0{,}6\ W}{0{,}5\ V \cdot Target\ current}$$

Higher entries will be ignored.

Due to the high self-heating in PWM operation, the theoretically maximum calculated value for the duty cycle cannot always be reached. The following figure shows the correlation between the maximum possible duty cycle in combination with the output voltage for 5 different output current values at an ambient temperature of 55 °C. At higher duty cycle settings, the EL2596 would have switched off temporarily due to overtemperature. The following figure gives guide values and does not necessarily apply to every application.

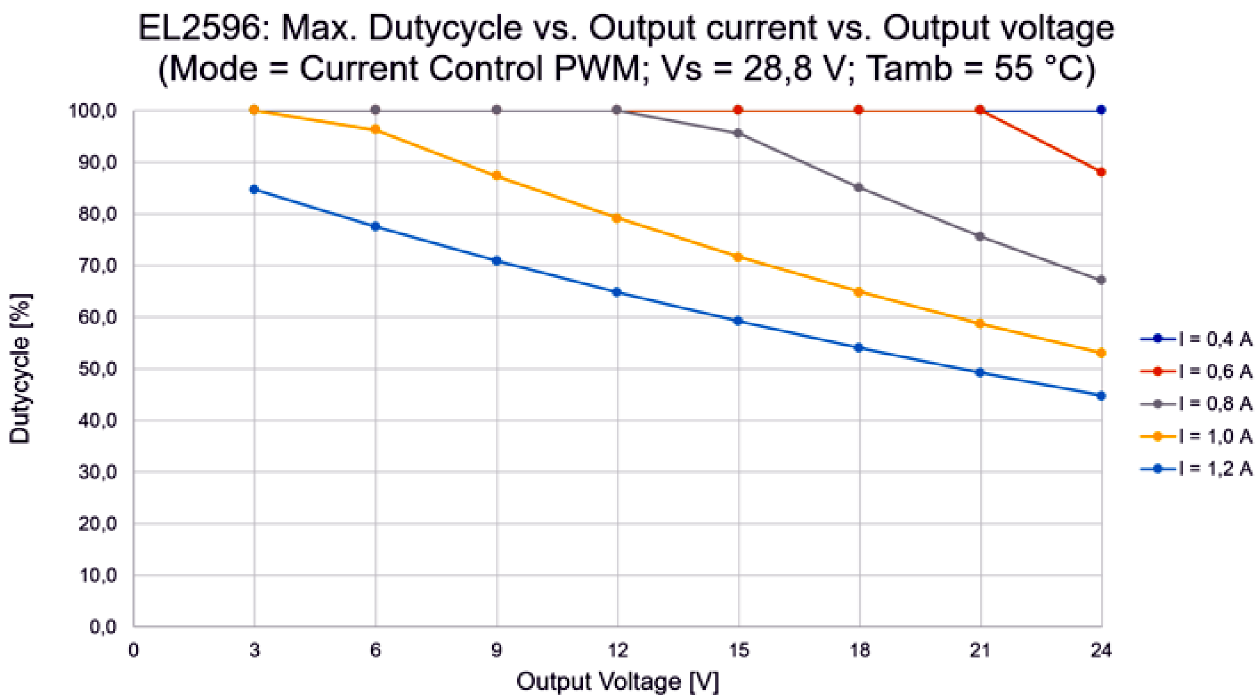


Fig. 164: EL2596 | maximum duty cycle in current control PWM mode

For the EL2596-0010 for operation without fan in Current control PWM mode the following limitation for the maximum duty cycle must be considered:

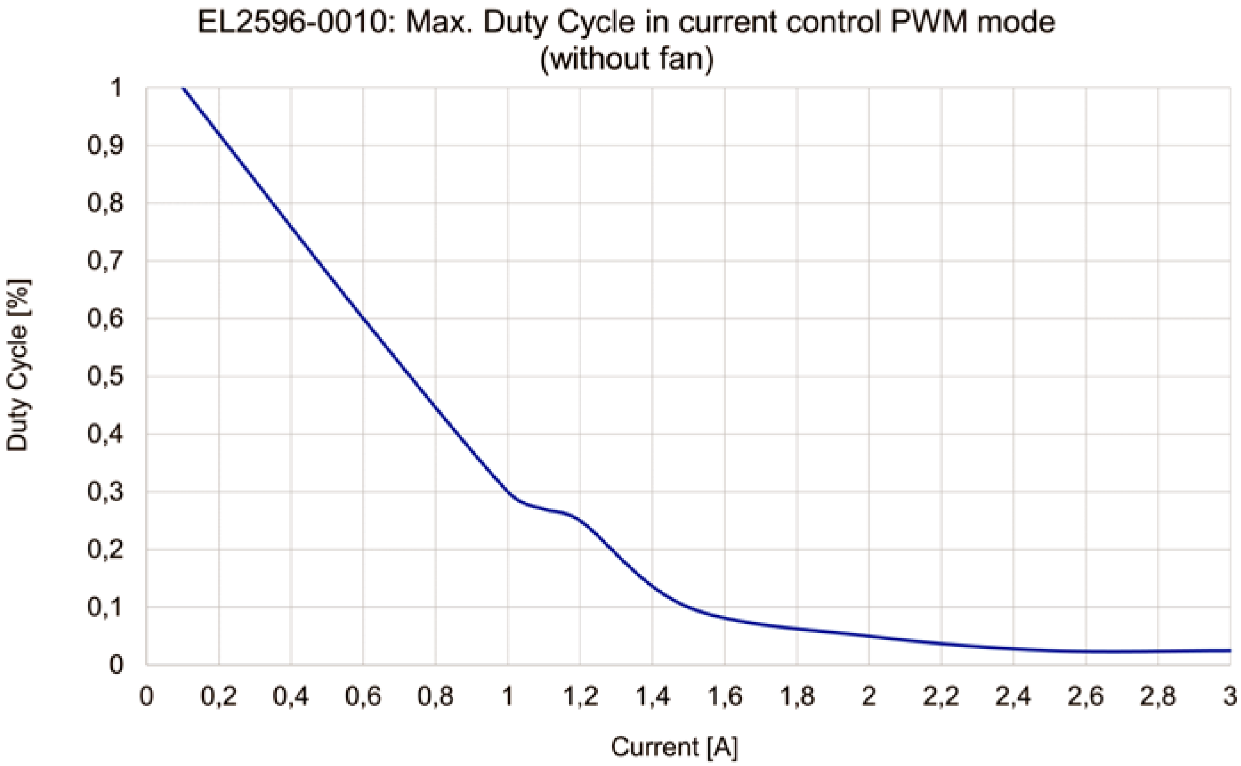


Fig. 165: EL2596-0010 | maximum duty cycle in current control PWM mode

For true-color dimming, the set limit current in the target current and the desired output current should correspond to the nominal current of the LED.

In this operation mode, it is possible to configure the trigger input as an enable input so that it can be used as an external switch. An output can then only be actuated with a specified signal at the trigger input. More detailed information and commissioning can be found in the chapter Hardware enable [▶ 178].

The specific parameters for controlling the LED output via PWM are described below.

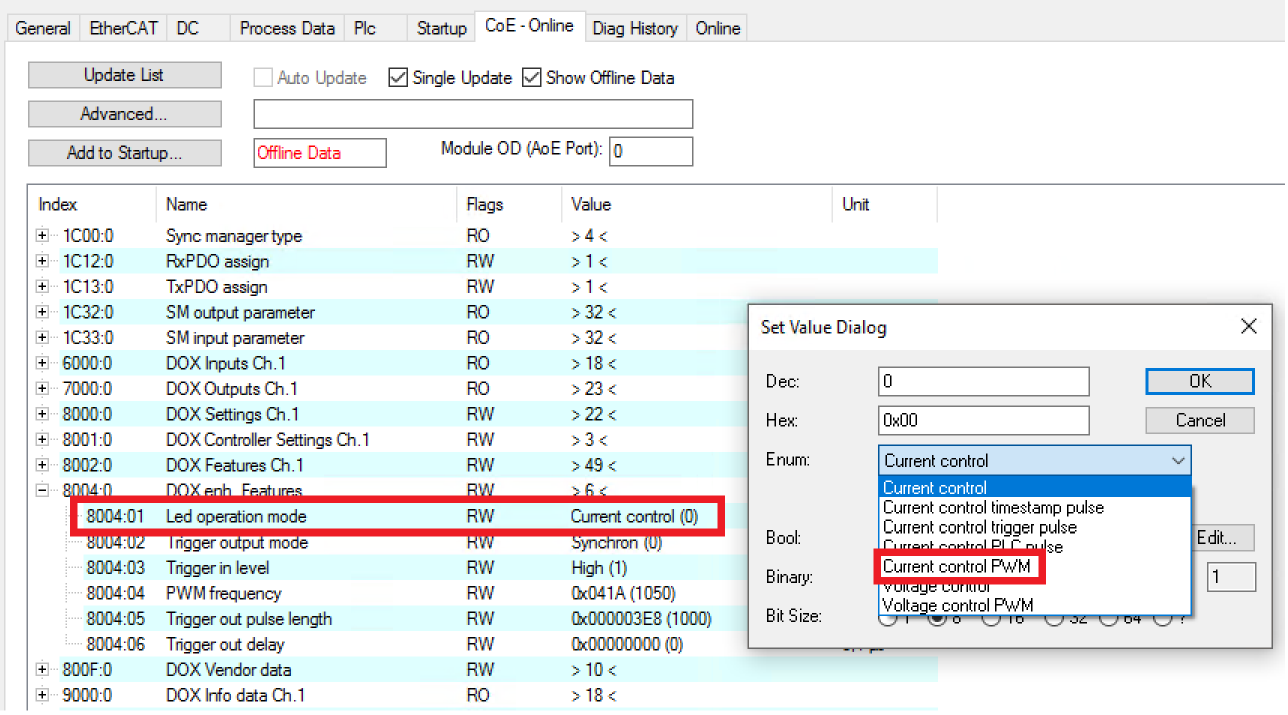
1. Nominal/limit current of the LED in the unit mA in index 0x8000:02 [▸ 217] “Target current”
2. Input voltage in the unit 0.01 V in the index 0x8000:03 [▸ 217] “Supply voltage”
3. Desired output voltage in the unit 0.01 V (max. U_{IN} - 2 V) in the index 0x8000:04 [▸ 217] “Output voltage”. In this operation mode, the value set here corresponds to the output on the controller. If the forward voltage of the connected LED is lower than the value set here, the remaining voltage drops internally in the EL2596. In many cases, this leads to overtemperature in the terminal and thus to the switching off of the LED output. As the desired output voltage, therefore, the forward voltage at the desired output current should be set here. The TeachIn function [▸ 168] can also be used to determine the output voltage.
4. The default PWM frequency is 1050 Hz. If necessary, the value can be adjusted in index 0x8004:04 [▸ 219].

⚠ **WARNING**

Stroboscopic effects in PWM mode

Due to the high-frequency light switching, there is a risk of a stroboscopic effect. If the frequency of the emitted light is in phase with the movement frequency of a rotating machine part, for example, a stroboscopic effect can make it appear as if the machine is stationary despite it moving. This can lead to a misinterpretation by an operator who may intervene due to the apparently stationary machine part. This can lead to serious injury or death.

5. Set the operation mode in the CoE directory in the index 0x8004:01 [▸ 219] to “Current Control PWM”



☐ Auto Update ☑ Single Update ☑ Show Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
1C00:0	Sync manager type	RO	> 4 <	
1C12:0	RxPDO assign	RW	> 1 <	
1C13:0	TxPDO assign	RW	> 1 <	
1C32:0	SM output parameter	RO	> 32 <	
1C33:0	SM input parameter	RO	> 32 <	
6000:0	DOX Inputs Ch.1	RO	> 18 <	
7000:0	DOX Outputs Ch.1	RO	> 23 <	
8000:0	DOX Settings Ch.1	RW	> 22 <	
8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
8002:0	DOX Features Ch.1	RW	> 49 <	
8004:0	DOX enh. Features	RW	> 6 <	
8004:01	Led operation mode	RW	Current control (0)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	
8004:06	Trigger out delay	RW	0x00000000 (0)	
800F:0	DOX Vendor data	RW	> 10 <	
9000:0	DOX Info data Ch.1	RO	> 18 <	

Dec: 0

Hex: 0x00

Enum: Current control

- Current control
- Current control timestamp pulse
- Current control trigger pulse
- Current control PLC pulse
- Current control PWM
- Voltage control
- Voltage control PWM

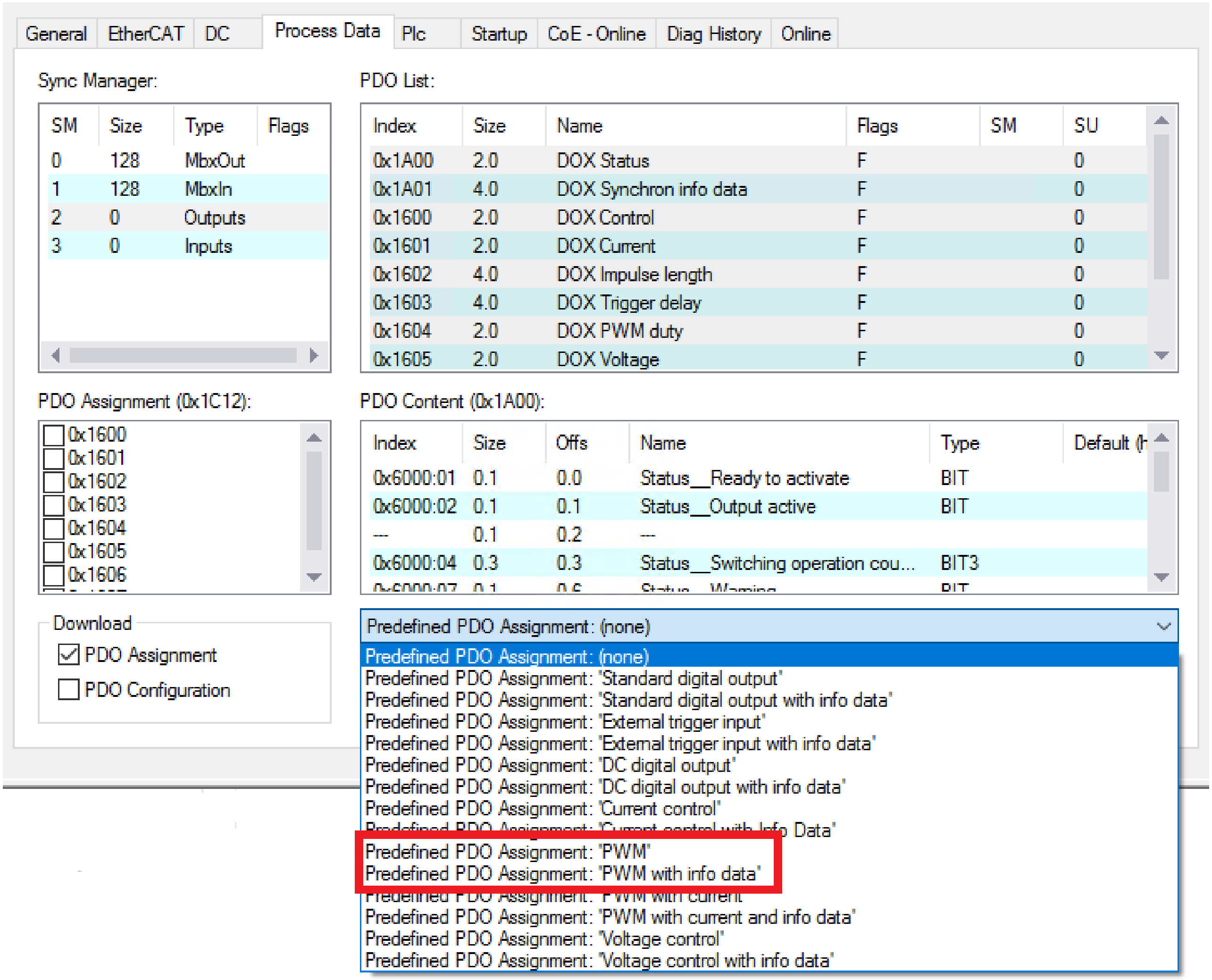
Bool:

Binary: 1

Bit Size: ○ 1 ● 8 ○ 16 ○ 32 ○ 64 ○ ?

Fig. 166: Operation mode setting “Current control PWM”

6. Set the predefined PDO assignments to “PWM (with info data)” or “PWM with current (and info data)”



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	0	Outputs	
3	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F		0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F		0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0

PDO Assignment (0x1C12):

□ 0x1600
□ 0x1601
□ 0x1602
□ 0x1603
□ 0x1604
□ 0x1605
□ 0x1606
□ 0x1607

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	

Download

☑ PDO Assignment

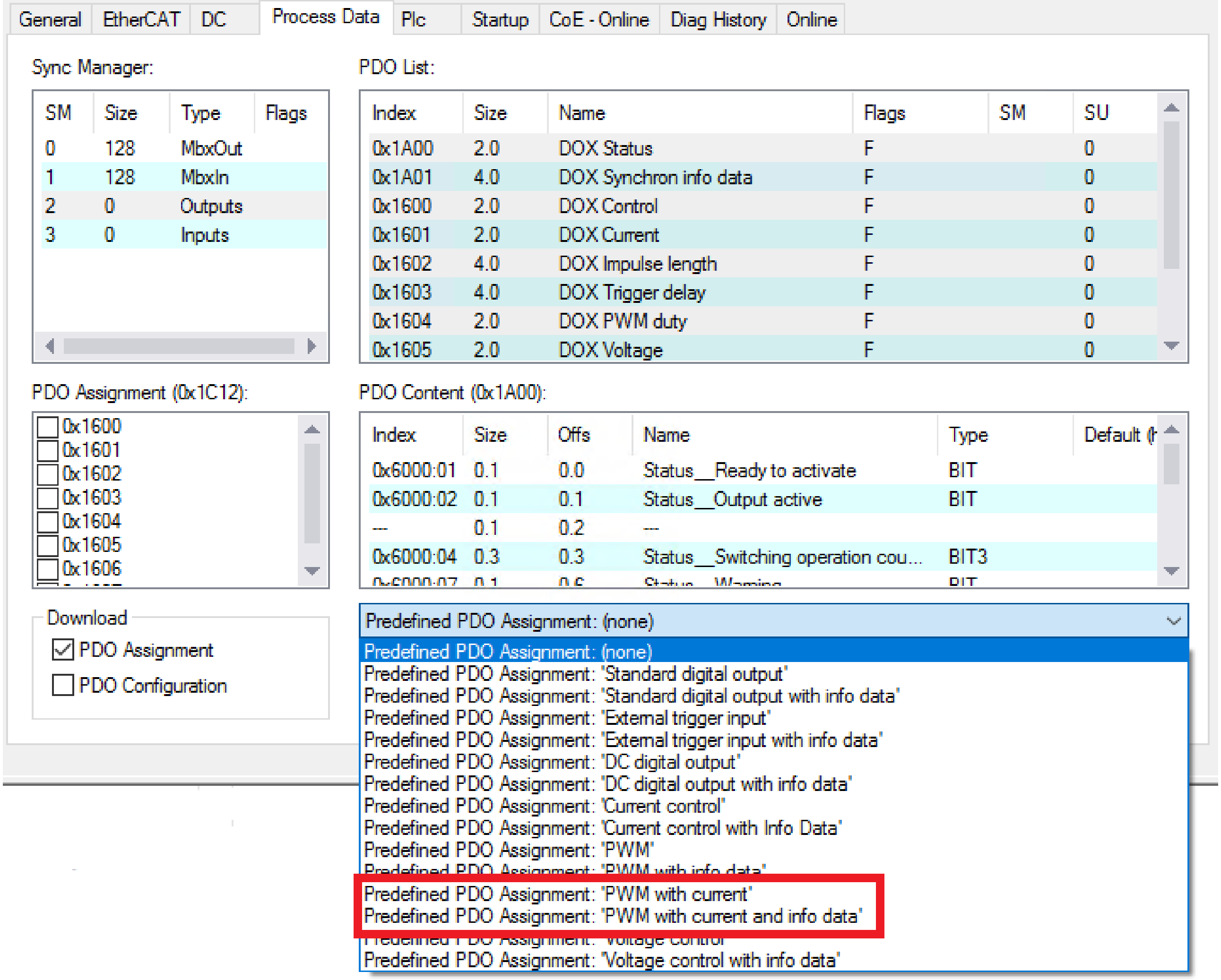
□ PDO Configuration

Predefined PDO Assignment: (none)

Predefined PDO Assignment: (none)
Predefined PDO Assignment: 'Standard digital output'
Predefined PDO Assignment: 'Standard digital output with info data'
Predefined PDO Assignment: 'External trigger input'
Predefined PDO Assignment: 'External trigger input with info data'
Predefined PDO Assignment: 'DC digital output'
Predefined PDO Assignment: 'DC digital output with info data'
Predefined PDO Assignment: 'Current control'
Predefined PDO Assignment: 'Current control with Info Data'
Predefined PDO Assignment: 'PWM'
Predefined PDO Assignment: 'PWM with info data'
Predefined PDO Assignment: 'PWM with current'
Predefined PDO Assignment: 'PWM with current and info data'
Predefined PDO Assignment: 'Voltage control'
Predefined PDO Assignment: 'Voltage control with info data'

Fig. 167: PDO setting “PWM (with info data)”

- **Only for Predefined PDO "PWM with current (and info data)":** Specify the set current in the unit mA via “DOX Current” → “Output Current”



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	0	Outputs	
3	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F		0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F		0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0

PDO Assignment (0x1C12):

- □ 0x1600
- □ 0x1601
- □ 0x1602
- □ 0x1603
- □ 0x1604
- □ 0x1605
- □ 0x1606

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	

Download

- ☑ PDO Assignment
- □ PDO Configuration

Predefined PDO Assignment: (none)

- Predefined PDO Assignment: (none)
- Predefined PDO Assignment: 'Standard digital output'
- Predefined PDO Assignment: 'Standard digital output with info data'
- Predefined PDO Assignment: 'External trigger input'
- Predefined PDO Assignment: 'External trigger input with info data'
- Predefined PDO Assignment: 'DC digital output'
- Predefined PDO Assignment: 'DC digital output with info data'
- Predefined PDO Assignment: 'Current control'
- Predefined PDO Assignment: 'Current control with Info Data'
- Predefined PDO Assignment: 'PWM'
- Predefined PDO Assignment: 'PWM with info data'
- Predefined PDO Assignment: 'PWM with current'
- Predefined PDO Assignment: 'PWM with current and info data'
- Predefined PDO Assignment: 'Voltage control'
- Predefined PDO Assignment: 'Voltage control with info data'

Fig. 168: PDO setting “PWM with current (and info data)”

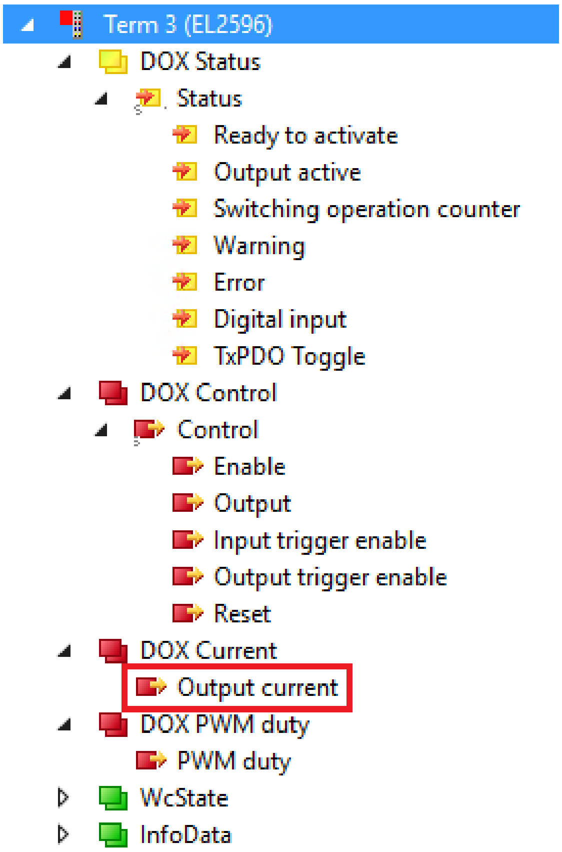
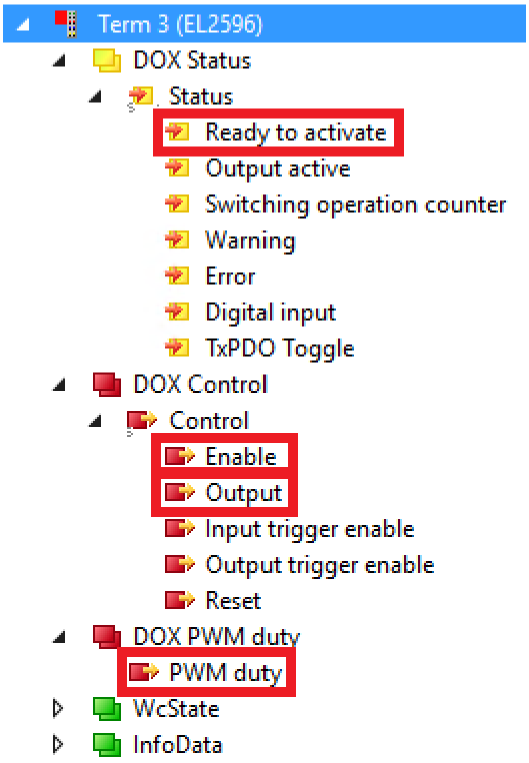


Fig. 169: Setting the set current

7. Specify “PWM duty cycle” under “DOX PWM duty” → “PWM duty”. If the PDO object 0x1604 “DOX PWM duty” is not mapped, the duty cycle is specified in the CoE index 0x8004:07 [▶ 219].
8. Check under “DOX Status” → “Status” whether the “Ready to activate” bit is 1
9. Activate the control under “DOX Control” → “Control” via the “Enable” bit
10. Switch on the LED output under “DOX Control” → “Control” by activating the “Output” bit



Term 3 (EL2596)
- DOX Status
  - Status
    - Ready to activate
    - Output active
    - Switching operation counter
    - Warning
    - Error
    - Digital input
    - TxPDO Toggle
- DOX Control
  - Control
    - Enable
    - Output
    - Input trigger enable
    - Output trigger enable
    - Reset
- DOX PWM duty
  - PWM duty
- WcState
- InfoData

Fig. 170: Activating the output in the operation mode “Current control PWM”

5.3.6 Voltage Control

The operation mode “Voltage Control” enables an LED to be driven by voltage control.

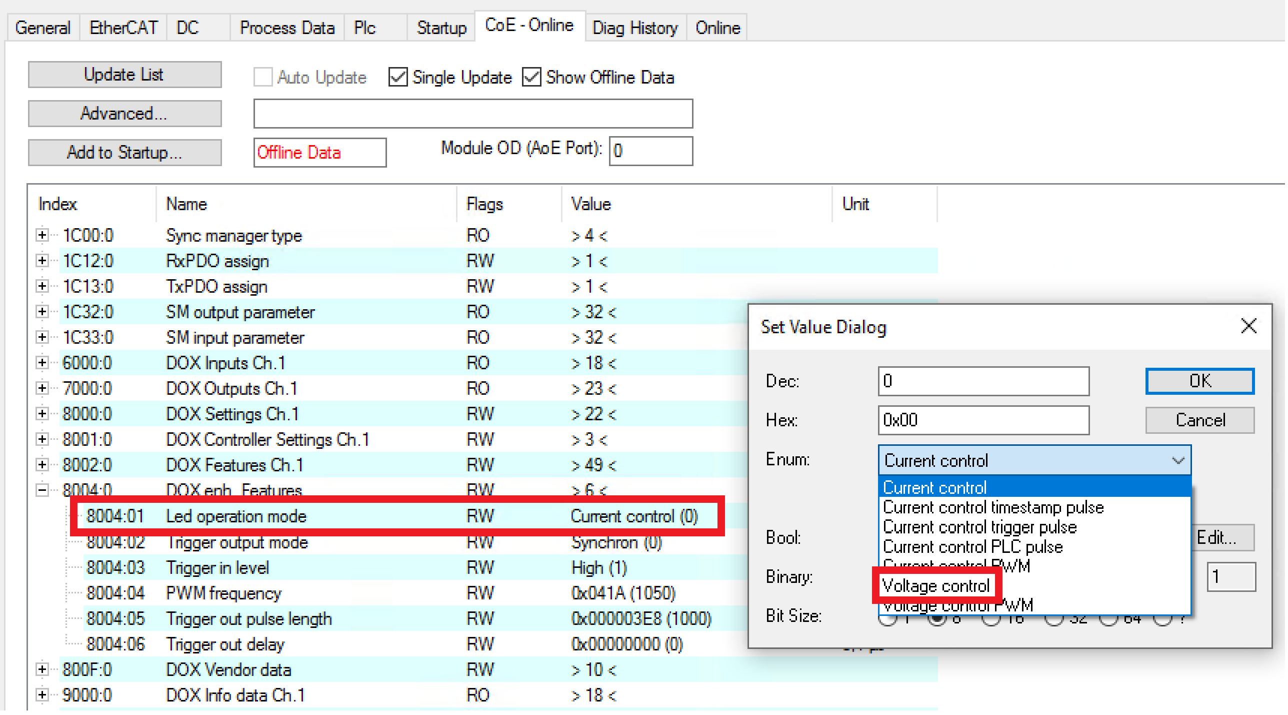
In this mode the maximum output current is 1.2 A up to an LED voltage of 12 V. Above 12 V the user must consider derating, as otherwise the module may switch off due to overtemperature if the power exceeds 14.4 W.

In this mode a set voltage is specified that is continuously controlled. The output current depends on the connected load. The output current can increase to 200% of the target current (limit current) in the CoE index 0x8000:02 [▸ 217]. If the resistance of the connected load is too low, so that the output current would have to rise to more than 200% of the set limit current in order to keep the output voltage constant at the specified value, the terminal switches the output off.

In this operation mode, it is possible to configure the trigger input as an enable input so that it can be used as an external switch. An output can then only be actuated with a prespecified signal at the trigger input. More detailed information and commissioning can be found in the chapter Hardware enable [▸ 178].

The specific parameters for setting the LED output as a voltage-controlled output are described below.

1. Nominal/limit current of the LED in index 0x8000:02 “Target current” in the unit mA
2. Input voltage in index 0x8000:03 “Supply voltage” in the unit 0.01 V
3. Set the operation mode in the CoE directory in the index 0x8004:01 [▸ 219] to “Voltage Control”



□ Auto Update ☑ Single Update ☑ Show Offline Data

Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
1C00:0	Sync manager type	RO	> 4 <	
1C12:0	RxPDO assign	RW	> 1 <	
1C13:0	TxPDO assign	RW	> 1 <	
1C32:0	SM output parameter	RO	> 32 <	
1C33:0	SM input parameter	RO	> 32 <	
6000:0	DOX Inputs Ch.1	RO	> 18 <	
7000:0	DOX Outputs Ch.1	RO	> 23 <	
8000:0	DOX Settings Ch.1	RW	> 22 <	
8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
8002:0	DOX Features Ch.1	RW	> 49 <	
8004:0	DOX enh. Features	RW	> 6 <	
8004:01	Led operation mode	RW	Current control (0)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	
8004:06	Trigger out delay	RW	0x00000000 (0)	
800F:0	DOX Vendor data	RW	> 10 <	
9000:0	DOX Info data Ch.1	RO	> 18 <	

Dec: 0

Hex: 0x00

Enum: Current control

- Current control
- Current control timestamp pulse
- Current control trigger pulse
- Current control PLC pulse
- Current control PWM
- Voltage control
- Voltage control PWM

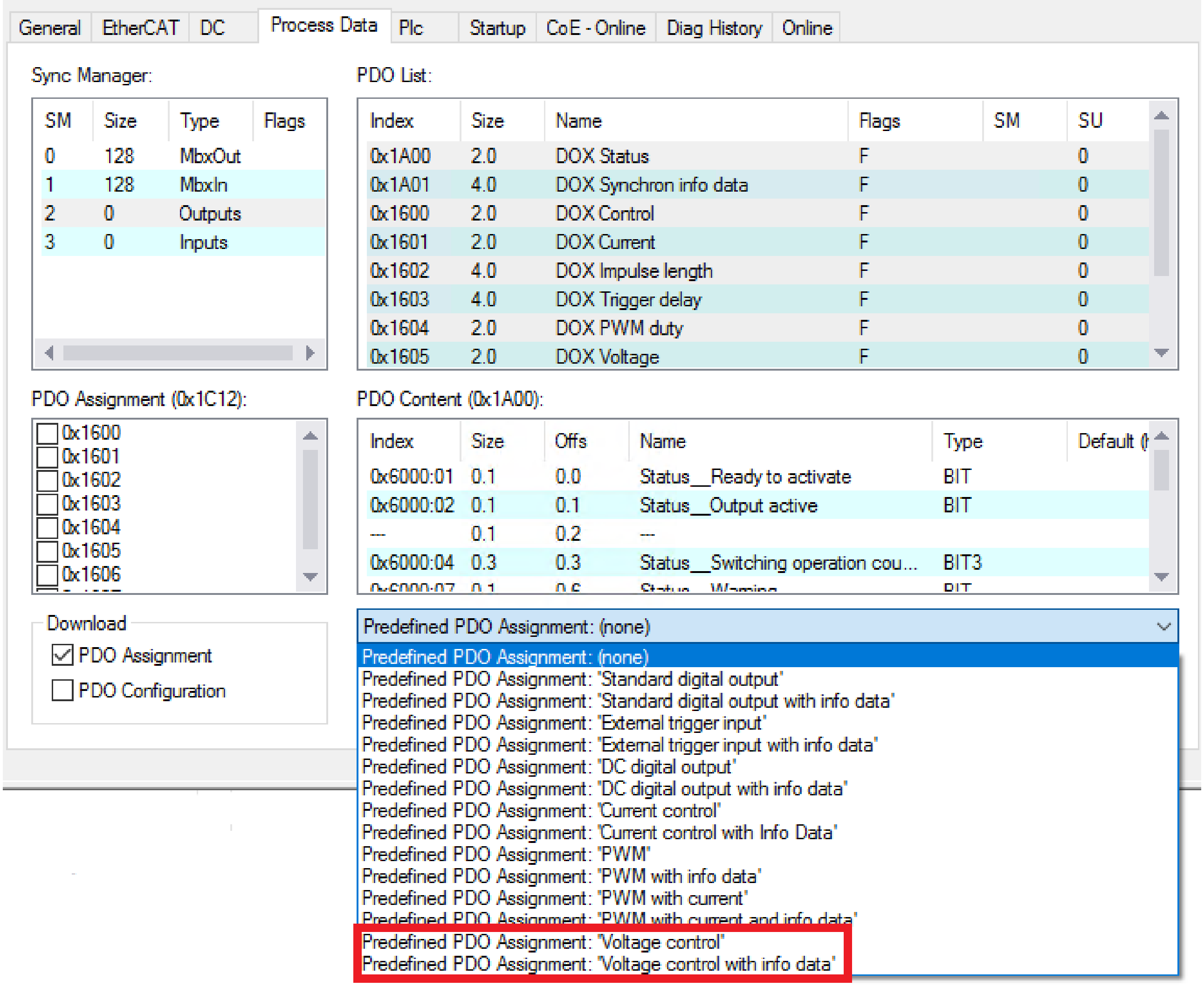
Bool:

Binary: 1

Bit Size: ○ 1 ● 8 ○ 16 ○ 32 ○ 64 ○ ?

Fig. 171: Operation mode setting “Voltage Control”

4. Set Predefined PDO Assignments to “Voltage control (with info data)”



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	0	Outputs	
3	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F		0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F		0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0

PDO Assignment (0x1C12):

□ 0x1600
□ 0x1601
□ 0x1602
□ 0x1603
□ 0x1604
□ 0x1605
□ 0x1606
□ 0x1607

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	

Download

☑ PDO Assignment

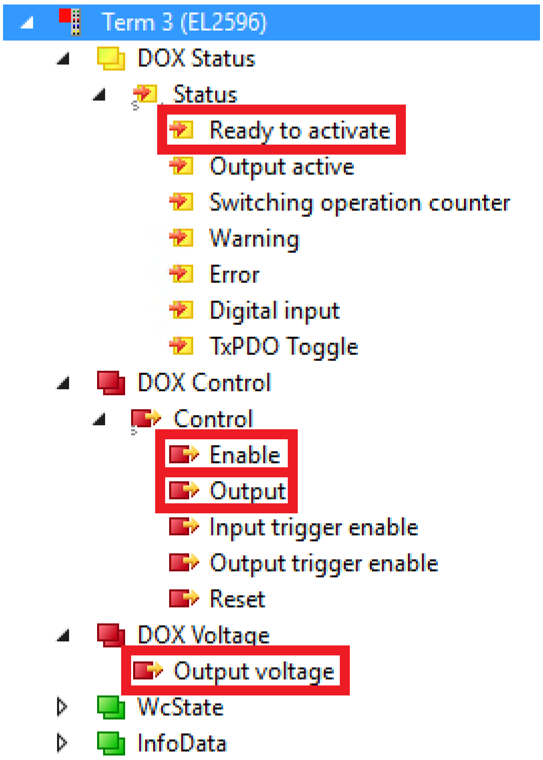
□ PDO Configuration

Predefined PDO Assignment: (none)

Predefined PDO Assignment: (none)
Predefined PDO Assignment: 'Standard digital output'
Predefined PDO Assignment: 'Standard digital output with info data'
Predefined PDO Assignment: 'External trigger input'
Predefined PDO Assignment: 'External trigger input with info data'
Predefined PDO Assignment: 'DC digital output'
Predefined PDO Assignment: 'DC digital output with info data'
Predefined PDO Assignment: 'Current control'
Predefined PDO Assignment: 'Current control with Info Data'
Predefined PDO Assignment: 'PWM'
Predefined PDO Assignment: 'PWM with info data'
Predefined PDO Assignment: 'PWM with current'
Predefined PDO Assignment: 'PWM with current and info data'
Predefined PDO Assignment: 'Voltage control'
Predefined PDO Assignment: 'Voltage control with info data'

Fig. 172: PDO setting “Voltage control (with info data)”

5. Specify the set voltage in the unit 0.01 V (max. U_{IN} - 0.5 V) via “DOX Voltage” → “Output” Voltage. The limit voltage of the output is specified in the unit 0.01 V in the CoE index 0x8000:04 [▶ 217] “Supply voltage”. If the set voltage specified in the PDO is greater than the limit voltage, the maximum limit voltage will nevertheless be output (without warning).
6. Check under “DOX Status” → “Status” whether the “Ready to activate” bit is 1.
7. Activate the control under “DOX Control” → “Control” via the “Enable” bit.
8. Switch on the LED output under “DOX Control” → “Control” by activating the “Output” bit.



Term 3 (EL2596)
DOX Status
Status
Ready to activate
Output active
Switching operation counter
Warning
Error
Digital input
TxPDO Toggle
DOX Control
Control
Enable
Output
Input trigger enable
Output trigger enable
Reset
DOX Voltage
Output voltage
WcState
InfoData

Fig. 173: Activating the output in the operation mode “Voltage control”

5.3.7 Voltage Control PWM

The operation mode “Voltage Control PWM” enables an LED to be driven by pulse width modulation of the voltage.

In this mode, the voltage is set to the value specified in the CoE index 0x8000:04 “Output voltage”. The output current depends on the connected load. The output current can increase to 200% of the “Target current” (limit current) in the CoE index 0x8000:02. If the resistance of the connected load is too low, so that the output current would have to rise to more than 200% of the set limit current in order to keep the output voltage constant at the specified value, the terminal switches the output off.

In this operation mode, it is possible to configure the trigger input as an enable input so that it can be used as an external switch. An output can then only be actuated with a prespecified signal at the trigger input. More detailed information and commissioning can be found in the chapter Hardware enable [▶ 178].

The specific parameters for setting the LED output as a voltage-PWM-controlled output are described below.

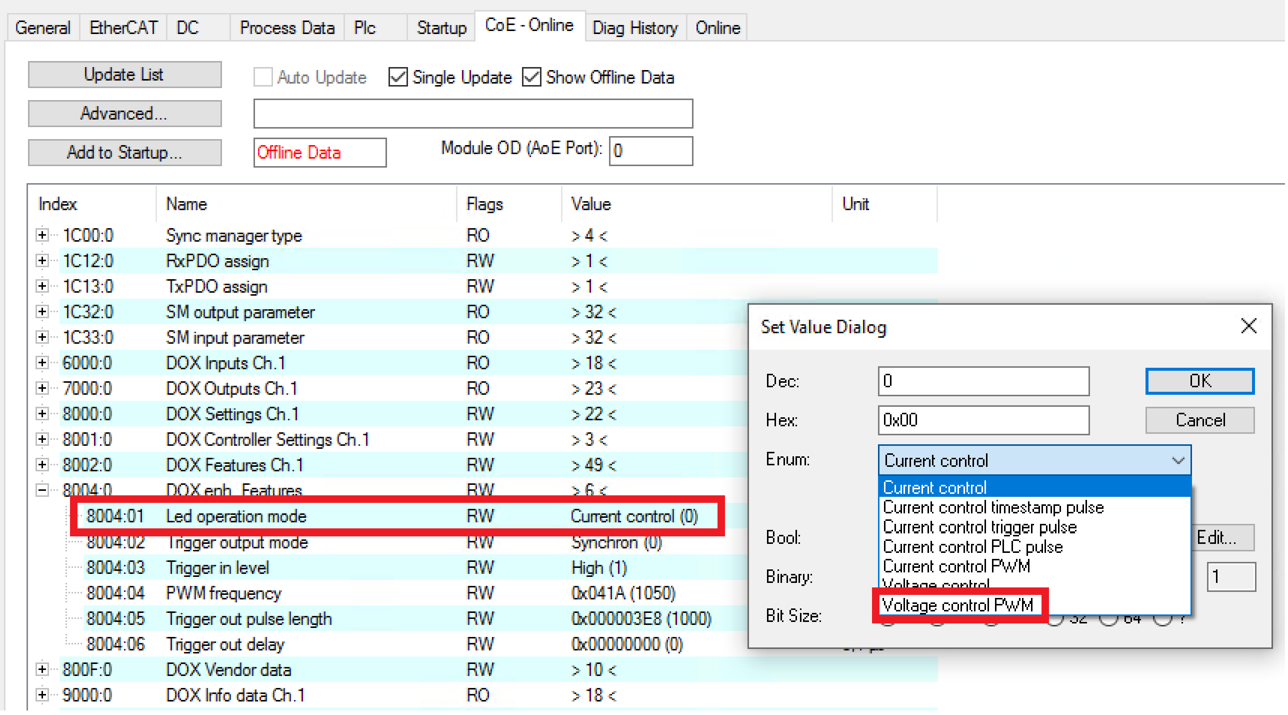
1. Nominal/limit current of the LED in index 0x8000:02 [▶ 217] “Target current” in the unit mA
2. Input voltage in index 0x8000:03 [▶ 217] “Supply voltage” in the unit 0.01 V
3. Desired output voltage in index 0x8000:04 [▶ 217] “Output voltage” in the unit 0.01 V (max. U_{IN} - 2 V).
4. The default PWM frequency is 1050 Hz. If necessary, the value can be adjusted in index 0x8004:04 [▶ 219].

⚠ **WARNING**

Stroboscopic effects in PWM mode

Due to the high-frequency light switching, there is a risk of a stroboscopic effect. If the frequency of the emitted light is in phase with the movement frequency of a rotating machine part, for example, a stroboscopic effect can make it appear as if the machine is stationary despite it moving. This can lead to a misinterpretation by an operator who may intervene due to the apparently stationary machine part. This can lead to serious injury or death.

5. Set the operation mode in the CoE directory in index 0x8004:01 [▶ 219] to “Voltage Control PWM”



General | EtherCAT | DC | Process Data | Plc | Startup | CoE - Online | Diag History | Online

Update List | □ Auto Update | ☑ Single Update | ☑ Show Offline Data

Advanced...

Add to Startup... | Offline Data | Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
+ 1C00:0	Sync manager type	RO	> 4 <	
+ 1C12:0	RxPDO assign	RW	> 1 <	
+ 1C13:0	TxPDO assign	RW	> 1 <	
+ 1C32:0	SM output parameter	RO	> 32 <	
+ 1C33:0	SM input parameter	RO	> 32 <	
+ 6000:0	DOX Inputs Ch.1	RO	> 18 <	
+ 7000:0	DOX Outputs Ch.1	RO	> 23 <	
+ 8000:0	DOX Settings Ch.1	RW	> 22 <	
+ 8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
+ 8002:0	DOX Features Ch.1	RW	> 49 <	
− 8004:0	DOX enh. Features	RW	> 6 <	
8004:01	Led operation mode	RW	Current control (0)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	
8004:06	Trigger out delay	RW	0x00000000 (0)	
+ 800F:0	DOX Vendor data	RW	> 10 <	
+ 9000:0	DOX Info data Ch.1	RO	> 18 <	

Set Value Dialog

Dec: 0

Hex: 0x00

Enum: Current control

- Current control
- Current control timestamp pulse
- Current control trigger pulse
- Current control PLC pulse
- Current control PWM
- Voltage control
- Voltage control PWM

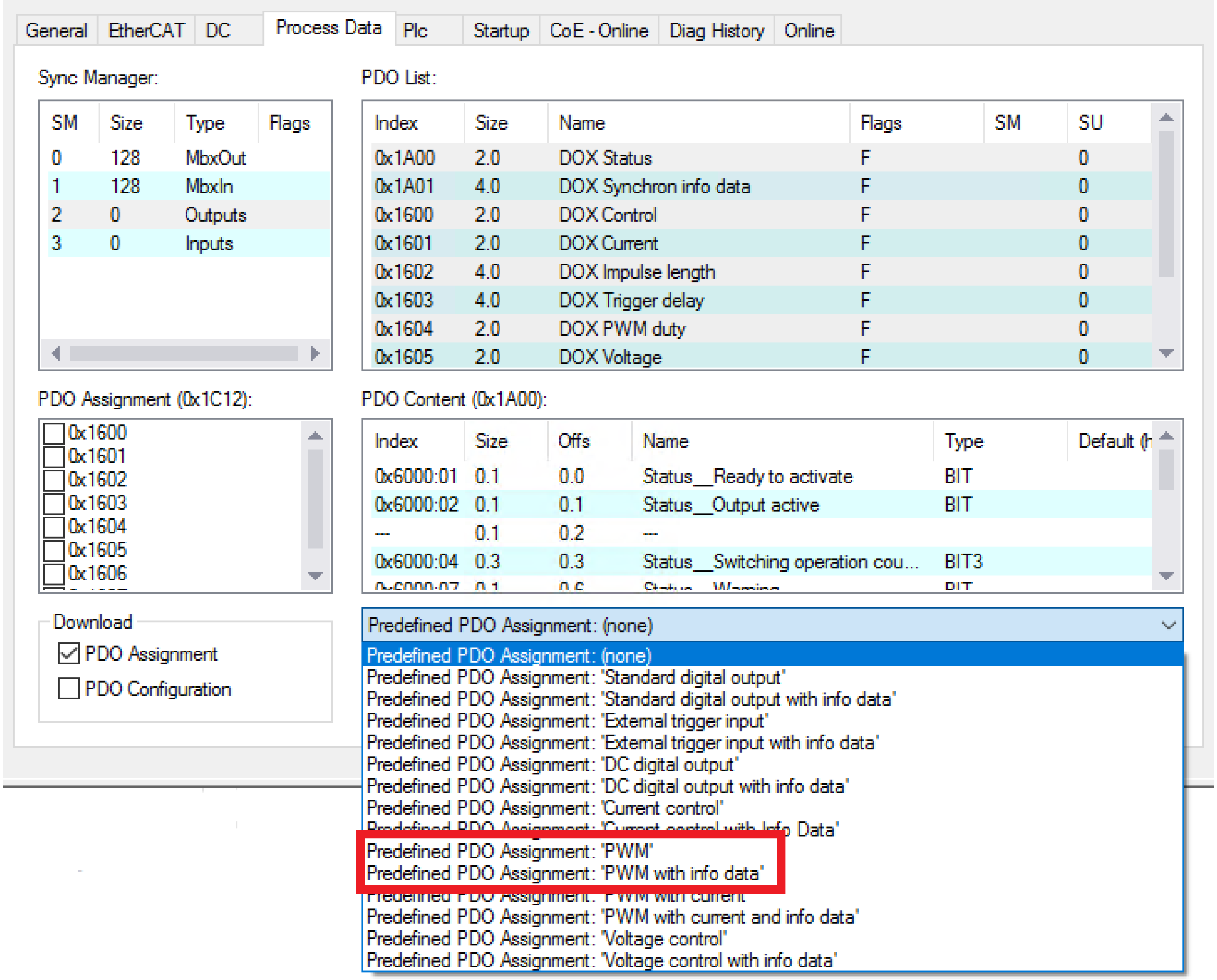
Bool:

Binary: 1

Bit Size: ○ 32 ○ 64 ○ ?

Fig. 174: Operation mode setting “Voltage control PWM”

6. Set Predefined PDO Assignments to “PWM (with info data)”



General | EtherCAT | DC | Process Data | Plc | Startup | CoE - Online | Diag History | Online

Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	0	Outputs	
3	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F		0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F		0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0

PDO Assignment (0x1C12):

- □ 0x1600
- □ 0x1601
- □ 0x1602
- □ 0x1603
- □ 0x1604
- □ 0x1605
- □ 0x1606

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	

Download

☑ PDO Assignment

□ PDO Configuration

Predefined PDO Assignment: (none)

- Predefined PDO Assignment: (none)
- Predefined PDO Assignment: 'Standard digital output'
- Predefined PDO Assignment: 'Standard digital output with info data'
- Predefined PDO Assignment: 'External trigger input'
- Predefined PDO Assignment: 'External trigger input with info data'
- Predefined PDO Assignment: 'DC digital output'
- Predefined PDO Assignment: 'DC digital output with info data'
- Predefined PDO Assignment: 'Current control'
- Predefined PDO Assignment: 'Current control with Info Data'
- Predefined PDO Assignment: 'PWM'
- Predefined PDO Assignment: 'PWM with info data'
- Predefined PDO Assignment: 'PWM with current'
- Predefined PDO Assignment: 'PWM with current and info data'
- Predefined PDO Assignment: 'Voltage control'
- Predefined PDO Assignment: 'Voltage control with info data'

Fig. 175: PDO setting “PWM (with info data)”

7. Specify PWM duty cycle under “DOX PWM duty” → “PWM duty”. If the PDO object 0x1604 “DOX PWM duty” is not mapped, the duty cycle is specified in the CoE index 0x8004:07 [▸ 219].
8. Check under “DOX Status” → “Status” whether the “Ready to activate” bit is 1.
9. Activate the control under “DOX Control” → “Control” via the “Enable” bit.
10. Switch on the LED output under “DOX Control” → “Control” by activating the “Output” bit.

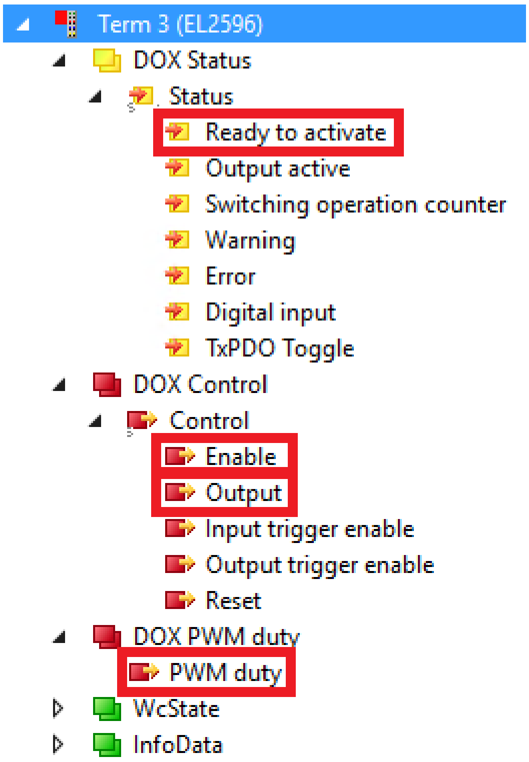


Fig. 176: Activating the output in the operation mode “Voltage control PWM”

5.3.8 Current Sink PWM

Available from FW04

The operation mode described in this chapter for controlling an LED with an external supply can only be used from FW04 onwards.

The operation mode “Current Sink PWM” is used when operating an LED with an external supply. This is the case, for example, with a multi-color Common Anode LED [▸ 190]. The individual LEDs are supplied via the common anode. One EL2596 is required for each cathode (equivalent to “for each color”). These EL2596 terminals are then operated in “Current Sink PWM” mode.

In this operation mode, there are a few requirements to consider:

- The LED must not be connected to the EL2596 until the operation mode has been set. The LED is **not** connected between LED- (1) and LED+ (9) as in the other operation modes. The anode of the LED is connected to an external supply source and the cathode is connected to LED- (1) of the EL2596.
- The operation mode must not be changed while the LED is connected.
- The external supply of the LED may only be switched on after the terminal has started up. If this is not directly possible in the application, the supply must be connected via a relay terminal (EL2622, EL2624, ...).
- The same restrictions regarding the maximum duty cycle as described in the chapter on Current control PWM [▸ 148] apply.

In this operation mode, it is possible to configure the trigger input as an enable input so that it can be used as an external switch. An output can then only be actuated with a prespecified signal at the trigger input. More detailed information and commissioning can be found in the chapter Hardware enable [▸ 178].

The commissioning of the “Current Sink PWM” mode is described step-by-step below.

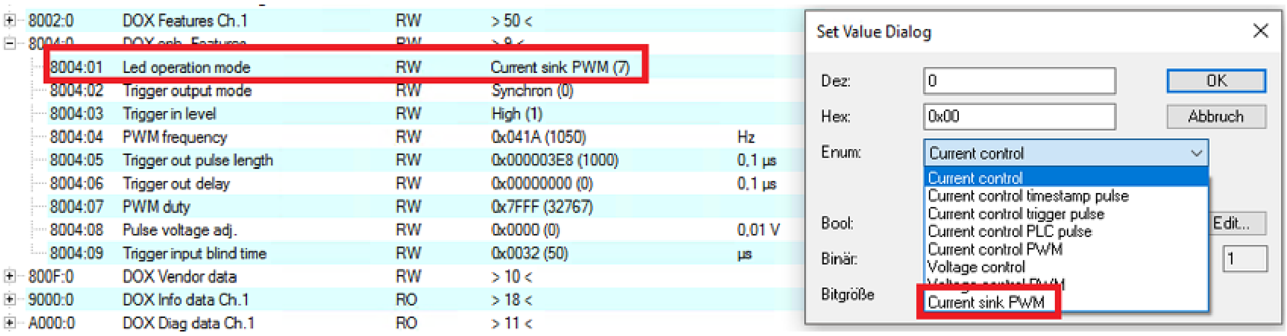
1. Nominal/limit current of the LED in index 0x8000:02 [▸ 217] “Target current” in the unit mA
2. Input voltage in index 0x8000:03 [▸ 217] “Supply voltage” in the unit 0.01 V
3. Desired output voltage in index 0x8000:04 [▸ 217] “Output voltage” in the unit 0.01 V (max. U_{IN} - 2 V). In this operation mode, the value set here corresponds to the output on the controller. If the forward voltage of the connected LED is lower than the value set here, the remaining voltage drops internally in the EL2596. In many cases, this leads to overtemperature in the terminal and thus to the switching off of the LED output. As the desired output voltage, therefore, the forward voltage at the desired output current should be set here. The TeachIn function [▸ 168] can also be used to determine the output voltage (the LED must not be connected to the EL2596 when switching to the necessary operation mode).
4. The default PWM frequency is 1050 Hz. If necessary, the value can be adjusted in index 0x8004:04 [▸ 219].

⚠ **WARNING**

Stroboscopic effects in PWM mode

Due to the high-frequency light switching, there is a risk of a stroboscopic effect. If the frequency of the emitted light is in phase with the movement frequency of a rotating machine part, for example, a stroboscopic effect can make it appear as if the machine is stationary despite it moving. This can lead to a misinterpretation by an operator who may intervene due to the apparently stationary machine part. This can lead to serious injury or death.

5. Set the operation mode in the CoE directory in index 0x8004:01 [▸ 219] to “Current Control PWM”



8002:0	DOX Features Ch.1	RW	> 50 <	
8004:0	DOX enh. Features	RW	> 9 <	
8004:01	Led operation mode	RW	Current sink PWM (7)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	Hz
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	0,1 µs
8004:06	Trigger out delay	RW	0x00000000 (0)	0,1 µs
8004:07	PWM duty	RW	0x7FFF (32767)	
8004:08	Pulse voltage adj.	RW	0x0000 (0)	0,01 V
8004:09	Trigger input blind time	RW	0x0032 (50)	µs
800F:0	DOX Vendor data	RW	> 10 <	
9000:0	DOX Info data Ch.1	RO	> 18 <	
A000:0	DOX Diag data Ch.1	RO	> 11 <	

Set Value Dialog

Dez: 0

Hex: 0x00

Enum: Current control

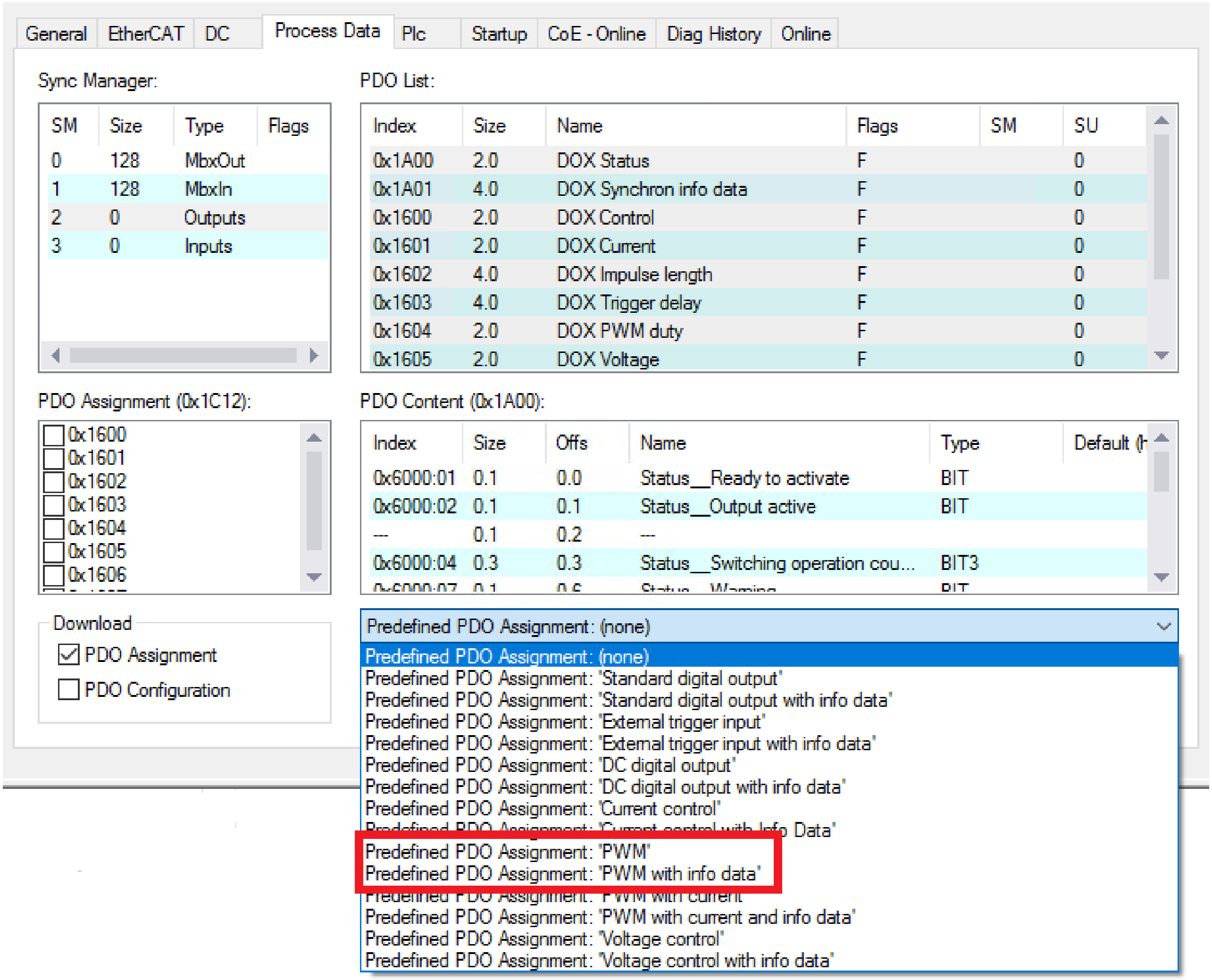
Bool:

Binär: 1

Bitgröße

Fig. 177: Operation mode setting “Current control PWM”

6. The LED can now be connected.
7. Set the predefined PDO assignments to “PWM (with info data)” or “PWM with current (and info data)”



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	0	Outputs	
3	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F		0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F		0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0

PDO Assignment (0x1C12):

- □ 0x1600
- □ 0x1601
- □ 0x1602
- □ 0x1603
- □ 0x1604
- □ 0x1605
- □ 0x1606
- □ 0x1607

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	

Download

☑ PDO Assignment

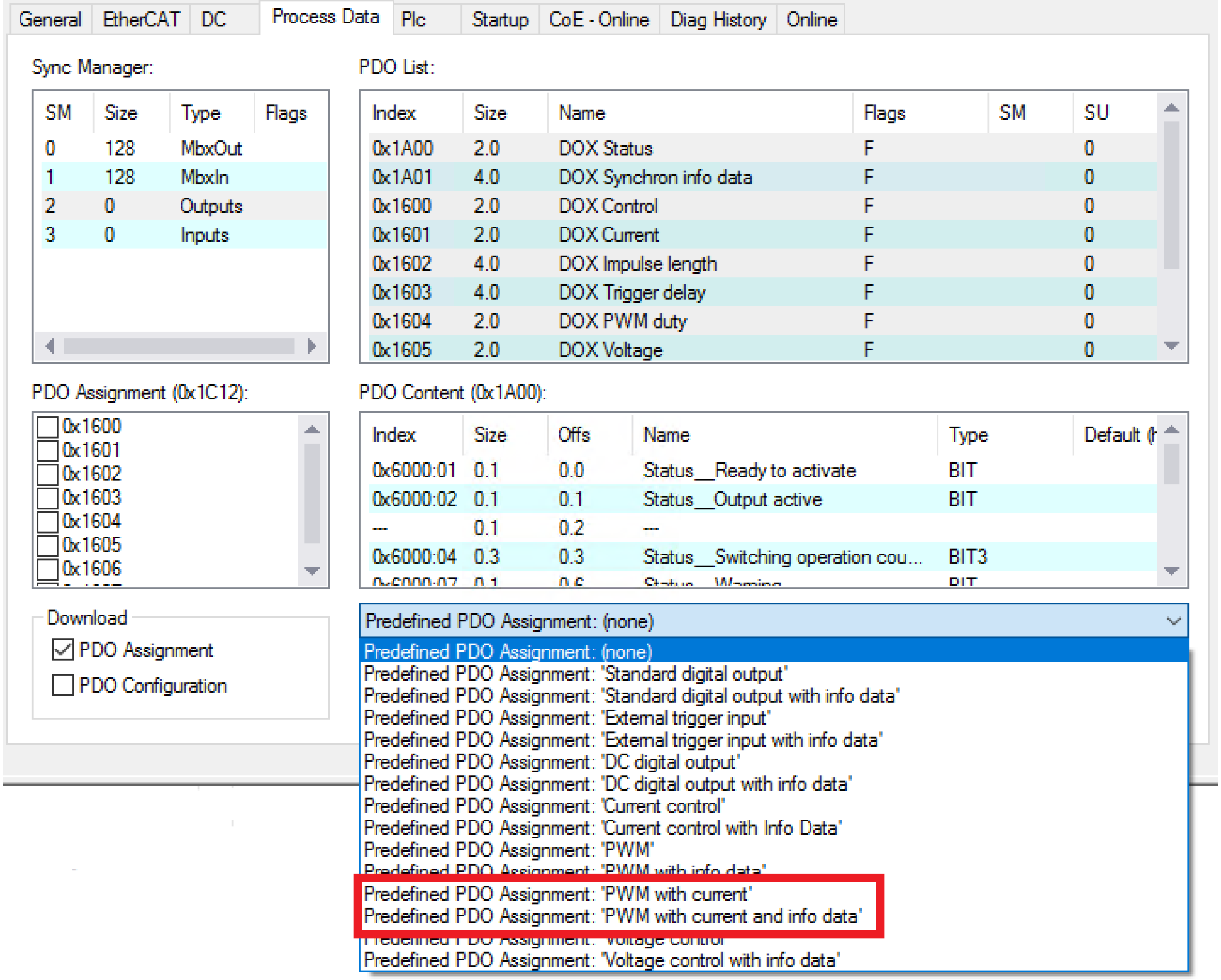
□ PDO Configuration

Predefined PDO Assignment: (none)

- Predefined PDO Assignment: (none)
- Predefined PDO Assignment: 'Standard digital output'
- Predefined PDO Assignment: 'Standard digital output with info data'
- Predefined PDO Assignment: 'External trigger input'
- Predefined PDO Assignment: 'External trigger input with info data'
- Predefined PDO Assignment: 'DC digital output'
- Predefined PDO Assignment: 'DC digital output with info data'
- Predefined PDO Assignment: 'Current control'
- Predefined PDO Assignment: 'Current control with Info Data'
- Predefined PDO Assignment: 'PWM'
- Predefined PDO Assignment: 'PWM with info data'
- Predefined PDO Assignment: 'PWM with current'
- Predefined PDO Assignment: 'PWM with current and info data'
- Predefined PDO Assignment: 'Voltage control'
- Predefined PDO Assignment: 'Voltage control with info data'

Fig. 178: PDO setting “PWM (with info data)”

- **Only for Predefined PDO “PWM with current (and info data)”:** Specify the set current in the unit mA via "DOX Current" → "Output Current"



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	0	Outputs	
3	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F		0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F		0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0

PDO Assignment (0x1C12):

- □ 0x1600
- □ 0x1601
- □ 0x1602
- □ 0x1603
- □ 0x1604
- □ 0x1605
- □ 0x1606

PDO Content (0x1A00):

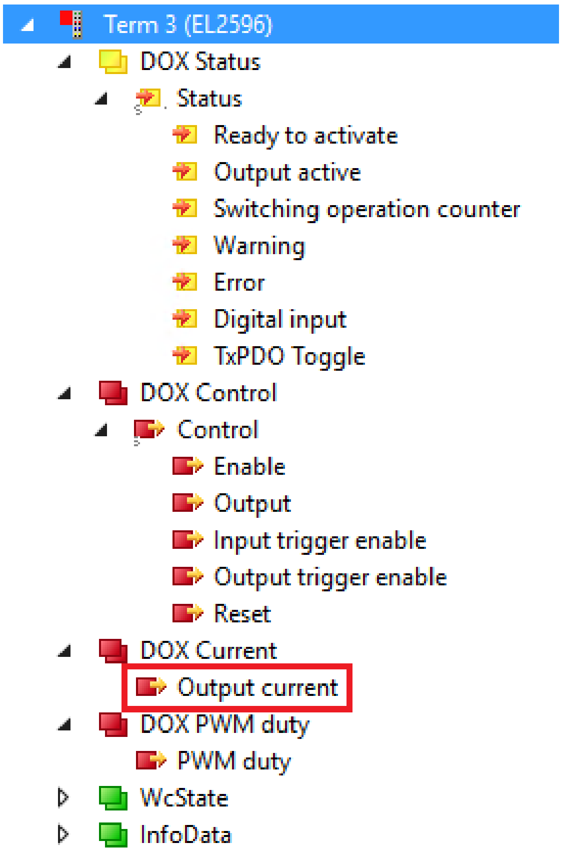
Index	Size	Offs	Name	Type	Default (h
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	

Download
- ☑ PDO Assignment
- □ PDO Configuration

Predefined PDO Assignment: (none)

- Predefined PDO Assignment: (none)
- Predefined PDO Assignment: 'Standard digital output'
- Predefined PDO Assignment: 'Standard digital output with info data'
- Predefined PDO Assignment: 'External trigger input'
- Predefined PDO Assignment: 'External trigger input with info data'
- Predefined PDO Assignment: 'DC digital output'
- Predefined PDO Assignment: 'DC digital output with info data'
- Predefined PDO Assignment: 'Current control'
- Predefined PDO Assignment: 'Current control with Info Data'
- Predefined PDO Assignment: 'PWM'
- Predefined PDO Assignment: 'PWM with info data'
- Predefined PDO Assignment: 'PWM with current'
- Predefined PDO Assignment: 'PWM with current and info data'
- Predefined PDO Assignment: 'Voltage control'
- Predefined PDO Assignment: 'Voltage control with info data'

Fig. 179: PDO setting “PWM with current (and info data)”



- Term 3 (EL2596)
  - DOX Status
    - Status
      - Ready to activate
      - Output active
      - Switching operation counter
      - Warning
      - Error
      - Digital input
      - TxPDO Toggle
  - DOX Control
    - Control
      - Enable
      - Output
      - Input trigger enable
      - Output trigger enable
      - Reset
  - DOX Current
    - Output current
  - DOX PWM duty
    - PWM duty
  - WcState
  - InfoData

Fig. 180: Setting the set current

8. Specify “PWM duty cycle” under “DOX PWM duty” → “PWM duty”. If the PDO object 0x1604 “DOX PWM duty” is not mapped, the duty cycle is specified in the CoE index 0x8004:07 [▶ 219].
9. Check under “DOX Status” → “Status” whether the “Ready to activate” bit is 1
10. Switch on the LED output under “DOX Control” → “Control” by activating the “Output” bit

5.4 Determination of the output parameters

This chapter describes an example of the parameterization of the EL2596-xxxx. The case is considered where one or more LEDs are supplied with a constant current by the EL2596-xxxx in the “Current Control” operation mode.

First of all, the forward voltage U_F is to be determined at the operating point of the LED. To this end there is a characteristic curve in the data sheet for the LED that shows the forward current I_F as a function of the forward voltage U_F. If a current other than the nominal current of the diodes is desired, the corresponding forward voltage is to be taken from the diagram that shows the forward voltage plotted against the forward current of the LED.

For a calculation example, let's assume that the nominal current is 1 A, resulting in a forward voltage of 2.35 V.

Several LEDs of the same type are to be connected in series in the example under consideration.

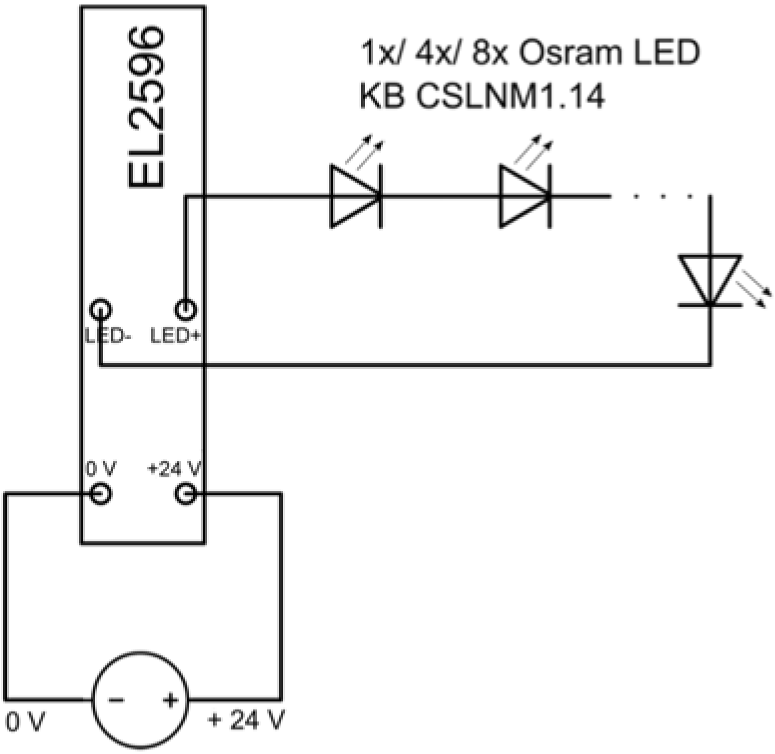


Fig. 181: Series connection of LEDs to the EL2596

Thus, the resulting output voltage of the EL2596 U_{out} corresponds to the number of series-connected LEDs multiplied by the forward voltage U_F at the set output current I_F.

$$U_{out} = n_{LEDs} \cdot U_F$$

With the set output current I_F and the output voltage, the output power P_{out} of the EL2596 can now be calculated:

$$P_{out} = U_{out} \cdot I_F$$

For a different number of LEDs, the following parameters now result for the EL2596:

Number of LEDs	Forward current I_F	Output voltage U_{out}	Output power P_{out}
1	1 A	2.35 V	2.35 W
4	1 A	9.4 V	9.4 W
8	1 A	18.8 V	18.8 W

Once the parameters for the desired circuit have been calculated, check whether the operating range of the EL2596 is complied with. The operating range is limited by three parameters:

- the maximum continuous output current, which is limited by internal components
- the maximum output voltage of the terminal, which is 28.3 V (continuous light) or 26.8 V (pulse mode) with a maximum supply voltage of 28.8 V
- the maximum output power of 14.4 W, which correlates to the internal power loss (the internal power loss can be read from the info data in CoE index 0xF900:13 [▶ 221])

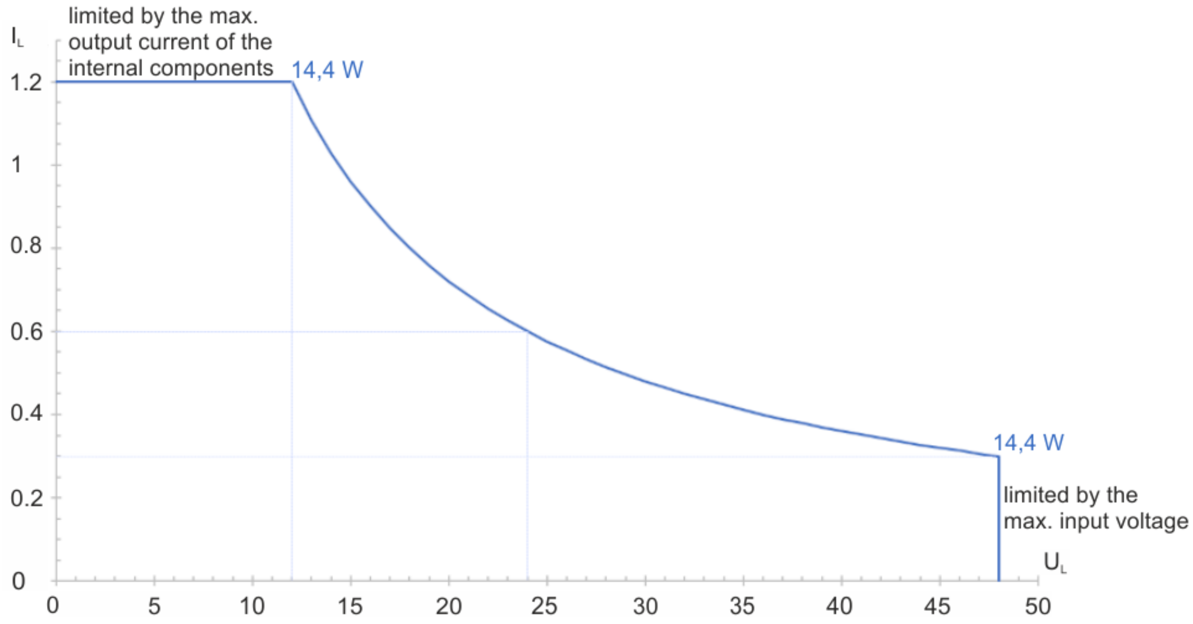


Fig. 182: Operating range of EL2596 in “Current control” mode

As can be seen in the illustration, the parameters for one and four LEDs lie within the permissible operating range of the EL2596. The maximum output power of the EL2596 is exceeded with eight LEDs, therefore the circuit is not permissible. The parallel connection of several EL2596 terminals for the purpose of increasing the power is not permitted due to the internal circuit in the EL2596.

The output power can be reduced by lowering the output current until the permissible operating range of the EL2596 is complied with.

5.5 Automatic determination of the output voltage and display of the characteristic curve

Available from FW04

The function described in this chapter for the automatic determination of the output voltage and output of the characteristic curve can only be used from FW04.

With the EL2596, it is possible from firmware 04 onwards to automatically determine the output voltage when the set current is set and to save it in the CoE either temporarily or permanently. In general, an automatic determination of the output voltage is recommended, as this minimizes the losses in the terminal. The forward voltage specified in the data sheet is influenced by manufacturing tolerances, so that the optimum forward voltage can only be determined by automatic determination or another measurement.

In addition, it is possible to record the characteristic curve of a connected LED up to a predetermined set current and to save the values as a CSV file or as a curve in HTML.

The various options and procedures are described below. All options must be performed in the operation mode “Current Control”.

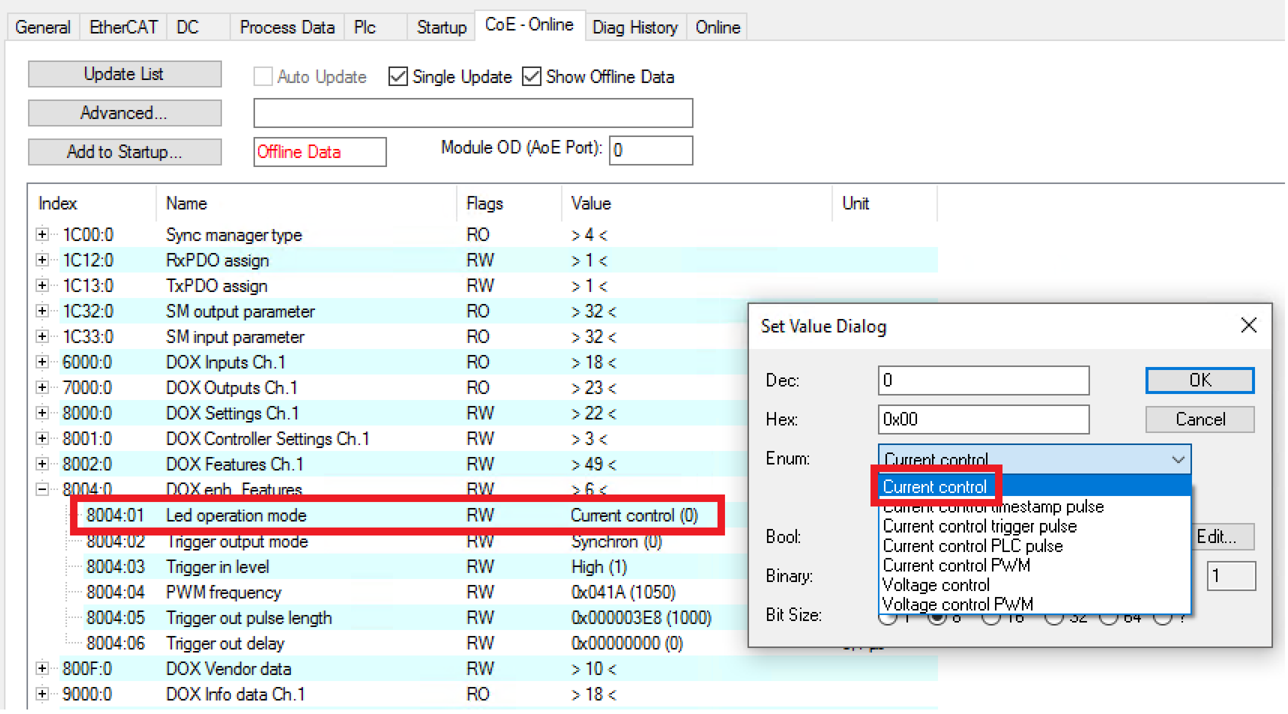
5.5.1 Saving the output voltage in operation

One way to determine and save the output voltage in the CoE index 0x8004:04 with a prespecified set current is to determine it during operation. With this method, the LED is switched on in "Current Control" mode with a prespecified set current and the output voltage is determined and saved in the CoE index 0x8000:04 [▸ 217] "Output voltage".

Since for this method the LED must be turned on in order to manually set a command that saves the voltage value, this method is relatively slow and is not suitable if the LED is to be overdriven in operation with currents greater than the nominal current. It is recommended to use this method if the LED is operated later in continuous light mode with currents ≥ the nominal current.

The procedure is described below:

1. Set the operation mode in the CoE index 0x8004:01 [▸ 219] “Led operation mode” to “Current Control”



□ Auto Update ☑ Single Update ☑ Show Offline Data

Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
1C00:0	Sync manager type	RO	> 4 <	
1C12:0	RxPDO assign	RW	> 1 <	
1C13:0	TxPDO assign	RW	> 1 <	
1C32:0	SM output parameter	RO	> 32 <	
1C33:0	SM input parameter	RO	> 32 <	
6000:0	DOX Inputs Ch.1	RO	> 18 <	
7000:0	DOX Outputs Ch.1	RO	> 23 <	
8000:0	DOX Settings Ch.1	RW	> 22 <	
8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
8002:0	DOX Features Ch.1	RW	> 49 <	
8004:0	DOX enh. Features	RW	> 6 <	
8004:01	Led operation mode	RW	Current control (0)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	
8004:06	Trigger out delay	RW	0x00000000 (0)	
800F:0	DOX Vendor data	RW	> 10 <	
9000:0	DOX Info data Ch.1	RO	> 18 <	

Set Value Dialog

Dec: 0

Hex: 0x00

Enum: Current control

- Current control
- Current control timestamp pulse
- Current control trigger pulse
- Current control PLC pulse
- Current control PWM
- Voltage control
- Voltage control PWM

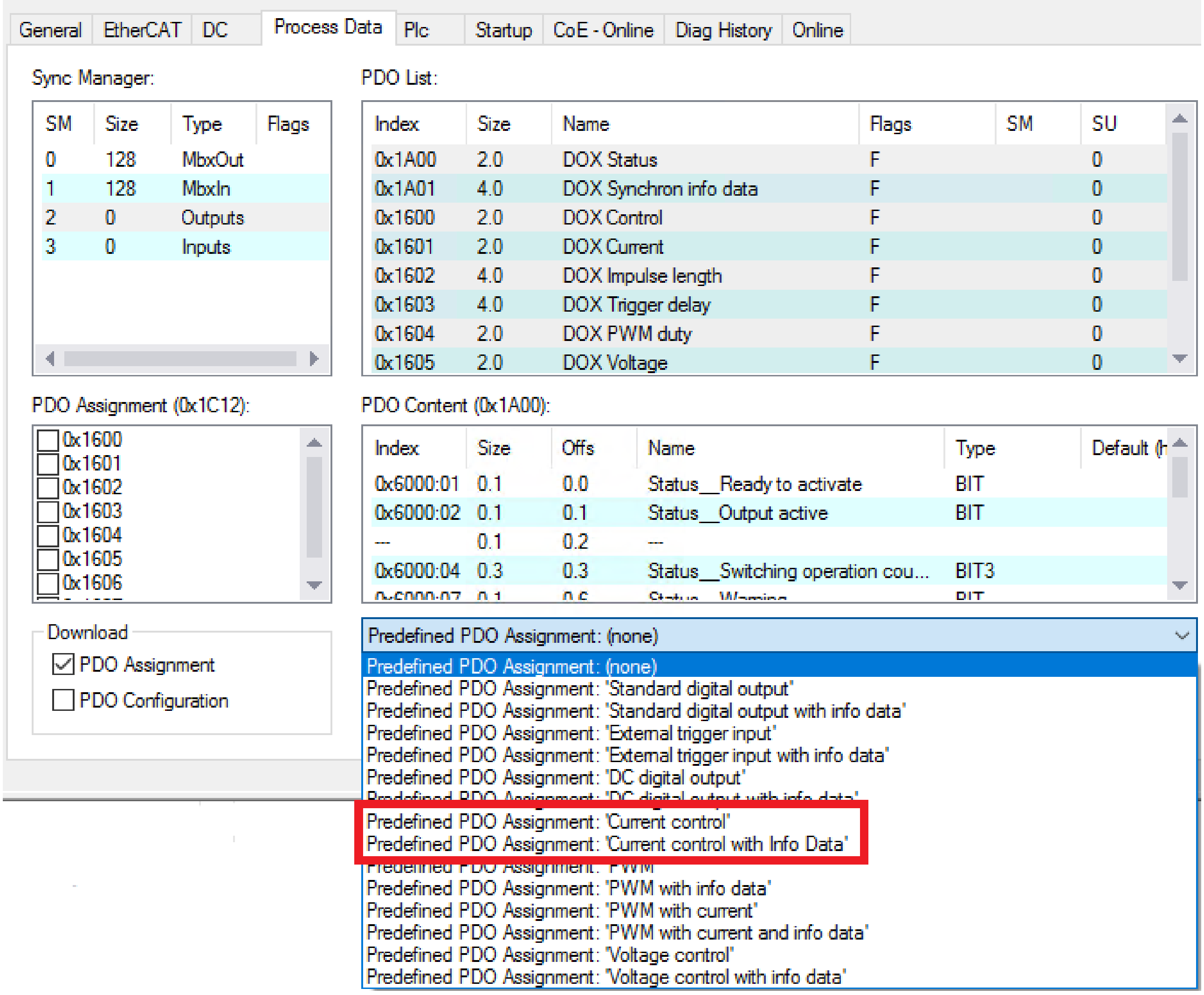
Bool:

Binary: 1

Bit Size: ○ 1 ● 8 ○ 16 ○ 32 ○ 64 ○ ?

Fig. 183: CoE object 0x8004:01 “Current Control”

2. Set the Predefined PDOs in the PDO Tab to “Current control (with info data)”



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	0	Outputs	
3	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F		0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F		0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0

PDO Assignment (0x1C12):

□ 0x1600
□ 0x1601
□ 0x1602
□ 0x1603
□ 0x1604
□ 0x1605
□ 0x1606

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	

Download

☑ PDO Assignment

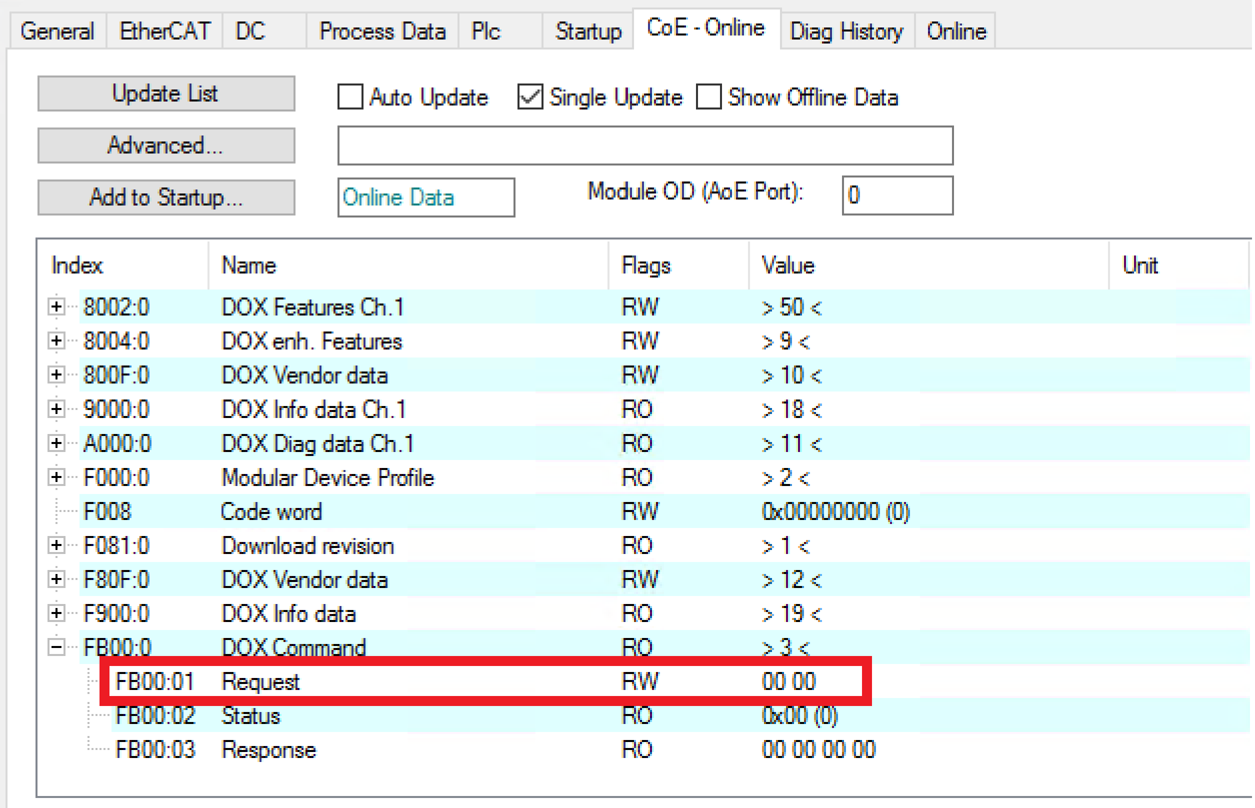
□ PDO Configuration

Predefined PDO Assignment: (none)

Predefined PDO Assignment: (none)
Predefined PDO Assignment: 'Standard digital output'
Predefined PDO Assignment: 'Standard digital output with info data'
Predefined PDO Assignment: 'External trigger input'
Predefined PDO Assignment: 'External trigger input with info data'
Predefined PDO Assignment: 'DC digital output'
Predefined PDO Assignment: 'DC digital output with info data'
Predefined PDO Assignment: 'Current control'
Predefined PDO Assignment: 'Current control with Info Data'
Predefined PDO Assignment: 'PWM'
Predefined PDO Assignment: 'PWM with info data'
Predefined PDO Assignment: 'PWM with current'
Predefined PDO Assignment: 'PWM with current and info data'
Predefined PDO Assignment: 'Voltage control'
Predefined PDO Assignment: 'Voltage control with info data'

Fig. 184: Select Predefined PDO “Current Control (with info data)”

3. Preset the set current in the process data under “DOX Current” → “Output Current”
4. Enable the LED output via “DOX Control” → “Control” → “Enable”
5. Switch on the LED output via “DOX Control” → “Control” → “Output”
6. Specify the command 0x0501 to store the output voltage in the CoE index 0xFB00:01 [▶ 222] “Request”. The stored voltage is then entered in the CoE index 0x8000:04 [▶ 217] “Output voltage”.



□ Auto Update ☑ Single Update □ Show Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
+ 8002:0	DOX Features Ch.1	RW	> 50 <	
+ 8004:0	DOX enh. Features	RW	> 9 <	
+ 800F:0	DOX Vendor data	RW	> 10 <	
+ 9000:0	DOX Info data Ch.1	RO	> 18 <	
+ A000:0	DOX Diag data Ch.1	RO	> 11 <	
+ F000:0	Modular Device Profile	RO	> 2 <	
F008	Code word	RW	0x00000000 (0)	
+ F081:0	Download revision	RO	> 1 <	
+ F80F:0	DOX Vendor data	RW	> 12 <	
+ F900:0	DOX Info data	RO	> 19 <	
− FB00:0	DOX Command	RO	> 3 <	
FB00:01	Request	RW	00 00	
FB00:02	Status	RO	0x00 (0)	
FB00:03	Response	RO	00 00 00 00	

Fig. 185: Entering the command for storing the output voltage

5.5.2 Storage of the output voltage temporarily or in the EEPROM

Another way to determine and store the output voltage in the CoE index 0x8000:04 [▸ 217] (temporarily or permanently) with a prespecified set current is the automatic determination. With this method, the LED is switched on with a prespecified set current and the output voltage is determined and saved in the CoE index 0x8000:04 [▸ 217] “Output voltage”.

If a “TeachIn” is performed using the commands described in this chapter, the terminal is automatically operated in “Current control” mode for a short switching process until the set current is reached. This is independent of the actual set operation mode, which is reactivated afterwards. This is followed by a brief flash of the LED. Depending on the command, the value is then permanently stored in the EEPROM or only temporarily until the next startup. When storing the voltage value in the EEPROM, it must be borne in mind that only limited write access is possible to EEPROM memory cells.

After the specifying of the set current, the setting of the command and a rising edge on the output bit of the EL2596, the output voltage is determined. Since the LED does not need to be constantly switched on for this method and the determination is automatic after the command has been given, this method is fast. It is also suitable if the LED is to be overdriven with currents greater than the nominal current during operation. During the determination, the current is adjusted up to the specified set current, which is why the LED flashes for a short time. This method can therefore be used if the LED is to be operated later in a pulsed mode with overdriving at a current ≥ the nominal current.

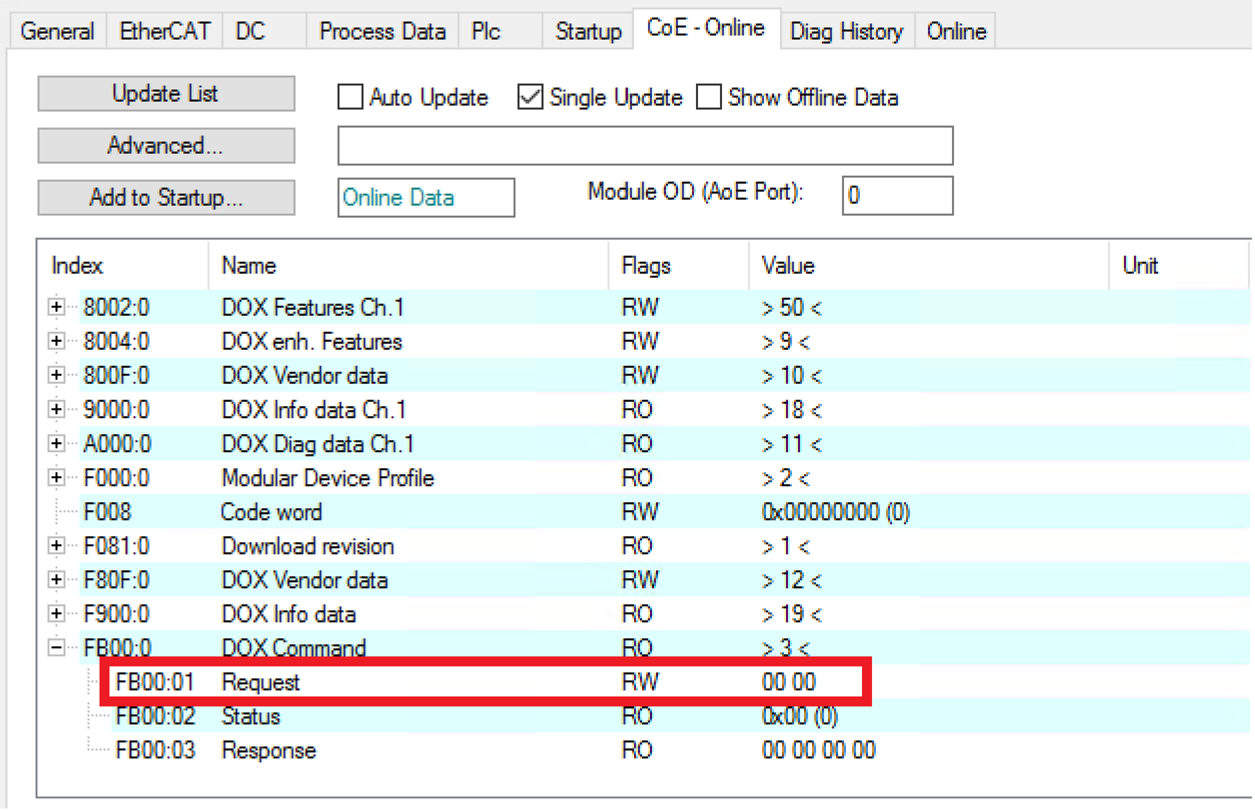
Especially in the current controlled pulse operation modes the use of the automatic determination of the output voltage is recommended, because possibly for the generation of high current pulses at the output a high pre-voltage is needed to generate steep edges. These may be higher than voltages specified in the LED data sheet.

> **NOTE**
>
> **TeachIn in pulsed mode at operating temperature**
>
> Automatic determination of the output voltage should be performed for all pulsing modes under operating conditions. Above all, the ambient temperature during teach-in is important and should correspond to the operating temperature.
>
> The reason for this is the decreasing forward resistance of the LED. If the resistance is reduced by rising temperatures at constant current at the LED output of the EL2596, less voltage drops across the LED than in the cold state. The voltage difference between the set output voltage and the voltage drop across the LED then drops internally in the terminal, causing the terminal to heat up. This can lead to the output being switched off due to overtemperature.
>
> If the teach-in is performed at operating temperature, internal losses can be avoided.
>
> For optimal parameterization, it is therefore recommended to perform two TeachIns. The first one in cold state to operate the LED first and to set it into operating conditions. The second TeachIn can then be performed under the achieved operating conditions.

The function described in this chapter can be used in any current-controlled operation mode except “Current Sink PWM” (“Current Control”, “Current Control Timestamp pulse”, “Current Control PLC pulse”, “Current Control Trigger pulse”).

The procedure is described below:

1. Specify the maximum output voltage in the CoE index 0x8000:04 [▶ 217] “Output voltage” (especially if overdriving is desired, this value should be greater than the nominal voltage)
2. Preset the set current in the process data under “DOX Current” → “Output Current”
3. Specify the command to store the output voltage in the CoE index 0xFB00:01 [▶ 222] “Request”
   - 0x0502: permanent storage of the value in the EEPROM
   - 0x0503: temporary storage of the value



☐ Auto Update ☑ Single Update ☐ Show Offline Data

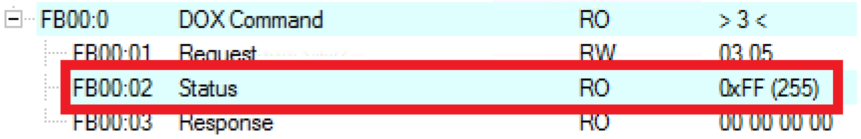
Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
+ 8002:0	DOX Features Ch.1	RW	> 50 <	
+ 8004:0	DOX enh. Features	RW	> 9 <	
+ 800F:0	DOX Vendor data	RW	> 10 <	
+ 9000:0	DOX Info data Ch.1	RO	> 18 <	
+ A000:0	DOX Diag data Ch.1	RO	> 11 <	
+ F000:0	Modular Device Profile	RO	> 2 <	
F008	Code word	RW	0x00000000 (0)	
+ F081:0	Download revision	RO	> 1 <	
+ F80F:0	DOX Vendor data	RW	> 12 <	
+ F900:0	DOX Info data	RO	> 19 <	
− FB00:0	DOX Command	RO	> 3 <	
FB00:01	Request	RW	00 00	
FB00:02	Status	RO	0x00 (0)	
FB00:03	Response	RO	00 00 00 00	

Fig. 186: Entering the command for storing the output voltage

4. Apply a rising edge on ”DOX Control” → “Control” → “Output” (the enable bit “DOX Control” → “Control” → “Enable” must remain 0). The output is then adjusted up to the set current and switched off when the set current is reached, then the output voltage is stored.

5. After successful determination, the status of the command in the CoE index 0xFB00:02 [▸ 222] “Status" is 0xFF and “DOX Control” → “Control” → “Switching operation counter” is incremented. The stored voltage is then entered in the CoE index 0x8000:04 [▸ 217] “Output voltage”.



FB00:0	DOX Command	RO	> 3 <
FB00:01	Request	RW	03 05
FB00:02	Status	RO	0xFF (255)
FB00:03	Response	RO	00 00 00 00

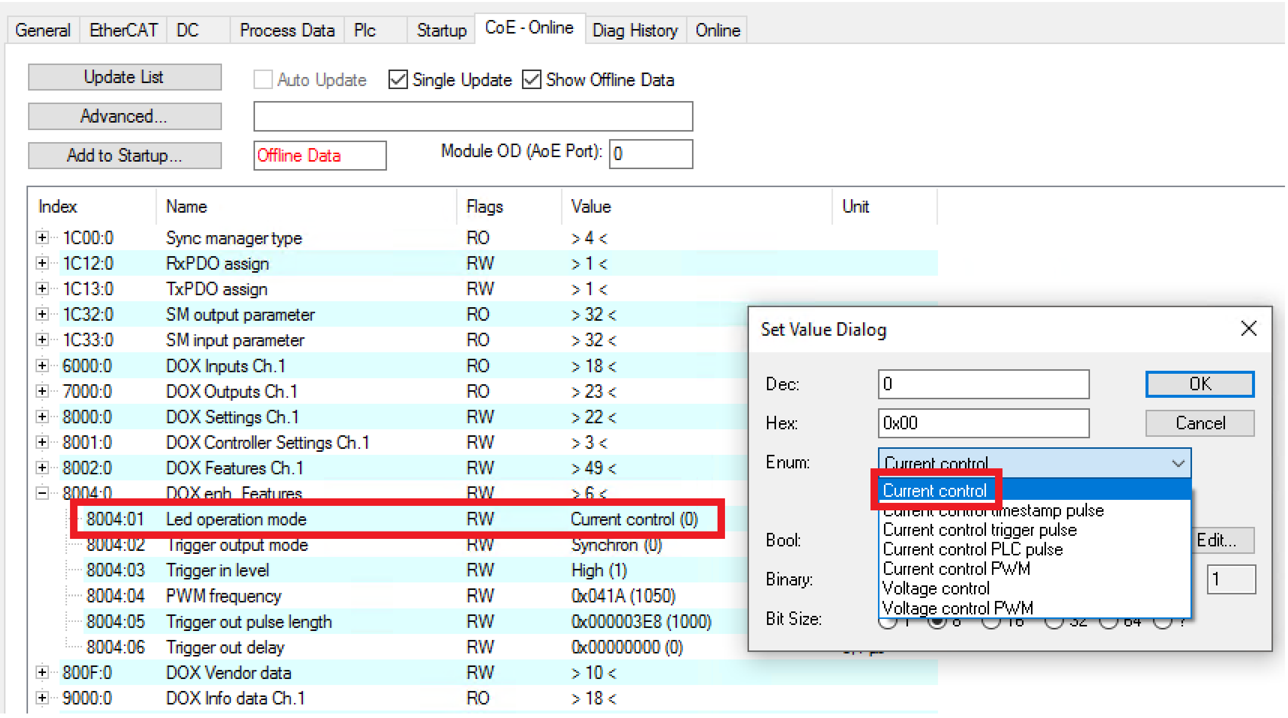
Fig. 187: Command status in case of successful storage of the output voltage

5.5.3 Storage of the characteristic curve of the LED as a CSV file

With the EL2596 it is also possible to determine the characteristic curve of a connected LED. Using the EtherCAT mailbox protocol FoE (File access over EtherCAT), the characteristic curve can then be saved as a CSV file (separator: semicolon, space). An upload of the LED characteristic curve is available if a "TeachIn", as described in the previous two chapters Saving the output voltage in operation [▸ 166] and Storage of the output voltage temporarily or in the EEPROM [▸ 168], has been carried out (points 1 - 6 can then be skipped). Alternatively, the entire method described below can be performed to save the characteristic curve.

The procedure is described below:

1. Set the operation mode in the CoE index 0x8004:01 [▸ 219] “Led operation mode” to “Current Control”



□ Auto Update ☑ Single Update ☑ Show Offline Data

Offline Data Module OD (AoE Port): 0

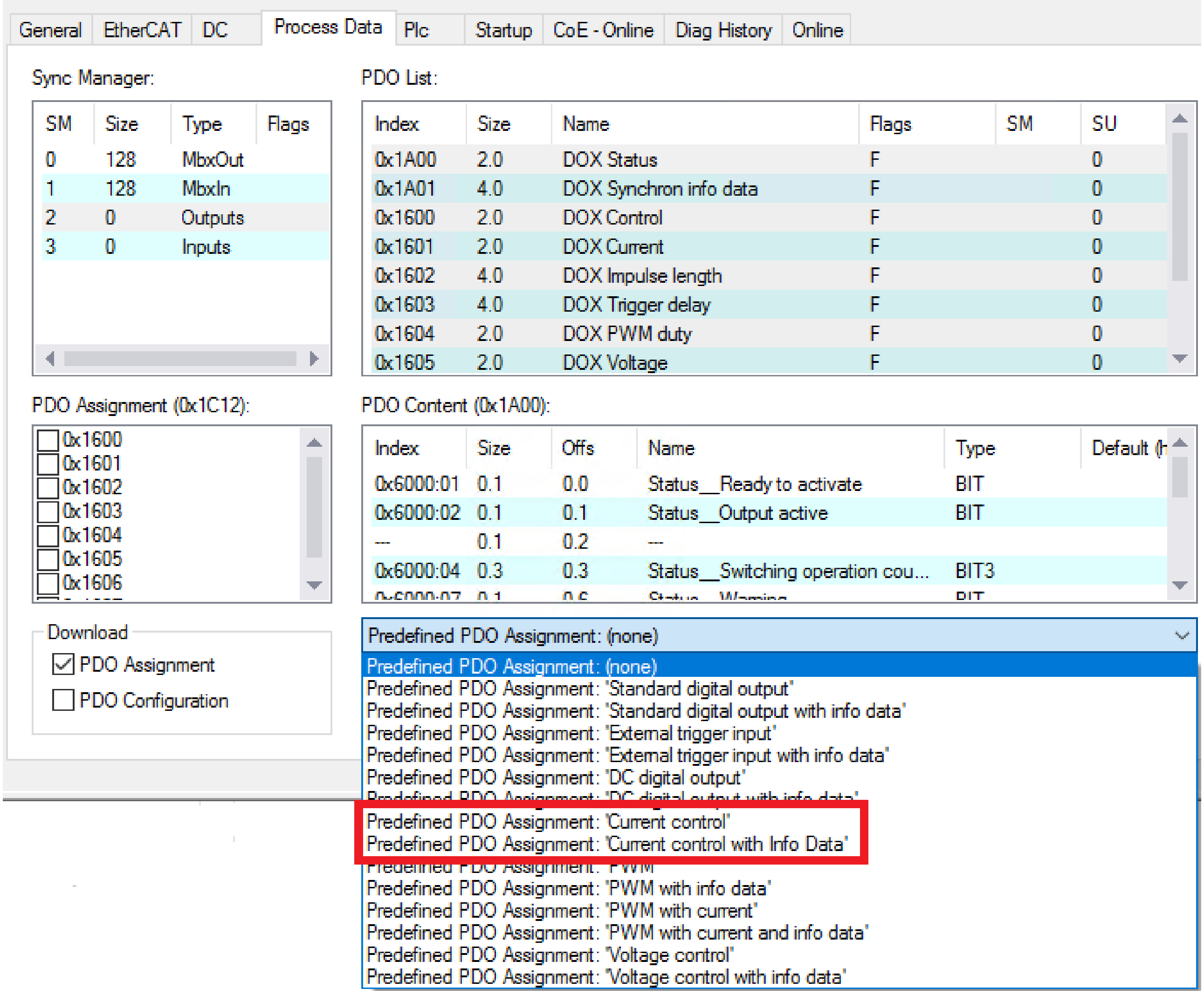
Index	Name	Flags	Value	Unit
1C00:0	Sync manager type	RO	> 4 <	
1C12:0	RxPDO assign	RW	> 1 <	
1C13:0	TxPDO assign	RW	> 1 <	
1C32:0	SM output parameter	RO	> 32 <	
1C33:0	SM input parameter	RO	> 32 <	
6000:0	DOX Inputs Ch.1	RO	> 18 <	
7000:0	DOX Outputs Ch.1	RO	> 23 <	
8000:0	DOX Settings Ch.1	RW	> 22 <	
8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
8002:0	DOX Features Ch.1	RW	> 49 <	
8004:0	DOX enh. Features	RW	> 6 <	
8004:01	Led operation mode	RW	Current control (0)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	
8004:06	Trigger out delay	RW	0x00000000 (0)	
800F:0	DOX Vendor data	RW	> 10 <	
9000:0	DOX Info data Ch.1	RO	> 18 <	

Dec: 0
Hex: 0x00
Enum: Current control
Bool:
Binary: 1
Bit Size: ○ ● 8 ○ 16 ○ 32 ○ 64 ○ ?

Current control
Current control timestamp pulse
Current control trigger pulse
Current control PLC pulse
Current control PWM
Voltage control
Voltage control PWM

Fig. 188: CoE object 0x8004:01 “Current Control”

2. Set the Predefined PDOs in the PDO Tab to “Current control (with info data)”



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	0	Outputs	
3	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F		0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F		0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0

PDO Assignment (0x1C12):

- □ 0x1600
- □ 0x1601
- □ 0x1602
- □ 0x1603
- □ 0x1604
- □ 0x1605
- □ 0x1606

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	

Download

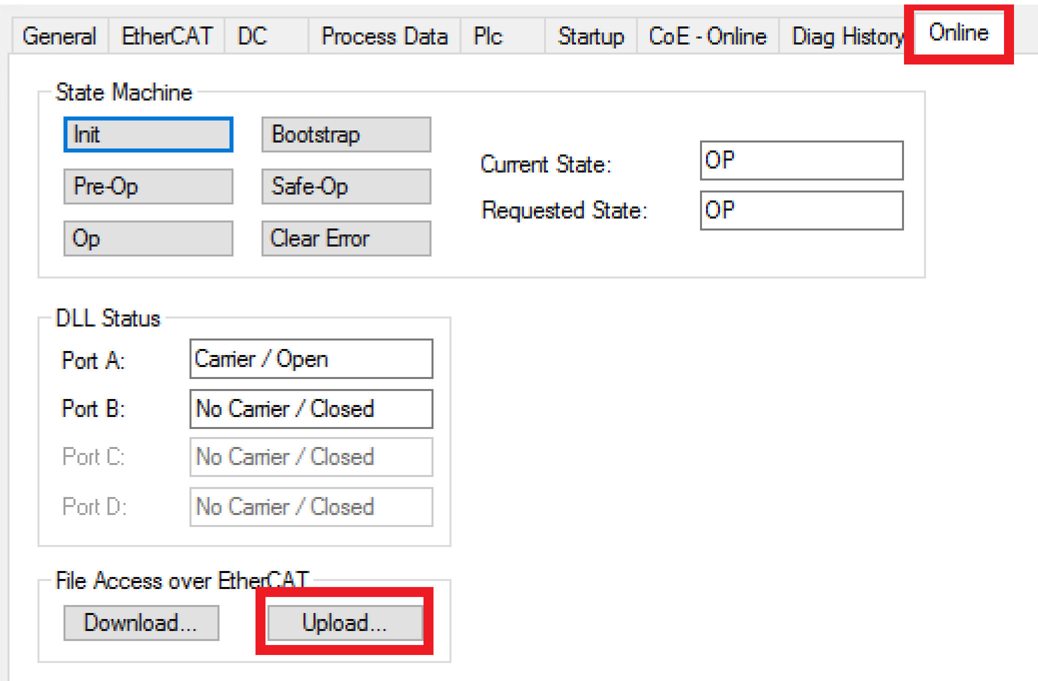
- ☑ PDO Assignment
- □ PDO Configuration

Predefined PDO Assignment: (none)

Predefined PDO Assignment: (none)
Predefined PDO Assignment: 'Standard digital output'
Predefined PDO Assignment: 'Standard digital output with info data'
Predefined PDO Assignment: 'External trigger input'
Predefined PDO Assignment: 'External trigger input with info data'
Predefined PDO Assignment: 'DC digital output'
Predefined PDO Assignment: 'DC digital output with info data'
Predefined PDO Assignment: 'Current control'
Predefined PDO Assignment: 'Current control with Info Data'
Predefined PDO Assignment: 'PWM'
Predefined PDO Assignment: 'PWM with info data'
Predefined PDO Assignment: 'PWM with current'
Predefined PDO Assignment: 'PWM with current and info data'
Predefined PDO Assignment: 'Voltage control'
Predefined PDO Assignment: 'Voltage control with info data'

Fig. 189: Select Predefined PDO “Current Control (with info data)”

3. Preset the set current in the process data under “DOX Current” → “Output Current”
4. Enable the LED output via “DOX Control” → “Control” → “Enable”
5. Switch on the LED output via “DOX Control” → “Control” → “Output”
6. Switch off the LED output via “DOX Control” → “Control” → “Output”
7. Start the upload via FoE Upload in the Online tab of the EL2596



Current State: OP

Requested State: OP

Port A: Carrier / Open

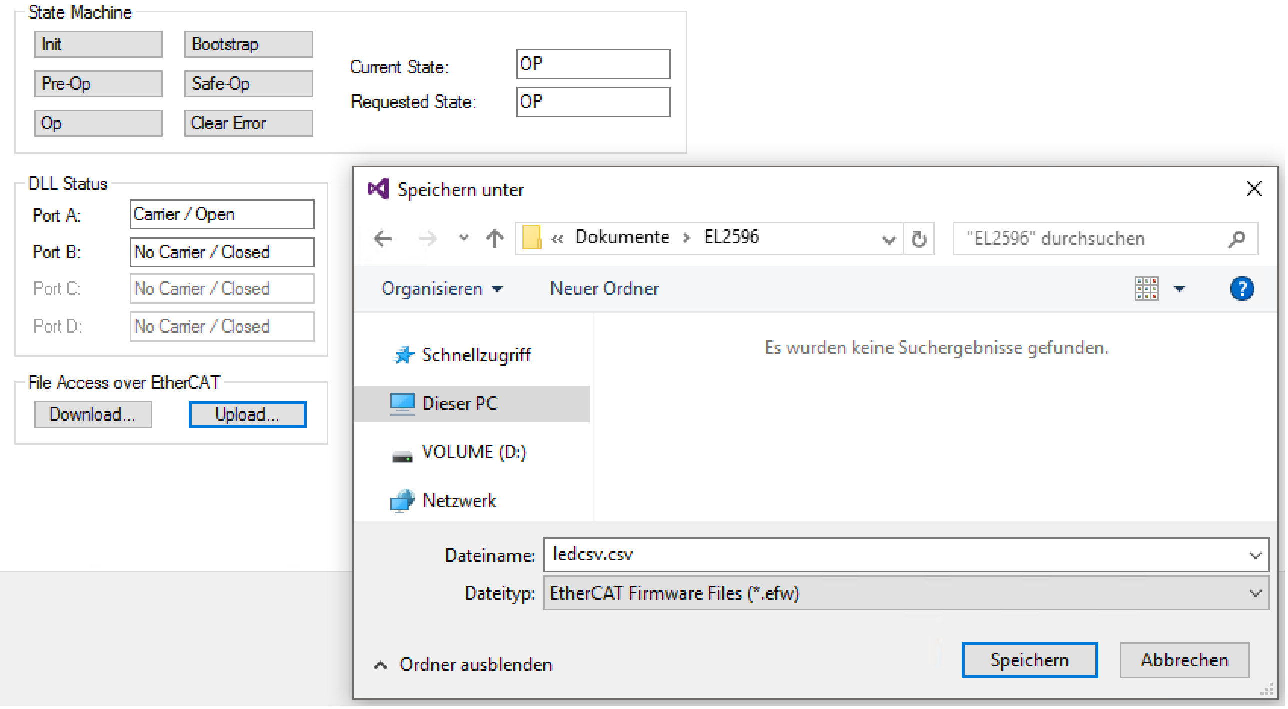
Port B: No Carrier / Closed

Port C: No Carrier / Closed

Port D: No Carrier / Closed

Fig. 190: Upload via FoE

8. Name the file “ledcsv.csv” and save it



Current State: OP

Requested State: OP

Port A: Carrier / Open

Port B: No Carrier / Closed

Port C: No Carrier / Closed

Port D: No Carrier / Closed

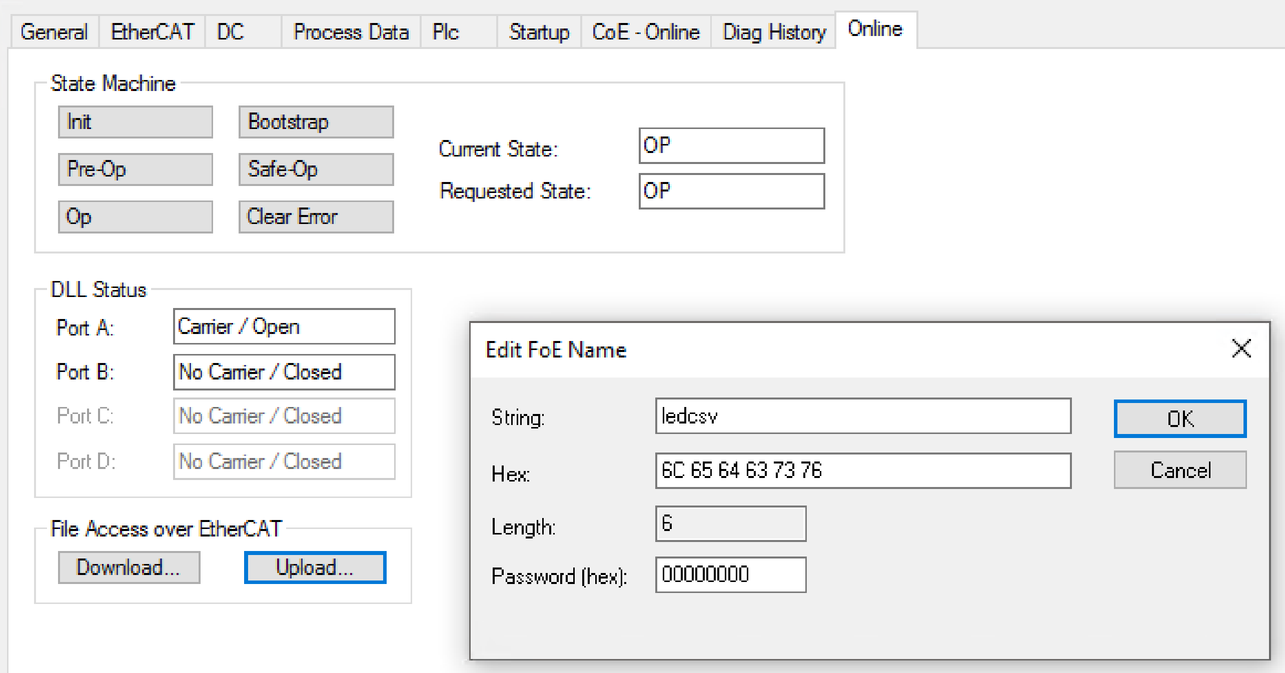
Es wurden keine Suchergebnisse gefunden.

Dateiname: ledcsv.csv

Dateityp: EtherCAT Firmware Files (*.efw)

Fig. 191: Create the file “ledcsv.csv”

9. Enter the FoE name “ledcsv” in the window that opens



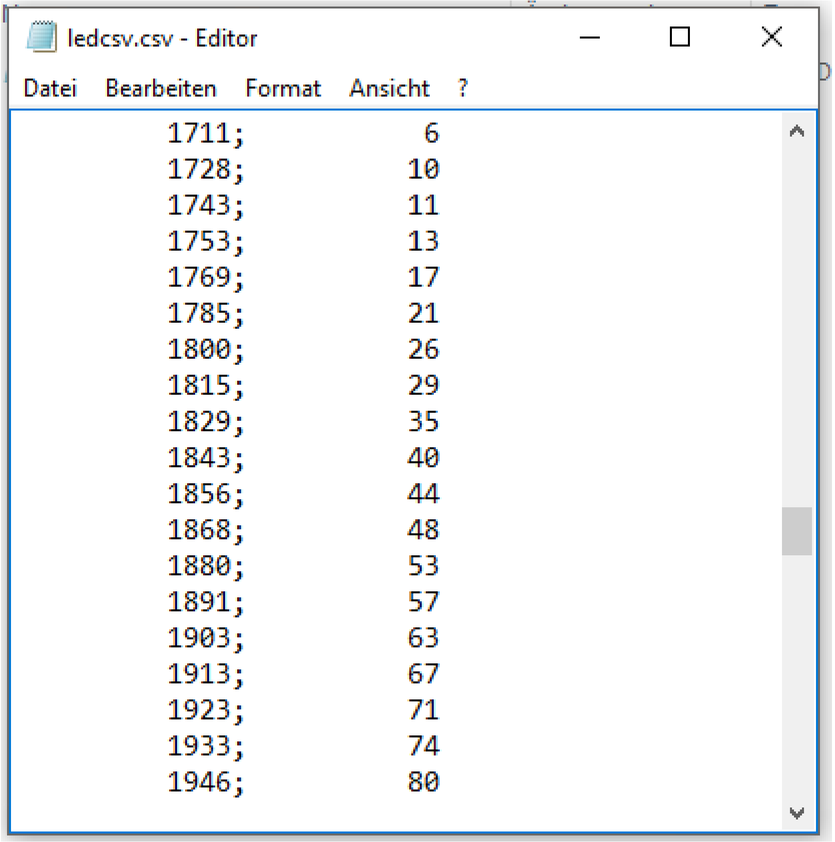
Current State: OP
Requested State: OP

Port A: Carrier / Open
Port B: No Carrier / Closed
Port C: No Carrier / Closed
Port D: No Carrier / Closed

String: ledcsv
Hex: 6C 65 64 63 73 76
Length: 6
Password (hex): 00000000

Fig. 192: Upload the “ledcsv” file

10. The CSV file is then saved in the selected path and can be opened with any editor. In the first column, the forward voltage is shown in 0.01 V. The second column shows the forward current in mA.



```
1711;        6
1728;       10
1743;       11
1753;       13
1769;       17
1785;       21
1800;       26
1815;       29
1829;       35
1843;       40
1856;       44
1868;       48
1880;       53
1891;       57
1903;       63
1913;       67
1923;       71
1933;       74
1946;       80
```

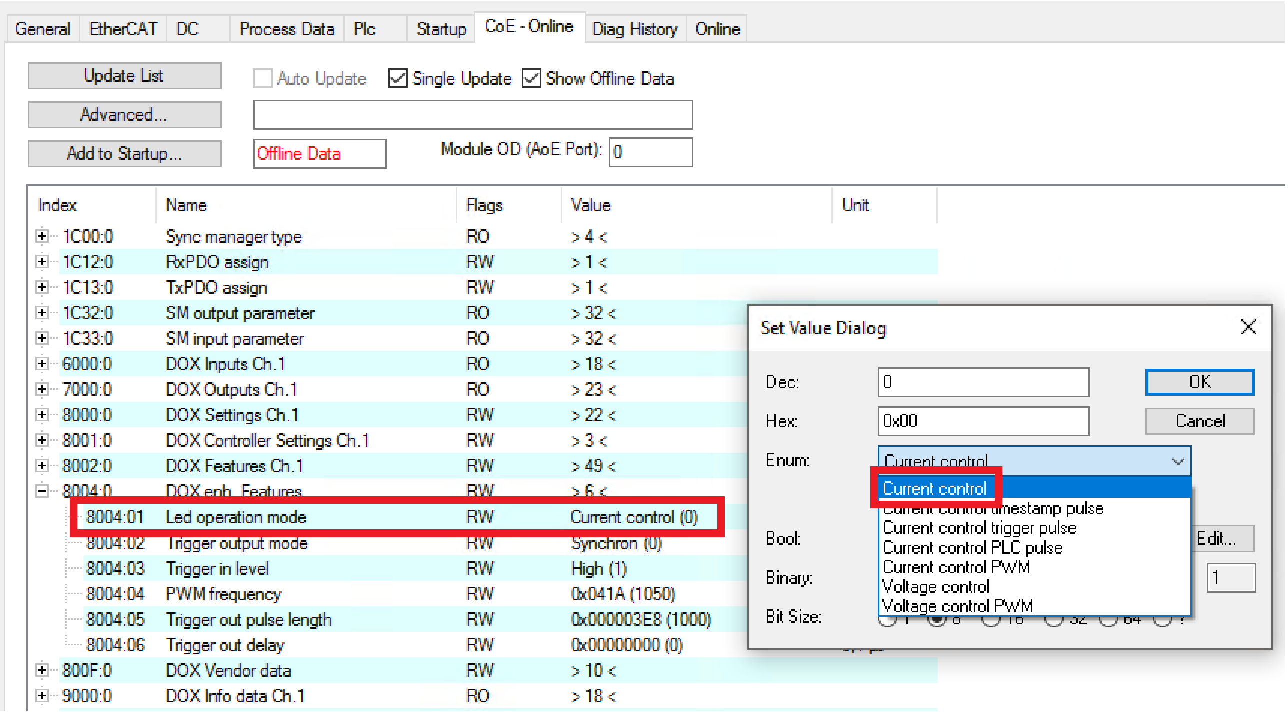
Fig. 193: ledcsv.csv file in the editor

5.5.4 Storage of the characteristic curve of the LED as a n HTML plot

With the EL2596 it is also possible to determine the characteristic curve of a connected LED. Using the EtherCAT mailbox protocol FoE (File access over EtherCAT), the characteristic curve can then be saved in HTML format. An upload of the LED characteristic curve is available if a "TeachIn", as described in the previous two chapters Storage of the output voltage in operation [▶ 166] and Storage of the output voltage temporarily or in the EEPROM [▶ 168], has been carried out (points 1 - 6 can then be skipped). Alternatively, the entire method described below can be performed to save the characteristic curve.

The procedure is described below:

1. Set the operation mode in the CoE index 0x8004:01 [▶ 219] “Led operation mode” to “Current Control”



Index	Name	Flags	Value	Unit
⊞ 1C00:0	Sync manager type	RO	> 4 <	
⊞ 1C12:0	RxPDO assign	RW	> 1 <	
⊞ 1C13:0	TxPDO assign	RW	> 1 <	
⊞ 1C32:0	SM output parameter	RO	> 32 <	
⊞ 1C33:0	SM input parameter	RO	> 32 <	
⊞ 6000:0	DOX Inputs Ch.1	RO	> 18 <	
⊞ 7000:0	DOX Outputs Ch.1	RO	> 23 <	
⊞ 8000:0	DOX Settings Ch.1	RW	> 22 <	
⊞ 8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
⊞ 8002:0	DOX Features Ch.1	RW	> 49 <	
⊟ 8004:0	DOX enh. Features	RW	> 6 <	
8004:01	Led operation mode	RW	Current control (0)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	
8004:06	Trigger out delay	RW	0x00000000 (0)	
⊞ 800F:0	DOX Vendor data	RW	> 10 <	
⊞ 9000:0	DOX Info data Ch.1	RO	> 18 <	

Set Value Dialog

Dec: 0

Hex: 0x00

Enum: Current control

- Current control
- Current control timestamp pulse
- Current control trigger pulse
- Current control PLC pulse
- Current control PWM
- Voltage control
- Voltage control PWM

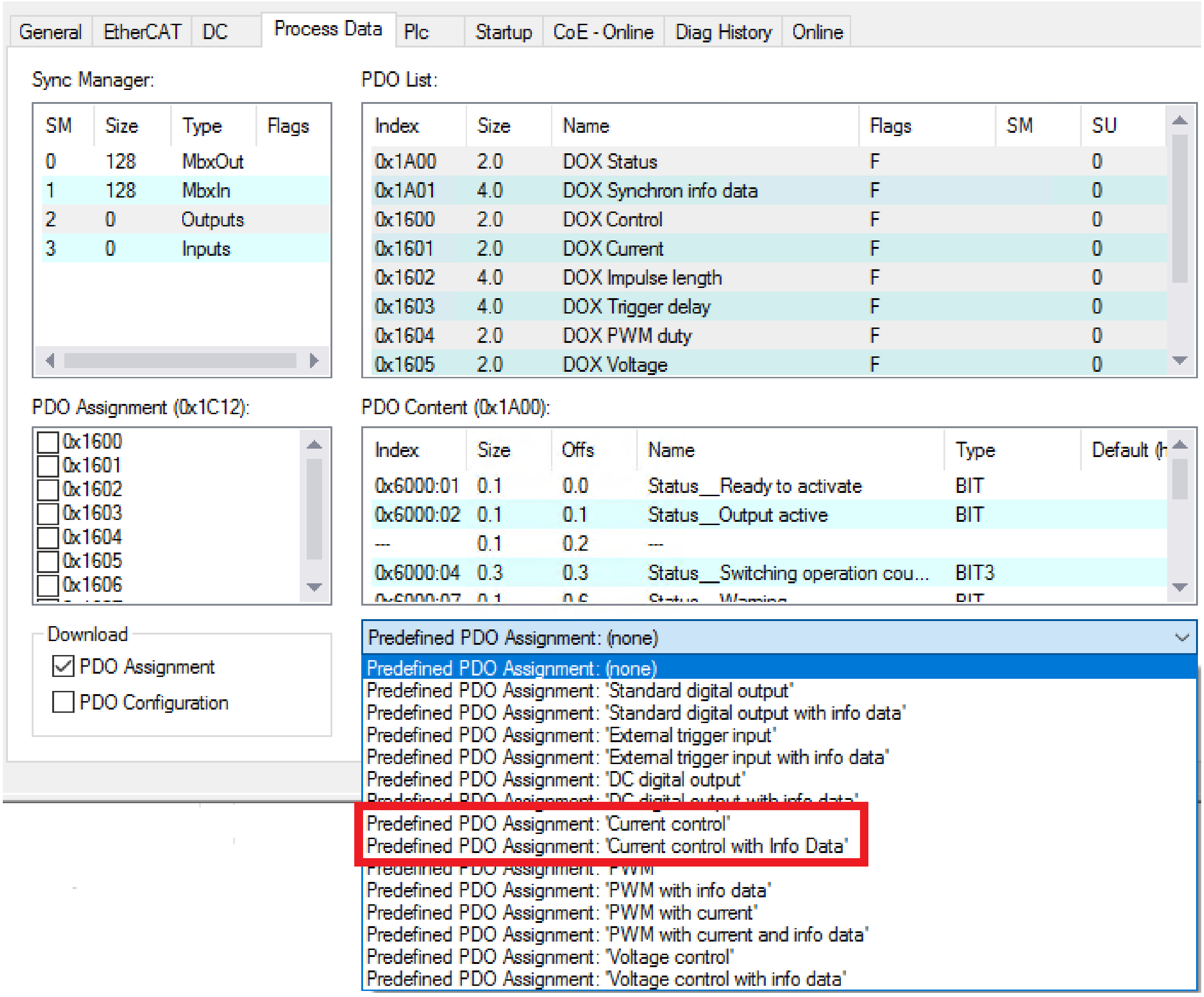
Bool:

Binary: 1

Bit Size: ○ 1 ● 8 ○ 16 ○ 32 ○ 64 ○ ?

Fig. 194: CoE index 0x8004:01 “Current Control”

2. Set the Predefined PDOs in the PDO Tab to “Current control (with info data)”



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	0	Outputs	
3	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F		0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F		0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0

PDO Assignment (0x1C12):

- □ 0x1600
- □ 0x1601
- □ 0x1602
- □ 0x1603
- □ 0x1604
- □ 0x1605
- □ 0x1606

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	

Download

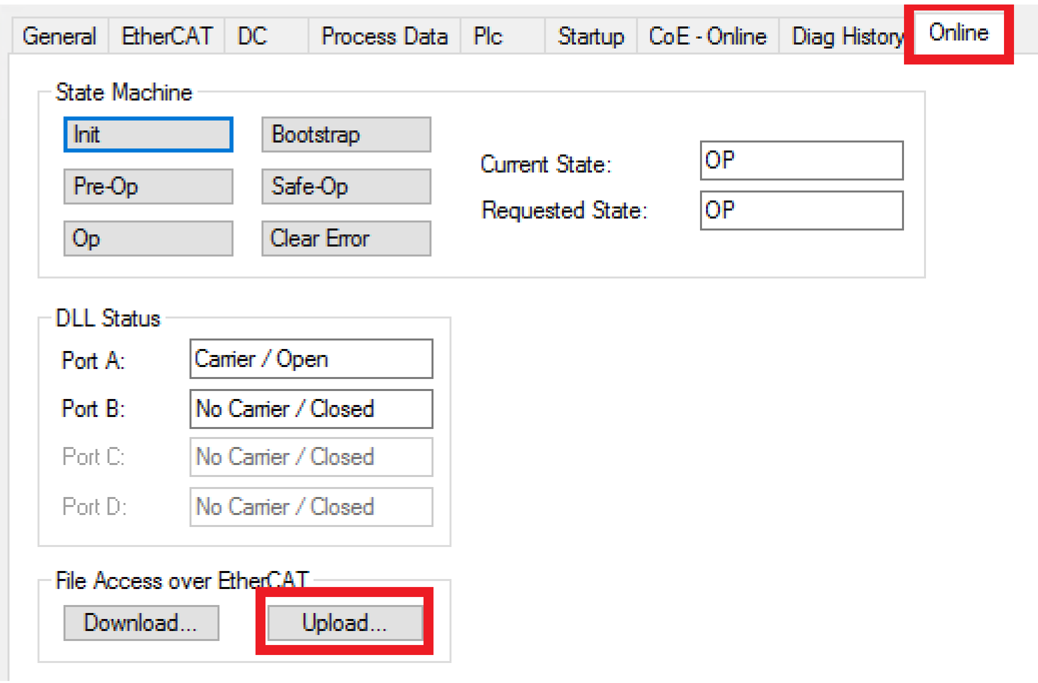
- ☑ PDO Assignment
- □ PDO Configuration

Predefined PDO Assignment: (none)

- Predefined PDO Assignment: (none)
- Predefined PDO Assignment: 'Standard digital output'
- Predefined PDO Assignment: 'Standard digital output with info data'
- Predefined PDO Assignment: 'External trigger input'
- Predefined PDO Assignment: 'External trigger input with info data'
- Predefined PDO Assignment: 'DC digital output'
- Predefined PDO Assignment: 'DC digital output with info data'
- Predefined PDO Assignment: 'Current control'
- Predefined PDO Assignment: 'Current control with Info Data'
- Predefined PDO Assignment: 'PWM'
- Predefined PDO Assignment: 'PWM with info data'
- Predefined PDO Assignment: 'PWM with current'
- Predefined PDO Assignment: 'PWM with current and info data'
- Predefined PDO Assignment: 'Voltage control'
- Predefined PDO Assignment: 'Voltage control with info data'

Fig. 195: Select Predefined PDO “Current Control (with info data)”

3. Preset the set current in the process data under “DOX Current” → “Output Current”
4. Enable the LED output via “DOX Control” → “Control” → “Enable”
5. Switch on the LED output via “DOX Control” → “Control” → “Output”
6. Switch off the LED output via “DOX Control” → “Control” → “Output”
7. Start the upload via FoE Upload in the Online tab of the EL2596



Current State: OP

Requested State: OP

Port A: Carrier / Open

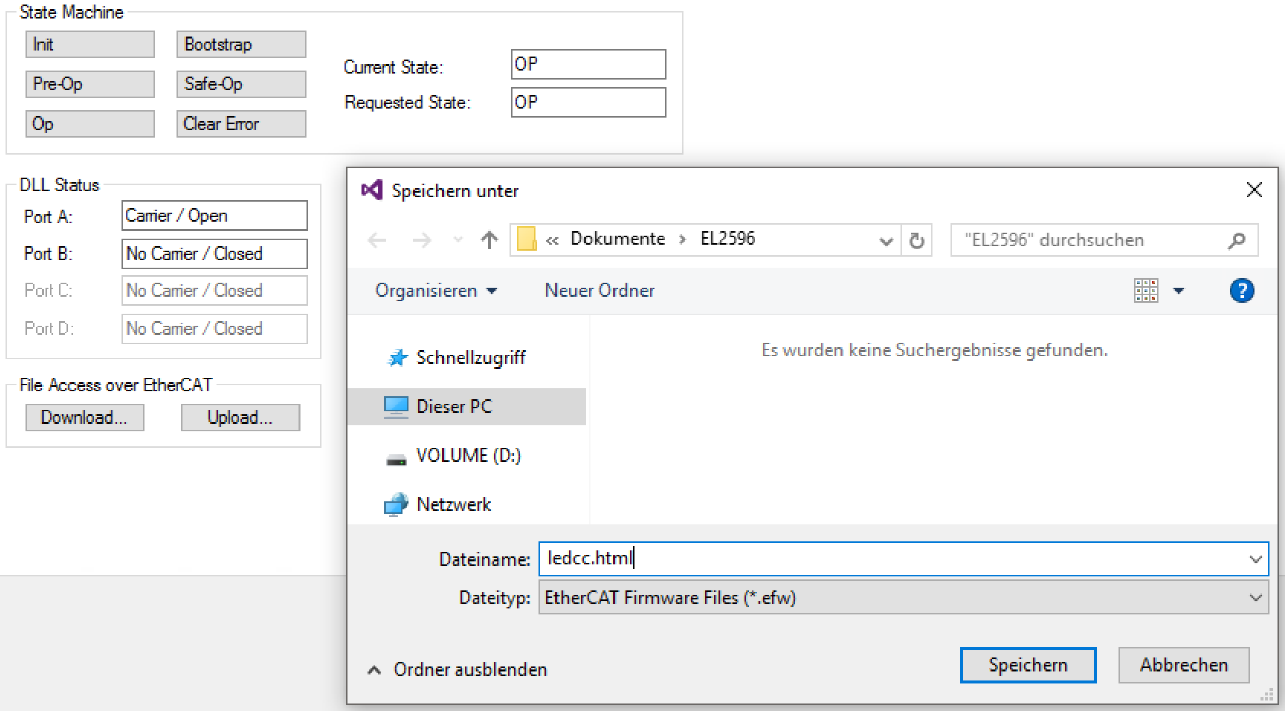
Port B: No Carrier / Closed

Port C: No Carrier / Closed

Port D: No Carrier / Closed

Fig. 196: Upload via FoE

8. Name the file “ledcc.html” and save it



Current State: OP

Requested State: OP

Port A: Carrier / Open

Port B: No Carrier / Closed

Port C: No Carrier / Closed

Port D: No Carrier / Closed

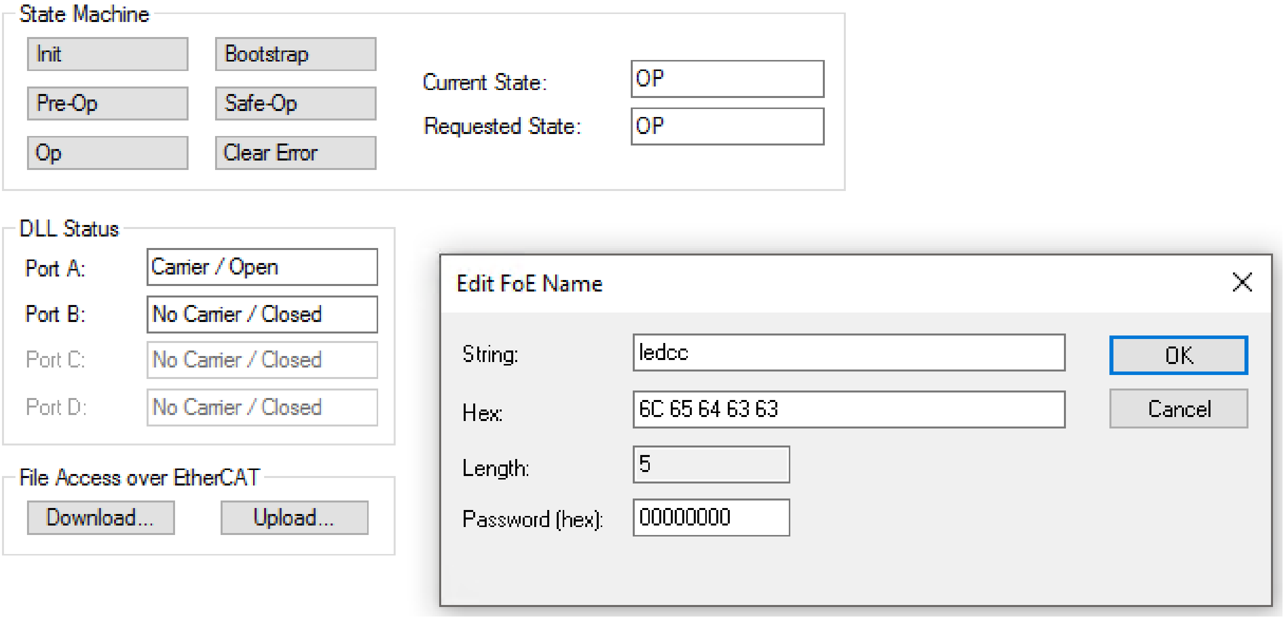
Es wurden keine Suchergebnisse gefunden.

Dateiname: ledcc.html

Dateityp: EtherCAT Firmware Files (*.efw)

Fig. 197: Create the file “ledcc.html”

9. Enter the FoE name “ledcc” in the window that opens



Current State: OP

Requested State: OP

Port A: Carrier / Open

Port B: No Carrier / Closed

Port C: No Carrier / Closed

Port D: No Carrier / Closed

String: ledcc

Hex: 6C 65 64 63 63

Length: 5

Password (hex): 00000000

Fig. 198: Upload the “ledcc” file

10. The HTML file is then saved in the selected path and can be opened with a browser. On the X-axis, the forward voltage is shown in 0.01 V. The Y axis shows the forward current in mA.

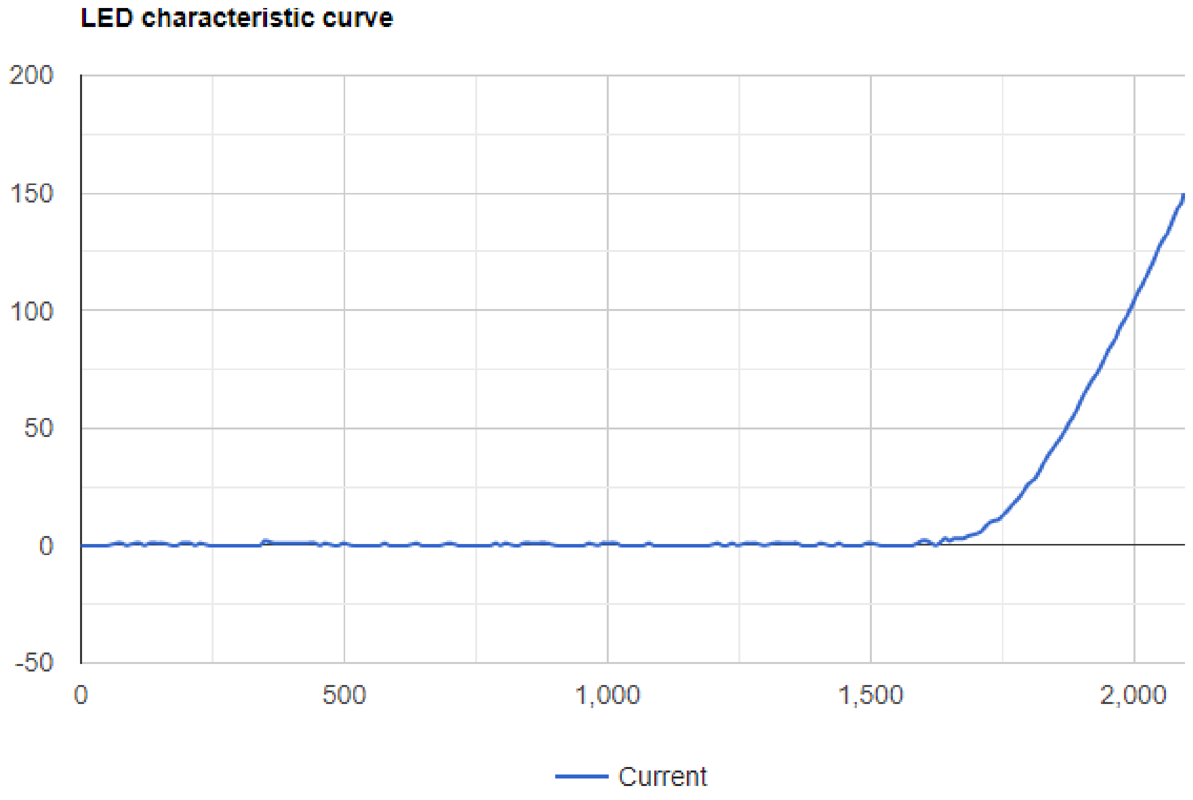


Fig. 199: ledcc.html file in the browser

5.6 Use of the trigger input

From FW04 the trigger input of the EL2596 has two functions. In the standard function, which is available from FW01, the trigger input in the “Current Control Trigger input” is used to output a pulse with a specified pulse duration, current and delay with an edge at the trigger input. In a further operation mode of the trigger input it can be used to enable the hardware. With a presettable signal (high/low) at the trigger input, the hardware of the EL2596 is then activated and outputs can be actuated.

5.6.1 Trigger input

By default, the trigger input is preset in “Trigger input” mode. In this setting, a signal at the trigger input only has a function in the “Current Control Trigger input” operation mode.

A pulse is then generated at the LED output on an edge at the trigger input. The parameters pulse duration, pulse delay and output current must be specified. A step-by-step description of the commissioning can be found in Current control trigger pulse [▸ 140].

5.6.2 Hardware enable

Available from FW04

The function described in this chapter for the use of the flash signal of a camera can only be used from FW04 onwards.

The trigger input also has an enable function for the hardware. If there is a preset signal (TRUE or FALSE) at the trigger input, the hardware of the EL2596 is activated and outputs can be actuated. One possible application is the “flash signals” of some cameras.

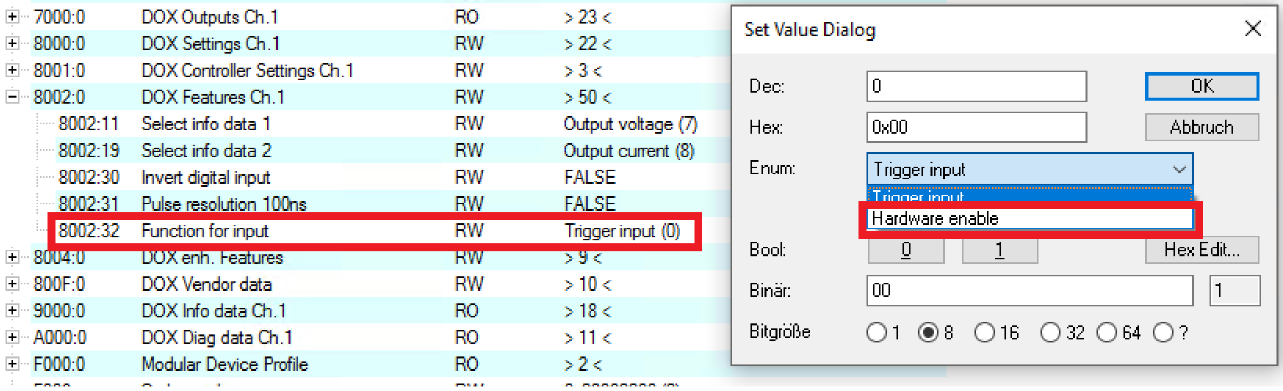
By default, some camera types have an output signal that is set to TRUE for the exposure time required. This signal from the camera is also referred to as a “flash signal”. If the flash signal of a camera is to be used as a trigger/enable input and the LED output is switched accordingly for as long as the trigger input of the flash signal of the camera is TRUE (LED output = flash signal), the hardware enable must be activated.

The “Hardware enable” function can be used in all operation modes. When using the “Hardware enable” in the operation mode “Current control trigger input” it should be noted, however, that the LED is then operated continuously as long as the signal is applied to the trigger input and not only for the specified pulse duration with the specified pulse delay. Consequently, the output is synchronous with the trigger input.

In addition to the LED output, the trigger output can also only be used in the trigger input mode “Hardware enable” if the hardware is activated by a signal at the trigger input.

To enable and use this function, the following settings of the EL2596xxxx are necessary:

1. Connect the flash signal to pins 5 (TrigIn+) and 13 (TrigIn-).
2. Select the desired operation mode in the CoE index 0x8004:01 [▸ 219].
3. Select “Hardware Enable” as the input function for the trigger input in the CoE index 0x8002:32 [▸ 218]



7000:0	DOX Outputs Ch.1	RO	> 23 <
8000:0	DOX Settings Ch.1	RW	> 22 <
8001:0	DOX Controller Settings Ch.1	RW	> 3 <
8002:0	DOX Features Ch.1	RW	> 50 <
8002:11	Select info data 1	RW	Output voltage (7)
8002:19	Select info data 2	RW	Output current (8)
8002:30	Invert digital input	RW	FALSE
8002:31	Pulse resolution 100ns	RW	FALSE
8002:32	Function for input	RW	Trigger input (0)
8004:0	DOX enh. Features	RW	> 9 <
800F:0	DOX Vendor data	RW	> 10 <
9000:0	DOX Info data Ch.1	RO	> 18 <
A000:0	DOX Diag data Ch.1	RO	> 11 <
F000:0	Modular Device Profile	RO	> 2 <

Set Value Dialog

Dec: 0

Hex: 0x00

Enum: Trigger input

Trigger input

Hardware enable

Binär: 00 1

Bitgröße ○ 1 ● 8 ○ 16 ○ 32 ○ 64 ○ ?

Fig. 200: Set the input mode “Hardware enable”

4. In the CoE index 0x8002:30 [▶ 218] “Invert digital input” set whether the hardware is to be enabled on a TRUE or FALSE signal
   - “Invert digital input” TRUE = enabled on FALSE at trigger input
   - “Invert digital input” FALSE = enabled on TRUE at trigger input
5. Set all mode-specific parameters in the PDOs as described in Setting the operation modes [▶ 131].
6. Activate the control under “DOX Control” → “Control” via the “Enable” bit.
7. Switch on the LED output under “DOX Control” → “Control” by activating the “Output” bit.
8. The LED output or trigger output is now TRUE as long as the signal is active at the trigger input.

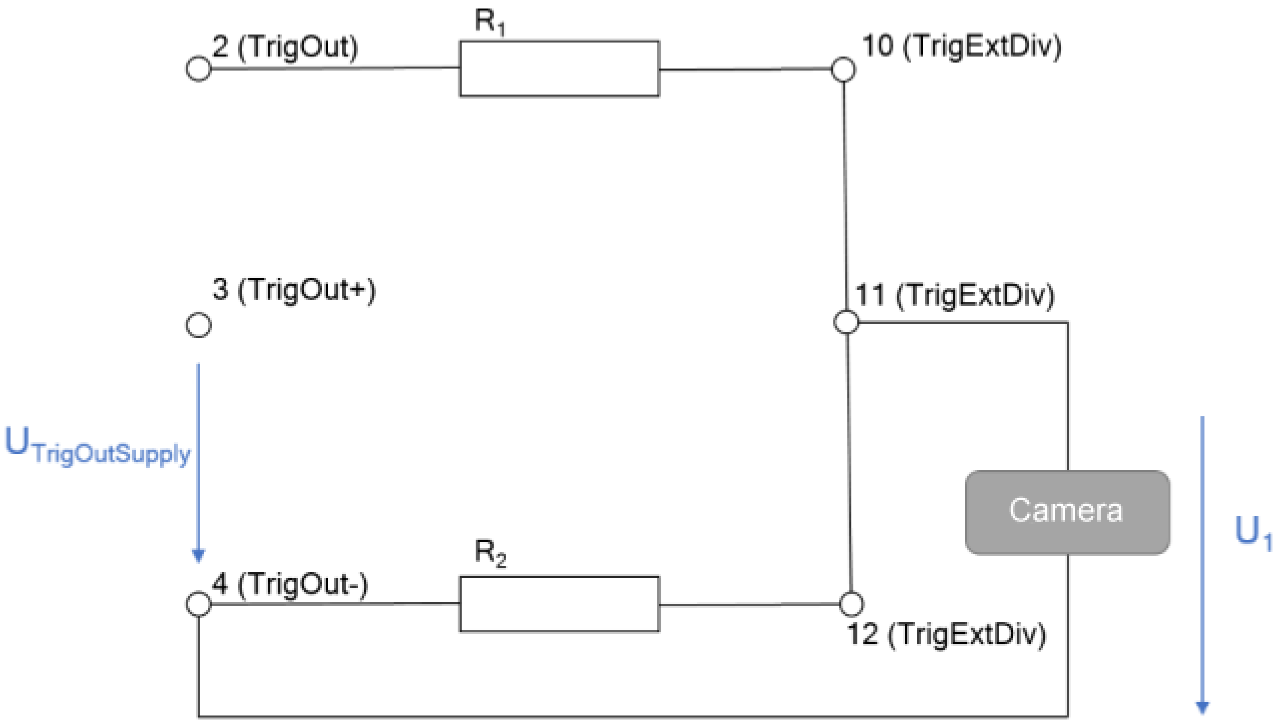
5.7 Use of the trigger output

5.7.1 Connection to the trigger output

The EL2596-xxxx has a trigger output for triggering a camera. This trigger output can be switched on in every pulse mode (Current control PLC pulse [▸ 145], Current control timestamp pulse [▸ 137] and Current control trigger pulse [▸ 140]).

The supply $U_{TrigOutSupply}$ for the trigger function between 10 V and 24 V is to be connected from outside to TrigOut+ and TrigOut-. The trigger input of the camera with its required voltage U_1 is connected

- to TrigOut if $U_1 = U_{TrigOutSupply}$ and can thus be switched directly through, or
- to TrigExtDiv (see diagram of external voltage dividers) if $U_1 < U_{TrigOutSupply}$ (e.g. for 5 V) and therefore has to be divided down



$$U_1 = \frac{R_2}{R_1 + R_2} \cdot U_{TrigOutSupply}$$

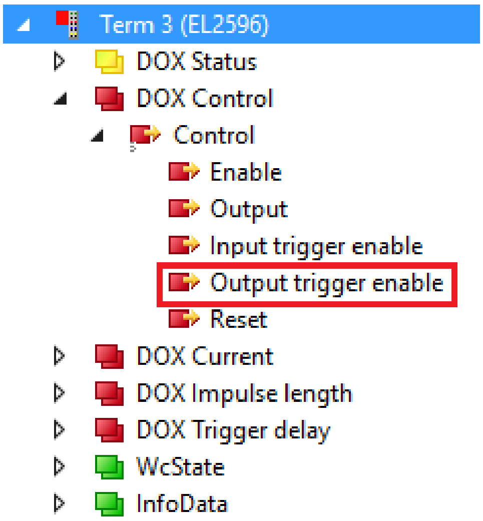
Fig. 201: Voltage divider for the trigger output

When dimensioning the resistors of the voltage divider R_1 and $R_{2,}$ make sure that $R_1 + R_2 > 2$ kΩ. With lower resistances, the power loss at the resistors would be high and would lead to strong heating.

5.7.2 Commissioning of the trigger output

The following steps are necessary in order to set the trigger output:

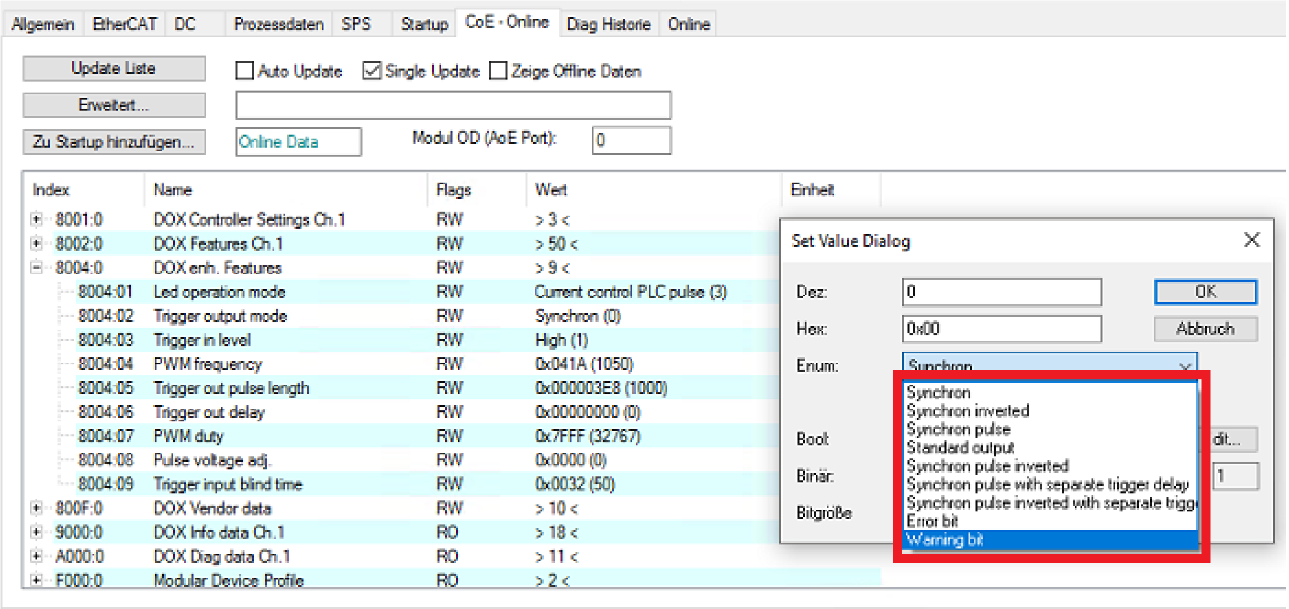
1. Set “DOX Control” → “Control” → “Output Trigger Enable” to 1



- Term 3 (EL2596)
  - DOX Status
  - DOX Control
    - Control
      - Enable
      - Output
      - Input trigger enable
      - Output trigger enable
      - Reset
  - DOX Current
  - DOX Impulse length
  - DOX Trigger delay
  - WcState
  - InfoData

Fig. 202: Activate the output trigger

2. Select the trigger mode in the CoE index 0x8004:02 [▶ 219]



□ Auto Update ☑ Single Update □ Zeige Offline Daten

Online Data

Modul OD (AoE Port): 0

Index	Name	Flags	Wert	Einheit
8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
8002:0	DOX Features Ch.1	RW	> 50 <	
8004:0	DOX enh. Features	RW	> 9 <	
8004:01	Led operation mode	RW	Current control PLC pulse (3)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	
8004:06	Trigger out delay	RW	0x00000000 (0)	
8004:07	PWM duty	RW	0x7FFF (32767)	
8004:08	Pulse voltage adj.	RW	0x0000 (0)	
8004:09	Trigger input blind time	RW	0x0032 (50)	
800F:0	DOX Vendor data	RW	> 10 <	
9000:0	DOX Info data Ch.1	RO	> 18 <	
A000:0	DOX Diag data Ch.1	RO	> 11 <	
F000:0	Modular Device Profile	RO	> 2 <	

Set Value Dialog

Dez: 0

Hex: 0x00

Enum: Synchron

Bool:

Binär: 1

Bitgröße:

- Synchron
- Synchron inverted
- Synchron pulse
- Standard output
- Synchron pulse inverted
- Synchron pulse with separate trigger delay
- Synchron pulse inverted with separate trigg
- Error bit
- Warning bit

Fig. 203: Select the trigger mode in the CoE index 0x8004:02

Element	Name	Description
0	Synchron	The trigger output is triggered when the LED output is TRUE.
1	Synchron inverted	The trigger output is triggered when the LED output is FALSE.
2	Synchron pulse	The trigger output is triggered if the LED output has a rising edge The pulse duration of the trigger output is set in 0.1 µs via the index 0x8004:05 [▶ 219].
3	Standard output	The trigger output can be used in the PLC as a digital output, independent of the LED output. This trigger mode is independent of the set LED output mode. The output can be set via the bit “Output trigger enable” (“DOX Control” → “Control” → “Output trigger enable”).
4	Synchron pulse inverted	The trigger output is triggered if the LED output has a falling edge. The pulse duration of the trigger output is set in 0.1 µs via the index 0x8004:05 [▶ 219].
5	Synchron pulse with separate trigger delay	If the LED output has a rising edge, the trigger output is triggered with a specified delay. The delay for the trigger output is specified in the unit µs in the PDO 0x1607. If it is not mapped, the delay must be specified in 0.1 µs in the index 0x8004:06 [▶ 219]. The pulse duration of the trigger output is set in 0.1 µs via the index 0x8004:05 [▶ 219].
6	Synchron pulse inverted with separate trigger delay	If the LED output has a falling edge, the trigger output is triggered with a specified delay. The delay for the trigger output is specified in the unit µs in the PDO 0x1607. If it is not mapped, the delay must be specified in 0.1 µs in the index 0x8004:06 [▶ 219]. The pulse duration of the trigger output is set in 0.1 µs via the index 0x8004:05 [▶ 219].
7	Error bit	The trigger output is TRUE as long as there is an error of the EL2596 (supply/ output voltage outside the value specified in the indices 0x8000:12, 0x8000:14, 0x800:16, [▶ 217] overtemperature, wire breakage, ...)
8	Warning bit	The trigger output is TRUE as long as there is a warning of the EL2596 (supply/ output voltage outside the value specified in the indices 0x8000:11, 0x8000:13, 0x800:15 [▶ 217], overtemperature, ...)

The timing behavior of the trigger modes 0, 1, 2, 4, 5 and 6 with the output bit and the trigger output is shown in the following figure.

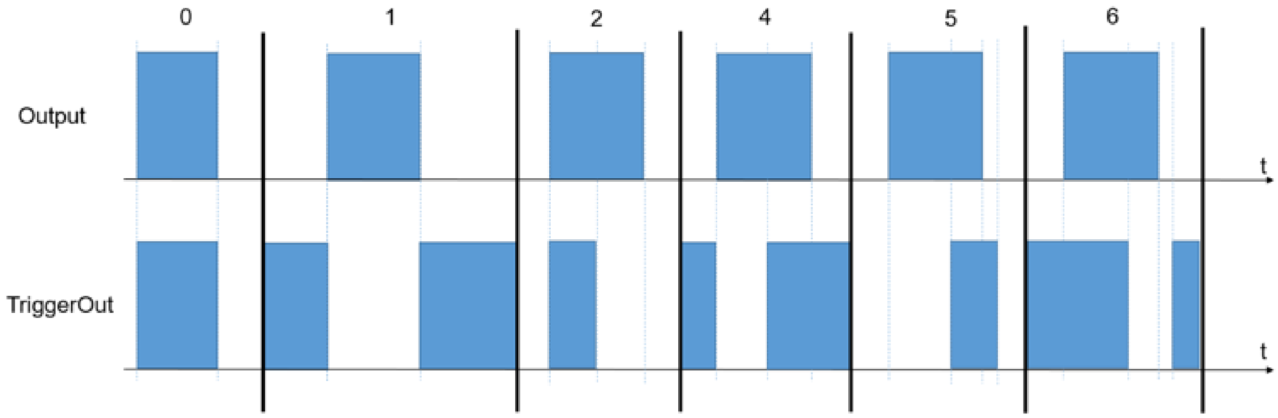


Fig. 204: Timing at the trigger output

In addition to the PDO values for the delay and pulse length, the EL2596 has a specific delay and a specific pulse length for the trigger output. These specific values in the CoE indices (0x8004:05 [▶ 219] “Trigger out pulse length”, 0x8004:06 [▶ 219] “Trigger out delay”) can only be used if “Trigger output mode” “Synchron pulse with separate trigger delay” (5) or “Synchron pulse inverted with separate trigger delay” (5) is enabled in the CoE index 0x8004:02 [▸ 219]. For all other set trigger modes in the CoE index 0x8004:02, [▸ 219] the values from the PDOs “Output length” and “Trigger delay” are used, or the delay value from the CoE index 0x8000:0B [▸ 217] is used, which are also valid for the LED output. In any case, the values from the PDOs or the delay value from the CoE index 0x8000:0B [▸ 217] are used for the LED output.

5.7.3 Using the trigger output with the trigger input

5.7.3.1 Commissioning instructions

The trigger output can be used as a response to the trigger input. An edge at the trigger input can thus generate a pulse at the trigger output. This function can be used in the “Current control trigger pulse” mode. In order to generate a reproducible behavior at the trigger and/or LED output via a signal at the trigger input, the signal at the trigger input should have a steep edge.

If the LED output and the trigger output are to be used, any trigger mode can be selected, except for the “Standard output” mode. If only the trigger output is to be used in response to the trigger input, only the trigger modes “Synchron”, “Synchron inverted”, “Synchron pulse with separate trigger delay” and “Synchron pulse inverted with separate trigger delay” can be used. With “Synchron” or “Synchron inverted” the pulse length and the delay of the pulse with respect to the trigger input are specified via the PDOs. In the trigger modes “Synchron pulse with separate trigger delay” and “Synchron pulse inverted with separate trigger delay”, the values for the pulse length and the delay of the pulse are specified at the trigger output.

The delay times can be specified via the CoE interface or in the PDOs. The delay times are then added together.

Possible from FW03

The function for using the trigger output in response to the trigger input described in this chapter can only be used from FW03 onwards.

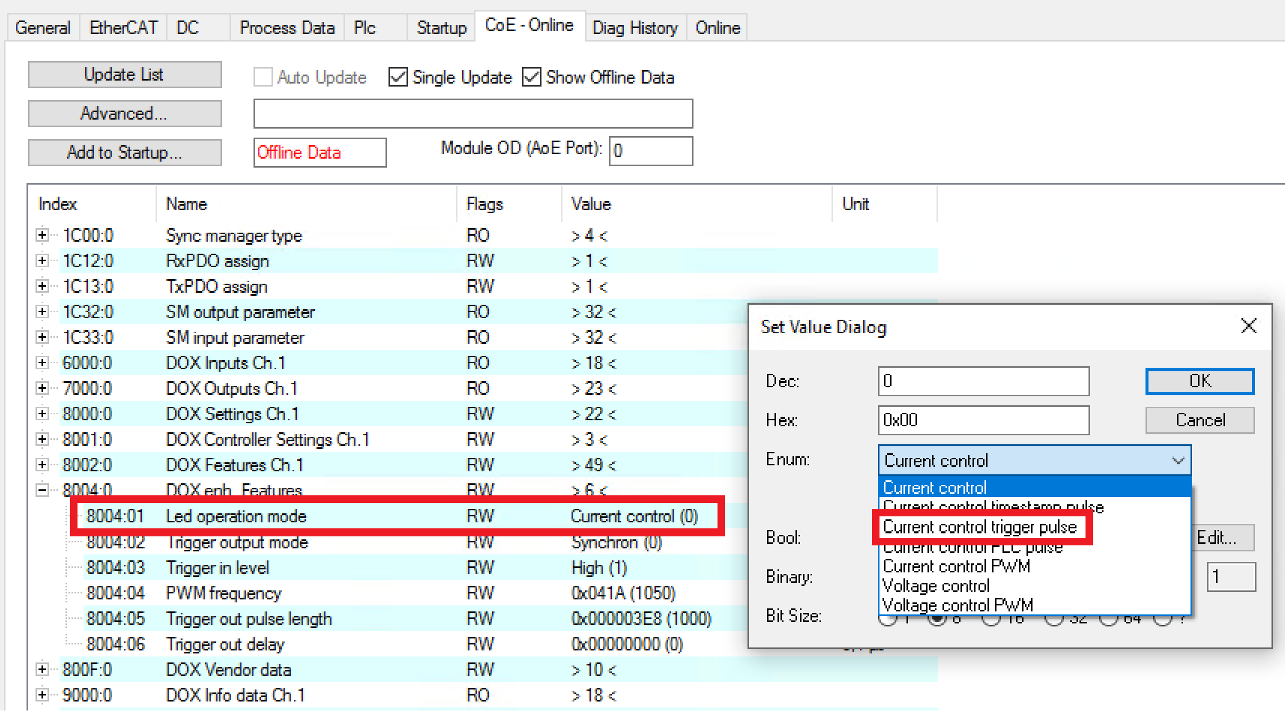
When using an external trigger, it should be noted that, based on internal terminal delays, the minimum delay between the trigger input and the signal output is approximately 2 µs. The offset of 2 µs must also be observed for all set delay values (regardless of whether they are set via PDO or CoE). The resulting delay is always

$$Real\ Delay = Set\ \ Delay + Offset$$

If the trigger output is used in response to the trigger input, it should be ensured that the signal at the trigger output has a jitter of <±1 µs.

5.7.3.2 Commissioning

1. Input voltage in index 0x8000:03 [▸ 217] “Supply voltage” in the unit 0.01 V
2. Set the operation mode in the CoE directory in index 0x8004:01 [▸ 219] to “Current Control trigger pulse”



□ Auto Update ☑ Single Update ☑ Show Offline Data

Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
1C00:0	Sync manager type	RO	> 4 <	
1C12:0	RxPDO assign	RW	> 1 <	
1C13:0	TxPDO assign	RW	> 1 <	
1C32:0	SM output parameter	RO	> 32 <	
1C33:0	SM input parameter	RO	> 32 <	
6000:0	DOX Inputs Ch.1	RO	> 18 <	
7000:0	DOX Outputs Ch.1	RO	> 23 <	
8000:0	DOX Settings Ch.1	RW	> 22 <	
8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
8002:0	DOX Features Ch.1	RW	> 49 <	
8004:0	DOX enh. Features	RW	> 6 <	
8004:01	Led operation mode	RW	Current control (0)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	
8004:06	Trigger out delay	RW	0x00000000 (0)	
800F:0	DOX Vendor data	RW	> 10 <	
9000:0	DOX Info data Ch.1	RO	> 18 <	

Set Value Dialog

Dec: 0

Hex: 0x00

Enum: Current control

- Current control
- Current control timestamp pulse
- Current control trigger pulse
- Current control PLC pulse
- Current control PWM
- Voltage control
- Voltage control PWM

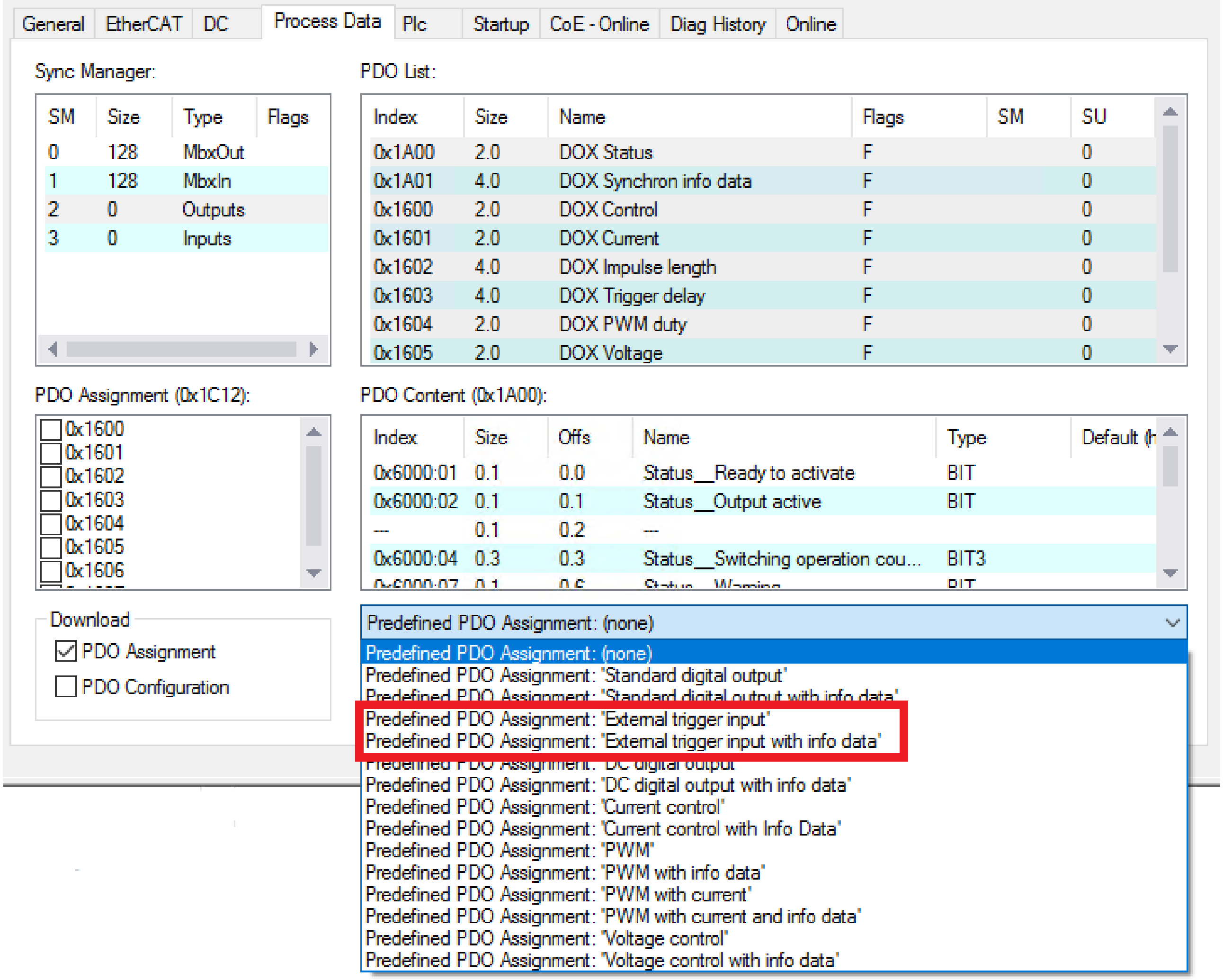
Bool:

Binary: 1

Bit Size: ○ 1 ● 8 ○ 16 ○ 32 ○ 64 ○ ?

Fig. 205: Operation mode setting “Current control trigger pulse”

3. Set Predefined PDO Assignments to "External trigger input (with info data)"



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	0	Outputs	
3	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F		0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F		0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0

PDO Assignment (0x1C12):

□ 0x1600
□ 0x1601
□ 0x1602
□ 0x1603
□ 0x1604
□ 0x1605
□ 0x1606

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	

Download

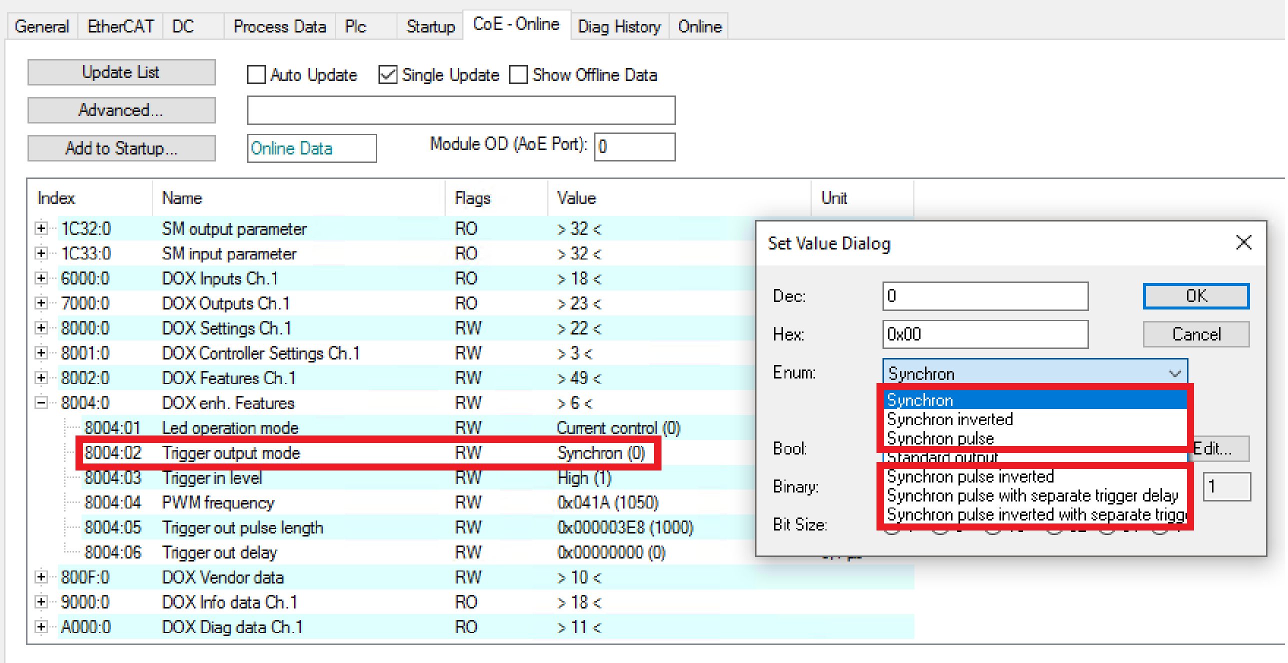
☑ PDO Assignment
□ PDO Configuration

Predefined PDO Assignment: (none)

- Predefined PDO Assignment: (none)
- Predefined PDO Assignment: 'Standard digital output'
- Predefined PDO Assignment: 'Standard digital output with info data'
- Predefined PDO Assignment: 'External trigger input'
- Predefined PDO Assignment: 'External trigger input with info data'
- Predefined PDO Assignment: 'DC digital output'
- Predefined PDO Assignment: 'DC digital output with info data'
- Predefined PDO Assignment: 'Current control'
- Predefined PDO Assignment: 'Current control with Info Data'
- Predefined PDO Assignment: 'PWM'
- Predefined PDO Assignment: 'PWM with info data'
- Predefined PDO Assignment: 'PWM with current'
- Predefined PDO Assignment: 'PWM with current and info data'
- Predefined PDO Assignment: 'Voltage control'
- Predefined PDO Assignment: 'Voltage control with info data'

Fig. 206: PDO setting “External trigger pulse (with info data)”

4. Set the parameters for the LED output (if used):
    - Specify the output current in the unit mA via “DOX Current” → “Output current”
    - Specify the pulse length in the unit µs via “DOX Impulse length” → “Impulse length”. The resolution of the time can be reduced from 1 µs to 100 ns using the CoE index 0x8002:31 [▶ 218] “Pulse resolution 100 ns”.
    - Specify the pulse delay in the unit µs via “DOX Trigger delay” → “Trigger delay”. The resolution of the time can be reduced from 1 µs to 100 ns using the CoE index 0x8002:31 [▶ 218] “Pulse resolution 100 ns”. Alternatively, the delay can be firmly specified in the unit 0.1 µs in the CoE index 0x8000:0B
5. Select the trigger mode for the trigger output in CoE index 0x8004:02 [▶ 219]
    - If the LED output and the trigger output are to be used, any trigger mode can be selected, except for the “Standard output” mode.
    - If only the trigger output is to be used in response to the trigger input, only the trigger modes “Synchron”, “Synchron inverted”, “Synchron pulse with separate trigger delay” and “Synchron pulse inverted with separate trigger delay” can be used.



□ Auto Update ☑ Single Update □ Show Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
1C32:0	SM output parameter	RO	> 32 <	
1C33:0	SM input parameter	RO	> 32 <	
6000:0	DOX Inputs Ch.1	RO	> 18 <	
7000:0	DOX Outputs Ch.1	RO	> 23 <	
8000:0	DOX Settings Ch.1	RW	> 22 <	
8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
8002:0	DOX Features Ch.1	RW	> 49 <	
8004:0	DOX enh. Features	RW	> 6 <	
8004:01	Led operation mode	RW	Current control (0)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	
8004:06	Trigger out delay	RW	0x00000000 (0)	
800F:0	DOX Vendor data	RW	> 10 <	
9000:0	DOX Info data Ch.1	RO	> 18 <	
A000:0	DOX Diag data Ch.1	RO	> 11 <	

Set Value Dialog

Dec: 0

Hex: 0x00

Enum: Synchron

- Synchron
- Synchron inverted
- Synchron pulse
- Standard output
- Synchron pulse inverted
- Synchron pulse with separate trigger delay
- Synchron pulse inverted with separate trigg

Bool:

Binary: 1

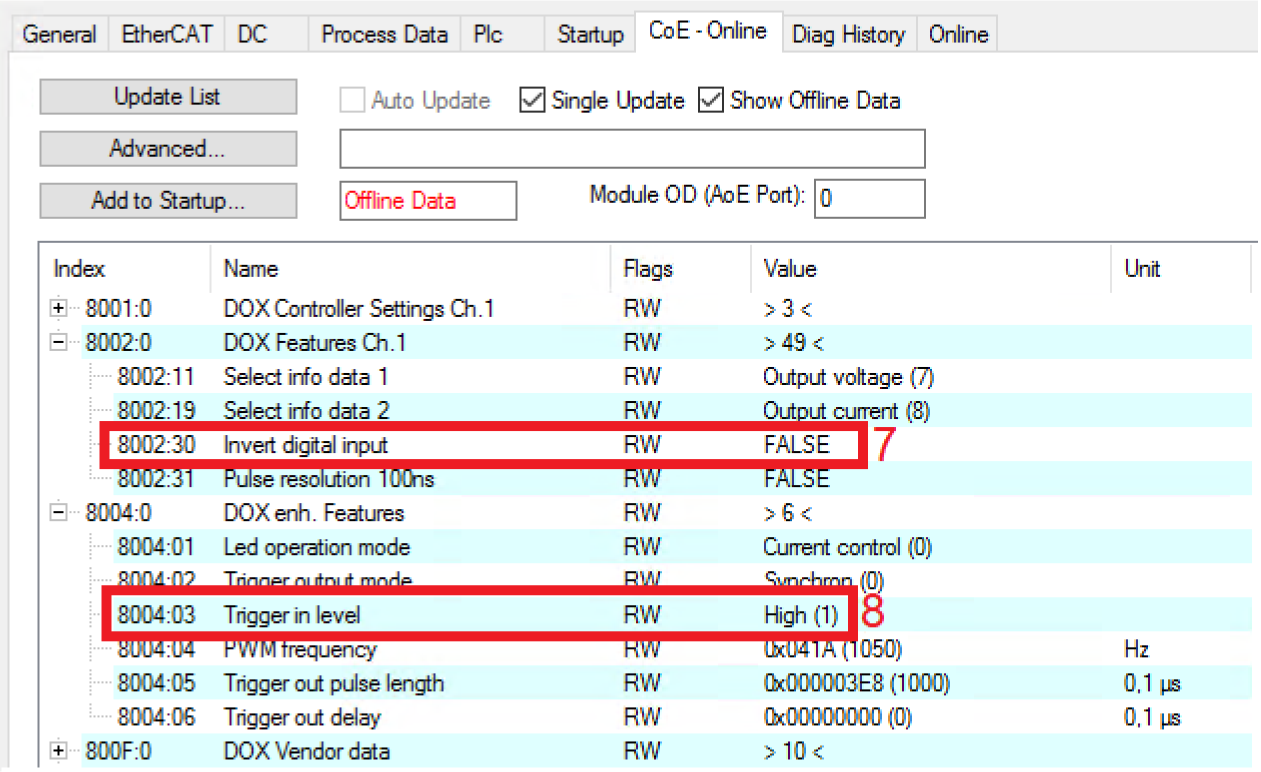
Bit Size:

Fig. 207: Setting the trigger mode

6. Setting the specific settings for the trigger mode

Trigger mode	Description	Settings
Synchron	Trigger output TRUE if LED output is TRUE	The PDO values for the LED output are adopted for delay and pulse length ◦ **Pulse length:** “DOX Impulse length” → “Impulse length” [µs] ◦ **Delay:** “DOX Trigger delay” → “Trigger delay” [µs] or CoE index 0x8000:0B [▶ 217] in [0.1 µs]
Synchron inverted	Trigger output TRUE if LED output is FALSE	
Synchron pulse	Trigger output has at the same time a rising edge like the LED output	The PDO value for the LED output is adopted for the delay ◦ **Delay:** “DOX Trigger delay” → “Trigger delay” [µs] or CoE index 0x8000:0B [▶ 217] in [0.1 µs] The separate pulse length for the trigger output is set in the CoE ◦ **Pulse length:** 0x8004:05 [▶ 219] “Trigger out pulse length” [0.1 µs]
Synchron pulse inverted	Trigger output has a falling edge when the LED output has a rising edge	
Synchron pulse with separate trigger delay	Trigger output has a rising edge with a specified delay when the LED output has a rising edge	The separate delay and pulse length for the trigger output are set in the CoE ◦ **Delay:** 0x8004:06 [▶ 219] “Trigger out delay” [0.1 µs] or PDO 0x1607 [µs] “Trigger out delay”, which must be mapped additionally ◦ **Pulse length:** 0x8004:05 [▶ 219] “Trigger out pulse length” [0.1 µs]
Synchron pulse inverted with separate trigger delay	Trigger output has a falling edge with a specified delay when the LED output has a rising edge	

7. In the CoE index 0x8002:30 [▶ 218], set whether the LED pulse should be generated by a rising or falling edge at the trigger input.
8. In the CoE index 0x8004:03 [▶ 219], specify the level at the trigger input at which the LED output should be switched (high ≥ 10 V, low ≥ 5 V).



□ Auto Update ☑ Single Update ☑ Show Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
+ 8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
− 8002:0	DOX Features Ch.1	RW	> 49 <	
8002:11	Select info data 1	RW	Output voltage (7)	
8002:19	Select info data 2	RW	Output current (8)	
8002:30	Invert digital input	RW	FALSE	
8002:31	Pulse resolution 100ns	RW	FALSE	
− 8004:0	DOX enh. Features	RW	> 6 <	
8004:01	Led operation mode	RW	Current control (0)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	Hz
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	0,1 µs
8004:06	Trigger out delay	RW	0x00000000 (0)	0,1 µs
+ 800F:0	DOX Vendor data	RW	> 10 <	

Fig. 208: Settings in the CoE objects for the combination of trigger input and output

9. Connect the trigger signal to the TrigIn+ (5) and TrigIn- (13) connections
10. Activate the trigger input under “DOX Control” → “Control” → “Input Trigger Enable”

11. Connect the device to be controlled (e.g. camera) to the TrigOut trigger output (2)
    - The external device to be controlled via the trigger output can be supplied with power via TrigOut+ (3) and TrigOut- (4).
    - An external voltage divider can be connected to the TrigExtDiv contacts (10, 11, 12) to reduce the supply voltage for the device in use (see Connection of trigger output [▶ 180])
12. Activate the trigger output under “DOX Control” → “Control” → “Output Trigger Enable”
13. Check under “DOX Status” → “Status” whether the “Ready to activate” bit is 1.
14. Activate the control under “DOX Control” → “Control” via the “Enable” bit.
15. Switch on the output under “DOX Control” → “Control” by activating the “Output” bit

5.7.3.3 Time behavior

The trigger output can also be used in combination with the trigger input. A signal at the trigger input can then enable both the LED output and the trigger output. There are various ways in which trigger inputs (TriggerIn), LED output (LED Out) and trigger output (TriggerOut) are combined with one another. Four possibilities of temporal behavior are shown in the following illustration. The various options are described below.

In addition to an edge at the trigger input, the TriggerIn signal can also correspond to a signal from the PLC (PLC Pulse) or a specified start time of the distributed clocks (DC Pulse).

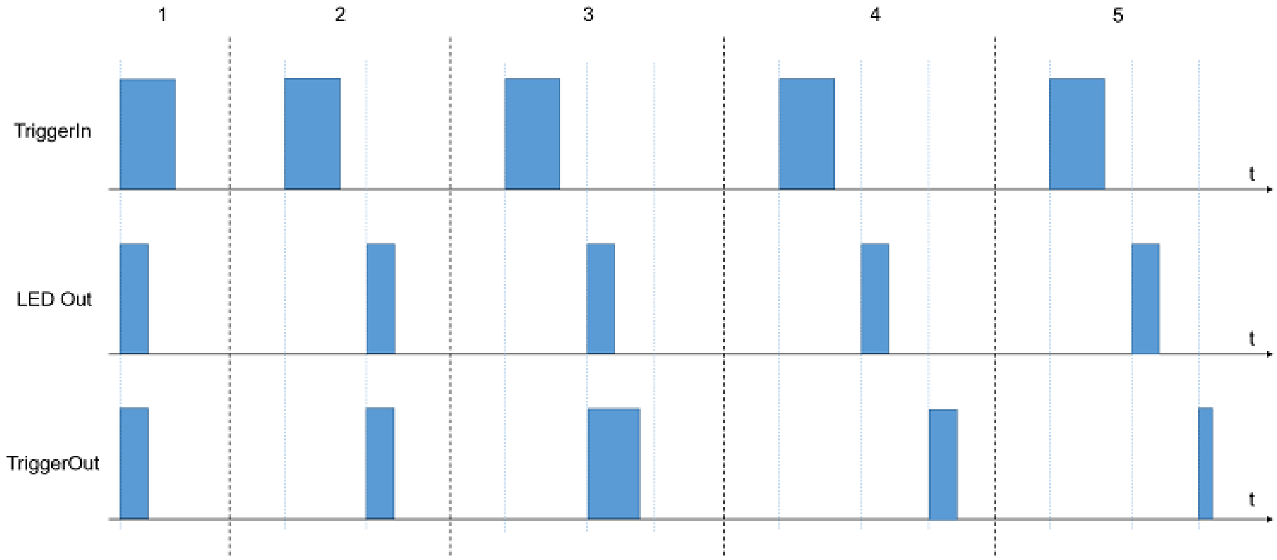


Fig. 209: Time behavior when combining trigger input and trigger output

1. With a rising edge at the trigger input, the LED output and the trigger output switch synchronously to the trigger input. Both outputs have the same pulse length.
2. The LED output and the trigger output have a uniform delay with respect to the rising edge at the trigger input. Both outputs (LED Out, TriggerOut) switch synchronously with the same pulse length.
3. The LED output and the trigger output have a uniform delay with respect to the rising edge at the trigger input. Both outputs (LED Out, TriggerOut) switch synchronously with different pulse lengths.
4. The LED output and the trigger output have delays with respect to the rising edge at the trigger input. However, the delays are configured differently for both outputs, so that the outputs do not switch synchronously. Both outputs have the same pulse length.
5. The LED output and the trigger output have delays with respect to the rising edge at the trigger input. However, the delays are configured differently for both outputs, so that the outputs do not switch synchronously. The outputs have different pulse lengths.

In the parameterization, it is therefore possible to set the different delays for the LED output, as well as the trigger output, as well as to set the pulse lengths of the two output signals independently of each other.

However, it should be noted that the delay between the trigger input and the trigger output cannot be zero due to internal factors in the terminal. There is always a delay of about 2 µs between the trigger input and the trigger output. This basic delay of 2 µs acts as an offset to the set delay. The offset of 2 µs must also be observed for all set delay values (regardless of whether they are set via PDO or CoE). The resulting delay is always

$$Real\ Delay = Set\ \ Delay + Offset$$

The basic delay described is visible in the following figure. In addition, the output at the trigger output has a jitter of ±500 ns, with the maximum of +500 ns occurring only rarely.

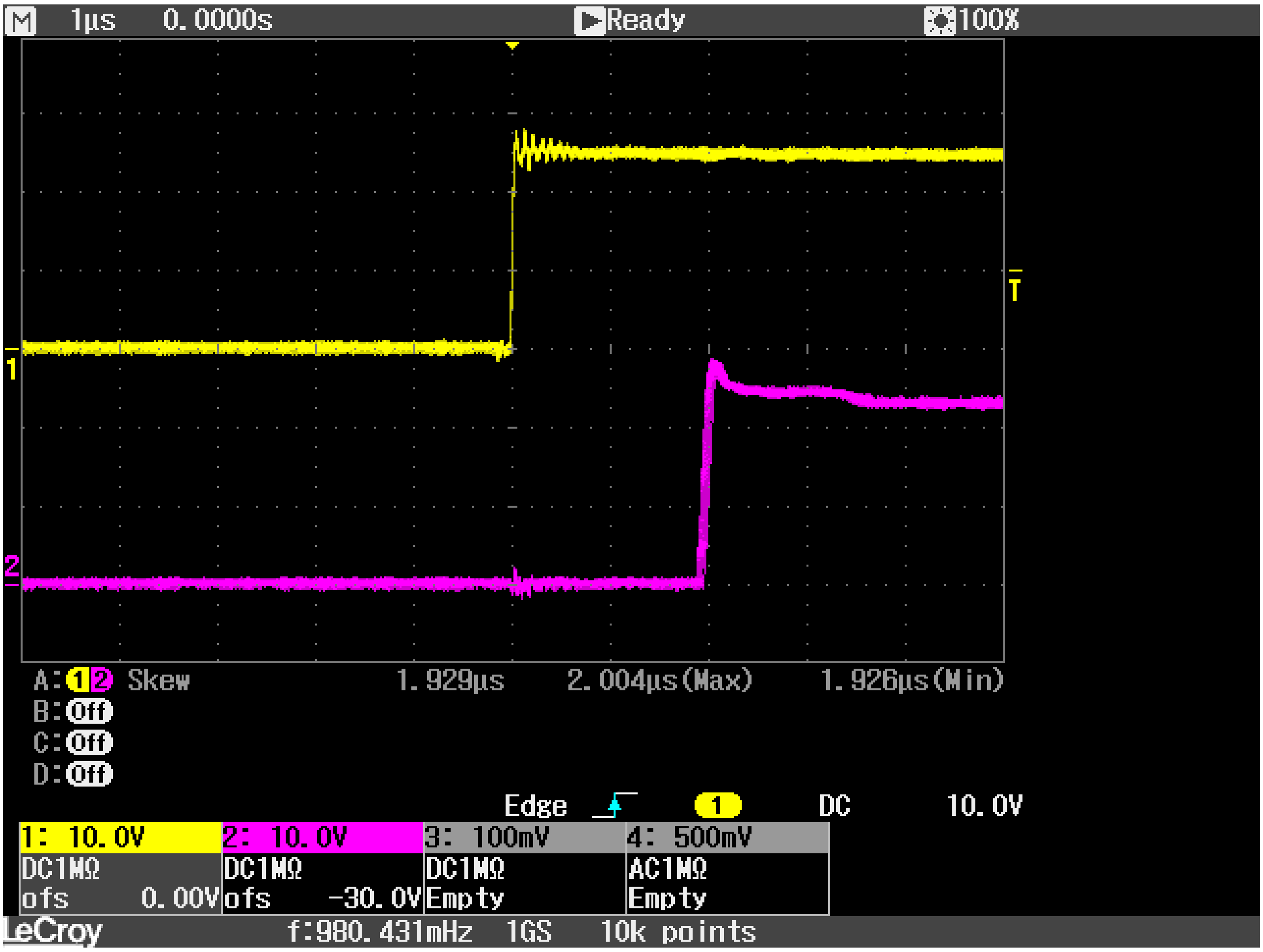


Fig. 210: Basic delay between trigger input and trigger output

The yellow signal (Channel 1) shows the input trigger. This signal is generated by an EL2202. In order to achieve reliable, reproducible behavior at the trigger output, it is imperative to use a signal with steep edges at the trigger input. The purple signal (Channel 2) shows the trigger output. The trigger output is parameterized so that the set delay of the trigger output is zero. Due to the basic delay described above, a delay in the range of 1.9 µs to 2.0 µs is set.

The commissioning procedure is described below:

1. Limit current in index 0x8000:02 [▸ 217] “Target current” in the unit mA
2. Input voltage in index 0x8000:03 [▸ 217] “Supply voltage” in the unit 0.01 V
3. Desired output voltage in index 0x8000:04 [▸ 217] “Output voltage” in the unit 0.01 V
4. Set the operation mode in the CoE directory in index 0x8004:01 [▸ 219] to “Current Control trigger pulse”
5. Set Predefined PDO Assignments to “External trigger input (with info data)”
6. Pulse length of the LED output (and the trigger output) specified via “DOX Impulse length” → “Impulse length” in the unit µs. The resolution of the time can be reduced from 1 µs to 100 ns using the CoE index 0x8002:31 [▸ 218] *Pulse resolution 100 ns*.

7. If you want to specify your own pulse length for the trigger output, “Trigger output mode”, “Synchron pulse with separate trigger delay” (5) or “Synchron pulse inverted with separate trigger delay” (6) must be enabled in the CoE index 0x8004:20 [▸ 219].
8. Only if 7. has been executed: A separate pulse length for the trigger output can then be specified via the CoE index 0x8004:05 [▸ 219] “Trigger out impulse length”.
9. Specify the delay of the LED output (and the output trigger) with respect to the trigger signal via “DOX Trigger delay” → “Trigger delay” in the unit µs. The resolution of the time can be reduced from 1 µs to 100 ns using the CoE index 0x8002:31 [▸ 218] *Pulse resolution 100 ns*. Alternatively, the delay can be firmly specified in the unit 0.1 µs in the CoE index 0x8000:0B [▸ 217].
10. Only if a delay for the trigger output is to be specified and 7. has been executed: A separate delay for the trigger output can then be specified via CoE index 0x8004:06 [▸ 219] “Trigger out delay” or via PDO 0x1607 “Trigger out delay”.
11. In the CoE index 0x8002:30 [▸ 218], specify whether the LED pulse should be generated by a rising or falling edge at the trigger input
12. In the CoE index 0x8004:03 [▸ 219], specify the level at the trigger input at which the LED output should be switched (high ≥ 10 V, low ≥ 5 V)
13. Connect the trigger signal to the TrigIn+ (5) and TrigIn- (13) connections
14. Activate the trigger input under “DOX Control” → “Control” → “Input Trigger Enable”
15. Connect the device to be controlled (e.g. camera) to the TrigOut trigger output (2). The external device to be controlled via the trigger output can be supplied with power via TrigOut+ (3) and TrigOut- (4). An external voltage divider [▸ 180] can be connected to the TrigExtDiv contacts (10, 11, 12) to reduce the supply voltage for the device in use.
16. Activate the trigger output under “DOX Control” → “Control” → “Output Trigger Enable”
17. Check under “DOX Status” → “Status” whether the “Ready to activate” bit is 1
18. Activate the control under “DOX Control” → “Control” via the “Enable” bit
19. Switch on the output under “DOX Control” → “Control” by activating the “Output” bit

5.8 Operation of a multi-color Common Anode LED

Multi-color LEDs usually consist of several individual, differently colored LEDs, which are interconnected with each other in an LED housing. There are different types of interconnection, which are described in more detail in the chapter Basics of LED technology [▶ 24] under Typical designs of multi-color LEDs [▶ 34].

A frequently used method of interconnection of multi-color LEDs is Common Anode. The anodes of the differently colored LEDs are connected to each other. Each LED has its own cathode fed to the outside, via which the individual differently colored LEDs in the multi-color LED can be controlled.

Only Common Anode LEDs can be controlled with the EL2596. In addition, the LED requires an EL2596 for each of its colors. An RGB LED has three colors (red, green, blue) and each color requires its own EL2596 for control, therefore three EL2596s are needed for an RGB LED.

Voltage control in the operation mode “Voltage control PWM” and current control from FW04 in the operation mode “Current Sink PWM” are both possible.

The connection of an RGB LED, which is also symbolic for RGB LED strips or multi-color LEDs with other/ more colors, is shown in the following figure.

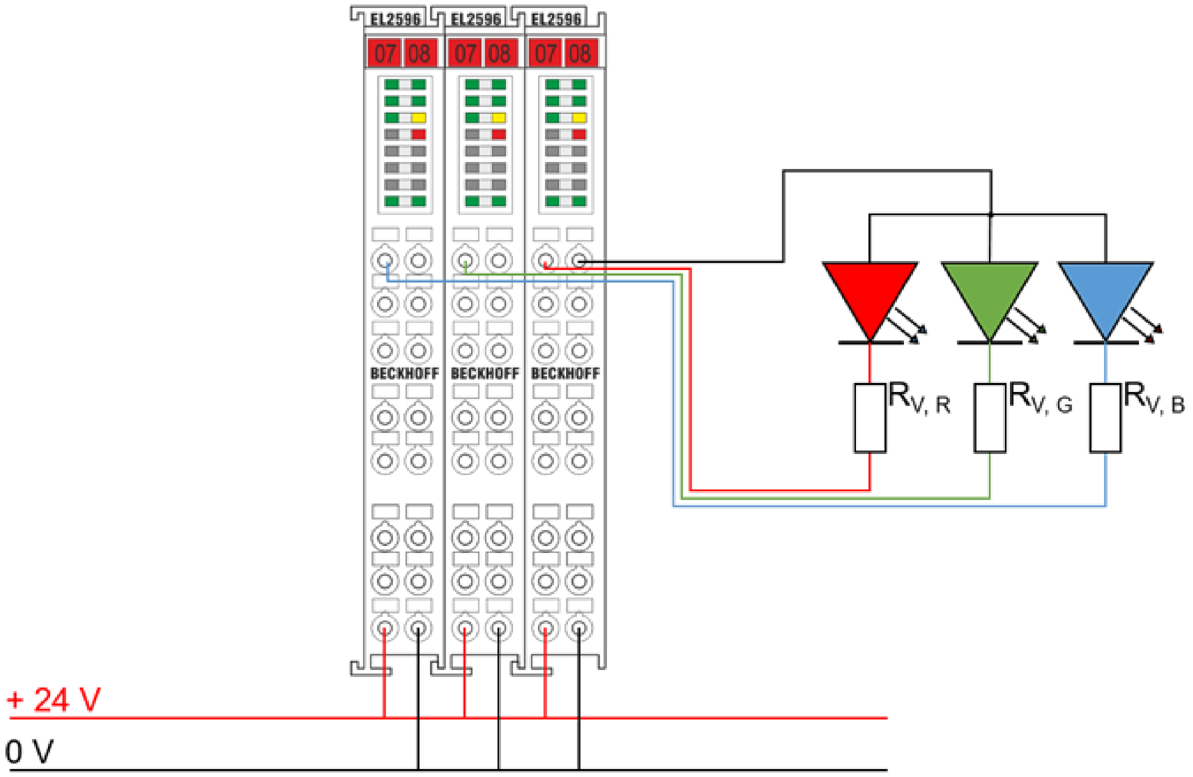


Fig. 211: Common Anode RGB LED on three EL2596 terminals

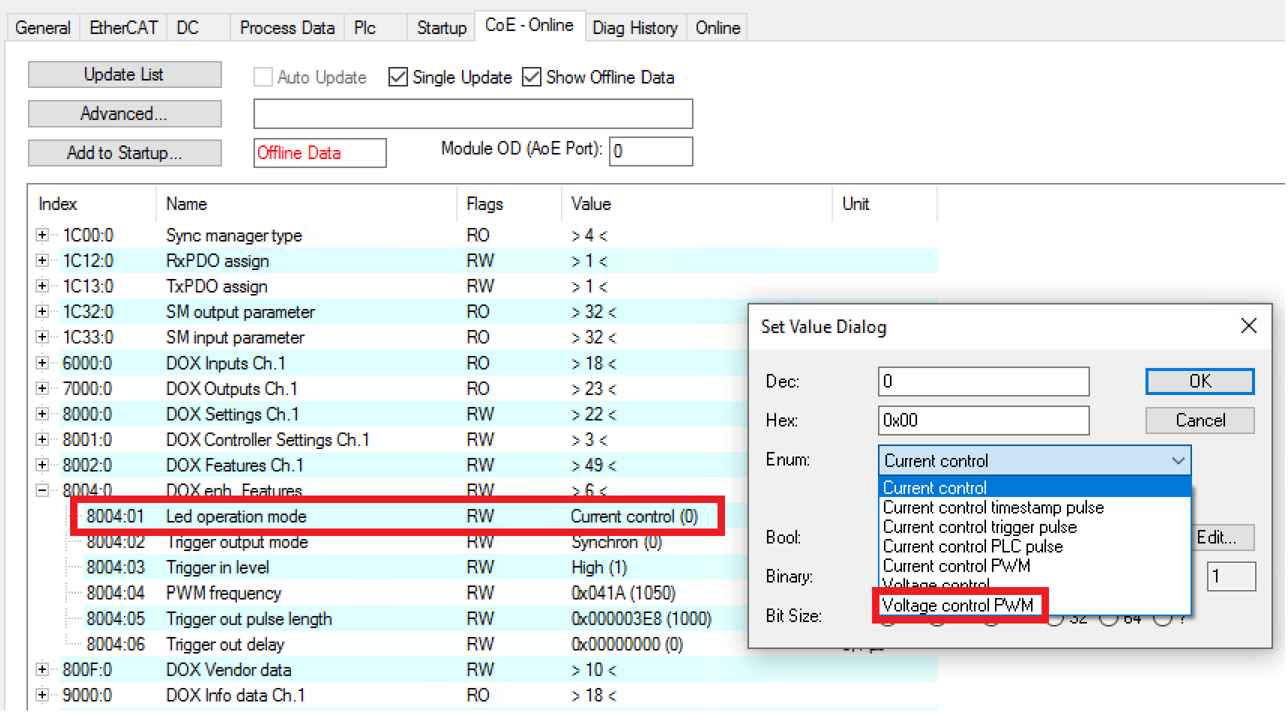
Operation of multicolor common anode LEDs in voltage mode

If a multicolor common anode LED or LED strip is to be voltage controlled, the EL2564 can also be used. This is a 4-channel LED output terminal with flexible output voltage between 5...48 V DC.

5.8.1 Voltage-controlled mode

In voltage-controlled mode, multi-color LEDs must always be used with a series resistor to protect the LEDs against destruction. The commissioning and control of a multi-color Common Anode LED in voltage-controlled mode is described below. Unless stated otherwise, the step described must be performed for all EL2596 terminals with the multi-color Common Anode LED.

1. Nominal/limit current of the LED in index 0x8000:02 [▸ 217] “Target current” in the unit mA.
2. Input voltage in index 0x8000:03 [▸ 217] “Supply voltage” in the unit 0.01 V.
3. Desired output voltage in index 0x8000:04 [▸ 217] “Output voltage” in the unit 0.01 V. **This value must be identical for all EL2596 terminals operated with the multi-color Common Anode LED.**
4. Set the operation mode in the CoE directory in index 0x8004:01 [▸ 219] to “Voltage Control PWM”



Index	Name	Flags	Value	Unit
1C00:0	Sync manager type	RO	> 4 <	
1C12:0	RxPDO assign	RW	> 1 <	
1C13:0	TxPDO assign	RW	> 1 <	
1C32:0	SM output parameter	RO	> 32 <	
1C33:0	SM input parameter	RO	> 32 <	
6000:0	DOX Inputs Ch.1	RO	> 18 <	
7000:0	DOX Outputs Ch.1	RO	> 23 <	
8000:0	DOX Settings Ch.1	RW	> 22 <	
8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
8002:0	DOX Features Ch.1	RW	> 49 <	
8004:0	DOX enh. Features	RW	> 6 <	
8004:01	Led operation mode	RW	Current control (0)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	
8004:06	Trigger out delay	RW	0x00000000 (0)	
800F:0	DOX Vendor data	RW	> 10 <	
9000:0	DOX Info data Ch.1	RO	> 18 <	

Set Value Dialog

Dec: 0

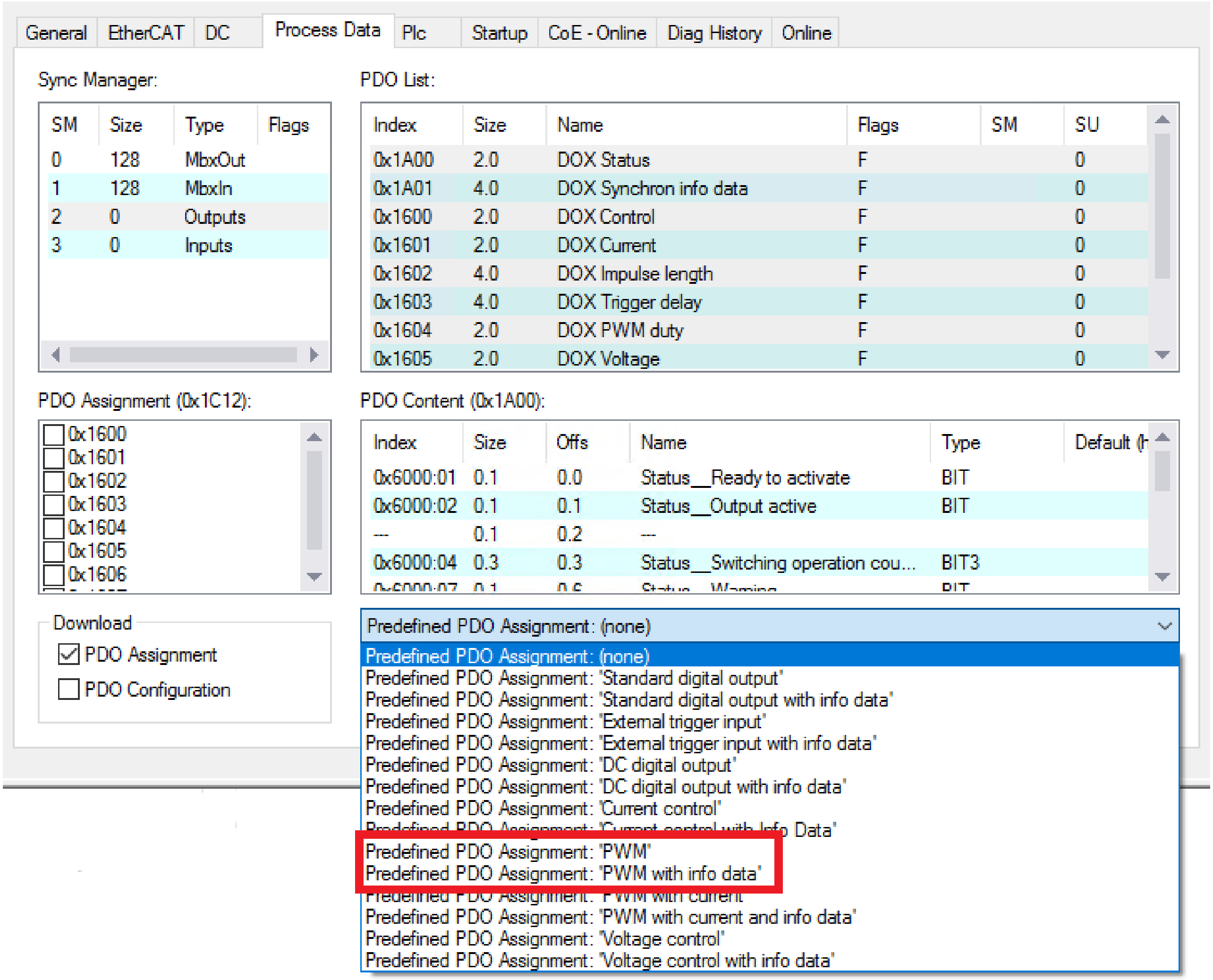
Hex: 0x00

Enum: Current control

- Current control
- Current control timestamp pulse
- Current control trigger pulse
- Current control PLC pulse
- Current control PWM
- Voltage control
- Voltage control PWM

Fig. 212: Operation mode setting “Voltage control PWM”

5. Set Predefined PDO Assignments to “PWM (with info data)”



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	0	Outputs	
3	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F		0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F		0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0

PDO Assignment (0x1C12):

- □ 0x1600
- □ 0x1601
- □ 0x1602
- □ 0x1603
- □ 0x1604
- □ 0x1605
- □ 0x1606

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	

Download

☑ PDO Assignment

□ PDO Configuration

Predefined PDO Assignment: (none)

- Predefined PDO Assignment: (none)
- Predefined PDO Assignment: 'Standard digital output'
- Predefined PDO Assignment: 'Standard digital output with info data'
- Predefined PDO Assignment: 'External trigger input'
- Predefined PDO Assignment: 'External trigger input with info data'
- Predefined PDO Assignment: 'DC digital output'
- Predefined PDO Assignment: 'DC digital output with info data'
- Predefined PDO Assignment: 'Current control'
- Predefined PDO Assignment: 'Current control with Info Data'
- Predefined PDO Assignment: 'PWM'
- Predefined PDO Assignment: 'PWM with info data'
- Predefined PDO Assignment: 'PWM with current'
- Predefined PDO Assignment: 'PWM with current and info data'
- Predefined PDO Assignment: 'Voltage control'
- Predefined PDO Assignment: 'Voltage control with info data'

Fig. 213: PDO setting “PWM (with info data)”

6. The PWM duty cycle can then be specified for each color (each EL2596) under “DOX PWM duty” → “PWM duty”. The higher the specified duty cycle, the brighter the color. The visible emitted color results from the mixture of colors in the multi-color LED.
7. Check under “DOX Status” → “Status” whether the “Ready to activate” bit is 1.
8. Activate the control under “DOX Control” → “Control” via the “Enable” bit.
9. Switch on the LED output under “DOX Control” → “Control” by activating the “Output” bit.

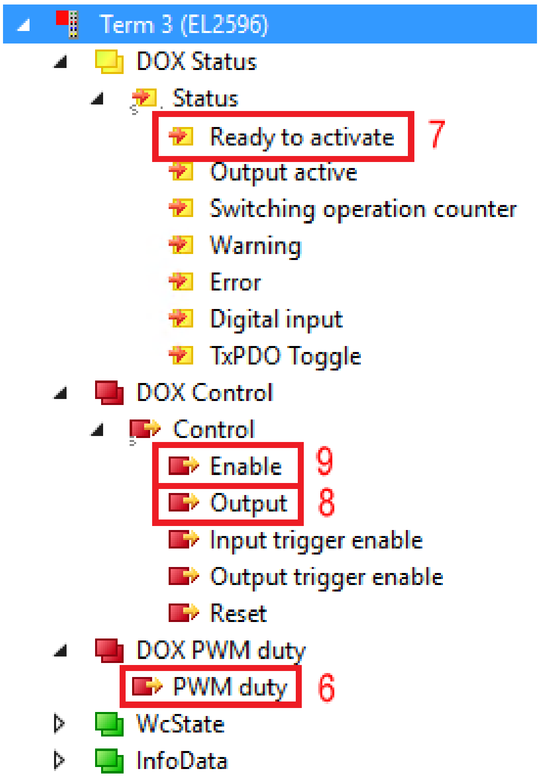


Fig. 214: Activating the output in the operation mode “Voltage control PWM”

5.8.2 Current-controlled mode

Current-controlled operation of multi-color Common Anode LEDs is also possible with the EL2596 from FW04. In this case, LEDs can also be operated without series resistors. This requires the “Current sink PWM” operation mode to be used. The commissioning is described step-by-step in the chapter Current Sink PWM [▶ 159].

The connection in this mode is possible as shown above. The color that requires the highest voltage should then also provide the voltage for the other colors. The anode is also connected to LED+ (9) at the terminal to which the cathode of the LED with the highest voltage is connected. This terminal must **not** be in “Current Sink PWM” mode, but in “Current Control” or “Current Control PWM”. The other LED colors are then each connected to one LED- (1) of an EL2596, which is operated in “Current Sink PWM” mode. Alternatively, the anode can be connected via an external supply, in which case all EL2596s are operated in “Current Sink PWM” mode.

5.9 Process data

The process data overview lists the detailed PDO selection. These data are not usually necessary for operation under TwinCAT, since they can be simply configured from the configuration interface via the process data preselection.

5.9.1 Process data overview

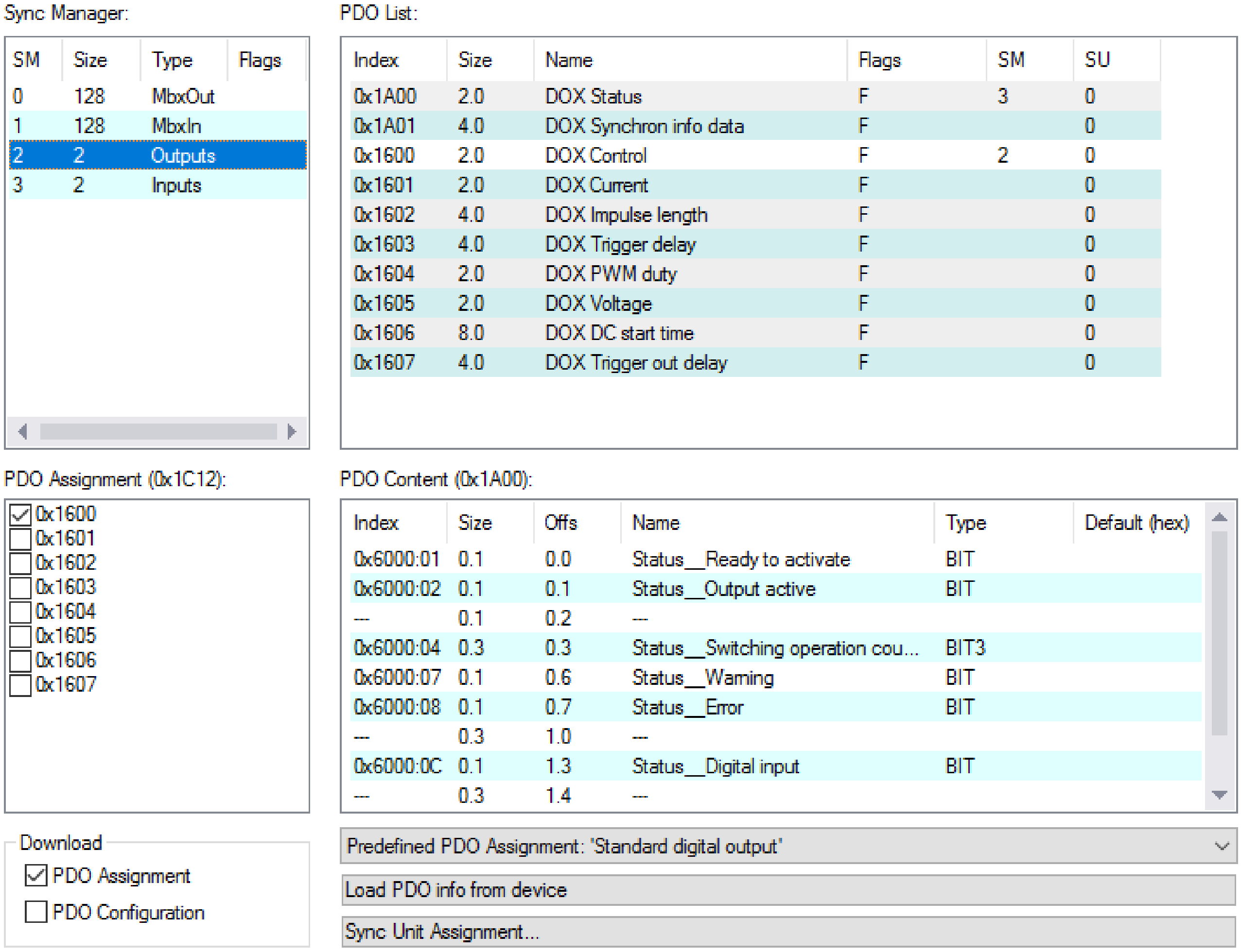
Manual process data assignment is necessary for TwinCAT up to version 2.10.

Sync Manager (SM)

The extent of the process data that is made available can be changed via the “Process data” tab (see Fig. *Process data tab SM2, EL2596 (default)*).
The PDOs from the range 0x160n (0x1600 to 0x1604) can be assigned to the Output SyncManager 2, see fig. *Process data tab SM2, EL2596 (default)*.

The PDOs from the range 0x1A0n (0x1A00 to 0x1A02) can be assigned to the Input SyncManager 3. See fig. *Process Data tab SM3, EL2596 (default)*.

Not all combinations are technically useful.



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	2	Outputs	
3	2	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F	3	0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F	2	0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0
0x1606	8.0	DOX DC start time	F		0
0x1607	4.0	DOX Trigger out delay	F		0

PDO Assignment (0x1C12):

☑ 0x1600
☐ 0x1601
☐ 0x1602
☐ 0x1603
☐ 0x1604
☐ 0x1605
☐ 0x1606
☐ 0x1607

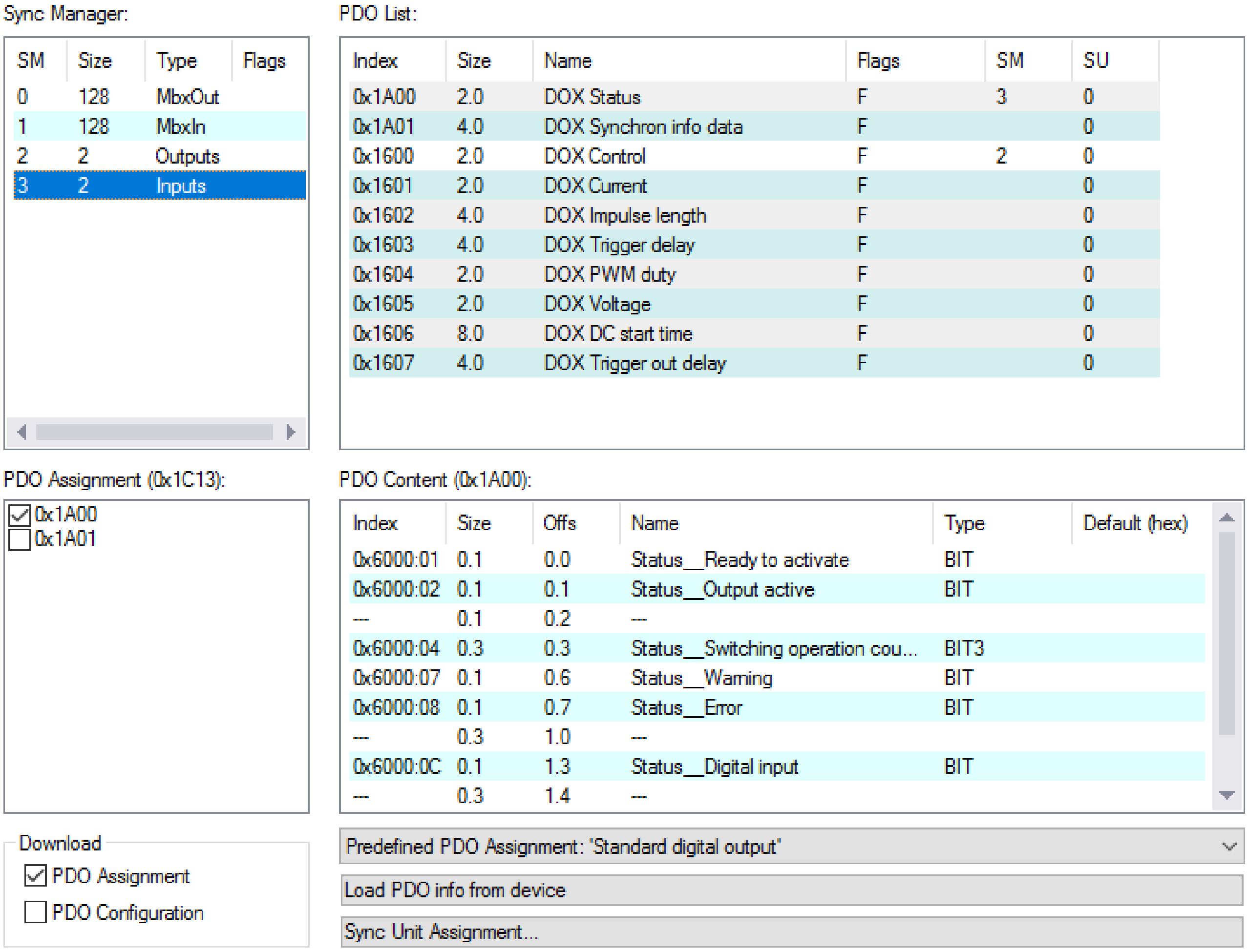
PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (hex)
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	
0x6000:08	0.1	0.7	Status__Error	BIT	
---	0.3	1.0	---		
0x6000:0C	0.1	1.3	Status__Digital input	BIT	
---	0.3	1.4	---		

Download
☑ PDO Assignment
☐ PDO Configuration

Predefined PDO Assignment: 'Standard digital output'
Load PDO info from device
Sync Unit Assignment...

Fig. 215: Process Data tab SM2, EL2596 (default)



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	2	Outputs	
3	2	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F	3	0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F	2	0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0
0x1606	8.0	DOX DC start time	F		0
0x1607	4.0	DOX Trigger out delay	F		0

PDO Assignment (0x1C13):

☑ 0x1A00
□ 0x1A01

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (hex)
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	
0x6000:08	0.1	0.7	Status__Error	BIT	
---	0.3	1.0	---		
0x6000:0C	0.1	1.3	Status__Digital input	BIT	
---	0.3	1.4	---		

Download
☑ PDO Assignment
□ PDO Configuration

Predefined PDO Assignment: 'Standard digital output'

Load PDO info from device

Sync Unit Assignment...

Fig. 216: Process Data tab SM3, EL2596 (default)

Manual PDO Assignment

To configure the process data, select the required Sync Manager (SM 2 + 3) in the “Sync Manager” field at the top left (see Fig. *Process data tab SM3, EL2596 (default)*). 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 Config mode (F4), causes the EtherCAT communication to restart, and the process data is transferred from the terminal.

SM2, PDO Assignment 0x1C12				
Index	**Index of excluded PDOs**	**Size (byte.bit)**	**Name**	**PDO Content Index - Name**
0x1600 (default)	-	2.0	DOX Control	0x7000:01 - Enable 0x7000:02 - Output 0x7000:04 - Input trigger enable 0x7000:05 - Input trigger enable 0x7000:08 - Reset
0x1601	0x1605	2.0	DOX Current	0x7000:11 - Output current
0x1602	-	4.0	DOX Impulse length	0x7000:12 - Impulse length
0x1603	-	4.0	DOX Trigger delay	0x7000:13 - Trigger delay
0x1604	-	2.0	DOX PWM duty	0x7000:14 - PWM duty
0x1605	0x1601	2.0	DOX Voltage	0x7000:15 - Output voltage
0x1606	-	8.0	DOX DC start time possible only in conjunction with DC operation mode!	0x7000:16 - DC Start time
0x1607	-	2.0	DOX Trigger out delay	0x7000:17 - Trigger out delay

SM3, PDO Assignment 0x1C13				
Index	**Index of excluded PDOs**	**Size (byte.bit)**	**Name**	**PDO Content Index - Name**
0x1A00 (default)	-	2.0	DOX Status	0x6000:01 - Ready to activate 0x6000:02 - Output active 0x6000:04 - Switching operation counter 0x6000:07 - Warning 0x6000:08 - Error 0x6000:0C - Digital input 0x6000:10 - TxPDO Toggle
0x1A01	-	4.0	DOX Synchron info data	0x6000:11 - Info data 1 0x6000:12 - Info data 2

5.9.2 Preselection of process data

An EtherCAT device usually offers several different process data objects (PDO) for input and output data, which can be configured in the System Manager, i.e. they can be activated or deactivated for cyclic transmission. See further below for the corresponding overview. Attention is thereby to be paid to the compatibility of input and output PDO.

From TwinCAT 2.11 with the EtherCAT devices intended for the purpose according to the ESI/XML description, the process data for input and output can be activated simultaneously by appropriate predefined sentences, “Predefined PDO”.

The EL2596 has on the “Process data” tab

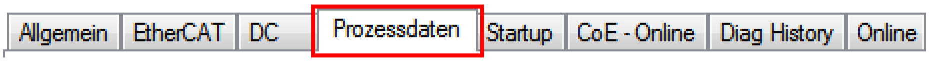
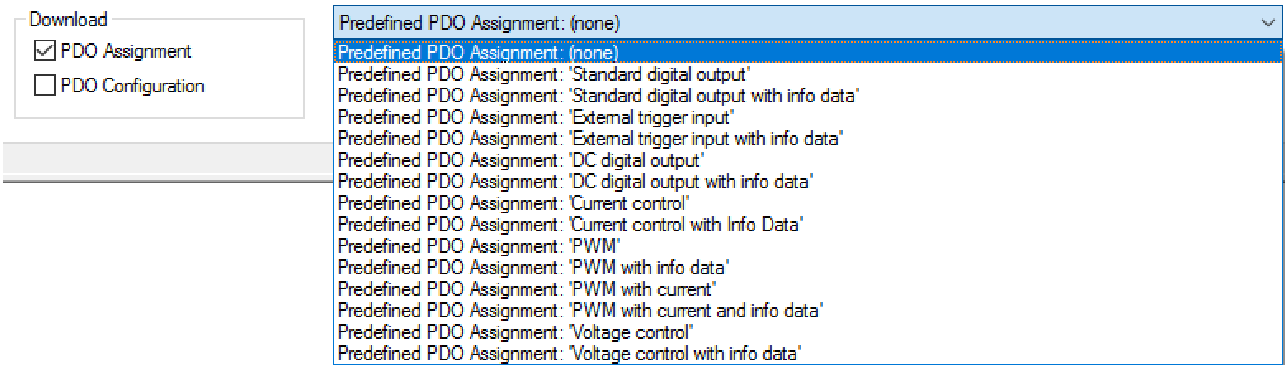


Fig. 217: “Process data” tab

the following “Predefined PDO” sentences:



Predefined PDO Assignment: (none)

Predefined PDO Assignment: (none)
Predefined PDO Assignment: 'Standard digital output'
Predefined PDO Assignment: 'Standard digital output with info data'
Predefined PDO Assignment: 'External trigger input'
Predefined PDO Assignment: 'External trigger input with info data'
Predefined PDO Assignment: 'DC digital output'
Predefined PDO Assignment: 'DC digital output with info data'
Predefined PDO Assignment: 'Current control'
Predefined PDO Assignment: 'Current control with Info Data'
Predefined PDO Assignment: 'PWM'
Predefined PDO Assignment: 'PWM with info data'
Predefined PDO Assignment: 'PWM with current'
Predefined PDO Assignment: 'PWM with current and info data'
Predefined PDO Assignment: 'Voltage control'
Predefined PDO Assignment: 'Voltage control with info data'

Fig. 218: TwinCAT System Manager with the PDO selection

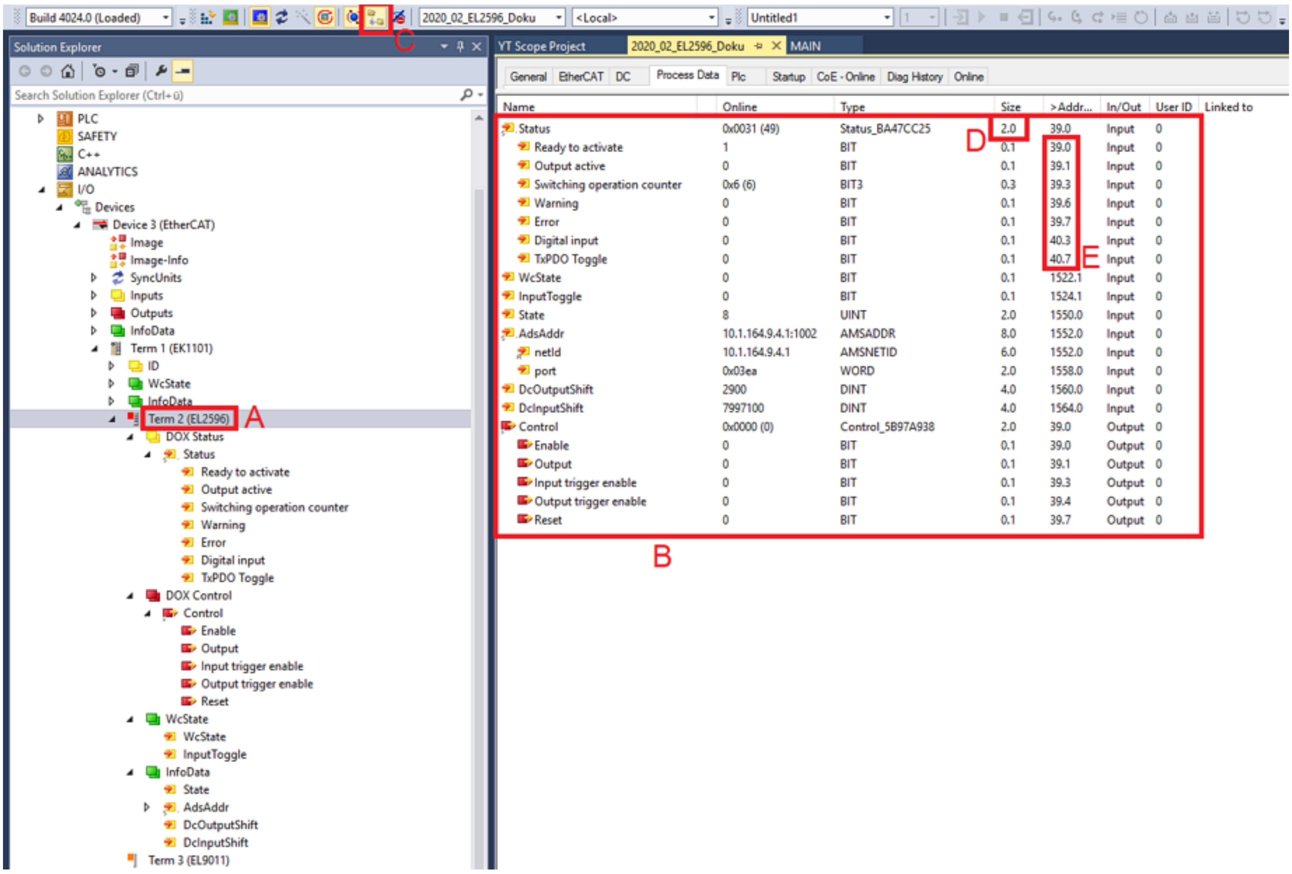
In detail the sentences are composed as follows:

Operation mode	Name	SM2, PDO assignment	SM3, PDO assignment
Current-controlled output	Standard digital output (default setting)	0x1600	0x1A00
	Standard digital output with info data	0x1600	0x1A00 0x1A01
	External trigger input	0x1600 0x1601 0x1602 0x1603	0x1A00
	External trigger input with info data	0x1600 0x1601 0x1602 0x1603	0x1A00 0x1A01
	DC digital output	0x1600 0x1601 0x1602 0x1606	0x1A00
	DC digital output with info data	0x1600 0x1601 0x1602 0x1606	0x1A00 0x1A01
	Current Control	0x1600 0x1601	0x1A00
	Current Control with info data	0x1600 0x1601	0x1A00 0x1A01
	PWM with current	0x1600 0x1601 0x1604	0x1A00
	PWM with current and info data	0x1600 0x1601 0x1604	0x1A00 0x1A01
Voltage-controlled output	Voltage control	0x1600 0x1605	0x1A00
	Voltage control with info data	0x1600 0x1601	0x1A00 0x1A01
	PWM	0x1600 0x1604	0x1A00
	PWM with info data	0x1600 0x1604	0x1A00 0x1A01

5.9.3 Explanation of the process data

Standard digital output (default)

The outputs can be written directly with this standard PDO assignment, i.e. the connected actuators can be switched frame-triggered directly.



Name	Online	Type	Size	>Addr...	In/Out	User ID	Linked to
Status	0x0031 (49)	Status_BA47CC25	2.0	39.0	Input	0	
Ready to activate	1	BIT	0.1	39.0	Input	0	
Output active	0	BIT	0.1	39.1	Input	0	
Switching operation counter	0x6 (6)	BIT3	0.3	39.3	Input	0	
Warning	0	BIT	0.1	39.6	Input	0	
Error	0	BIT	0.1	39.7	Input	0	
Digital input	0	BIT	0.1	40.3	Input	0	
TxPDO Toggle	0	BIT	0.1	40.7	Input	0	
WcState	0	BIT	0.1	1522.1	Input	0	
InputToggle	0	BIT	0.1	1524.1	Input	0	
State	8	UINT	2.0	1550.0	Input	0	
AdsAddr	10.1.164.9.4.1:1002	AMSADDR	8.0	1552.0	Input	0	
netId	10.1.164.9.4.1	AMSNETID	6.0	1552.0	Input	0	
port	0x03ea	WORD	2.0	1558.0	Input	0	
DcOutputShift	2900	DINT	4.0	1560.0	Input	0	
DcInputShift	7997100	DINT	4.0	1564.0	Input	0	
Control	0x0000 (0)	Control_5B97A938	2.0	39.0	Output	0	
Enable	0	BIT	0.1	39.0	Output	0	
Output	0	BIT	0.1	39.1	Output	0	
Input trigger enable	0	BIT	0.1	39.3	Output	0	
Output trigger enable	0	BIT	0.1	39.4	Output	0	
Reset	0	BIT	0.1	39.7	Output	0	

Fig. 219: Standard EL2596 process image

The EL2596 (A) has input and output variables. These can be seen by expanding the tree (A). They are also displayed in the detail view (B) if the appropriate display function (C) is activated.

The bit meaning i.e. offset position can then also be taken from the memory assignment display (E), taking into account the variable size (D).

Both the collective name e.g. *Status* and the individual bit variable e.g. *Output active* can be linked, but not both at the same time.

Input data			
Collective name	**Name**	**Description / function**	**Bit position [0..15]**
Status	Ready to activate	The terminal signals its operational readiness here. This bit is 0 if there is an error. For diagnostics, see EL2596-xxxx specific diagnostics [▶ 204])	0
	Output active	The output is actively switched.	1
	Switching operation counter		3
	Warning	A warning has occurred - > evaluate “Diag data” (index 0xA000).	6
	Error	An error has occurred and the output driver is deactivated - > evaluate "Diag data" (index 0xA000).	7
	Digital input	The status of the digital input is returned here.	11
	TxPDO Toggle	Changes its state each time process data are exchanged.	15
WcState		Setpoint during operation: 0 Each datagram of the EL2596 indicates its processing state here. This allows the EL2596 to be monitored for correct process data communication.	
InputToggle			
State		Setpoint during operation: 8 Status display of the “EtherCAT State Machine”	
AdsAddr		AMS address of the responsible EtherCAT Master in the format "0.0.0.0.0.0". In addition, the port number valid for this Slave. Required for acyclic accesses to the CoE at runtime.	

Output data			
Collective name	**Name**	**Description / function**	**Bit position [0..15]**
Control	Enable	Activate control	0
	Output	Switches the output active	1
	Input trigger enable	Activates the digital input as a trigger input	3
	Output trigger enable	Activates the digital output as a trigger output	
	Reset	Resets an error	7

Standard digital output with InfoData

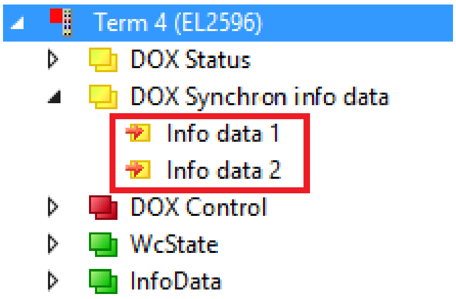


Fig. 220: Additional information data

Two further cyclic data words can be displayed per channel for more exact information about the states of the actuators or the driver stage. The respective selection is to be configured via the corresponding Index 0x8002:11 [▶ 218] or 0x8002:19 [▶ 218] in the CoE. Among other things, the interior temperature of the terminal or the momentary current through the connected actuator can be selected, for example.

Input data	
Name	**Description / function**
Info data 1	Additional channel information, definition in 0x8002:11
Info data 2	Additional channel information, definition in 0x8002:19

External trigger input

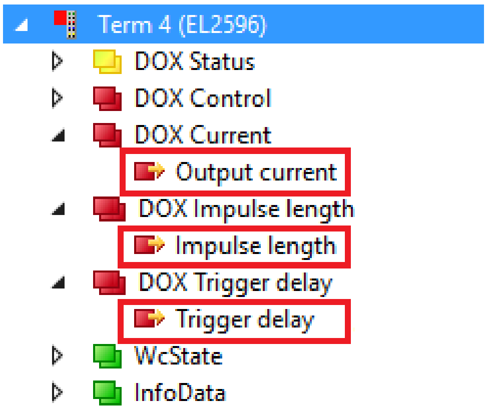


Fig. 221: Additional process data in the “External trigger input” mode

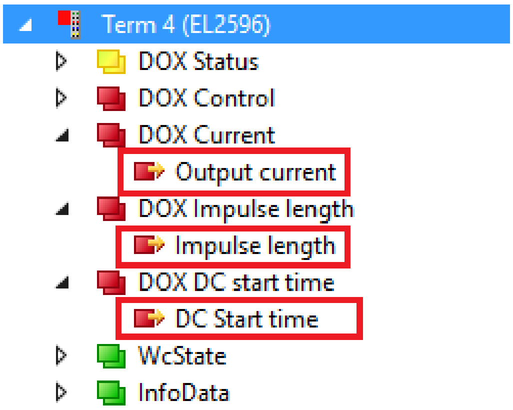
In addition to the variables of the “Standard digital output” mode, there are also the following variables:

Output data	
Name	**Description / function**
Output current	Specification of the set current for the current control in the unit mA
Impulse length	Specifies the length of the output pulses. The unit is 1 µs. The resolution of the time can be reduced from 1 µs to 100 ns using the CoE index 0x8002:31 [▶ 218] *Pulse resolution 100 ns*.
Trigger delay	Specifies the length of the delay from the time of the digital input to the switching of the output. The unit is 1 µs. The resolution of the time can be reduced from 1 µs to 100 ns using the CoE index 0x8002:31 [▶ 218] *Pulse resolution 100 ns*.

External trigger input with InfoData

Like both standard data types, additional information data can also be displayed in “External trigger input” mode. See above.

DC Digital output



Term 4 (EL2596)
- DOX Status
- DOX Control
- DOX Current
  - Output current
- DOX Impulse length
  - Impulse length
- DOX DC start time
  - DC Start time
- WcState
- InfoData

Fig. 222: Additional process data for Distributed Clocks mode

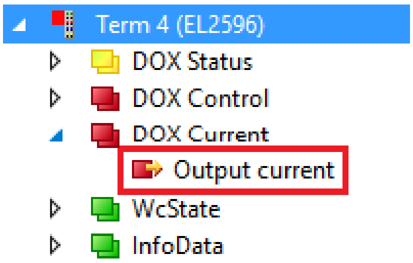
In Distributed Clocks mode the EL2596 works according to timestamp order. The process image is structured correspondingly, as in the fig. *Additional process data for Distributed Clocks mode*.

Output data	
Name	**Description / function**
Output current	Specification of the set current for the current control in the unit mA
Impulse length	Specifies the length of the output pulses. The unit is 1 µs. The resolution of the time specification can be reduced via the CoE object 0x8002:31 [▶ 218] *Pulse resolution 100 ns* from 1 µs to 100 ns.
DC Start time	64-bit value of the next desired switching event. The data of the DC time: • Start time 1.1.2000 • Resolution 1 bit = 1 ns

DC Digital output with InfoData

As with the standard data, additional information data can also be shown in the “DC digital output” mode. See above.

Current control



Term 4 (EL2596)
- DOX Status
- DOX Control
- DOX Current
  - Output current
- WcState
- InfoData

Fig. 223: Additional process data in “Current control” mode

In addition to the variables of the “Standard digital output” mode, there are also the following variables:

Output data	
Name	**Description / function**
Output current	Specification of the set current for the current control in the unit mA

Current control with InfoData

As with the standard data, additional information data can also be shown in the “Current control” mode. See above.

PWM

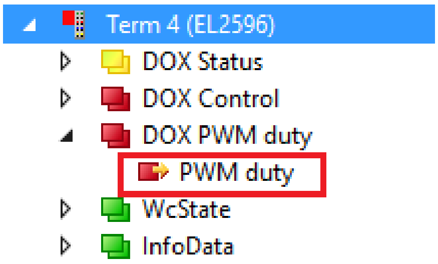


Fig. 224: Additional process data in “PWM” mode

In addition to the variables of the “Standard digital output” mode, there are also the following variables:

Output data	
Name	**Description / function**
PWM duty	Duty cycle of the pulse width modulation

PWM with InfoData

As with the standard data, additional information data can also be shown in the “PWM” mode. See above.

PWM with current

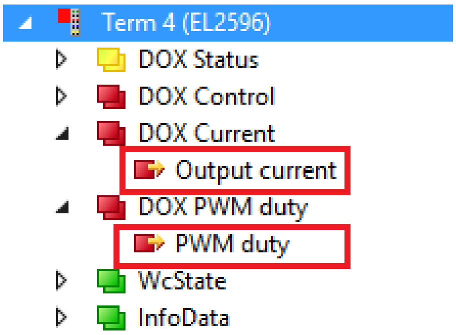


Fig. 225: Additional process data in “PWM with current” mode

In addition to the variables of the “Standard digital output” mode, there are also the following variables:

Output data	
Name	**Description / function**
Output current	Specification of the set current for the current control in the unit mA
PWM duty	Duty cycle of the pulse width modulation

PWM with current and InfoData

As with the standard data, additional information data can also be shown in the “PWM with current” mode. See above.

Voltage control

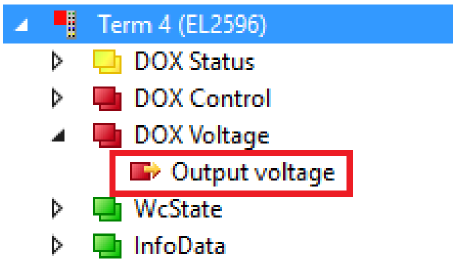


Fig. 226: Additional process data in “Voltage control” mode

In addition to the variables of the “Standard digital output” mode, there are also the following variables:

Output data	
Name	**Description / function**
Output voltage	Specification of the set voltage for the voltage control in the unit 0.01 mV

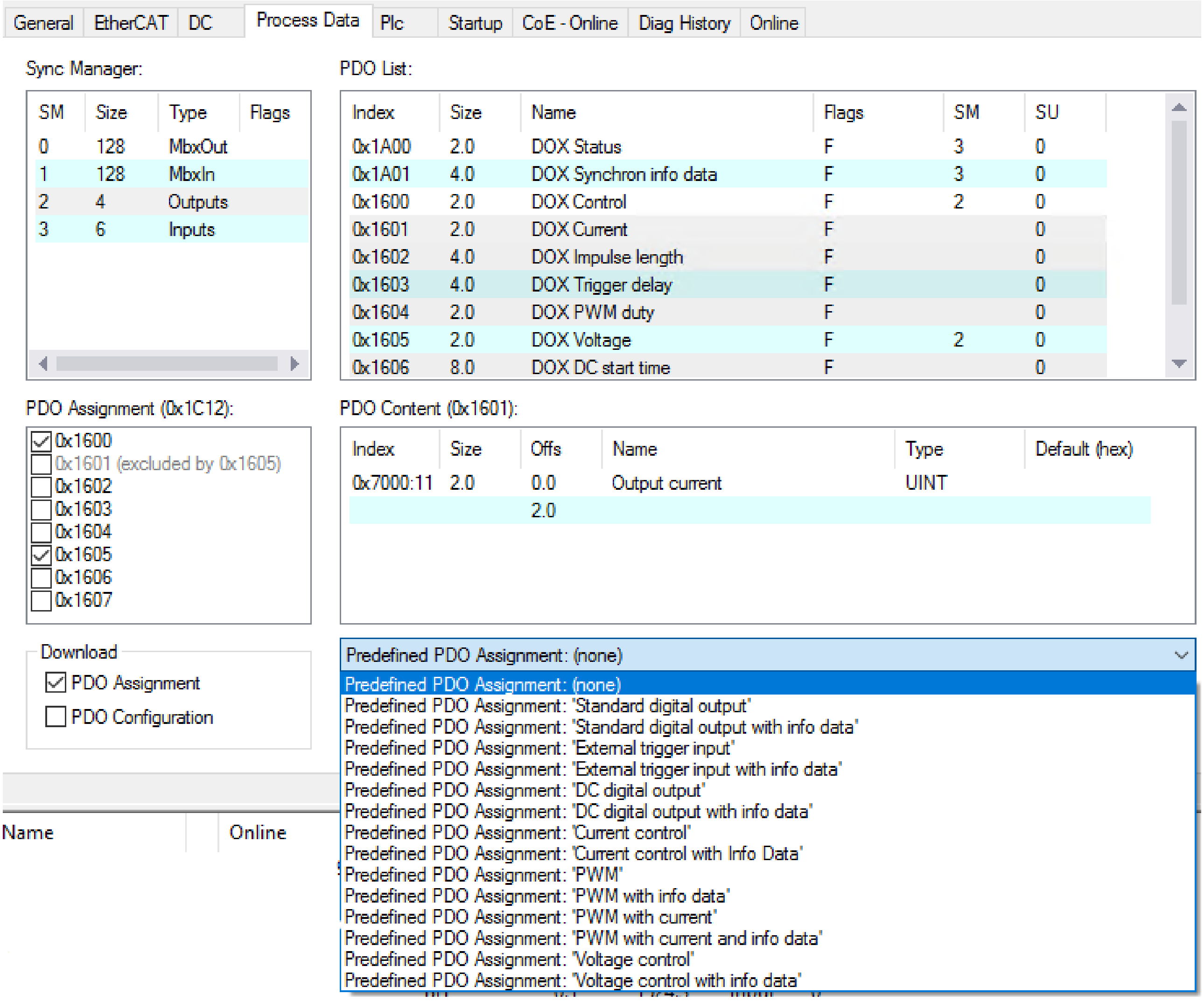
Voltage control with current and InfoData

As with the standard data, additional information data can also be shown in the “Voltage control” mode. See above.

5.10 EL2596-xxxx specific diagnostics

5.10.1 Diagnostics with the info data

For all Predefined PDOs, info data can be mapped to the standard PDOs via the selection “... with info data”.



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	4	Outputs	
3	6	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F	3	0
0x1A01	4.0	DOX Synchron info data	F	3	0
0x1600	2.0	DOX Control	F	2	0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F	2	0
0x1606	8.0	DOX DC start time	F		0

PDO Assignment (0x1C12):

☑ 0x1600
□ 0x1601 (excluded by 0x1605)
□ 0x1602
□ 0x1603
□ 0x1604
☑ 0x1605
□ 0x1606
□ 0x1607

PDO Content (0x1601):

Index	Size	Offs	Name	Type	Default (hex)
0x7000:11	2.0	0.0	Output current	UINT	
		2.0			

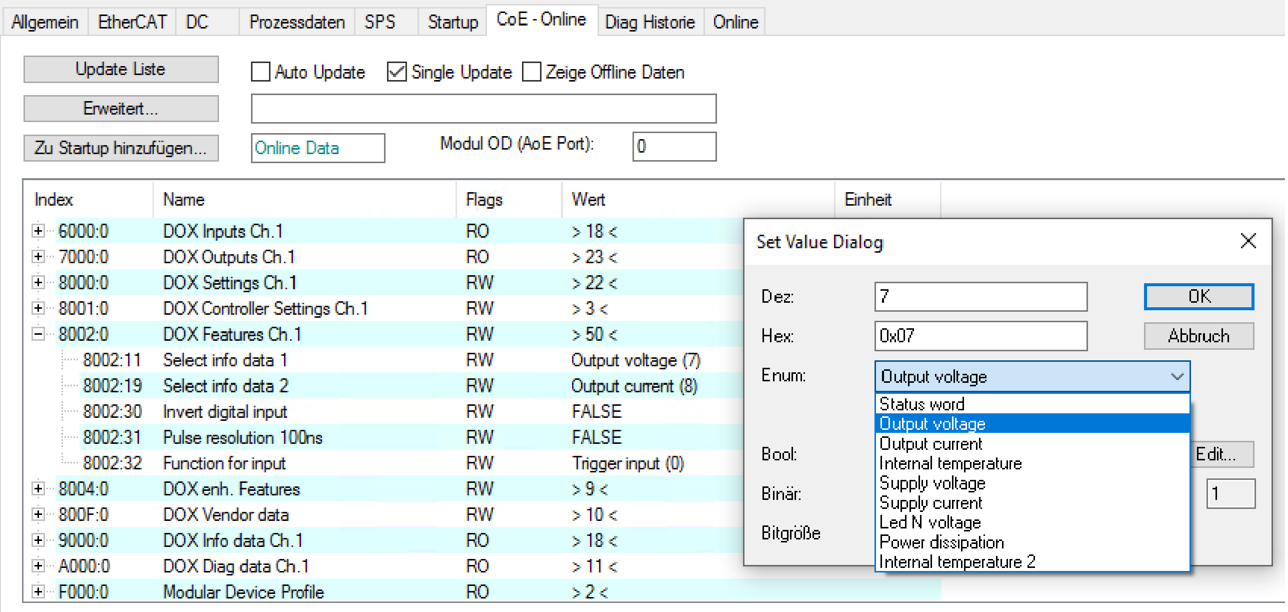
Download
☑ PDO Assignment
□ PDO Configuration

Predefined PDO Assignment: (none)

Predefined PDO Assignment: (none)
Predefined PDO Assignment: 'Standard digital output'
Predefined PDO Assignment: 'Standard digital output with info data'
Predefined PDO Assignment: 'External trigger input'
Predefined PDO Assignment: 'External trigger input with info data'
Predefined PDO Assignment: 'DC digital output'
Predefined PDO Assignment: 'DC digital output with info data'
Predefined PDO Assignment: 'Current control'
Predefined PDO Assignment: 'Current control with Info Data'
Predefined PDO Assignment: 'PWM'
Predefined PDO Assignment: 'PWM with info data'
Predefined PDO Assignment: 'PWM with current'
Predefined PDO Assignment: 'PWM with current and info data'
Predefined PDO Assignment: 'Voltage control'
Predefined PDO Assignment: 'Voltage control with info data'

Fig. 227: Predefined PDOs “... with info data”

These info data are two input variables that can display different internal values from the terminal. Which data the info data display can be selected in CoE index 0x8002:11 [▶ 218] (Select info data 1) and 0x8002:19 [▶ 218] (Select info data 2). Various input values (supply voltage, supply current), output values (output voltage, output current) or internal terminal values (status word, internal temperature) can be selected.



□ Auto Update ☑ Single Update □ Zeige Offline Daten

Modul OD (AoE Port): 0

Index	Name	Flags	Wert	Einheit
+ 6000:0	DOX Inputs Ch.1	RO	> 18 <	
+ 7000:0	DOX Outputs Ch.1	RO	> 23 <	
+ 8000:0	DOX Settings Ch.1	RW	> 22 <	
+ 8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
– 8002:0	DOX Features Ch.1	RW	> 50 <	
8002:11	Select info data 1	RW	Output voltage (7)	
8002:19	Select info data 2	RW	Output current (8)	
8002:30	Invert digital input	RW	FALSE	
8002:31	Pulse resolution 100ns	RW	FALSE	
8002:32	Function for input	RW	Trigger input (0)	
+ 8004:0	DOX enh. Features	RW	> 9 <	
+ 800F:0	DOX Vendor data	RW	> 10 <	
+ 9000:0	DOX Info data Ch.1	RO	> 18 <	
+ A000:0	DOX Diag data Ch.1	RO	> 11 <	
+ F000:0	Modular Device Profile	RO	> 2 <	

Set Value Dialog

Dez: 7

Hex: 0x07

Enum: Output voltage

- Status word
- Output voltage
- Output current
- Internal temperature
- Supply voltage
- Supply current
- Led N voltage
- Power dissipation
- Internal temperature 2

Bool:

Binär: 1

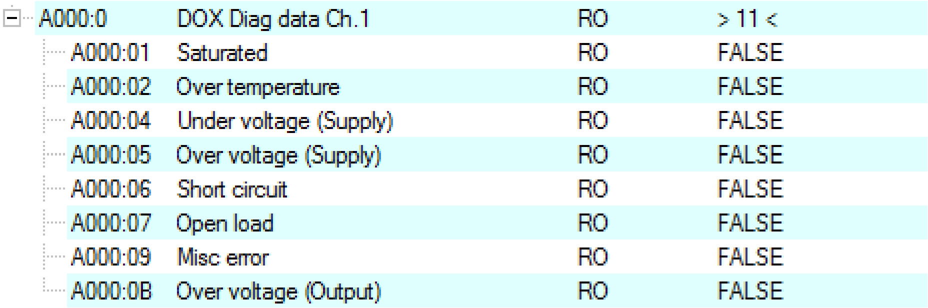
Bitgröße

Fig. 228: CoE index 0x8002 “Select info data”

The values from the info data can then be used for diagnostics from the PLC, so that in various cases your own warning or error messages from the PLC are displayed, for example, in a visualization. However, the values from the info data can also be recorded and made visible for continuous visualization with the TwinCAT Scope. In this way, for example, the curve of output current and output voltage can be compared in different use cases.

5.10.2 Diagnostics in the CoE

The EL2596 has internal diagnostics. Warnings and errors are also displayed by the LEDs. The diagnostics can be viewed in the CoE objects under 0xA000 [▶ 220] “DOX Diag data Ch. 1”.



– A000:0	DOX Diag data Ch.1	RO	> 11 <
A000:01	Saturated	RO	FALSE
A000:02	Over temperature	RO	FALSE
A000:04	Under voltage (Supply)	RO	FALSE
A000:05	Over voltage (Supply)	RO	FALSE
A000:06	Short circuit	RO	FALSE
A000:07	Open load	RO	FALSE
A000:09	Misc error	RO	FALSE
A000:0B	Over voltage (Output)	RO	FALSE

Fig. 229: Diagnostics in the CoE object 0xA000

The warning and error levels for the supply and output voltages can be adjusted in the CoE objects 0x8000:11 – 0x8000:16 [▶ 217].

In case of an error, bit 0 “Ready to activate” in the “DOX Status” goes to 0. The LED output can then no longer be set. An error must be rectified and reset via a rising edge on the “Reset” bit in the “DOX Control”. In case of a reset, the “Enable” and “Output” bits under “DOX Control” must be deactivated.

Object	Name	Description	Correction
A000:01	Saturated	Target voltage cannot be reached	Increase U_{in} or reduce set current
A000:02	Over temperature	Warning at internal temperature of 80 °C Error at internal temperature of 100 °C	EL2596 must cool down
A000:04	Under voltage (Supply)	Under voltage supply	Increase U_{in}
A000:05	Over voltage (Supply)	Over voltage supply	Reduce U_{in}
A000:06	Short circuit	Short circuit has occurred	Check whether the load is connected correctly and is functional. (not detected during voltage control)
A000:07	Open load	Wire breakage detected	Check whether the load is connected correctly and is functional. (not detected during voltage control)
A000:09	Misc error	Internal hardware error	
A000:0B	Over voltage (Output)	Overvoltage on the digital output	Reduce output voltage

5.10.3 Diagnostics - basic principles of diag messages

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

Definition

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

Each DiagMessage consists of

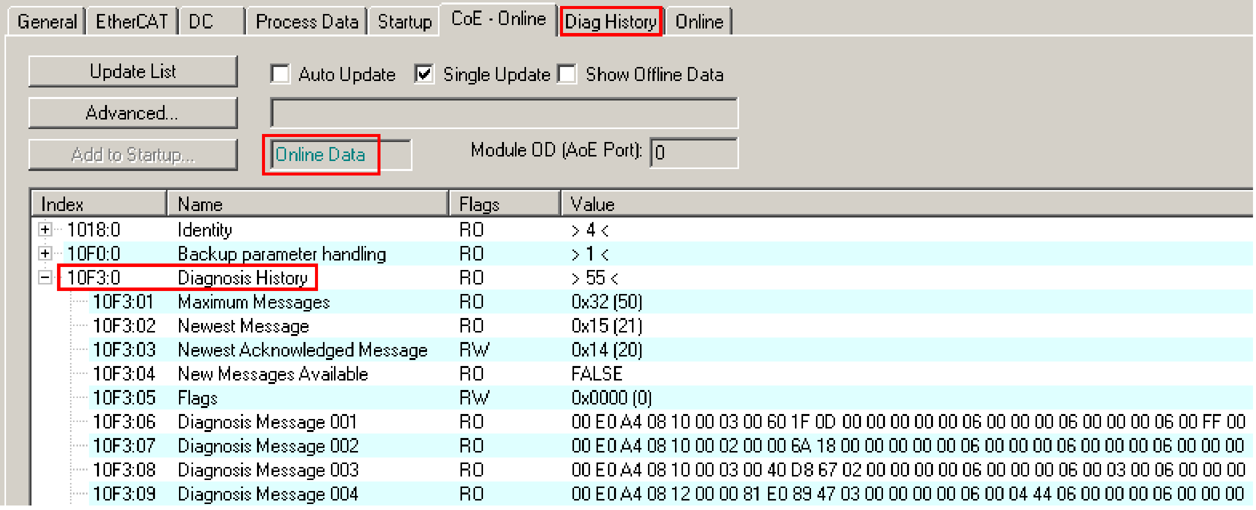
- Diag Code (4-byte)
- Flags (2-byte; info, warning or error)
- Text ID (2-byte; reference to explanatory text from the ESI/XML)
- Timestamp (8-byte, local slave time or 64-bit Distributed Clock time, if available)
- Dynamic parameters added by the firmware

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

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

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

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



□ Auto Update ☑ Single Update □ Show Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value
+ 1018:0	Identity	RO	> 4 <
+ 10F0:0	Backup parameter handling	RO	> 1 <
− 10F3:0	Diagnosis History	RO	> 55 <
10F3:01	Maximum Messages	RO	0x32 (50)
10F3:02	Newest Message	RO	0x15 (21)
10F3:03	Newest Acknowledged Message	RW	0x14 (20)
10F3:04	New Messages Available	RO	FALSE
10F3:05	Flags	RW	0x0000 (0)
10F3:06	Diagnosis Message 001	RO	00 E0 A4 08 10 00 03 00 60 1F 0D 00 00 00 00 00 06 00 00 00 06 00 00 00 06 00 FF 00
10F3:07	Diagnosis Message 002	RO	00 E0 A4 08 10 00 02 00 00 6A 18 00 00 00 00 00 06 00 00 00 06 00 00 00 06 00 00 00
10F3:08	Diagnosis Message 003	RO	00 E0 A4 08 10 00 03 00 40 D8 67 02 00 00 00 00 06 00 00 00 06 00 03 00 06 00 00 00
10F3:09	Diagnosis Message 004	RO	00 E0 A4 08 12 00 00 81 E0 89 47 03 00 00 00 00 06 00 04 44 06 00 00 00 06 00 00 00

Fig. 230: DiagMessages in the CoE

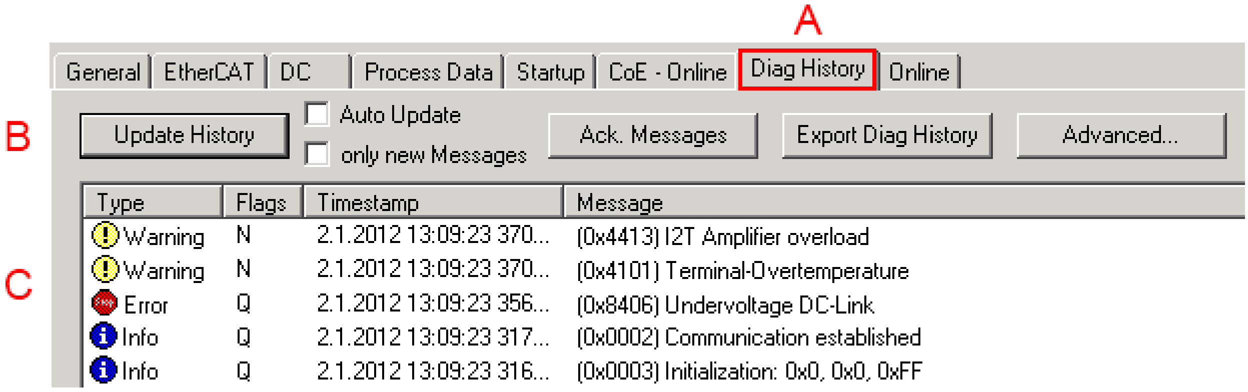
The subindex of the latest *DiagMessage* can be read under 0x10F3:02.

Support for commissioning

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

TwinCAT System Manager implementation

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



□ Auto Update
□ only new Messages

Type	Flags	Timestamp	Message
Warning	N	2.1.2012 13:09:23 370...	(0x4413) I2T Amplifier overload
Warning	N	2.1.2012 13:09:23 370...	(0x4101) Terminal-Overtemperature
Error	Q	2.1.2012 13:09:23 356...	(0x8406) Undervoltage DC-Link
Info	Q	2.1.2012 13:09:23 317...	(0x0002) Communication established
Info	Q	2.1.2012 13:09:23 316...	(0x0003) Initialization: 0x0, 0x0, 0xFF

Fig. 231: Implementation of the DiagMessage system in the TwinCAT System Manager

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

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

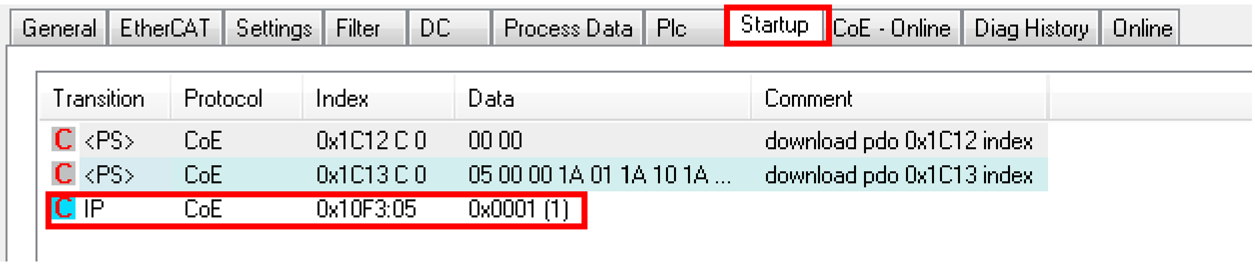
The meanings of the buttons are self-explanatory.

DiagMessages within the ADS Logger/Eventlogger

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

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

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



Transition	Protocol	Index	Data	Comment
C <PS>	CoE	0x1C12 C 0	00 00	download pdo 0x1C12 index
C <PS>	CoE	0x1C13 C 0	05 00 00 1A 01 1A 10 1A ...	download pdo 0x1C13 index
C IP	CoE	0x10F3:05	0x0001 (1)	

Fig. 232: Startup List

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

Reading messages into the PLC

- In preparation -

Interpretation

Time stamp

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

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

Structure of the Text ID

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

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

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

Overview of text IDs

Specific text IDs are listed in the device documentation.

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

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

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

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

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

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

Text ID	Type	Place	Text Message	Additional comment
0x8550	Error	Inputs	Zero crossing phase %X missing	Zero crossing phase %X missing
0x8551	Error	Inputs	Phase sequence Error	Wrong direction of rotation
0x8552	Error	Inputs	Overcurrent phase %X	Overcurrent phase %X
0x8553	Error	Inputs	Overcurrent neutral wire	Overcurrent neutral wire
0x8581	Error	Inputs	Wire broken Ch %D	Wire broken Ch %d
0x8600	Error	General IO	Wrong supply voltage range	Supply voltage not in the correct range
0x8601	Error	General IO	Supply voltage to low	Supply voltage too low
0x8602	Error	General IO	Supply voltage to high	Supply voltage too high
0x8603	Error	General IO	Over current of supply voltage	Overcurrent of supply voltage
0x8610	Error	General IO	Wrong output voltage range	Output voltage not in the correct range
0x8611	Error	General IO	Output voltage to low	Output voltage too low
0x8612	Error	General IO	Output voltage to high	Output voltage too high
0x8613	Error	General IO	Over current of output voltage	Overcurrent of output voltage
0x8700	Error		Channel/Interface not calibrated	Channel/interface not synchronized
0x8701	Error		Operating time was manipulated	Operating time was manipulated
0x8702	Error		Oversampling setting is not possible	Oversampling setting not possible
0x8703	Error		No slave controller found	No slave controller found
0x8704	Error		Slave controller is not in Bootstrap	Slave controller is not in bootstrap
0x8705	Error		Processor usage to high (>= 100%%)	Processor load too high (>= 100%%)
0x8706	Error		Channel in saturation	Channel in saturation
0x8707	Error		Channel overload	Channel overload
0x8708	Error		Overloadtime was manipulated	Overload time was manipulated
0x8709	Error		Saturationtime was manipulated	Saturation time was manipulated
0x870A	Error		Channel range error	Measuring range error for the channel
0x870B	Error		no ADC clock	No ADC clock available
0xFFFF	Information		Debug: 0x%X, 0x%X, 0x%X	Debug: 0x%X, 0x%X, 0x%X

5.11 Object description and parameterization

EtherCAT XML Device Description

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

Parameterization via the CoE list (CAN over EtherCAT)

The EtherCAT device is parameterized via the CoE-Online tab [▶ 87] (double-click on the respective object) or via the Process Data tab [▶ 84](allocation of PDOs). Please note the following general CoE notes [▶ 43] when using/manipulating the CoE parameters:

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

5.11.1 Profile-specific objects

Index 6000 DOX Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
6000:0	DOX Inputs Ch.1	Max. Subindex	UINT8	RO	0x12 (18_{dec})
6000:01	Ready to activate	Driver stage is ready for activation	BOOLEAN	RO	0x00 (0_{dec})
6000:02	Output active	Output is activated	BOOLEAN	RO	0x00 (0_{dec})
6000:07	Warning	a warning has occurred (see index 0xA000)	BOOLEAN	RO	0x00 (0_{dec})
6000:08	Error	an error has occurred (see index 0xA000)	BOOLEAN	RO	0x00 (0_{dec})
6000:0C	Digital input	Digital input	BOOLEAN	RO	0x00 (0_{dec})
6000:10	TxPDO Toggle	Toggle bit	BOOLEAN	RO	0x00 (0_{dec})
6000:11	Info data 1	Synchronous information (selection via subindex 0x8001:11)	UINT16	RO	0x0000 (0_{dec})
6000:12	Info data 2	Synchronous information (selection via subindex 0x8001:19)	UINT16	RO	0x0000 (0_{dec})

Index 7000 DOX Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
7000:0	DOX Outputs Ch.1	Max. Subindex	UINT8	RO	0x17 (23_{dec})
7000:01	Enable	Activates the buck controller for generating the output voltage.	BOOLEAN	RO	0x00 (0_{dec})
7000:02	Output	Activates the output.	BOOLEAN	RO	0x00 (0_{dec})
7000:04	Input trigger enable	The trigger input is enabled with this bit.	BOOLEAN	RO	0x00 (0_{dec})
7000:05	Output trigger enable	The trigger output is enabled or switched with this bit.	BOOLEAN	RO	0x00 (0_{dec})
7000:08	Reset	Following an error, or if the Ready bit in the Status is not set, a rising edge in the Reset bit resets or restarts the output stage.	BOOLEAN	RO	0x00 (0_{dec})
7000:11	Output current	Specification of the set current in mA.	UINT16	RO	0x0000 (0_{dec})
7000:12	Impulse length	Specification of the pulse duration in µs.	UINT32	RO	0x00000000 (0_{dec})
7000:13	Trigger delay	Specification of the delay time of the trigger input in µs.	UINT32	RO	0x00000000 (0_{dec})
7000:14	PWM duty	Specification of the duty cycle (32767=100%).	UINT16	RO	0x0000 (0_{dec})
7000:15	Output voltage	Specification of the output voltage in 0.01 V.	UINT16	RO	0x0000 (0_{dec})
7000:16	DC Start time	Timestamp for the start time in DC mode.	UINT64	RO	0x0000000000000000 (0_{dec})
7000:17	Tigger out delay	Specification of the delay time of the trigger output in µs.	UINT32	RO	0x00000000 (0_{dec})

Index 8000 DOX Settings Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
8000:0	DOX Settings Ch.1	Max. Subindex	UINT8	RO	0x16 (22$_{dec}$)
8000:02	Target current	Nominal current or maximum current intensity if PDO 0x1601 is mapped. This is then the maximum current intensity that can be set via 0x1601. Adjustable values: 0.. 3000 (0…3 A)	UINT16	RW	0x01F4 (500$_{dec}$)
8000:03	Supply voltage	Setting the input voltage for monitoring during operation. Adjustable values EL2596: 2000..2880 (0…28.8 V) EL2596-0010: 2000..5760 (0…57.6 V)	UINT16	RW	0x0960 (2400$_{dec}$)
8000:04	Output voltage	Setting the nominal LED voltage or output voltage. Use depends on the set operation mode. In pulse mode (OPV), the set voltage and an additional voltage swing are controlled. This is the nominal voltage of the LED output in “Current Control” mode. Adjustable values EL2596: 0..2780 (0…27.8 V) EL2596-0010: 0..4800 (0…48.0 V)	UINT16	RW	0x04B0 (1200$_{dec}$)
8000:0B	Trigger delay (switch on)	Trigger input delay in 100 ns. Adjustable values: 0.. 100 000 000 (0…10 s)	UINT32	RW	0x00000000 (0$_{dec}$)
8000:11	Warning level (supply voltage)	This entry sets the warning level as a percentage of the input voltage. If the input voltage falls below the warning value, a message is displayed Adjustable values: 0..100 %	UINT8	RW	0x05 (5$_{dec}$)
8000:12	Error level (supply voltage)	This entry sets the error level as a percentage of the input voltage. If the input voltage falls below the error value, an error is displayed Adjustable values: 0..100 %	UINT8	RW	0x14 (20$_{dec}$)
8000:13	Positive warning level (output voltage)	This entry sets the warning level as a percentage of the output voltage. Adjustable values: 0..100 %	UINT8	RW	0x05 (5$_{dec}$)
8000:14	Positive error level (output voltage)	This entry sets the error level as a percentage of the output voltage. (serves to detect a wire breakage) Adjustable values: 0..100 %	UINT8	RW	0x14 (20$_{dec}$)
8000:15	Negative warning level (output voltage)	The negative warning level for the detection of a short-circuit is specified with this entry. Adjustable values: 0..100 %	UINT8	RW	0x05 (5$_{dec}$)
8000:16	Negative error level (output voltage)	The negative error level for the detection of a short-circuit is specified with this entry. Adjustable values: 0..100 %	UINT8	RW	0x0A (10$_{dec}$)

Index 8001 DOX Controller Settings Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
8001:0	DOX Controller Settings Ch.1	Max. Subindex	UINT8	RO	0x03 (3$_{dec}$)
8001:01	Kp factor (curr.)	P control parameter should only be adjusted in exceptional cases	UINT16	RW	0x0064 (100$_{dec}$)
8001:02	Ki factor (curr.)	I control parameter should only be adjusted in exceptional cases.	UINT16	RW	0x0000 (0$_{dec}$)
8001:03	Kd factor (curr.)	D control parameter should only be adjusted in exceptional cases.	UINT16	RW	0x0014 (20$_{dec}$)

Index 8002 DOX Features Ch.1

Index (hex)	Name	Meaning		Data type	Flags	Default
8002:0	DOX Features Ch.1	Max. Subindex		UINT8	RO	0x35 (50_{dec})
8002:11	Select info data 1	permitted values:		UINT8	RW	0x07 (7_{dec})
		0	Status word			
		7	Output voltage			
		8	Output current			
		101	Internal temperature (at the processor)			
		104	Supply voltage			
		105	Supply current			
		106	Led N voltage			
		107	Power dissipation			
		108	Internal temperature 2 (at the LED driver)			
8002:19	Select info data 2	permitted values:		UINT8	RW	0x08 (8_{dec})
		0	Status word			
		7	Output voltage			
		8	Output current			
		101	Internal temperature (at the processor)			
		104	Supply voltage			
		105	Supply current			
		106	Led N voltage			
		107	Power dissipation			
		108	Internal temperature 2 (at the LED driver)			
8002:30	Invert digital input	Activates the inversion of the digital input		BOOLEAN	RW	0x00 (0_{dec})
8002:31	Pulse resolution 100 ns	The resolution of the PDO pulse and delay time specifications in 0x1602, 0x1603 and 0x1607 is 1 µs if FALSE and 100 ns if TRUE.		BOOLEAN	RW	0x00 (0_{dec})
8002:32	Function for input	permitted values:		BIT4	RW	0x00 (0_{dec})
		0	Trigger input			
		1	Hardware enable			

Index 8004 DOX enh. Features

Index (hex)	Name	Meaning	Data type	Flags	Default
8004:0	DOX enh. Features	Max. Subindex	UINT8	RO	0x09 (9_{dec})
8004:01	LED operation mode	permitted values: 0 Current control 1 Current control timestamp pulse 2 Current control trigger pulse 3 Current control PLC pulse 4 Current control PWM 5 Voltage control 6 Voltage control PWM 7 Current sink PWM	UINT8	RW	0x00 (0_{dec})
8004:02	Trigger output mode	permitted values: 0 Synchron 1 Synchron inverted 2 Synchron pulse 3 Standard output 4 Synchron pulse inverted 5 Synchron pulse with separate trigger delay 6 Synchron pulse inverted with separate trigger delay 7 Error bit 8 Warning bit	BIT4	RW	0x00 (0_{dec})
8004:03	Trigger in level	permitted values: 0 Low 1 High	BIT4	RW	0x01 (1_{dec})
8004:04	PWM frequency	Frequency specification in PWM mode. Adjustable values: 1..10000 Hz	UINT16	RW	0x041A (1050_{dec})
8004:05	Trigger out pulse length	Pulse width if the operation mode is set in the trigger output mode. Resolution 100 ns Adjustable values: 100.. 100 000 000 (10 µs…10 s)	UINT32	RW	0x000003E8 (1000_{dec})
8004:06	Trigger out delay	Delay of the trigger output in 100 ns if it is configured via Trigger output mode (0x8004:02) as an output with separate delay time and the latter is not mapped as PDO. Adjustable values: 0.. 100 000 000 (0…10 s)	UINT32	RW	0x00000000 (0_{dec})
8004:07	PWM duty	Specification of the Duty Cycle if not mapped in PDO. Adjustable values: 0..32767	UINT16	RW	0x7FFF (32767_{dec})
8004:08	Pulse voltage adj.	Additional voltage increase for shaping the output current in pulse mode. Intended, for example, for commissioning an LED via the TeachIn command. With this command, the voltage required for the set current, which can be different from LED to LED even in the same series, is automatically determined and saved. The value is added to the necessary automatic voltage increase of 2..3 V. Adjustable values: 0..500 (0..5 V)	UINT16	RW	0x0000 (0_{dec})
8004:09	Trigger input blind time	The inactive time of the trigger input from the trigger start time to the end of the pulse output plus “Trigger input blind time” [µs] Adjustable values: 20..65535 µs	UINT16	RW	0x32 (50_{dec})

Index 800F DOX Vendor data

Index (hex)	Name	Meaning	Data type	Flags	Default
800F:0	DOX Vendor data	Max. Subindex	UINT8	RO	0x0A (10_{dec})
800F:01	Offset (output voltage)	Vendor-specific calibration parameter	INT16	RW	0x0000 (0_{dec})
800F:02	Gain (output voltage)	Vendor-specific calibration parameter	UINT16	RW	0x433F (17215_{dec})
800F:03	Offset (output current)	Vendor-specific calibration parameter	INT16	RW	0x0000 (0_{dec})
800F:04	Gain (output current)	Vendor-specific calibration parameter	UINT16	RW	0x2E18 (11800_{dec})
800F:05	Offset (pulse current)	Vendor-specific calibration parameter	INT16	RW	0x0168 (360_{dec})
800F:06	Gain (pulse current)	Vendor-specific calibration parameter	UINT16	RW	0x3CE3 (15587_{dec})
800F:07	Offset (supply voltage)	Vendor-specific calibration parameter	INT16	RW	0x0000 (0_{dec})
800F:08	Gain (supply voltage)	Vendor-specific calibration parameter	UINT16	RW	0x4000 (16384_{dec})
800F:09	Offset (supply current)	Vendor-specific calibration parameter	INT16	RW	0x0000 (0_{dec})
800F:0A	Gain (supply current)	Vendor-specific calibration parameter	UINT16	RW	0x4000 (16384_{dec})

Index 9000 DOX Info data Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
9000:0	DOX Info data Ch.1	Max. Subindex	UINT8	RO	0x12 (18_{dec})
9000:01	Status word	The DOX Diag data 0xA000 exists here as a word.	UINT16	RO	0x0000 (0_{dec})
9000:08	Output voltage	The present output voltage in 0.01 V.	UINT16	RO	0x0000 (0_{dec})
9000:09	Output current	The present output current in mA.	UINT16	RO	0x0000 (0_{dec})
9000:11	Operating hour counter	Operating hour counter of the LED output in minutes.	UINT32	RO	0x00000000 (0_{dec})
9000:12	Switching operation counter	Counter for the LED output switching procedures.	UINT32	RO	0x00000000 (0_{dec})

Index A000 DOX Diag data Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
A000:0	DOX Diag data Ch.1	Max. Subindex	UINT8	RO	0x0B (11_{dec})
A000:01	Saturated	The controller is saturated; a further increase in current/ voltage is not possible.	BOOLEAN	RO	0x00 (0_{dec})
A000:02	Over temperature	A temperature error has occurred.	BOOLEAN	RO	0x00 (0_{dec})
A000:04	Under voltage (Supply)	The input voltage is too low.	BOOLEAN	RO	0x00 (0_{dec})
A000:05	Over voltage (Supply)	The input voltage is too high.	BOOLEAN	RO	0x00 (0_{dec})
A000:06	Short circuit	A short-circuit has been detected.	BOOLEAN	RO	0x00 (0_{dec})
A000:07	Open load	A wire breakage has been detected.	BOOLEAN	RO	0x00 (0_{dec})
A000:09	Misc error	Internal error	BOOLEAN	RO	0x00 (0_{dec})
A000:0B	Over voltage (Output)	The necessary voltage for the LED output for operation with the set current is too high.	BOOLEAN	RO	0x00 (0_{dec})

Index F000 Modular device profile

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

Index F008 Code word

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

Index F081 Download revision

Index (hex)	Name	Meaning	Data type	Flags	Default
F081:0	Download revision	Max. Subindex	UINT8	RO	0x01 (1_{dec})
F081:01	Revision number	The subindex 0xF081:01 (Download revision) describes the revision level of the terminal / module.	UINT32	RW	0x00000000 (0_{dec})

Index F80F DOX Vendor data

Index (hex)	Name	Meaning	Data type	Flags	Default
F80F:0	DOX Vendor data	Max. Subindex	UINT8	RO	0x0C (12_{dec})
F80F:01	PWM Frequency Buck	Frequency with which the FET half bridge is driven.	UINT32	RW	0x0001D4C0 (120000_{dec})
F80F:02	Deadtime	Dead time of the two buck FETs	UINT16	RW	0x0014 (20_{dec})
F80F:04	Warning temperature	On reaching the temperature, a warning is displayed via LEDs and the status byte. The HW remains active.	INT8	RW	0x50 (80_{dec})
F80F:05	Switch off temperature	On reaching the temperature, an error is displayed via LEDs and the status byte. The HW is deactivated.	INT8	RW	0x64 (100_{dec})
F80F:07	SYNC 0 shift time		INT16	RW	0x0000 (0_{dec})
F80F:0B	Current filter	Number of values over which the output current is filtered.	UINT16	RW	0x000A (10_{dec})
F80F:0C	Voltage filter	Number of values over which the output voltage is filtered.	UINT16	RW	0x0014 (20_{dec})

Index F900 DOX Info data

Index (hex)	Name	Meaning	Data type	Flags	Default
F900:0	DOX Info data	Max. Subindex	UINT8	RO	0x13 (19_{dec})
F900:02	Internal temperature	Internal terminal temperature in °C (at the processor)	INT8	RO	0x00 (0_{dec})
F900:03	Internal temperature 2	Internal terminal temperature in °C (at the LED driver)	INT8	RO	0x00 (0_{dec})
F900:05	Supply voltage	Supply voltage	UINT16	RO	0x0000 (0_{dec})
F900:06	Supply current	Supply current	UINT16	RO	0x0000 (0_{dec})
F900:07	Buck CC Unit	Resolution of the buck timer	UINT16	RO	0x0000 (0_{dec})
F900:08	Temp1 Raw Value	Raw value of the MC temperature measurement	UINT16	RO	0x0000 (0_{dec})
F900:09	Temp2 Raw Value	Raw value of the MC temperature measurement	UINT16	RO	0x0000 (0_{dec})
F900:0A	Voltage Raw Value	Raw value of the MC supply voltage measurement.	UINT16	RO	0x0000 (0_{dec})
F900:0B	Current Raw Value	Raw value of the MC supply current measurement.	UINT16	RO	0x0000 (0_{dec})
F900:0C	MC temperature	Internal temperature of the STM32	INT16	RO	0x0000 (0_{dec})
F900:0D	MC reference	Internal reference voltage of the STM32	UINT16	RO	0x0000 (0_{dec})
F900:0E	U buck raw	unfiltered raw value of the output voltage measurement	UINT32	RO	0x00000000 (0_{dec})
F900:0F	I led raw	unfiltered raw value of the current measurement	UINT32	RO	0x00000000 (0_{dec})
F900:10	DAC raw	unfiltered raw value of the current measurement	UINT16	RO	0x0000 (0_{dec})
F900:11	U Led N raw	unfiltered raw value of the output voltage measurement	UINT32	RO	0x00000000 (0_{dec})
F900:12	Led N voltage	Voltage at the LED output of the terminal that drops in pulse/PWM operation. A high voltage leads to a corresponding power loss in the terminal. This value can be optimized via a TeachIn. The measuring range extends from 0 to approx. 4 V	UINT16	RO	0x0000 (0_{dec})
F900:13	Power dissipation	Power loss of the Buck controller	UINT16	RO	0x0000 (0_{dec})

Index FB00 DOX Command

Index (hex)	Name	Meaning	Data type	Flags	Default
FB00:0	DOX Command	Terminal-specific commands can be executed via DOX Command.	UINT8	RO	0x03 (3_{dec})
FB00:01	Request	• Commands: • 0x0001: Saving of the operating hour counter • 0x0002: Clearing of the operating hour counter • 0x0501: the output voltage determined at the set current is stored (in “Current Control” mode) • 0x0502: Perform Teachin of an LED with storage of the voltage in the EEPROM (in 8000:04) • 0x0503: Teachin of an LED without storing the voltage in the EEPROM, i.e. temporarily	OCTET-STRING[2]	RW	{0}
FB00:02	Status		UINT8	RO	0x00 (0_{dec})
FB00:03	Response		OCTET STRING[4]	RO	{0}

5.11.2 Standard objects

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default
1000:0	Device type	Device type of the EtherCAT slave: The Lo-Word contains the CoE profile used (5001). The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	0x00D21389 (13767561_{dec})

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EL2596

Index 1009 Hardware version

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

Index 100A Software version

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

Index 100B Bootloader version

Index (hex)	Name	Meaning	Data type	Flags	Default
100B:0	Bootloader version	Bootloader version of the EtherCAT slave	STRING	RO	N/A

Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters [▸ 260]	Restore default parameters	UINT8	RO	0x01 (1_{dec})
1011:01	SubIndex 001	If this object is set to "**0x64616F6C**" in the set value dialog, all backup objects are reset to their delivery state.	UINT32	RW	0x00000000 (0_{dec})

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Information for identifying the slave	UINT8	RO	0x04 (4_{dec})
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2_{dec})
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x0A233052 (170078290_{dec})
1018:03	Revision	Revision number of the EtherCAT slave; the Low Word (bit 0-15) indicates the special terminal number, the High Word (bit 16-31) refers to the device description	UINT32	RO	0x00000000 (0_{dec})
1018:04	Serial number	Serial number of the EtherCAT slave; the Low Byte (bit 0-7) of the Low Word contains the year of production, the High Byte (bit 8-15) of the Low Word contains the week of production, the High Word (bit 16-31) is 0	UINT32	RO	0x00000000 (0_{dec})

Index 10F0 Backup parameter handling

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

Index 10F3 Diagnosis History

Index (hex)	Name	Meaning	Data type	Flags	Default
10F3:0	Diagnosis History	Max. Subindex	UINT8	RO	0x37 (55_{dec})
10F3:01	Maximum Messages	Maximum number of stored messages A maximum of 50 messages can be stored	UINT8	RO	0x00 (0_{dec})
10F3:02	Newest Message	Subindex of the latest message	UINT8	RO	0x00 (0_{dec})
10F3:03	Newest Acknowledged Message	Subindex of the last confirmed message	UINT8	RW	0x00 (0_{dec})
10F3:04	New Messages Available	Indicates that a new message is available	BOOLEAN	RO	0x00 (0_{dec})
10F3:05	Flags	not used	UINT16	RW	0x0000 (0_{dec})
10F3:06	Diagnosis Message 001	Message 1	OCTET STRING[28]	RO	{0}
...	...	...	OCTET STRING[28]	RO	{0}
10F3:23	Diagnosis Message 030	Message 30	OCTET STRING[28]	RO	{0}

Index 10F8 Actual Time Stamp

Index (hex)	Name	Meaning	Data type	Flags	Default
10F8:0	Actual Time Stamp	Timestamp	UINT64	RO	

Index 1401 DOX RxPDO-Par Current

Index (hex)	Name	Meaning	Data type	Flags	Default
1401:0	DOX RxPDO-Par Current	PDO Parameter RxPDO 2	UINT64	RO	0x06 (6_{dec})
1401:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 2	OCTET-STRING[2]	RO	05 16

Index 1405 DOX RxPDO-Par Voltage

Index (hex)	Name	Meaning	Data type	Flags	Default
1405:0	DOX RxPDO-Par Voltage	PDO Parameter RxPDO 6	UINT64	RO	0x06 (6_{dec})
1405:06	Exclude RxPDOs	Specifies the RxPDOs (index of RxPDO mapping objects) that must not be transferred together with RxPDO 6	OCTET-STRING[2]	RO	01 16

Index 1600 DOX RxPDO-Map Control

Index (hex)	Name	Meaning	Data type	Flags	Default
1600:0	DOX RxPDO-Map Control	PDO Mapping RxPDO 1	UINT8	RO	0x08 (8_{dec})
1600:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (DOX Outputs Ch.1), entry 0x01 (Enable))	UINT32	RO	0x7000:01, 1
1600:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (DOX Outputs Ch.1), entry 0x02 (Output))	UINT32	RO	0x7000:02, 1
1600:03	SubIndex 003	3. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1600:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (DOX Outputs Ch.1), entry 0x04 (Input trigger enable))	UINT32	RO	0x7000:04, 1
1600:05	SubIndex 005	5. PDO Mapping entry (object 0x7000 (DOX Outputs Ch.1), entry 0x05 (Output trigger enable))	UINT32	RO	0x7000:05, 1
1600:06	SubIndex 006	6. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1600:07	SubIndex 007	7. PDO Mapping entry (object 0x7000 (DOX Outputs Ch.1), entry 0x08 (Reset))	UINT32	RO	0x7000:08, 1
1600:08	SubIndex 008	8. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8

Index 1601 DOX RxPDO-Map Current

Index (hex)	Name	Meaning	Data type	Flags	Default
1601:0	DOX RxPDO-Map Current	PDO Mapping RxPDO 2	UINT8	RO	0x01 (1_{dec})
1601:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (DOX Outputs Ch.1), entry 0x11 (Output current))	UINT32	RO	0x7000:11, 16

Index 1602 DOX RxPDO-Map Impulse length

Index (hex)	Name	Meaning	Data type	Flags	Default
1602:0	DOX RxPDO-Map Impulse length	PDO Mapping RxPDO 3	UINT8	RO	0x01 (1_{dec})
1602:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (DOX Outputs Ch.1), entry 0x12 (Impulse length))	UINT32	RO	0x7000:12, 32

Index 1603 DOX RxPDO-Map Trigger delay

Index (hex)	Name	Meaning	Data type	Flags	Default
1603:0	DOX RxPDO-Map Trigger delay	PDO Mapping RxPDO 4	UINT8	RO	0x01 (1_{dec})
1603:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (DOX Outputs Ch.1), entry 0x13 (Trigger delay))	UINT32	RO	0x7000:13, 32

Index 1604 DOX RxPDO-Map PWM duty

Index (hex)	Name	Meaning	Data type	Flags	Default
1604:0	DOX RxPDO-Map PWM duty	PDO Mapping RxPDO 5	UINT8	RO	0x01 (1_{dec})
1604:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (DOX Outputs Ch.1), entry 0x14 (PWM duty))	UINT32	RO	0x7000:14, 16

Index 1605 DOX RxPDO-Map Voltage

Index (hex)	Name	Meaning	Data type	Flags	Default
1605:0	DOX RxPDO-Map Voltage	PDO Mapping RxPDO 6	UINT8	RO	0x01 (1_{dec})
1605:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (DOX Outputs Ch.1), entry 0x15 (Output voltage))	UINT32	RO	0x7000:15, 16

Index 1606 DOX RxPDO-Map DC start time

Index (hex)	Name	Meaning	Data type	Flags	Default
1606:0	DOX RxPDO-Map DC start time	PDO Mapping RxPDO 7	UINT8	RO	0x01 (1_{dec})
1606:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (DOX Outputs Ch.1), entry 0x16 (DC Start time))	UINT32	RO	0x7000:16, 64

Index 1607 DOX RxPDO-Map Trigger out delay

Index (hex)	Name	Meaning	Data type	Flags	Default
1607:0	DOX RxPDO-Map Trigger out delay	PDO Mapping RxPDO 8	UINT8	RO	0x01 (1_{dec})
1607:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (DOX Outputs Ch.1), entry 0x17 (Trigger out delay))	UINT32	RO	0x7000:17, 16

Index 1A00 DOX TxPDO-Map Status

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00:0	DOX TxPDO-Map Status	PDO Mapping TxPDO 1	UINT8	RO	0x0A (10_{dec})
1A00:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (DOX Inputs Ch.1), entry 0x01 (Ready to activate))	UINT32	RO	0x6000:01, 1
1A00:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (DOX Inputs Ch.1), entry 0x02 (Output active))	UINT32	RO	0x6000:02, 1
1A00:03	SubIndex 003	3. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A00:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (DOX Inputs Ch.1), entry 0x04 (Switching operation counter))	UINT32	RO	0x6000:04, 3
1A00:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (DOX Inputs Ch.1), entry 0x07 (Warning))	UINT32	RO	0x6000:07, 1
1A00:06	SubIndex 006	6. PDO Mapping entry (object 0x6000 (DOX Inputs Ch.1), entry 0x08 (Error))	UINT32	RO	0x6000:08, 1
1A00:07	SubIndex 007	7. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1A00:08	SubIndex 008	8. PDO Mapping entry (object 0x6000 (DOX Inputs Ch.1), entry 0x0C (Digital input))	UINT32	RO	0x6000:0C, 1
1A00:09	SubIndex 009	9. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1A00:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (DOX Inputs Ch.1), entry 0x10 (TxPDO Toggle))	UINT32	RO	0x6000:10, 1

Index 1A01 DOX TxPDO-Map Synchron info data

Index (hex)	Name	Meaning	Data type	Flags	Default
1A01:0	DOX TxPDO-Map Synchron info data	PDO Mapping TxPDO 2	UINT8	RO	0x02 (2_{dec})
1A01:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (DOX Inputs Ch.1), entry 0x11 (Info data 1))	UINT32	RO	0x6000:11, 16
1A01:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (DOX Inputs Ch.1), entry 0x12 (Info data 2))	UINT32	RO	0x6000:12, 16

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x01 (1_{dec})
1C12:01	Subindex 001	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1600 (5632_{dec})
1C12:02	Subindex 002	2. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C12:03	Subindex 003	3. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C12:04	Subindex 004	4. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C12:05	Subindex 005	5. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C12:06	Subindex 006	6. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C12:07	Subindex 007	7. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x01 (1_{dec})
1C13:01	Subindex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A00 (6656_{dec})
1C13:02	Subindex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})

Index 1C32 SM output parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20 (32_{dec})
1C32:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 2 Event • 2: DC-Mode - Synchron with SYNC0 Event • 3: DC-Mode - Synchron with SYNC1 Event	UINT16	RW	0x0001 (1_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchron with SM 2 Event: Master cycle time • DC mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x000F4240 (1000000_{dec})
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0 = 1: free run is supported • Bit 1 = 1: Synchron with SM 2 Event is supported • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 10: Output Shift with SYNC1 event (only DC mode) • Bit 14 = 1: dynamic times (measurement through writing of 1C32:08)	UINT16	RO	0x0002 (2_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x000186A0 (100000_{dec})
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:07	Minimum delay time		UINT32	RO	0x00000000 (0_{dec})
1C32:08	Get Cycle Time	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 0x1C32:03 [▸ 227], 0x1C32:05 [▸ 227], 0x1C32:06 [▸ 227], 0x1C32:09 [▸ 227], 0x1C33:03, 0x1C33:06 [▸ 227], 0x1C33:09 are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RW	0x0000 (0_{dec})
1C32:09	Maximum delay time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0_{dec})
1C32:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	0x00 (0_{dec})

Index 1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32_{dec})
1C33:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 3 Event (no outputs available) • 2: DC - Synchron with SYNC0 Event • 3: DC - Synchron with SYNC1 Event • 34: Synchron with SM 2 Event (outputs available)	UINT16	RW	0x0022 (34_{dec})
1C33:02	Cycle time	as 0x1C32:02	UINT32	RW	0x000F4240 (1000000_{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: free run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron with SM 3 Event is supported (no outputs available) • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 01: Input Shift through local event (outputs available) • Bit 4-5 = 10: Input Shift with SYNC1 Event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08 or 0x1C33:08 [▶ 228])	UINT16	RO	0x0002 (2_{dec})
1C33:05	Minimum cycle time	as 0x1C32:05	UINT32	RO	0x000186A0 (100000_{dec})
1C33:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:07	Minimum delay time		UINT32	RO	0x00000000 (0_{dec})
1C33:08	Command	as 0x1C32:08	UINT16	RW	0x0000 (0_{dec})
1C33:09	Maximum delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:0B	SM event missed counter	as 0x1C32:11	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	as 0x1C32:12	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	as 0x1C32:13	UINT16	RO	0x0000 (0_{dec})
1C33:20	Sync error	as 0x1C32:32	BOOLEAN	RO	0x00 (0_{dec})

5.12 Sample 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.

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.

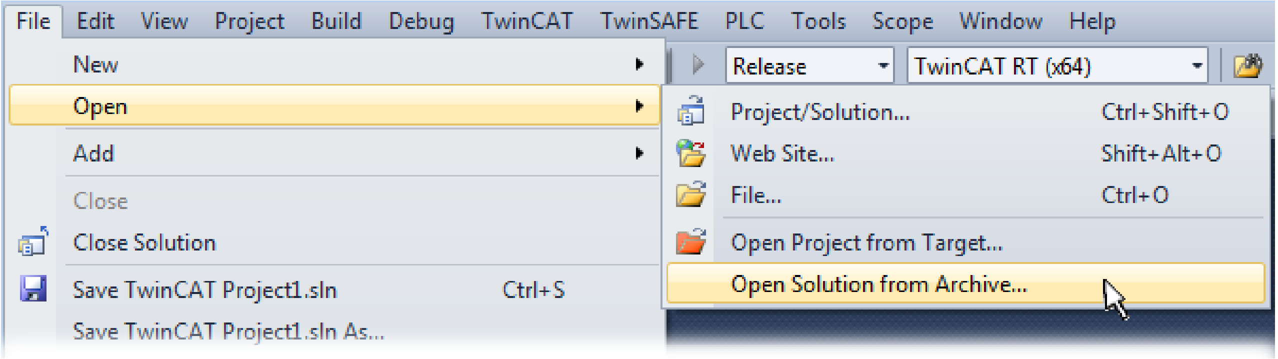
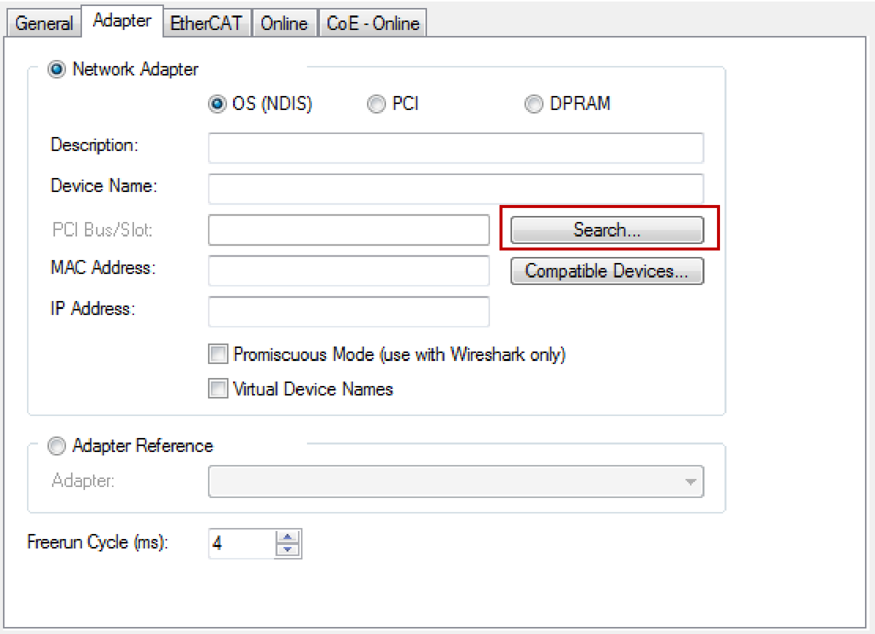


Fig. 233: Opening the *. tnzip archive

- 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...”:



● Network Adapter

● OS (NDIS) ○ PCI ○ DPRAM

□ Promiscuous Mode (use with Wireshark only)

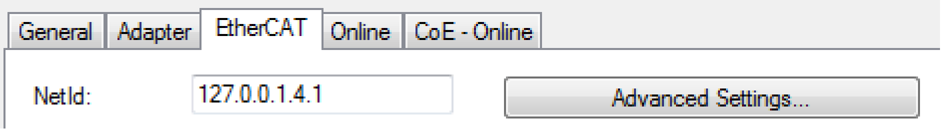
□ Virtual Device Names

○ Adapter Reference

Freerun Cycle (ms): 4

Fig. 234: 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:



NetId: 127.0.0.1.4.1

  The first four numbers must 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 must be entered; 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

5.12.1 Sample program 1 - short-circuit detection (current-controlled)

https://infosys.beckhoff.com/content/1033/el2596/Resources/8397411083/.zip

Program description and function

In addition to the classic use for controlling LEDs, the EL2596 can also be used for testing electrical contacts and other special applications. The current-controlled LED output can detect short-circuits or general current changes within a few milliseconds, display them as an error message and, if necessary, switch off. The test specimen to be tested, for example an electrical contact, must be connected potential-free to the LED output (connections 1 and 9).

In this sample, a device is connected to the EL2596-0000 (FW03), which in the normal state after enabling the output conducts a quiescent current of about 100 mA. If a short circuit also occurs, an error message is generated and the LED output of the EL2596 switches off after a short time.

The current/voltage curve during the short circuit can be seen in the following scope recording. The values “Output-Current” and “Output-Voltage” are read from the info data of the terminal. The time required to completely switch off the output is about 3 ms, the current increases briefly to approx. 700 mA.

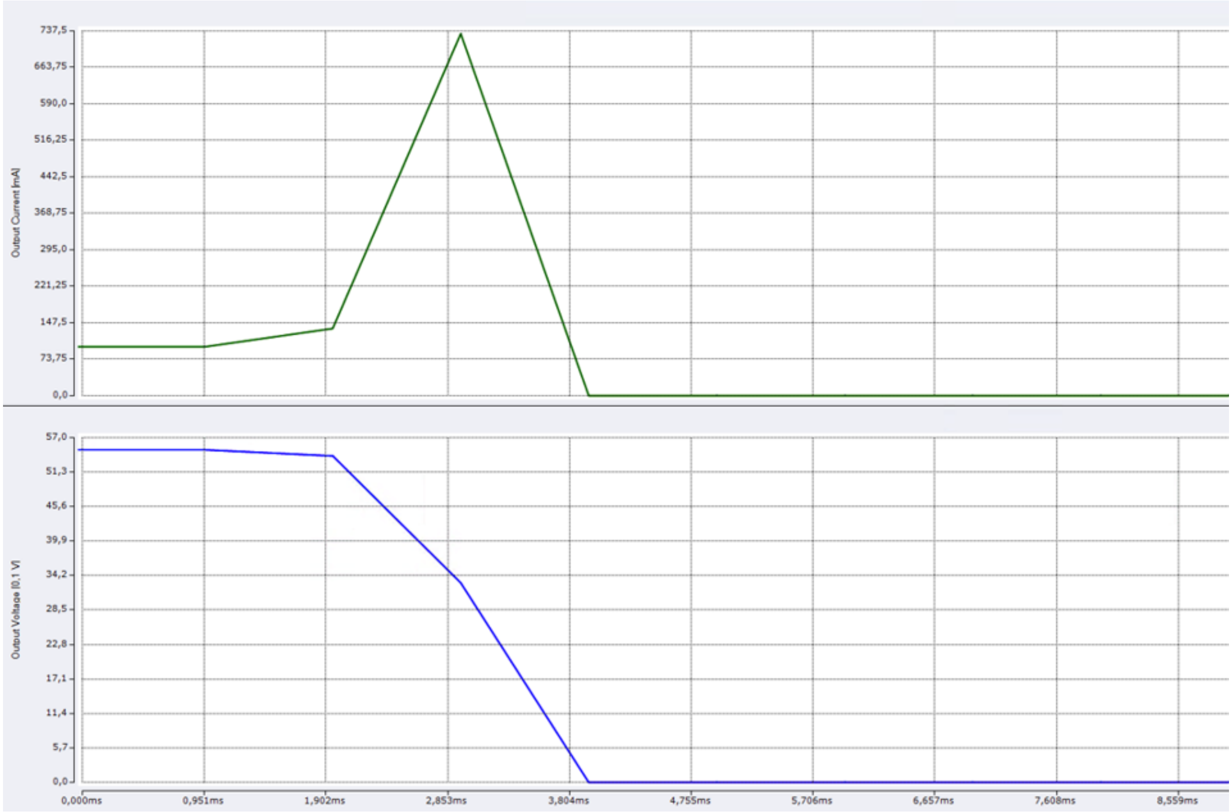
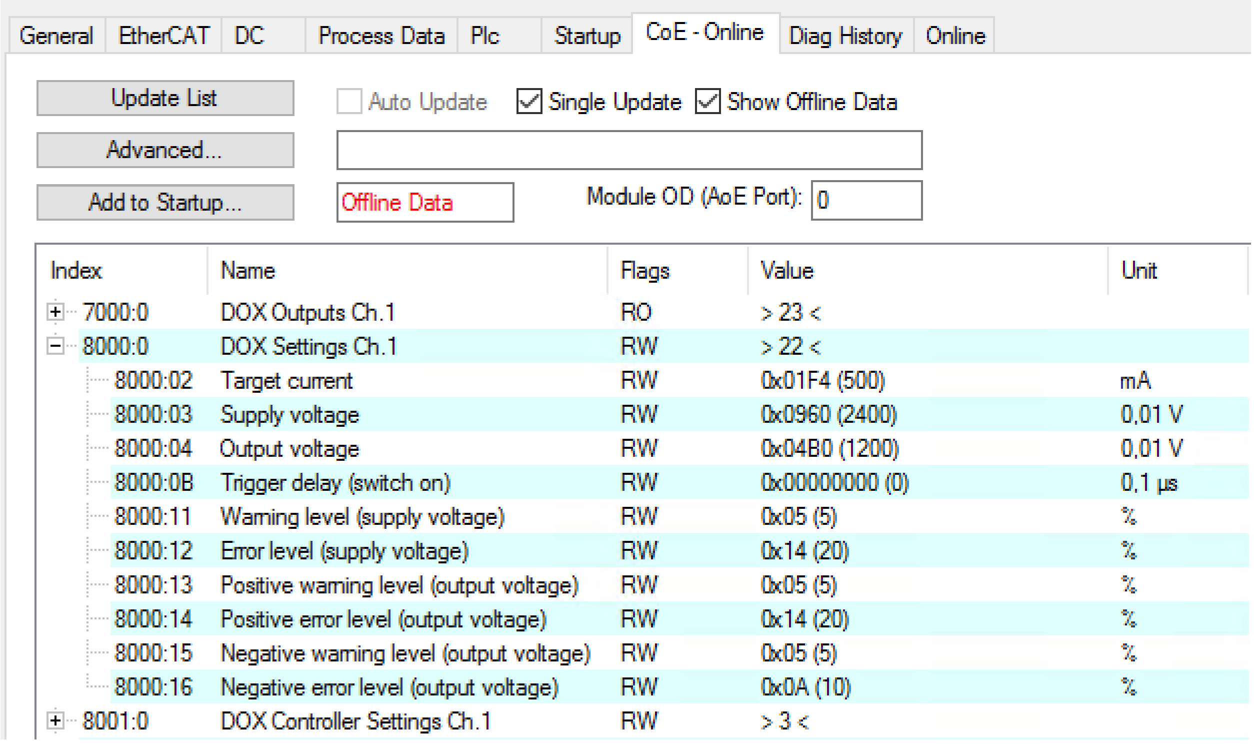


Fig. 235: Time for detection of a short-circuit

With this sample program, a short circuit can be detected at the LED output with the help of the EL2596. For this purpose, the EL2596 outputs a specified current at the LED output (operation mode: “Current Control”). If a short circuit is detected at the LED output, an error message is generated, which can be read and processed with this sample program.

For this purpose, as described in the chapter Commissioning [▶ 129] the values for the supply and output voltage, as well as the maximum output current must be specified in the CoE data.



□ Auto Update ☑ Single Update ☑ Show Offline Data

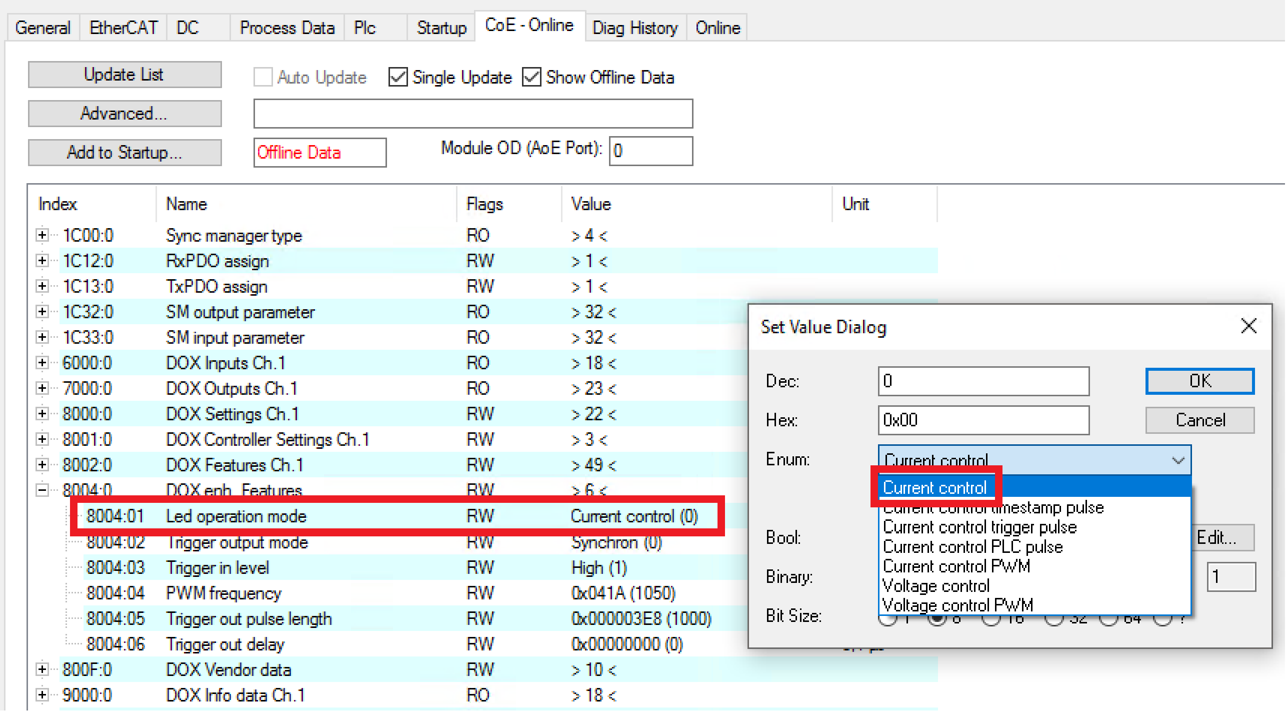
Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
+ 7000:0	DOX Outputs Ch.1	RO	> 23 <	
− 8000:0	DOX Settings Ch.1	RW	> 22 <	
8000:02	Target current	RW	0x01F4 (500)	mA
8000:03	Supply voltage	RW	0x0960 (2400)	0,01 V
8000:04	Output voltage	RW	0x04B0 (1200)	0,01 V
8000:0B	Trigger delay (switch on)	RW	0x00000000 (0)	0,1 µs
8000:11	Warning level (supply voltage)	RW	0x05 (5)	%
8000:12	Error level (supply voltage)	RW	0x14 (20)	%
8000:13	Positive warning level (output voltage)	RW	0x05 (5)	%
8000:14	Positive error level (output voltage)	RW	0x14 (20)	%
8000:15	Negative warning level (output voltage)	RW	0x05 (5)	%
8000:16	Negative error level (output voltage)	RW	0x0A (10)	%
+ 8001:0	DOX Controller Settings Ch.1	RW	> 3 <	

Fig. 236: Setting current and voltage values in the CoE

The specified output values are then output at the output of the LED terminal. “Current Control” mode must be selected as the output mode (CoE index 0x8004:01).



□ Auto Update ☑ Single Update ☑ Show Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
1C00:0	Sync manager type	RO	> 4 <	
1C12:0	RxPDO assign	RW	> 1 <	
1C13:0	TxPDO assign	RW	> 1 <	
1C32:0	SM output parameter	RO	> 32 <	
1C33:0	SM input parameter	RO	> 32 <	
6000:0	DOX Inputs Ch.1	RO	> 18 <	
7000:0	DOX Outputs Ch.1	RO	> 23 <	
8000:0	DOX Settings Ch.1	RW	> 22 <	
8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
8002:0	DOX Features Ch.1	RW	> 49 <	
8004:0	DOX enh. Features	RW	> 6 <	
8004:01	Led operation mode	RW	Current control (0)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	
8004:06	Trigger out delay	RW	0x00000000 (0)	
800F:0	DOX Vendor data	RW	> 10 <	
9000:0	DOX Info data Ch.1	RO	> 18 <	

Dec: 0

Hex: 0x00

Enum: Current control

- Current control
- Current control timestamp pulse
- Current control trigger pulse
- Current control PLC pulse
- Current control PWM
- Voltage control
- Voltage control PWM

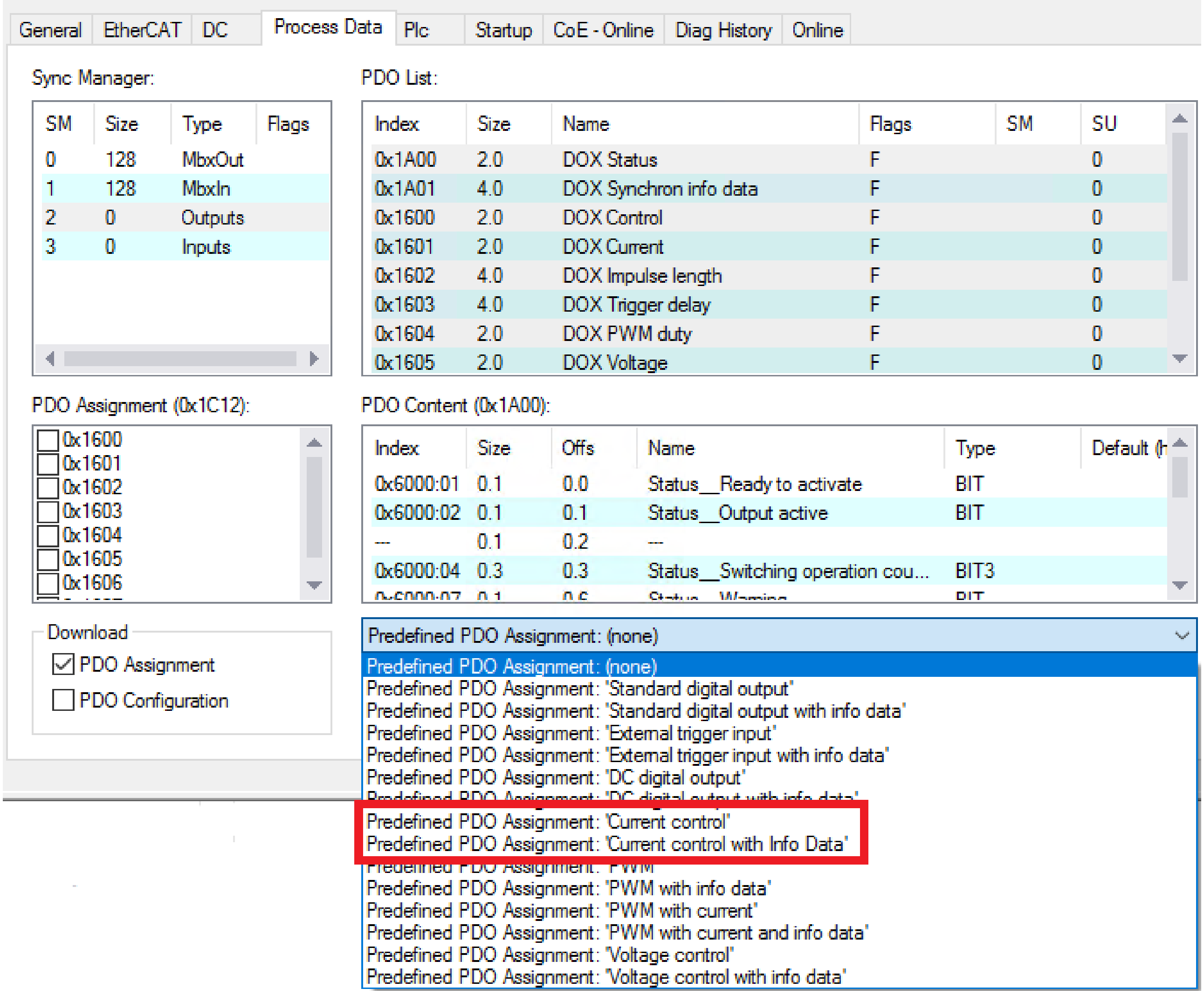
Bool:

Binary: 1

Bit Size: ○ 1 ● 8 ○ 16 ○ 32 ○ 64 ○ ?

Fig. 237: Selecting the “Current Control” operation mode

“Current Control (with info data)” is used as the Predefined PDO.



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	0	Outputs	
3	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F		0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F		0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0

PDO Assignment (0x1C12):

☐ 0x1600
☐ 0x1601
☐ 0x1602
☐ 0x1603
☐ 0x1604
☐ 0x1605
☐ 0x1606

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	

Download

☑ PDO Assignment

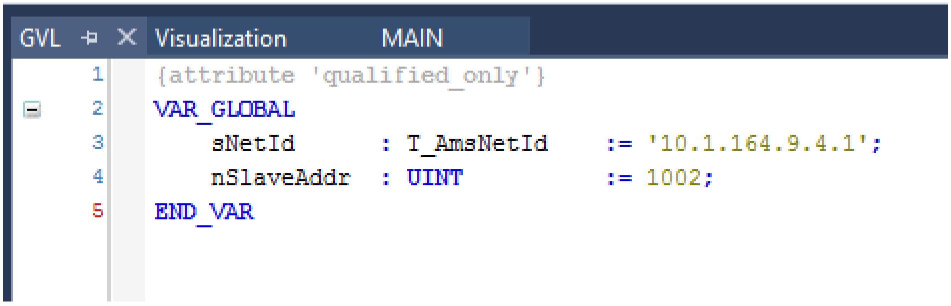
☐ PDO Configuration

Predefined PDO Assignment: (none)

Predefined PDO Assignment: (none)
Predefined PDO Assignment: 'Standard digital output'
Predefined PDO Assignment: 'Standard digital output with info data'
Predefined PDO Assignment: 'External trigger input'
Predefined PDO Assignment: 'External trigger input with info data'
Predefined PDO Assignment: 'DC digital output'
Predefined PDO Assignment: 'DC digital output with info data'
Predefined PDO Assignment: 'Current control'
Predefined PDO Assignment: 'Current control with Info Data'
Predefined PDO Assignment: 'PWM'
Predefined PDO Assignment: 'PWM with info data'
Predefined PDO Assignment: 'PWM with current'
Predefined PDO Assignment: 'PWM with current and info data'
Predefined PDO Assignment: 'Voltage control'
Predefined PDO Assignment: 'Voltage control with info data'

Fig. 238: Selection of the Predefined PDOs “Current Control with info data”

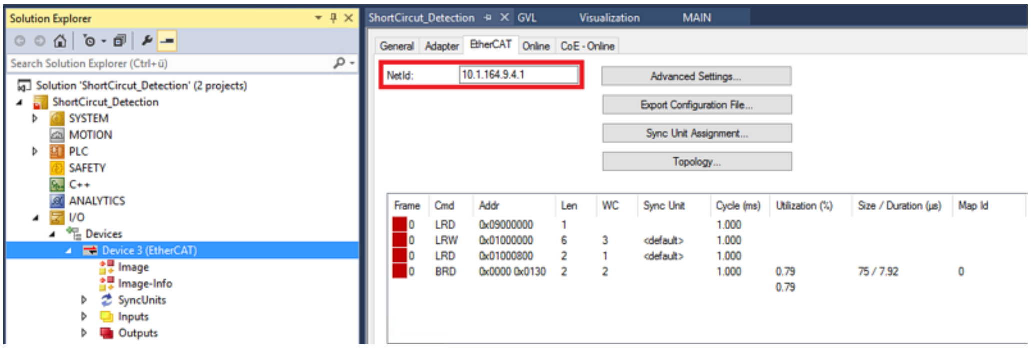
In the sample program, diagnostic data are read from the diagnostic messages of the terminal. To do this, the NetID of the EtherCAT network and the ID of the EL2596 must be specified in the global variable list (GVL).



```
{attribute 'qualified_only'}
VAR_GLOBAL
    sNetId       : T_AmsNetId    := '10.1.164.9.4.1';
    nSlaveAddr   : UINT          := 1002;
END_VAR
```

Fig. 239: Sample program 1 - GVL

The NetID can be displayed in TwinCAT under the EtherCAT Master on the “EtherCAT” tab. The ID must be specified in the GVL as sNetID in inverted commas as a string.

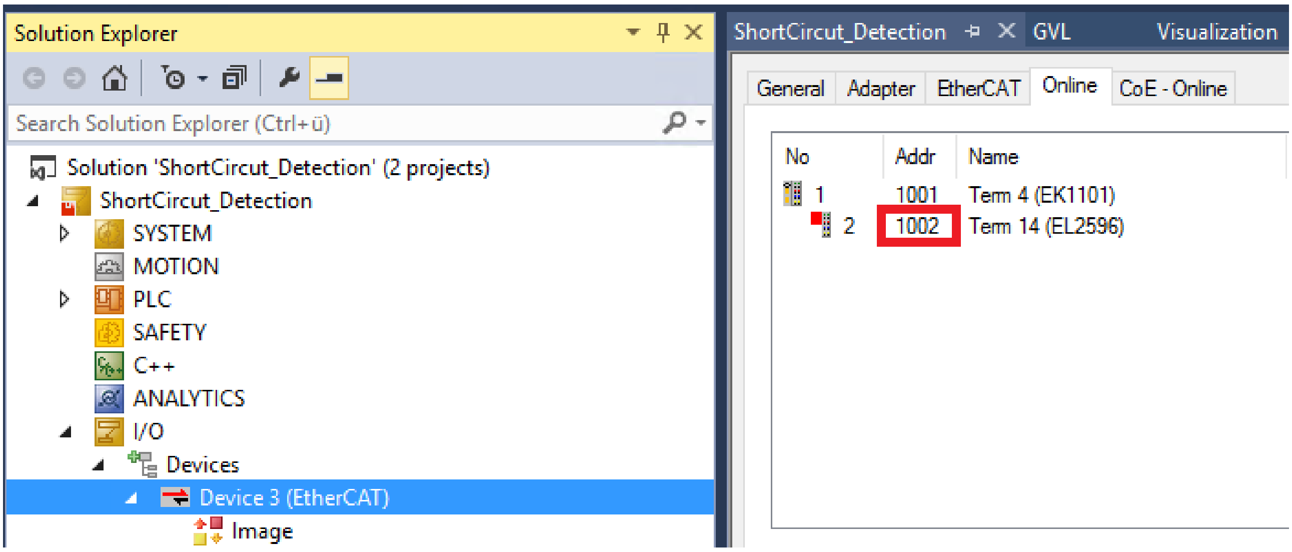


NetId: 10.1.164.9.4.1

Frame	Cmd	Addr	Len	WC	Sync Unit	Cycle (ms)	Utilization (%)	Size / Duration (µs)	Map Id
0	LRD	0x09000000	1			1.000			
0	LRW	0x01000000	6	3	<default>	1.000			
0	LRD	0x01000800	2	1	<default>	1.000			
0	BRD	0x0000 0x0130	2	2		1.000	0.79	75 / 7.92	0
							0.79		

Fig. 240: Displaying the NetID

The ID of the slave can also be found in the EtherCAT master. The “Online” tab must be selected in order to do this. This value must then be specified as nSlaveAddr.



No	Addr	Name
1	1001	Term 4 (EK1101)
2	1002	Term 14 (EL2596)

Fig. 241: Displaying the slave address

The sample program can then be started. To use the sample program, the included visualization should be opened.

- If the program is started and the terminal is not in an error state, “ReadyToActivate” is active and the lamp in the visualization is green.
- The desired current at the output must be set in the “Set Output Current” box. This can also be changed while the short-circuit detection is running.
- The short-circuit detection can then be started in the visualization via the “Start” button.
- The output is then switched on from the PLC, so that the “Output” lamp also turns green.
- The “Output Voltage” and “Output Current” text boxes display the present values of the output.
- If a falling edge occurs on the “ReadyToActivate” bit in the PLC program, the reading of the diagnostic information from the CoE objects 0xA000 and 0x10F3 is started. The output switches off immediately.
- If a short circuit has occurred, this is recognizable from the diagnostic data and the “Shortcircuit” lamp in the visualization lights up red.
- If there is no longer a short-circuit at the LED output, the EL2596 must be reset manually via the “Reset” button, which is linked to the “Reset” bit of the EL2596.
- The LED output is then reactivated.
- The short-circuit detection can be stopped manually via the “Stop” button; the LED output is then switched off.

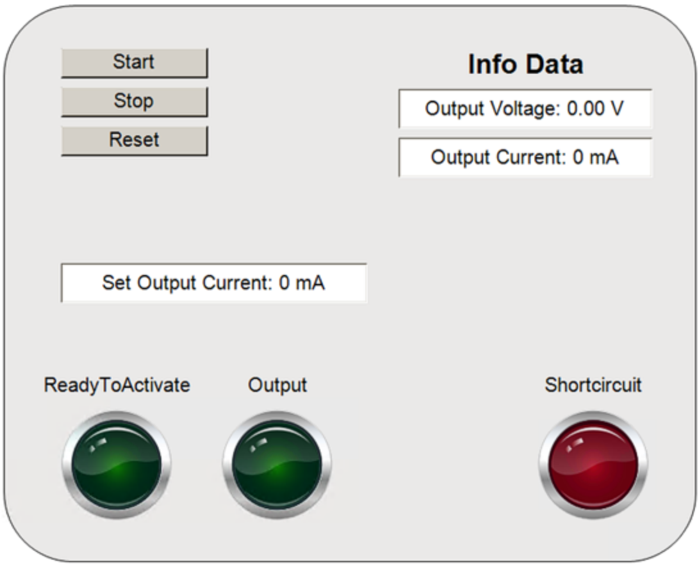


Fig. 242: Sample program 1 - Visualization

Limiting the peak current in case of short-circuit

If a short-circuit occurs at the LED output, currents >6 A can flow for a short time. In order to limit the peak output current, a resistor in the low ohm range can be connected in series with the component on which a short-circuit is to be detected.

5.12.2 Sample program 2 - short-circuit detection (voltage-controlled)

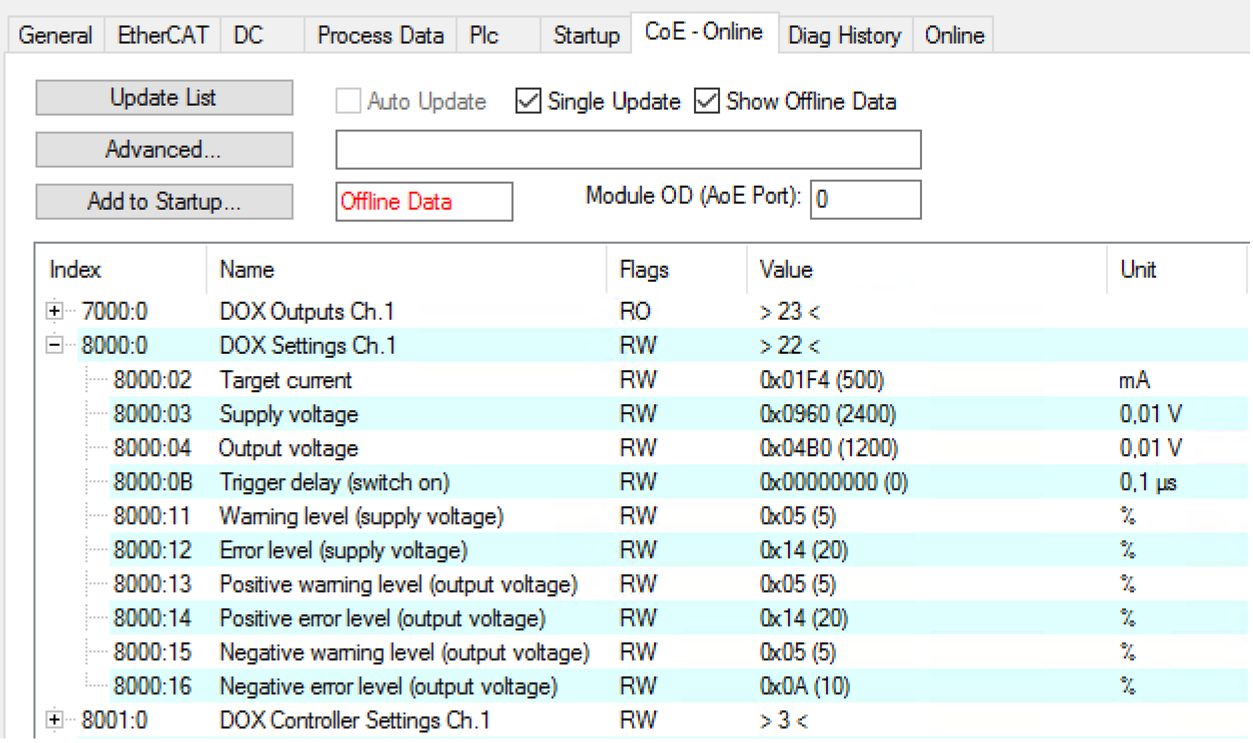
https://infosys.beckhoff.com/content/1033/el2596/Resources/8397414539/.zip

Program description and function

This sample program has the same function as sample program 1. Here, too, a short-circuit is to be detected with the help of the LED output of the EL2596.

However, this sample program 2 uses the “Voltage Control” operation mode, so that a specified voltage is output at the LED output.

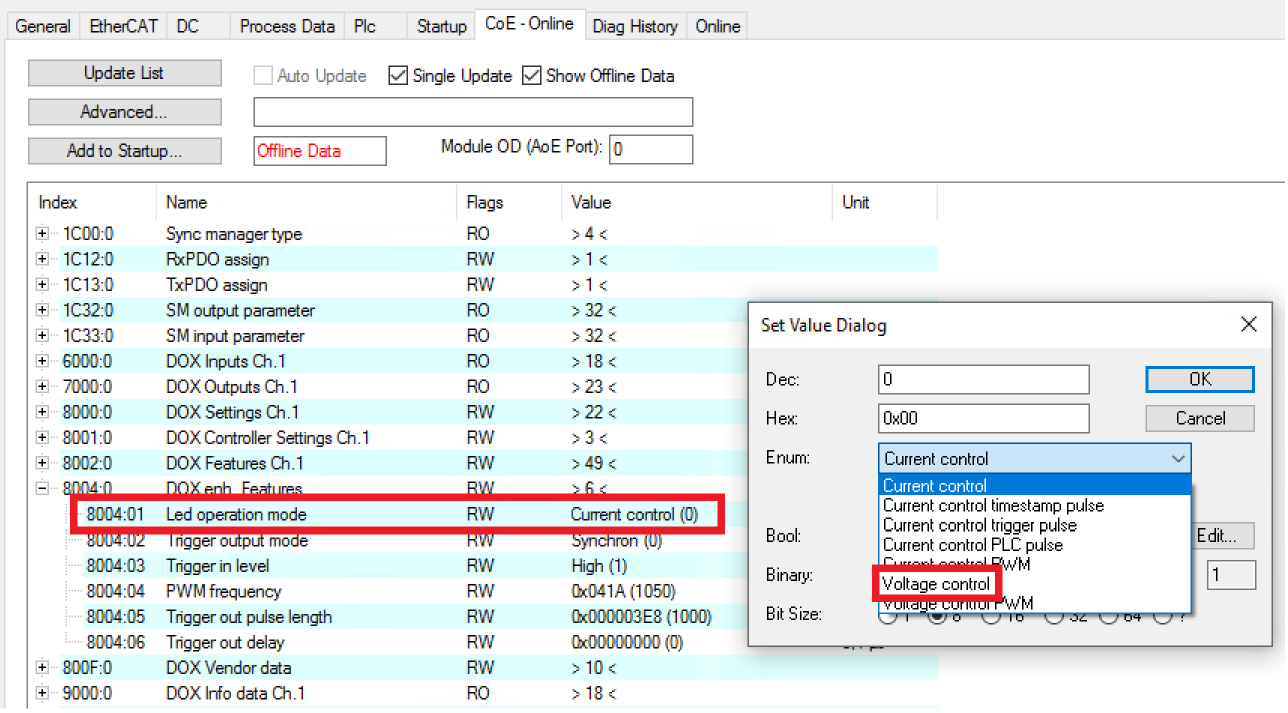
For this purpose, as described in the chapter Commissioning [▸ 129], the values for the supply and output voltage, as well as the maximum output current must also be specified in the CoE data.



Index	Name	Flags	Value	Unit
7000:0	DOX Outputs Ch.1	RO	> 23 <	
8000:0	DOX Settings Ch.1	RW	> 22 <	
8000:02	Target current	RW	0x01F4 (500)	mA
8000:03	Supply voltage	RW	0x0960 (2400)	0,01 V
8000:04	Output voltage	RW	0x04B0 (1200)	0,01 V
8000:0B	Trigger delay (switch on)	RW	0x00000000 (0)	0,1 µs
8000:11	Warning level (supply voltage)	RW	0x05 (5)	%
8000:12	Error level (supply voltage)	RW	0x14 (20)	%
8000:13	Positive warning level (output voltage)	RW	0x05 (5)	%
8000:14	Positive error level (output voltage)	RW	0x14 (20)	%
8000:15	Negative warning level (output voltage)	RW	0x05 (5)	%
8000:16	Negative error level (output voltage)	RW	0x0A (10)	%
8001:0	DOX Controller Settings Ch.1	RW	> 3 <	

Fig. 243: Setting current and voltage values in the CoE

The specified output values are then output at the output of the LED terminal. “Voltage Control” mode must be selected as the output mode (CoE index 0x8004:01).



Index	Name	Flags	Value	Unit
1C00:0	Sync manager type	RO	> 4 <	
1C12:0	RxPDO assign	RW	> 1 <	
1C13:0	TxPDO assign	RW	> 1 <	
1C32:0	SM output parameter	RO	> 32 <	
1C33:0	SM input parameter	RO	> 32 <	
6000:0	DOX Inputs Ch.1	RO	> 18 <	
7000:0	DOX Outputs Ch.1	RO	> 23 <	
8000:0	DOX Settings Ch.1	RW	> 22 <	
8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
8002:0	DOX Features Ch.1	RW	> 49 <	
8004:0	DOX enh. Features	RW	> 6 <	
8004:01	Led operation mode	RW	Current control (0)	
8004:02	Trigger output mode	RW	Synchron (0)	
8004:03	Trigger in level	RW	High (1)	
8004:04	PWM frequency	RW	0x041A (1050)	
8004:05	Trigger out pulse length	RW	0x000003E8 (1000)	
8004:06	Trigger out delay	RW	0x00000000 (0)	
800F:0	DOX Vendor data	RW	> 10 <	
9000:0	DOX Info data Ch.1	RO	> 18 <	

Set Value Dialog

Dec: 0

Hex: 0x00

Enum: Current control

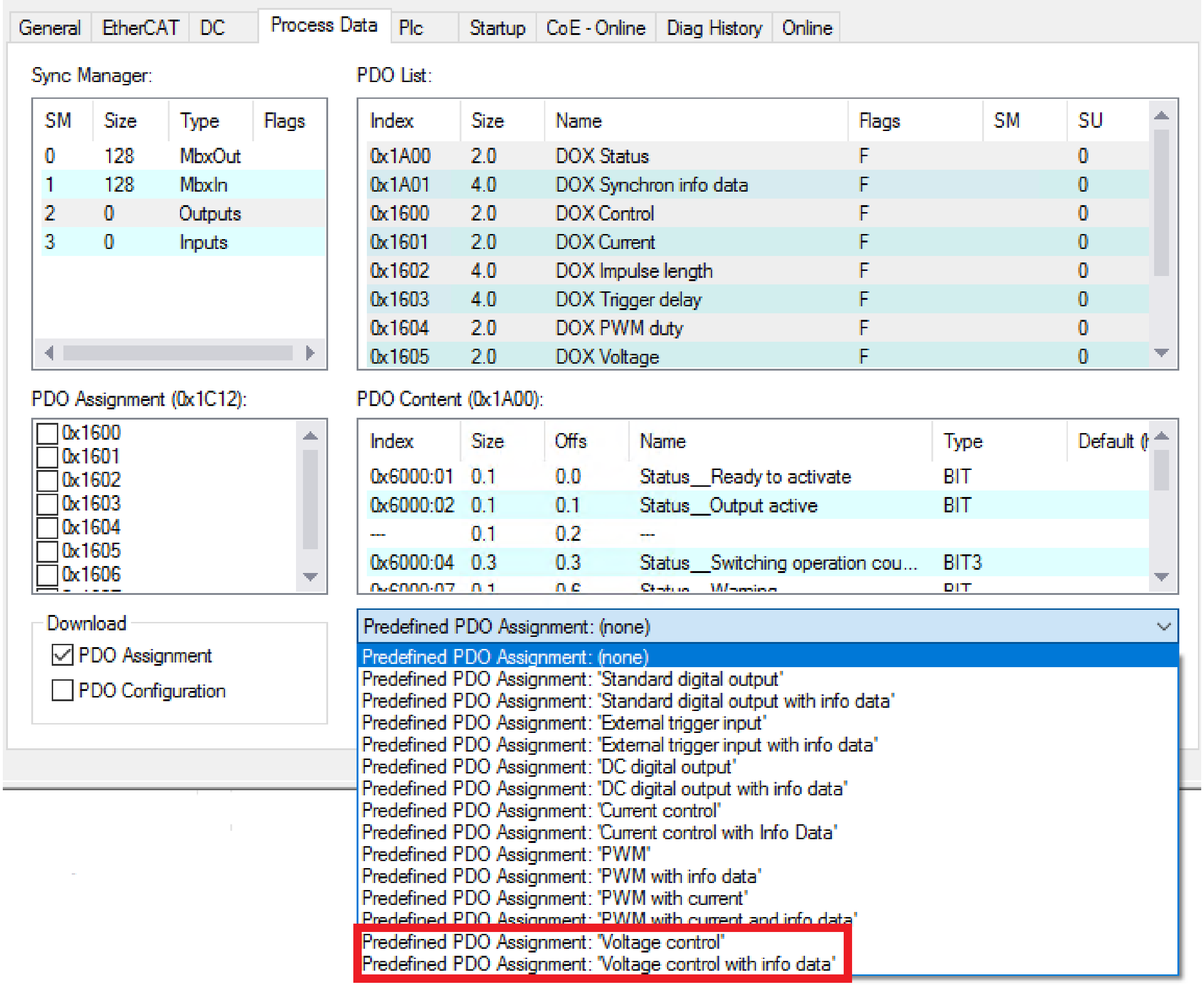
Bool:

Binary: 1

Bit Size: ○ 1 ● 8 ○ 16 ○ 32 ○ 64 ○ ?

Fig. 244: Selecting the “Voltage Control” operation mode

“Standard digital output (with info data)” is used as the predefined PDO.



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	0	Outputs	
3	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F		0
0x1A01	4.0	DOX Synchron info data	F		0
0x1600	2.0	DOX Control	F		0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	2.0	DOX PWM duty	F		0
0x1605	2.0	DOX Voltage	F		0

PDO Assignment (0x1C12):

- ☐ 0x1600
- ☐ 0x1601
- ☐ 0x1602
- ☐ 0x1603
- ☐ 0x1604
- ☐ 0x1605
- ☐ 0x1606

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h
0x6000:01	0.1	0.0	Status__Ready to activate	BIT	
0x6000:02	0.1	0.1	Status__Output active	BIT	
---	0.1	0.2	---		
0x6000:04	0.3	0.3	Status__Switching operation cou...	BIT3	
0x6000:07	0.1	0.6	Status__Warning	BIT	

Download

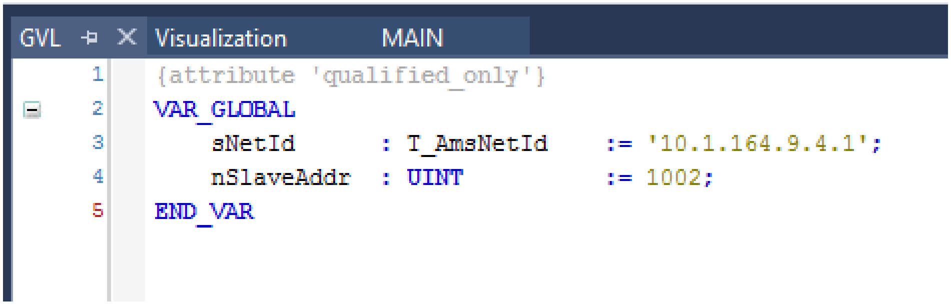
- ☑ PDO Assignment
- ☐ PDO Configuration

Predefined PDO Assignment: (none)

- Predefined PDO Assignment: (none)
- Predefined PDO Assignment: 'Standard digital output'
- Predefined PDO Assignment: 'Standard digital output with info data'
- Predefined PDO Assignment: 'External trigger input'
- Predefined PDO Assignment: 'External trigger input with info data'
- Predefined PDO Assignment: 'DC digital output'
- Predefined PDO Assignment: 'DC digital output with info data'
- Predefined PDO Assignment: 'Current control'
- Predefined PDO Assignment: 'Current control with Info Data'
- Predefined PDO Assignment: 'PWM'
- Predefined PDO Assignment: 'PWM with info data'
- Predefined PDO Assignment: 'PWM with current'
- Predefined PDO Assignment: 'PWM with current and info data'
- Predefined PDO Assignment: 'Voltage control'
- Predefined PDO Assignment: 'Voltage control with info data'

Fig. 245: Selection of the Predefined PDOs “Current Control (with info data)”

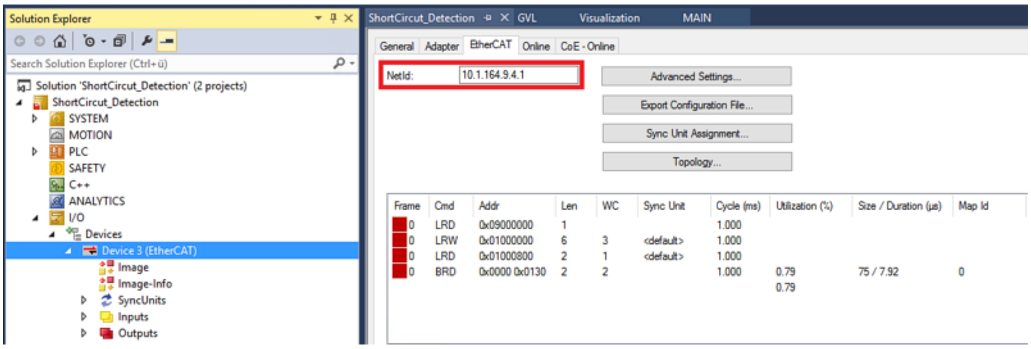
In the sample program, diagnostic data are read from the diagnostic messages of the terminal. To do this, the NetID of the EtherCAT network and the ID of the EL2596 must be specified in the global variable list (GVL).



```
{attribute 'qualified_only'}
VAR_GLOBAL
    sNetId        : T_AmsNetId     := '10.1.164.9.4.1';
    nSlaveAddr    : UINT           := 1002;
END_VAR
```

Fig. 246: Sample program 1 - GVL

The NetID can be displayed in TwinCAT under the EtherCAT Master on the “EtherCAT” tab. The ID must be specified in the GVL as sNetID in inverted commas as a string.

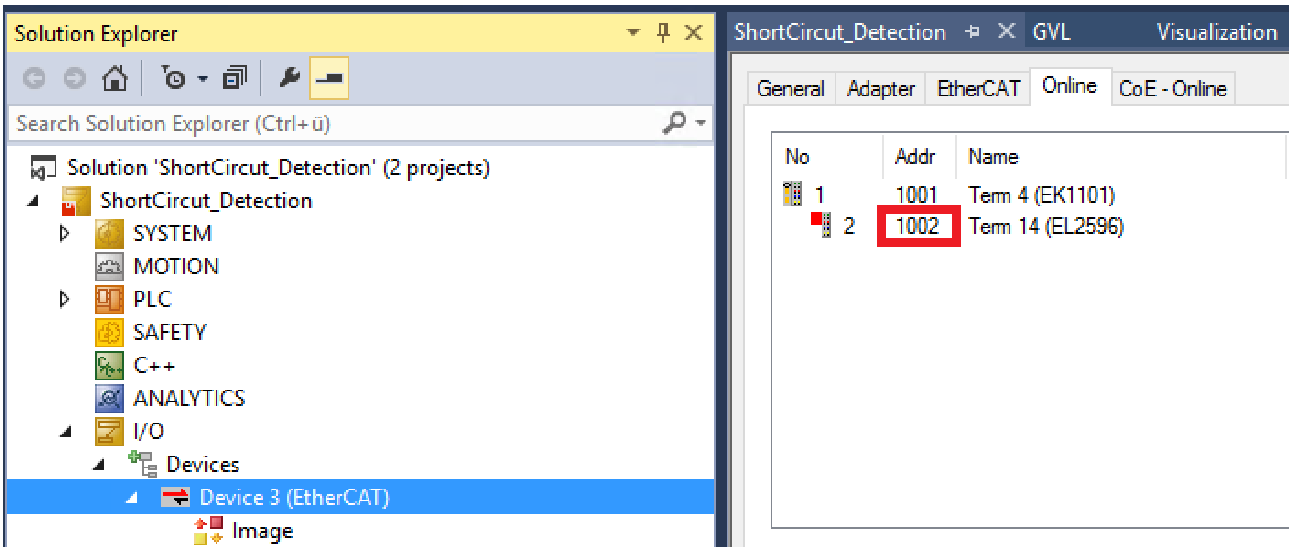


NetId: 10.1.164.9.4.1

Frame	Cmd	Addr	Len	WC	Sync Unit	Cycle (ms)	Utilization (%)	Size / Duration (µs)	Map Id
0	LRD	0x09000000	1			1.000			
0	LRW	0x01000000	6	3	<default>	1.000			
0	LRD	0x01000800	2	1	<default>	1.000			
0	BRD	0x0000 0x0130	2	2		1.000	0.79	75 / 7.92	0
							0.79		

Fig. 247: Displaying the NetID

The ID of the slave can also be found in the EtherCAT master. The “Online” tab must be selected in order to do this. This value must then be specified as nSlaveAddr.

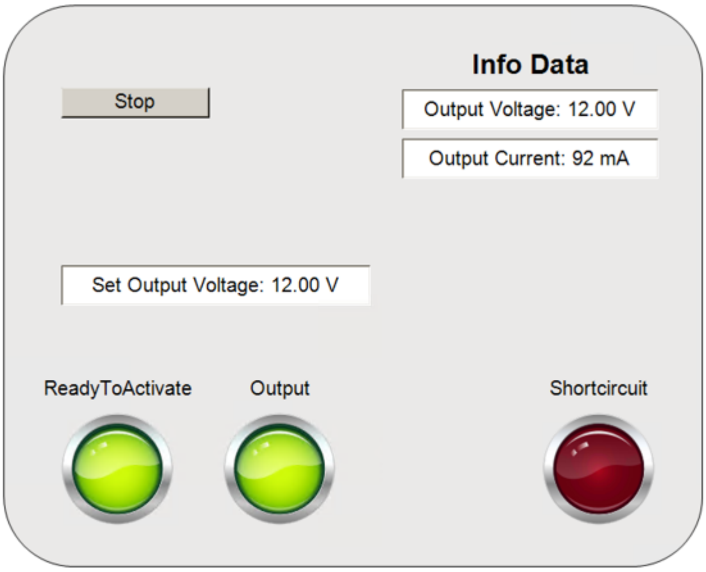


No	Addr	Name
1	1001	Term 4 (EK1101)
2	1002	Term 14 (EL2596)

Fig. 248: Displaying the slave address

The sample program can then be started. To use the sample program, the included visualization should be opened.

- If the program is started and the terminal is not in an error state, “ReadyToActivate” is active and the lamp in the visualization is green.
- The desired voltage at the output must be set in the “Set Output Voltage” box. This can also be changed while the short-circuit detection is running.
- The short-circuit detection can then be started in the visualization via the “Start” button.
- The output is then switched on from the PLC, so that the “Output” lamp also turns green.
- The “Output Voltage” and “Output Current” text boxes display the present values of the output.
- If a falling edge occurs on the “ReadyToActivate” bit in the PLC program, the reading of the diagnostic information from the CoE objects 0xA000 and 0x10F3 is started. The output switches off immediately.
- Any short-circuit that has occurred is recognizable from the diagnostic data, and the “Shortcircuit” lamp in the visualization lights up red.
- If there is no longer a short-circuit at the LED output, the EL2596 must be reset manually via the “Reset” button, which is linked to the “Reset” bit of the EL2596.
- The LED output is then reactivated.
- The short-circuit detection can be stopped manually via the “Stop” button; the LED output is then switched off.



Output Voltage: 12.00 V

Output Current: 92 mA

Set Output Voltage: 12.00 V

Fig. 249: Sample program 2 - Visualization

Limiting the peak current in case of short-circuit

If a short-circuit occurs at the LED output, currents >6 A can flow for a short time. In order to limit the peak output current, a resistor in the low ohm range can be connected in series with the component on which a short-circuit is to be detected.

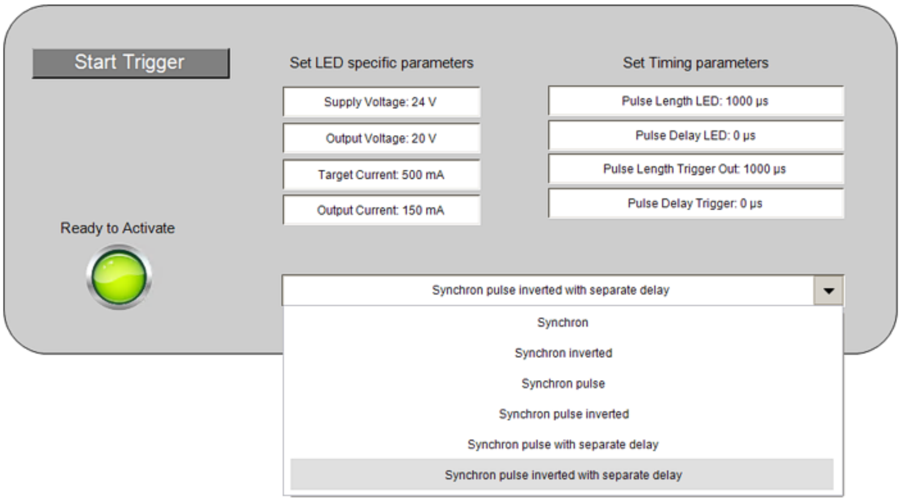
5.12.3 Sample program 3 - combination of input and output triggers

https://infosys.beckhoff.com/content/1033/el2596/Resources/8316880011/.zip

Program description and function

As already described in the chapter Using the trigger output with the trigger input [▶ 183] the trigger output of the EL2596-xxxx can be used as a response to the trigger input.

In this sample program, a signal is generated at the input trigger with an EL2202. The EL2202 is a digital output terminal with a switch-on time of <1 µs. This terminal is used because a steep edge is then applied to the trigger input of the EL2596. In the sample program, a visualization can then be used to specify which trigger mode (described in the table in the chapter Activating the trigger output) [▶ 183] is to be used. In addition, all LED-specific parameters, such as output current, etc. and all time-specific parameters, such as the delay at the LED and/or trigger output, can be specified.



Supply Voltage: 24 V

Output Voltage: 20 V

Target Current: 500 mA

Output Current: 150 mA

Pulse Length LED: 1000 µs

Pulse Delay LED: 0 µs

Pulse Length Trigger Out: 1000 µs

Pulse Delay Trigger: 0 µs

Synchron pulse inverted with separate delay

Fig. 250: Sample program 2 - Visualization

All characteristic values for the LED and trigger outputs are specified via the visualization. When starting the trigger output via the “Start Trigger” button in the visualization, all specified values are transferred to the CoE and the process data. A pulsating signal is then started at the output of the EL2202. With each rising edge at the trigger input of the EL2202, a pulse with the specified length is output at the LED or trigger output. If new parameters are to be set, the output of the EL2202 must be stopped manually via the “Stop Trigger” button. Then the parameters can be adjusted via the visualization. When the trigger output is restarted, all parameters are transferred to the CoE and the process data.

An EL2202 digital output terminal 24 V and an EL2596 1-channel LED strobe control terminal are required in order to use the sample. The trigger output can be measured with an oscilloscope, for example. The wiring of the hardware for the sample program is shown in the following figure.

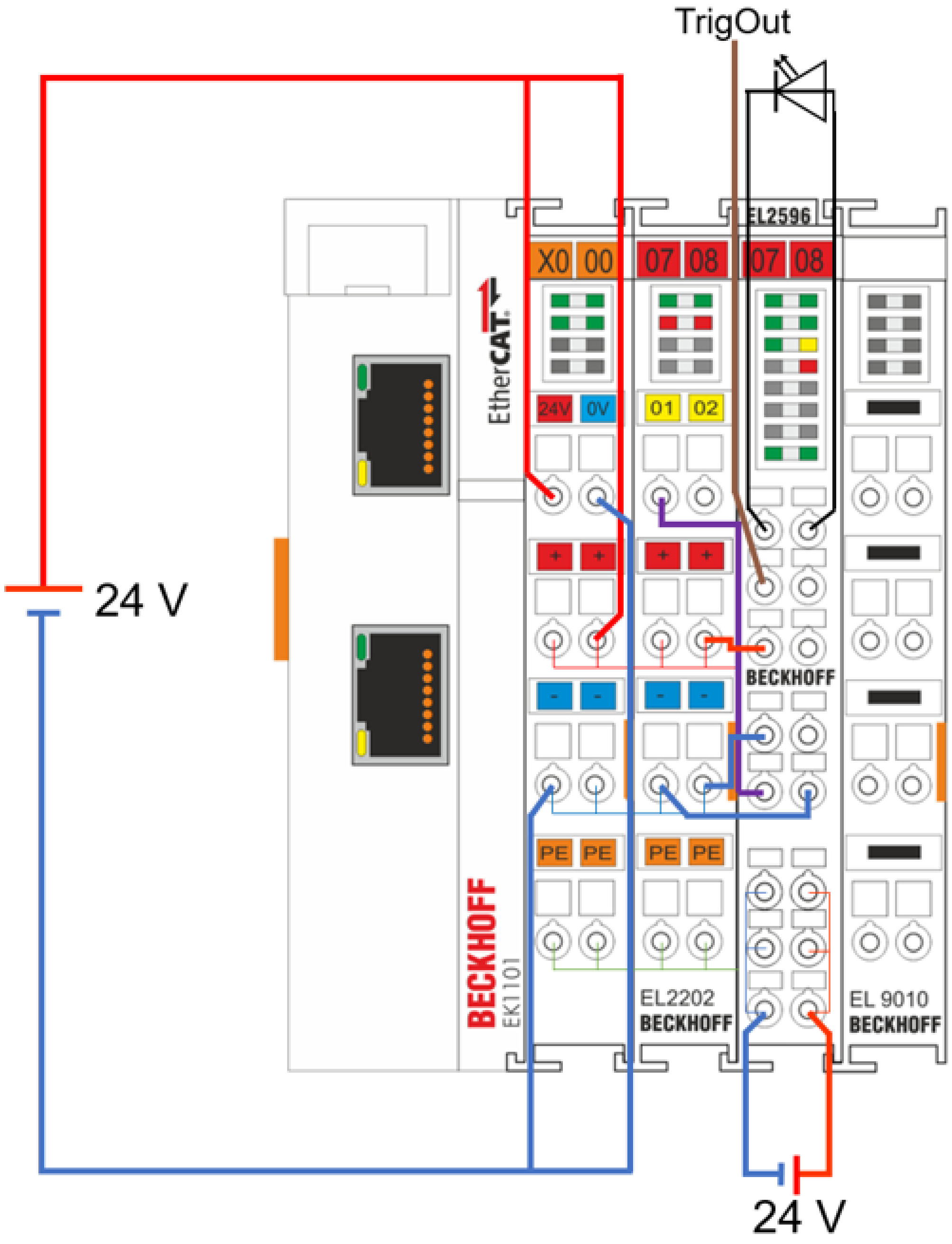


Fig. 251: Sample program 2 - Hardware wiring

5.13 General Commissioning Instructions for an EtherCAT Slave

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

Diagnosis in real time: WorkingCounter, EtherCAT State and Status

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

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

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

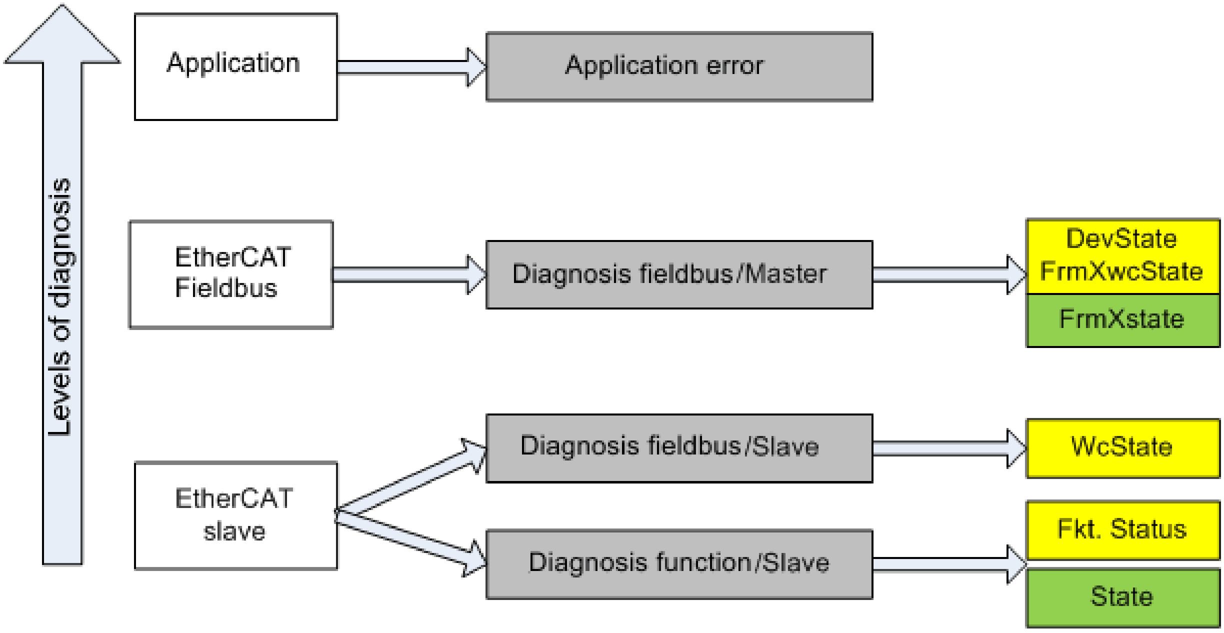


Fig. 252: Selection of the diagnostic information of an EtherCAT Slave

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.

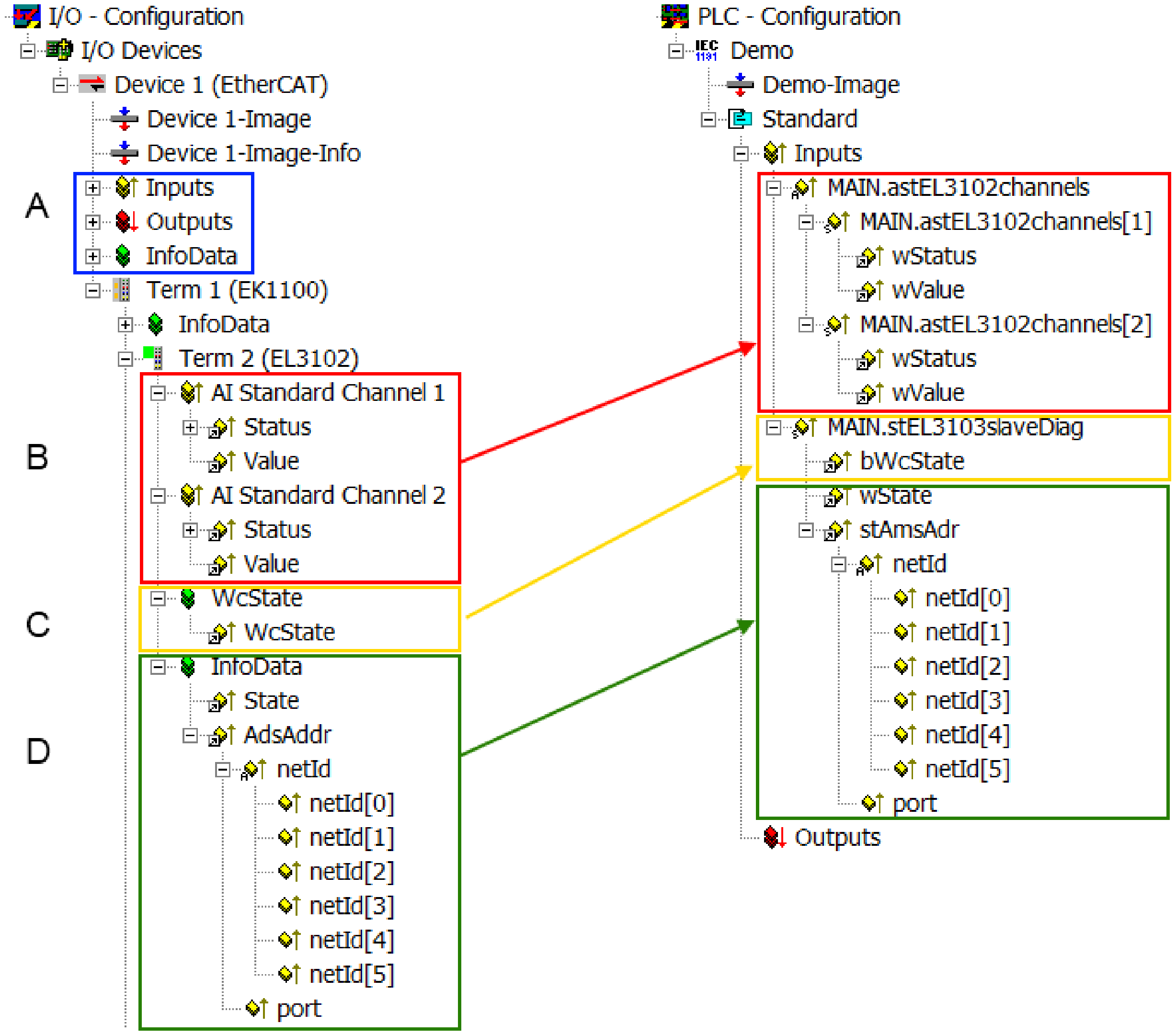


Fig. 253: Basic EtherCAT Slave Diagnosis in the PLC

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.

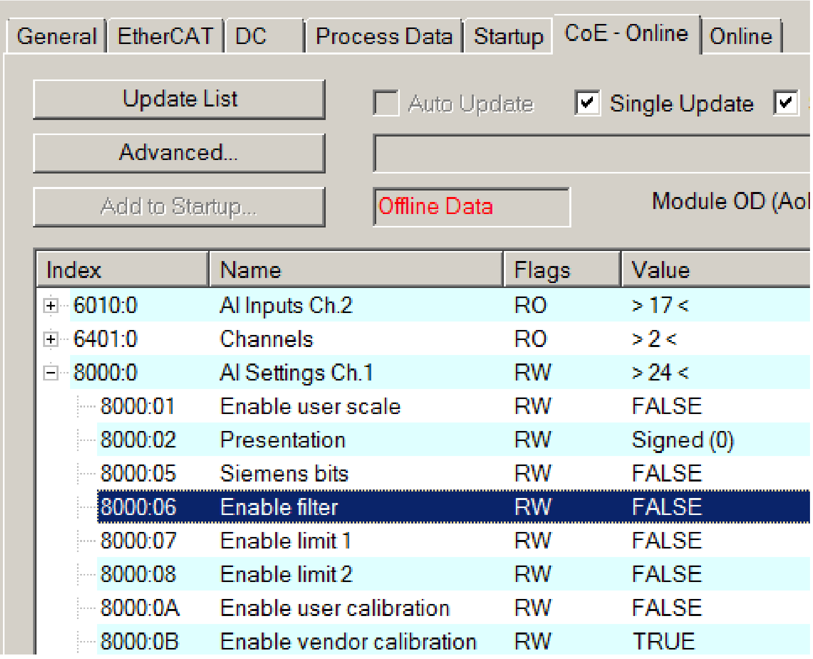
NOTE

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*:



Index	Name	Flags	Value
⊞ 6010:0	AI Inputs Ch.2	RO	> 17 <
⊞ 6401:0	Channels	RO	> 2 <
⊟ 8000:0	AI Settings Ch.1	RW	> 24 <
8000:01	Enable user scale	RW	FALSE
8000:02	Presentation	RW	Signed (0)
8000:05	Siemens bits	RW	FALSE
8000:06	Enable filter	RW	FALSE
8000:07	Enable limit 1	RW	FALSE
8000:08	Enable limit 2	RW	FALSE
8000:0A	Enable user calibration	RW	FALSE
8000:0B	Enable vendor calibration	RW	TRUE

Fig. 254: EL3102, CoE directory

EtherCAT System Documentation

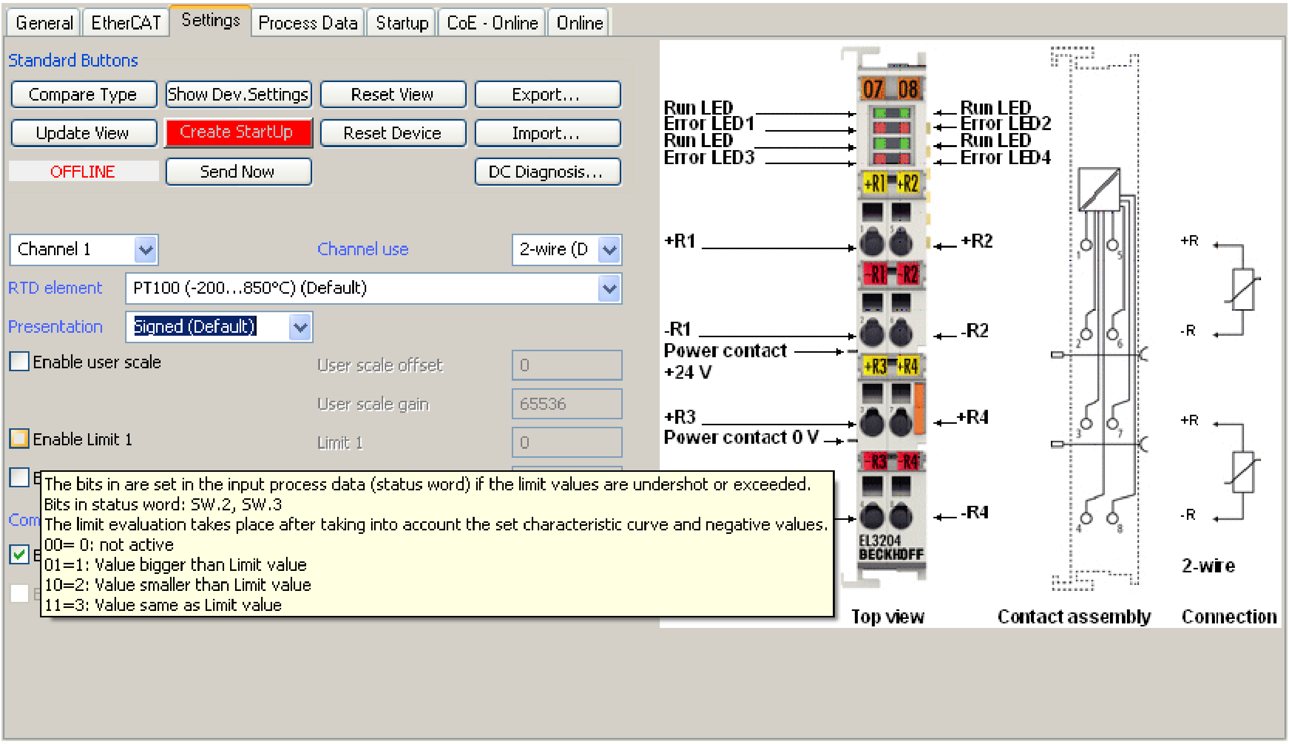
The comprehensive description in the EtherCAT System Documentation (EtherCAT Basics --> CoE Interface) must be observed!

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.



Channel 1 — Channel use: 2-wire (D

RTD element: PT100 (-200…850°C) (Default)

Presentation: Signed (Default)

□ Enable user scale — User scale offset: 0 — User scale gain: 65536

□ Enable Limit 1 — Limit 1: 0

The bits in are set in the input process data (status word) if the limit values are undershot or exceeded.
Bits in status word: SW.2, SW.3
The limit evaluation takes place after taking into account the set characteristic curve and negative values.
00= 0: not active
01=1: Value bigger than Limit value
10=2: Value smaller than Limit value
11=3: Value same as Limit value

Fig. 255: 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 [▶ 41]" in this connection. Depending how much configuration has to be done, and on the overall communication, booting can take up to a few seconds.

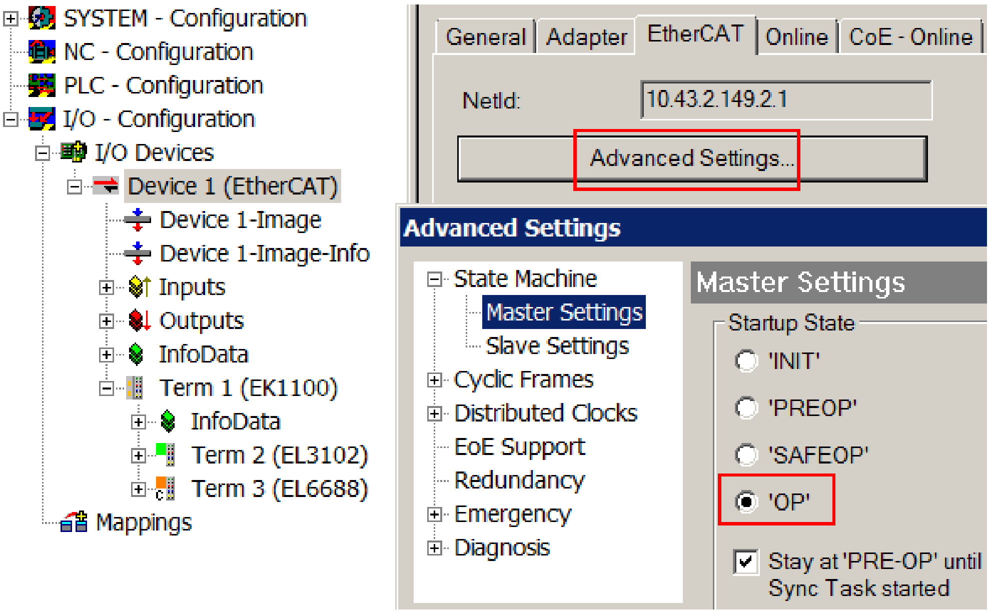
The EtherCAT Master itself must go through these routines when starting, until it has reached at least the OP target state.

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.



NetId: 10.43.2.149.2.1

Startup State

○ 'INIT'

○ 'PREOP'

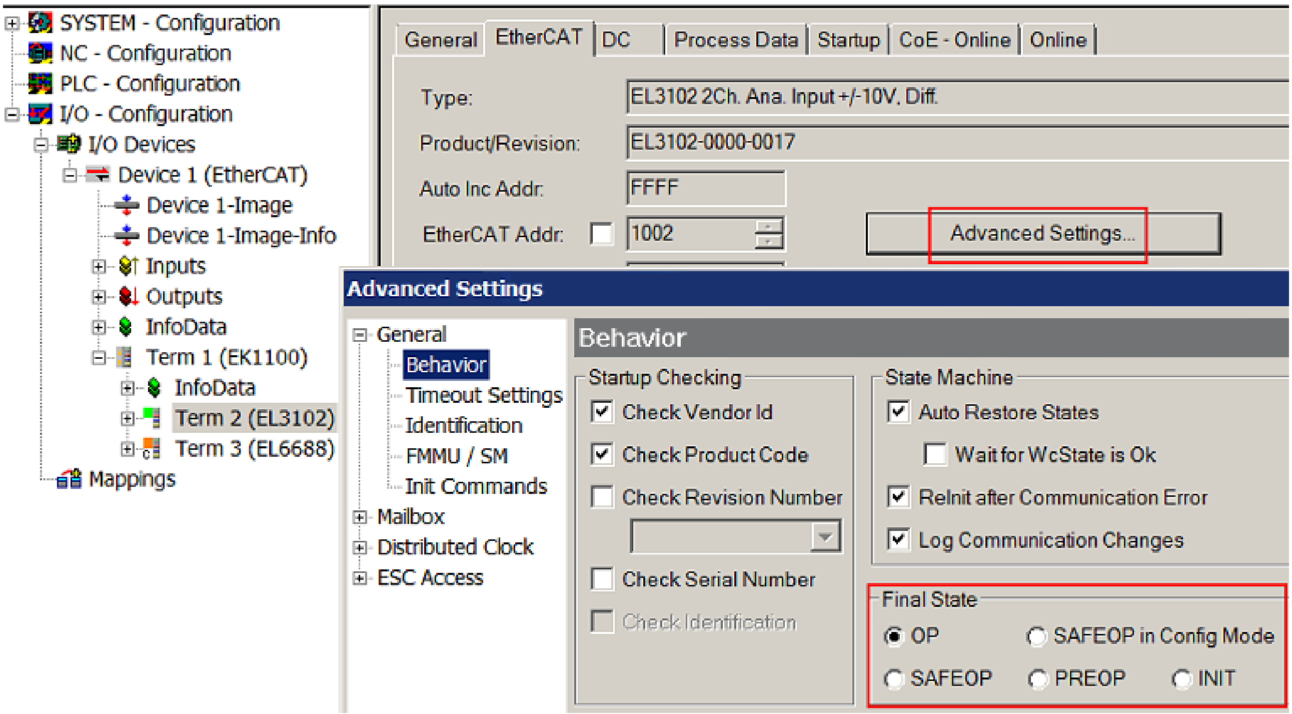
○ 'SAFEOP'

● 'OP'

☑ Stay at 'PRE-OP' until Sync Task started

Fig. 256: 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.



Type: EL3102 2Ch. Ana. Input +/-10V, Diff.

Product/Revision: EL3102-0000-0017

Auto Inc Addr: FFFF

EtherCAT Addr: □ 1002

Startup Checking

☑ Check Vendor Id

☑ Check Product Code

□ Check Revision Number

□ Check Serial Number

□ Check Identification

State Machine

☑ Auto Restore States

□ Wait for WcState is Ok

☑ ReInit after Communication Error

☑ Log Communication Changes

Final State

● OP ○ SAFEOP in Config Mode

○ SAFEOP ○ PREOP ○ INIT

Fig. 257: Default target state in the Slave

Manual Control

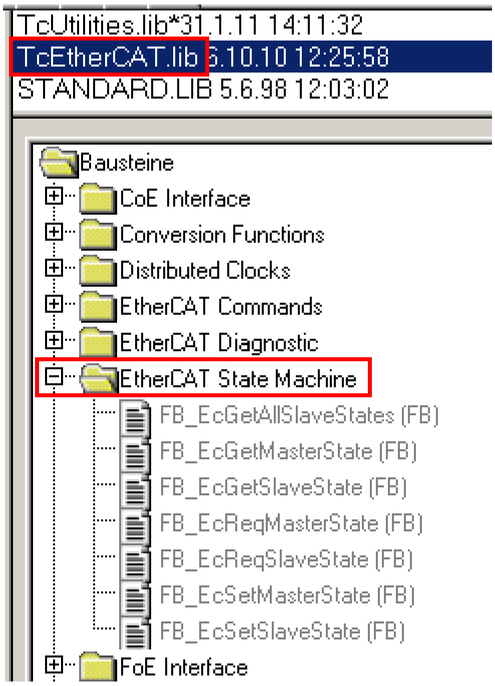
There are particular reasons why it may be appropriate to control the states from the application/task/PLC. For instance:

- 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.



TcUtilities.lib*31.1.11 14:11:32
TcEtherCAT.lib 6.10.10 12:25:58
STANDARD.LIB 5.6.98 12:03:02

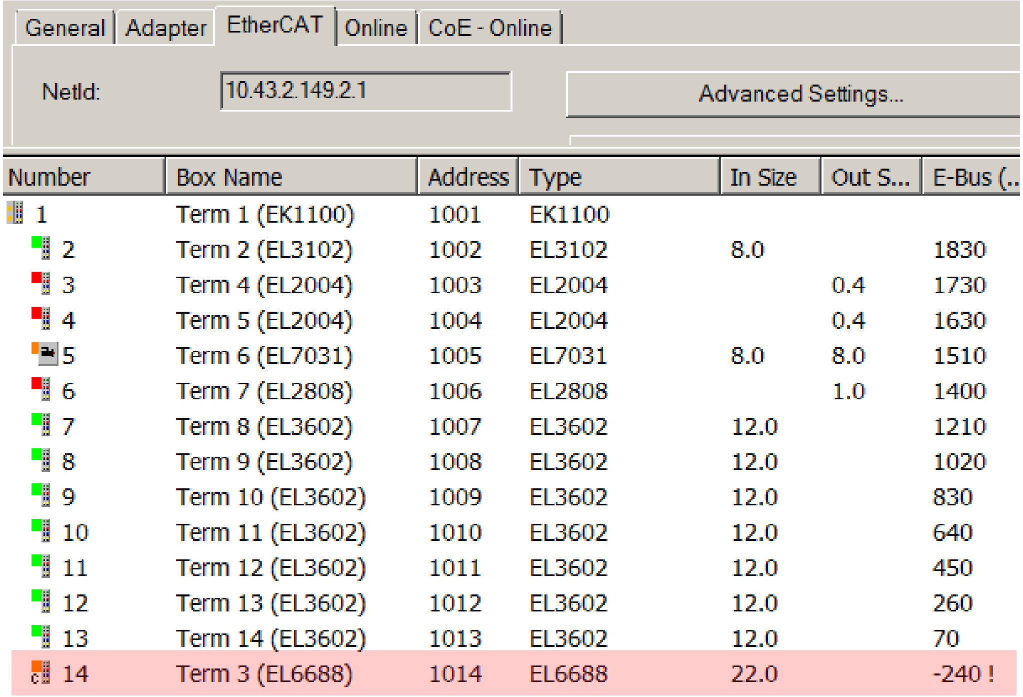
- Bausteine
  - CoE Interface
  - Conversion Functions
  - Distributed Clocks
  - EtherCAT Commands
  - EtherCAT Diagnostic
  - EtherCAT State Machine
    - FB_EcGetAllSlaveStates (FB)
    - FB_EcGetMasterState (FB)
    - FB_EcGetSlaveState (FB)
    - FB_EcReqMasterState (FB)
    - FB_EcReqSlaveState (FB)
    - FB_EcSetMasterState (FB)
    - FB_EcSetSlaveState (FB)
  - FoE Interface

Fig. 258: 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.

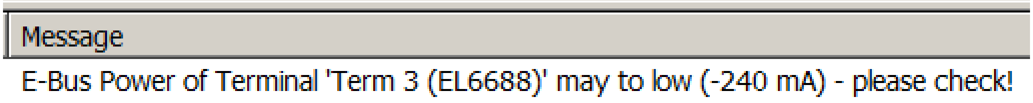


NetId: 10.43.2.149.2.1

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. 259: 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:



Message
E-Bus Power of Terminal 'Term 3 (EL6688)' may to low (-240 mA) - please check!

Fig. 260: Warning message for exceeding E-Bus current

NOTE

Caution! Malfunction possible!

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

6 Appendix

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

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

NOTE

Only use TwinCAT 3 software!

A firmware update of Beckhoff IO devices must only be performed with a TwinCAT 3 installation. It is recommended to build as up-to-date as possible, available for free download on the Beckhoff website https://www.beckhoff.com/en-us/.

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

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

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

Storage locations

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

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

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

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

Simplified update by bundle firmware

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

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

Following the update, its success should be verified

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

NOTE

Risk of damage to the device!

✓ Note the following when downloading new device files

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

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

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

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

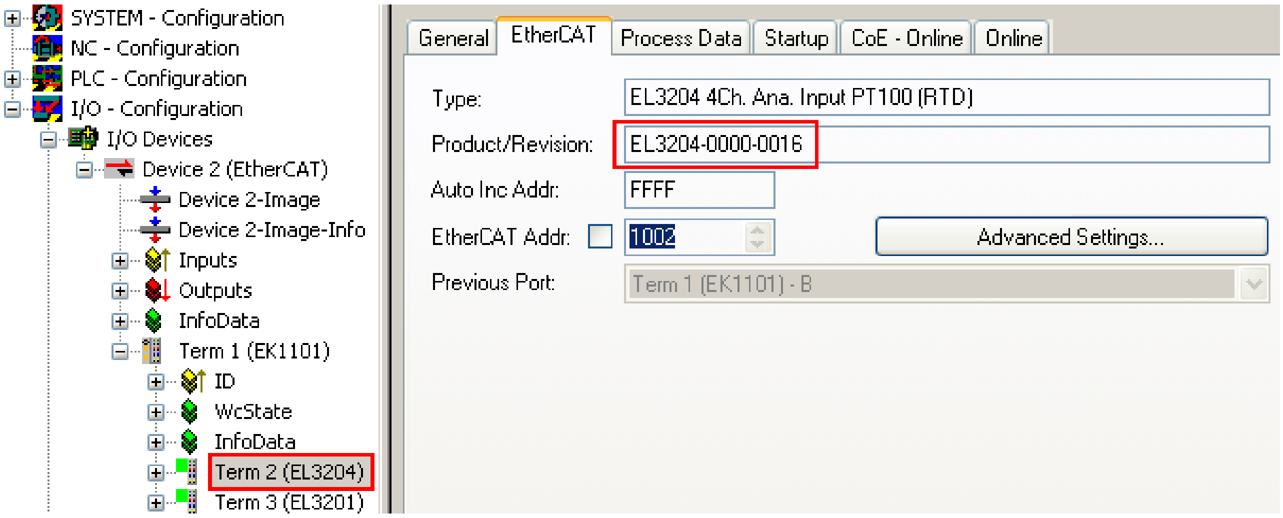
6.1.1 Device description ESI file/XML

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:



Type: EL3204 4Ch. Ana. Input PT100 (RTD)

Product/Revision: EL3204-0000-0016

Auto Inc Addr: FFFF

EtherCAT Addr: □ 1002

Previous Port: Term 1 (EK1101) - B

Fig. 261: 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:

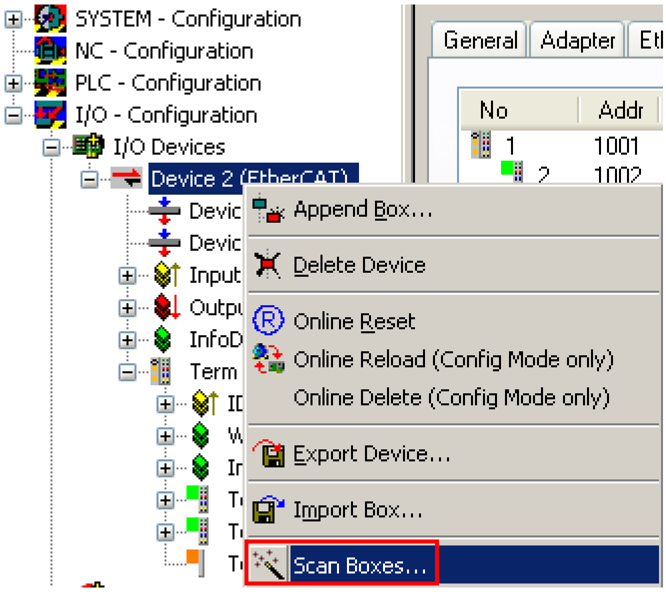
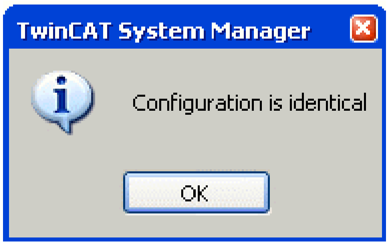


Fig. 262: Scan the subordinate field by right-clicking on the EtherCAT device

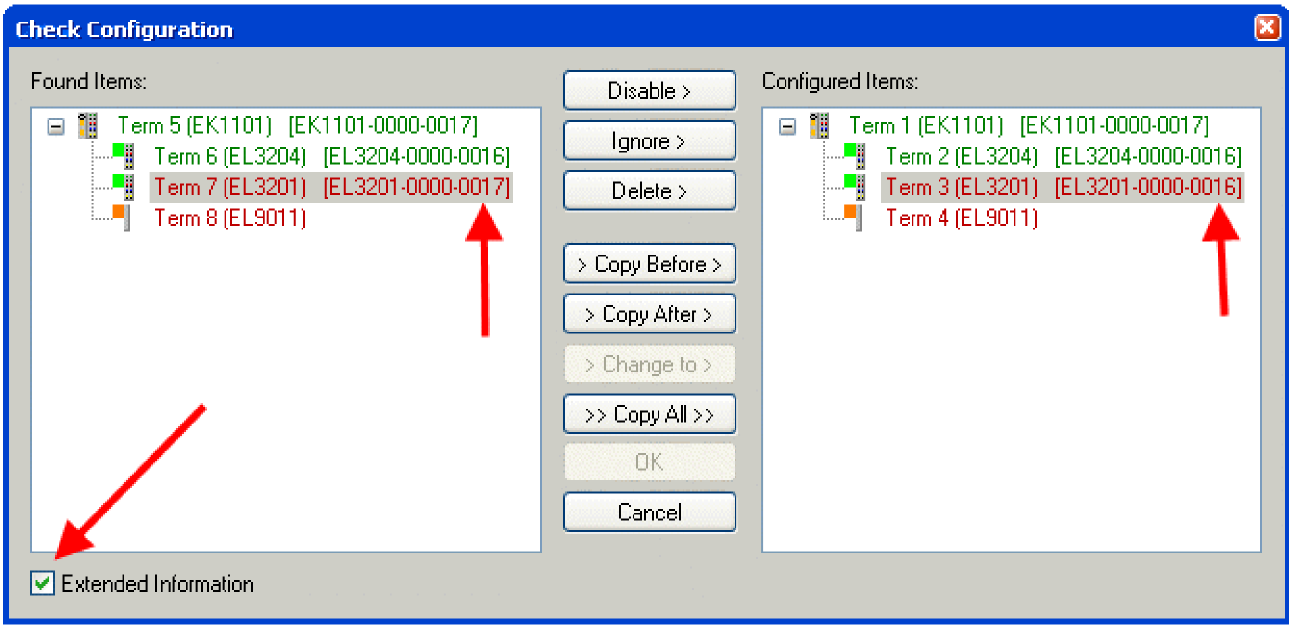
If the found field matches the configured field, the display shows


TwinCAT System Manager

Configuration is identical

Fig. 263: Configuration is identical

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


Check Configuration

Found Items:

- Term 5 (EK1101) [EK1101-0000-0017]
  - Term 6 (EL3204) [EL3204-0000-0016]
  - Term 7 (EL3201) [EL3201-0000-0017]
  - Term 8 (EL9011)

Configured Items:

- Term 1 (EK1101) [EK1101-0000-0017]
  - Term 2 (EL3204) [EL3204-0000-0016]
  - Term 3 (EL3201) [EL3201-0000-0016]
  - Term 4 (EL9011)

☑ Extended Information

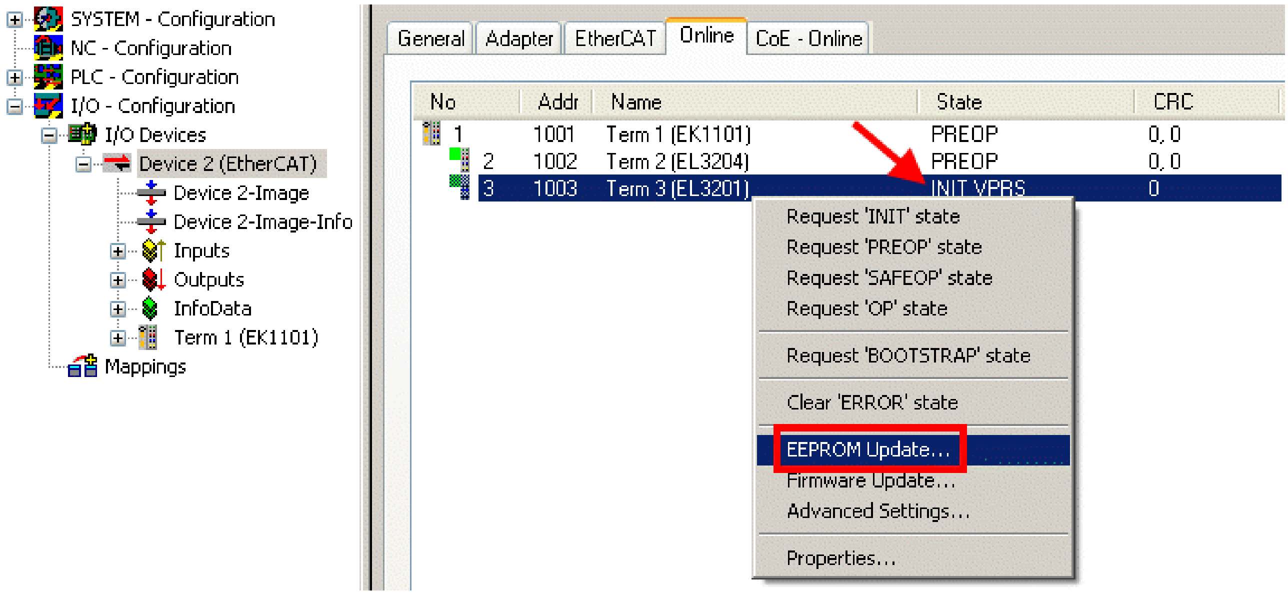
Fig. 264: Change dialog

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*



No	Addr	Name	State	CRC
1	1001	Term 1 (EK1101)	PREOP	0, 0
2	1002	Term 2 (EL3204)	PREOP	0, 0
3	1003	Term 3 (EL3201)	INIT VPRS	0

Fig. 265: 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.

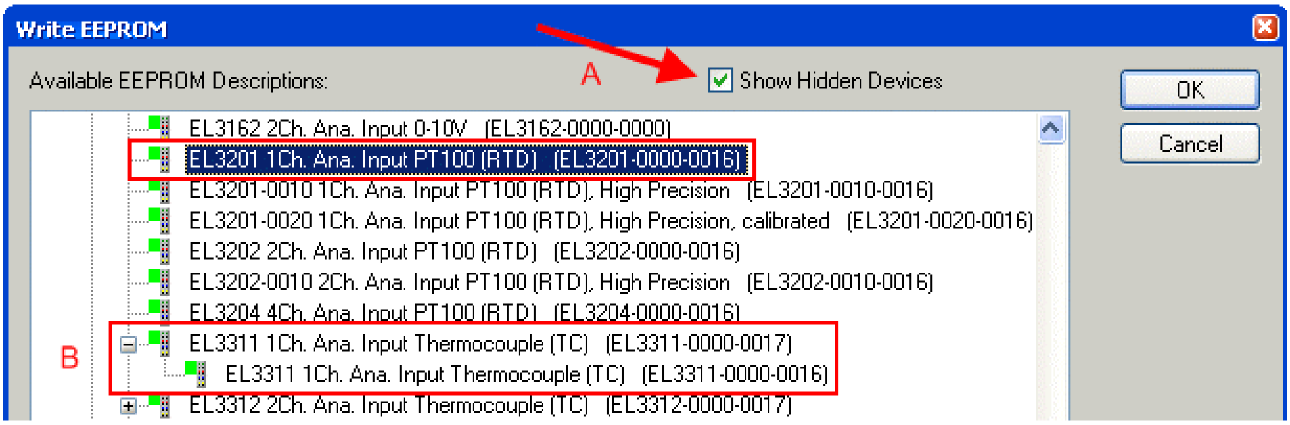

Write EEPROM
Available EEPROM Descriptions:
A
☑ Show Hidden Devices
OK
Cancel
EL3162 2Ch. Ana. Input 0-10V (EL3162-0000-0000)
EL3201 1Ch. Ana. Input PT100 (RTD) (EL3201-0000-0016)
EL3201-0010 1Ch. Ana. Input PT100 (RTD), High Precision (EL3201-0010-0016)
EL3201-0020 1Ch. Ana. Input PT100 (RTD), High Precision, calibrated (EL3201-0020-0016)
EL3202 2Ch. Ana. Input PT100 (RTD) (EL3202-0000-0016)
EL3202-0010 2Ch. Ana. Input PT100 (RTD), High Precision (EL3202-0010-0016)
EL3204 4Ch. Ana. Input PT100 (RTD) (EL3204-0000-0016)
B
EL3311 1Ch. Ana. Input Thermocouple (TC) (EL3311-0000-0017)
EL3311 1Ch. Ana. Input Thermocouple (TC) (EL3311-0000-0016)
EL3312 2Ch. Ana. Input Thermocouple (TC) (EL3312-0000-0017)

Fig. 266: 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.

6.1.2 Firmware explanation

Determining the firmware version

Determining the version via the System Manager

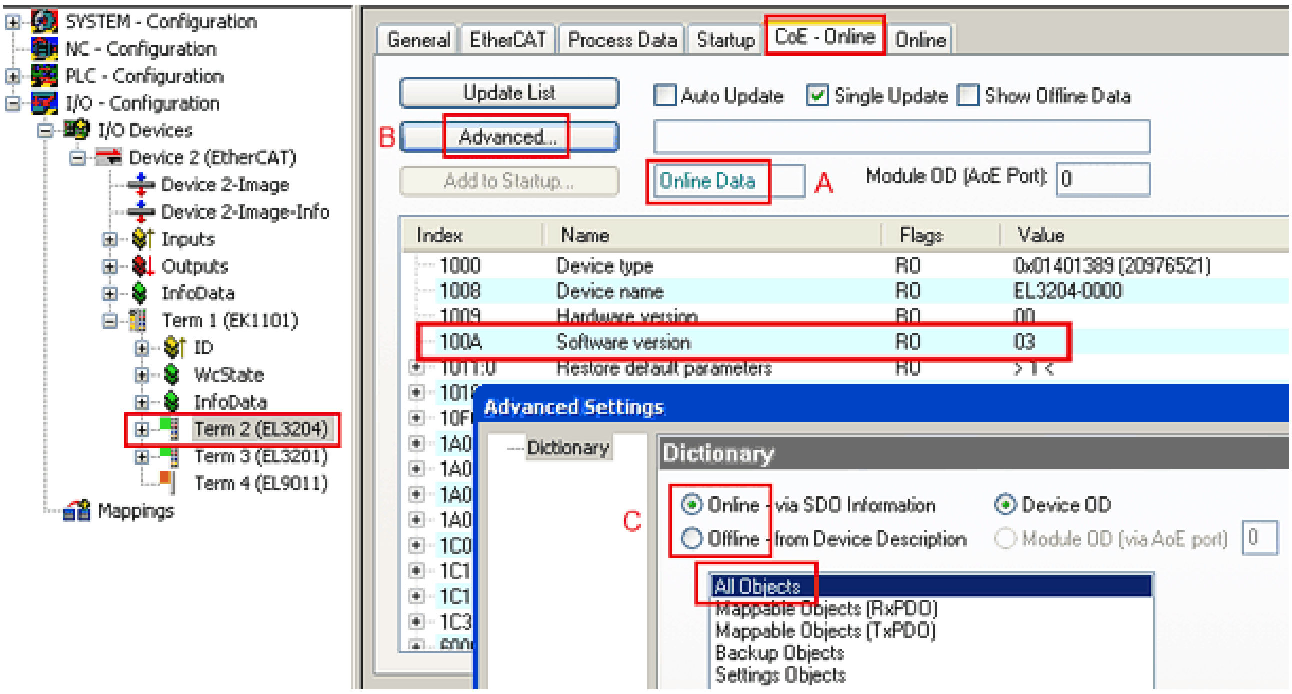
The TwinCAT System Manager shows the version of the controller firmware if the master can access the slave online. Click on the E-Bus Terminal whose controller firmware you want to check (in the example terminal 2 (EL3204)) and select the tab *CoE Online* (CAN over EtherCAT).

CoE Online and Offline CoE

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

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

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



Index	Name	Flags	Value
1000	Device type	RO	0x01401389 (20976521)
1008	Device name	RO	EL3204-0000
1009	Hardware version	RO	00
100A	Software version	RO	03
1011:0	Restore default parameters	RO	>1<

Fig. 267: Display of EL3204 firmware version

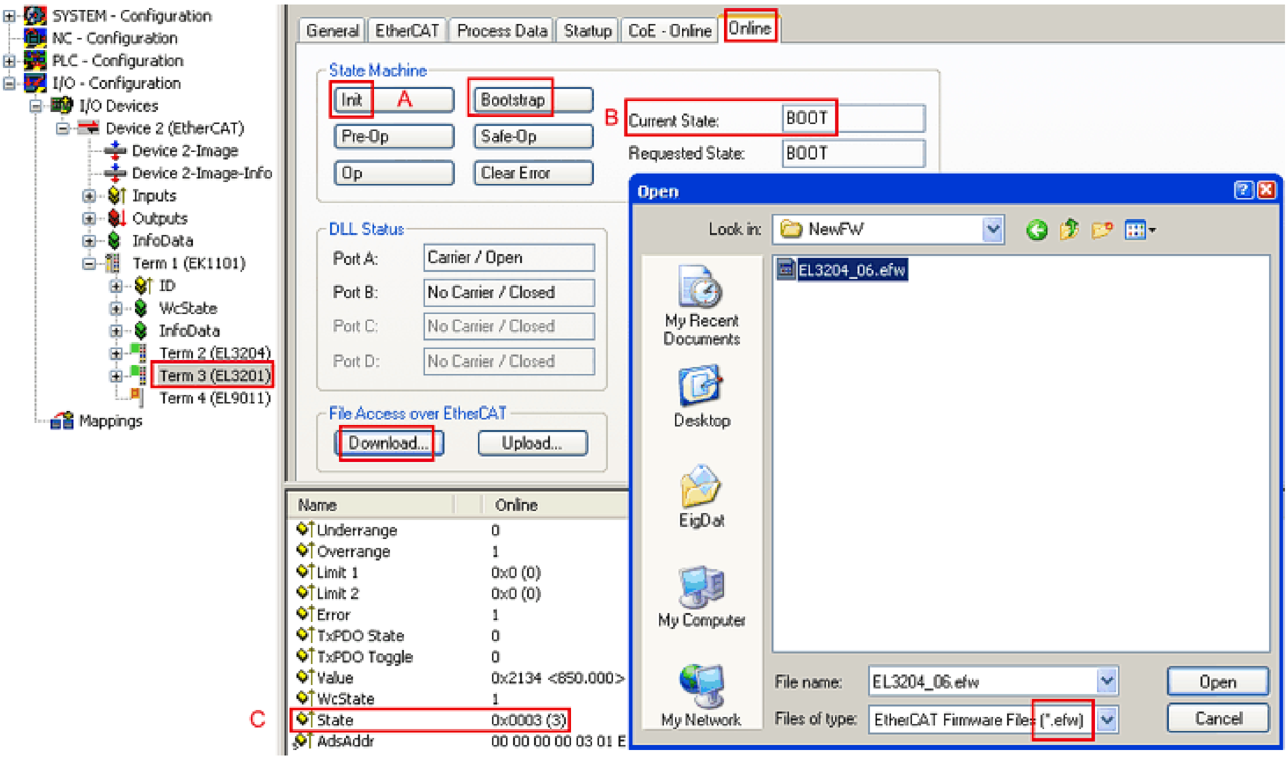
In (A) TwinCAT 2.11 shows that the Online CoE directory is currently displayed. If this is not the case, the Online directory can be loaded via the *Online* option in Advanced Settings (B) and double-clicking on *AllObjects*.

6.1.3 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.*



Current State: BOOT
Requested State: BOOT

Port A: Carrier / Open
Port B: No Carrier / Closed
Port C: No Carrier / Closed
Port D: No Carrier / Closed

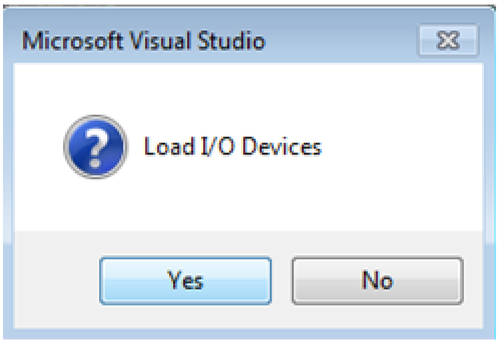
Name	Online
Underrange	0
Overrange	1
Limit 1	0x0 (0)
Limit 2	0x0 (0)
Error	1
TxPDO State	0
TxPDO Toggle	0
Value	0x2134 <850.000>
WcState	1
State	0x0003 (3)
AdsAddr	00 00 00 00 03 01 E

Look in: NewFW
EL3204_06.efw
File name: EL3204_06.efw
Files of type: EtherCAT Firmware Files (*.efw)

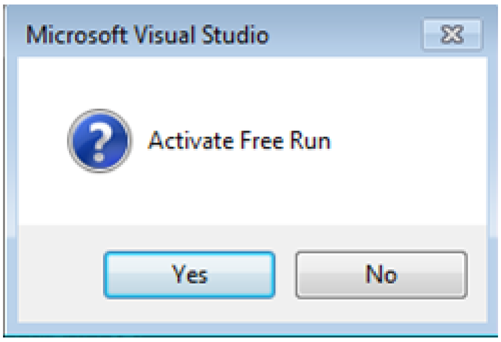
Fig. 268: Firmware Update

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.

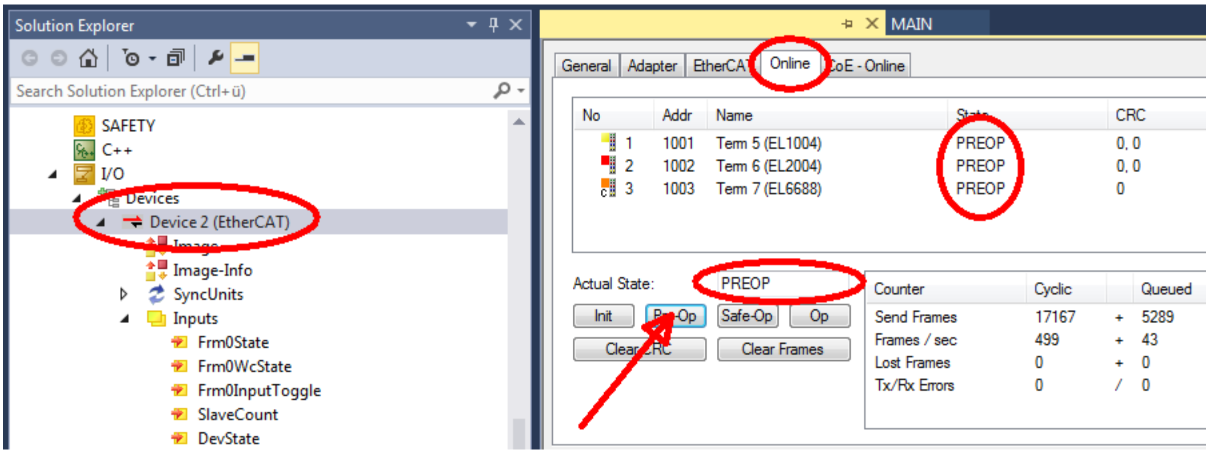

Microsoft Visual Studio

Load I/O Devices


Microsoft Visual Studio

Activate Free Run

- Switch EtherCAT Master to PreOP



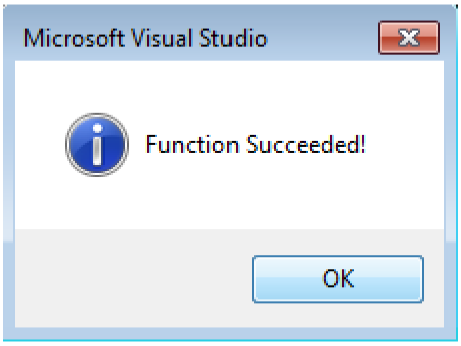
No	Addr	Name	State	CRC
1	1001	Term 5 (EL1004)	PREOP	0, 0
2	1002	Term 6 (EL2004)	PREOP	0, 0
3	1003	Term 7 (EL6688)	PREOP	0

Actual State: PREOP

Counter	Cyclic		Queued
Send Frames	17167	+	5289
Frames / sec	499	+	43
Lost Frames	0	+	0
Tx/Rx Errors	0	/	0

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

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



Microsoft Visual Studio

Function Succeeded!

- 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.

6.1.4 FPGA firmware *.rbf

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

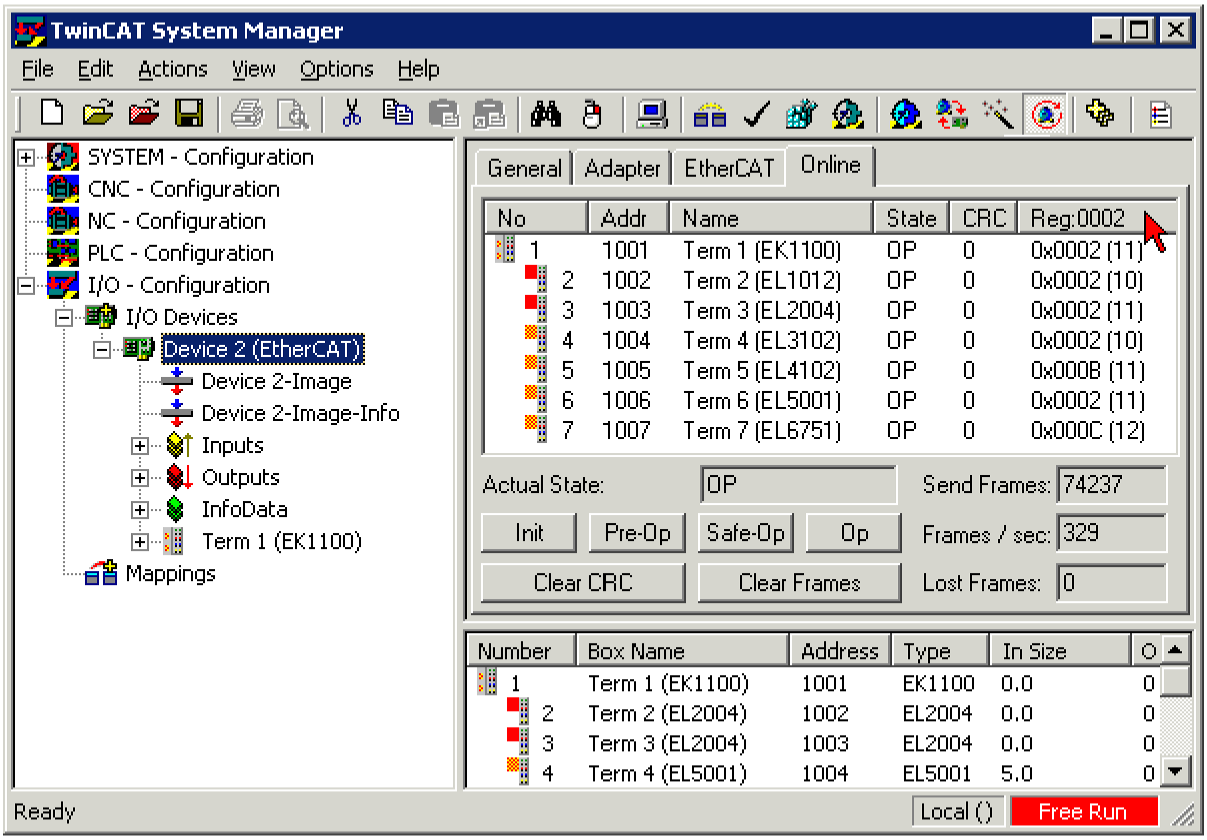
- Controller firmware for processing I/O signals
- FPGA firmware for EtherCAT communication (only for terminals with FPGA)

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.



No	Addr	Name	State	CRC	Reg:0002
1	1001	Term 1 (EK1100)	OP	0	0x0002 (11)
2	1002	Term 2 (EL1012)	OP	0	0x0002 (10)
3	1003	Term 3 (EL2004)	OP	0	0x0002 (11)
4	1004	Term 4 (EL3102)	OP	0	0x0002 (10)
5	1005	Term 5 (EL4102)	OP	0	0x000B (11)
6	1006	Term 6 (EL5001)	OP	0	0x0002 (11)
7	1007	Term 7 (EL6751)	OP	0	0x000C (12)

Actual State: OP
Send Frames: 74237
Frames / sec: 329
Lost Frames: 0

Number	Box Name	Address	Type	In Size	O
1	Term 1 (EK1100)	1001	EK1100	0.0	0
2	Term 2 (EL2004)	1002	EL2004	0.0	0
3	Term 3 (EL2004)	1003	EL2004	0.0	0
4	Term 4 (EL5001)	1004	EL5001	5.0	0

Fig. 269: FPGA firmware version definition

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

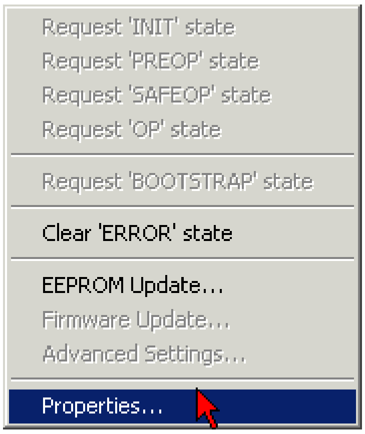
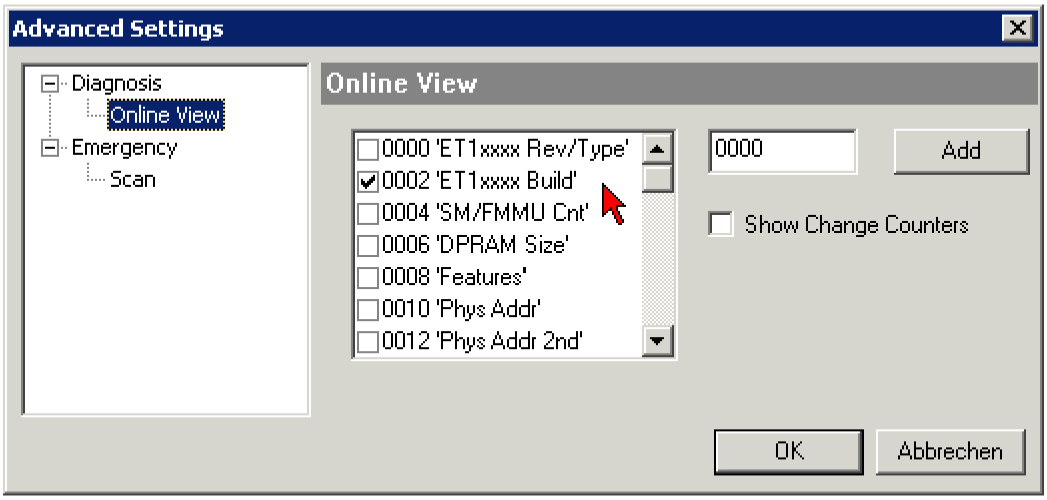


Fig. 270: Context menu *Properties*

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.



□ 0000 'ET1xxxx Rev/Type'
☑ 0002 'ET1xxxx Build'
□ 0004 'SM/FMMU Cnt'
□ 0006 'DPRAM Size'
□ 0008 'Features'
□ 0010 'Phys Addr'
□ 0012 'Phys Addr 2nd'

0000

□ Show Change Counters

Fig. 271: 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.

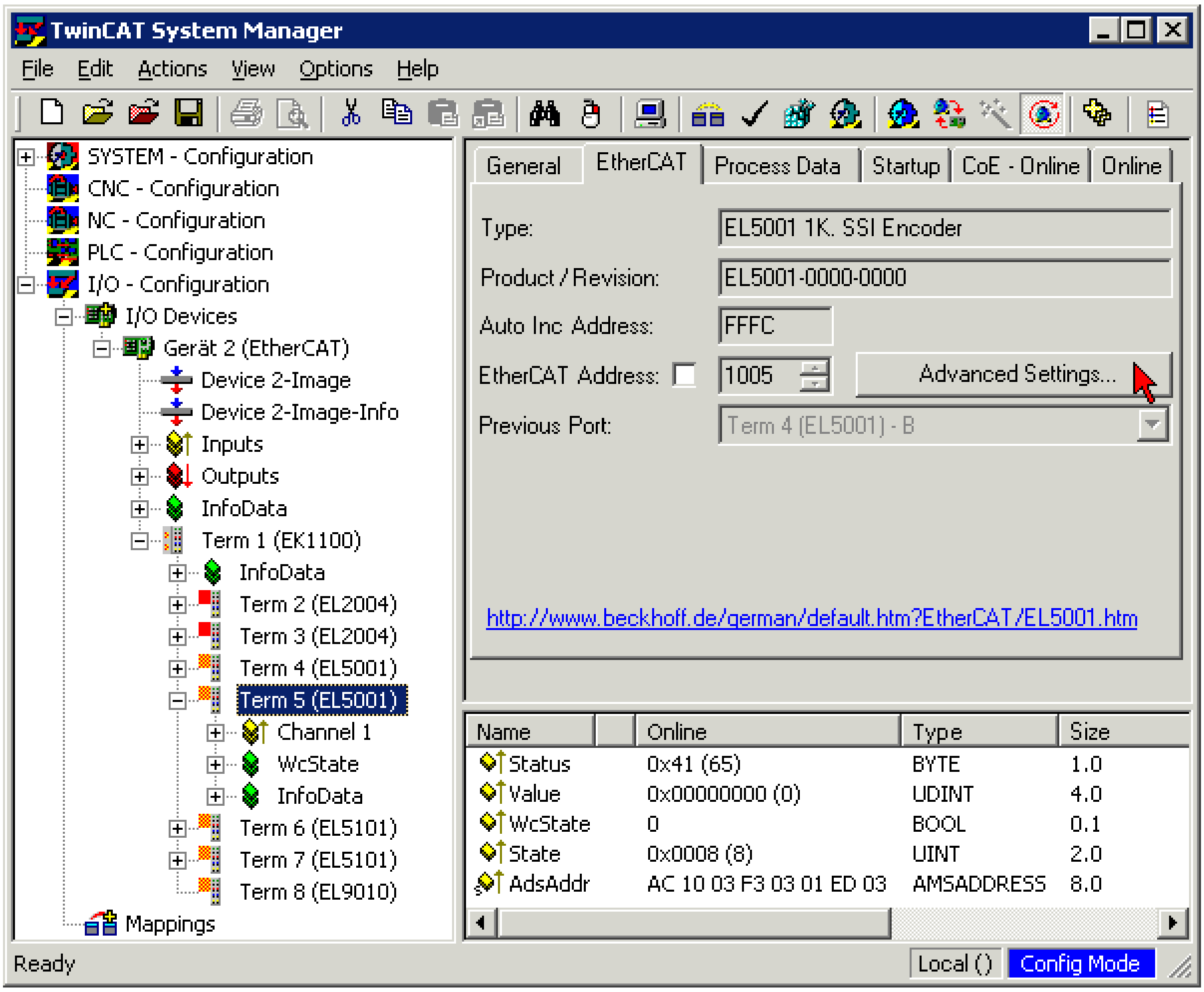
Older firmware versions can only be updated by the manufacturer!

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:



Type: EL5001 1K. SSI Encoder

Product / Revision: EL5001-0000-0000

Auto Inc Address: FFFC

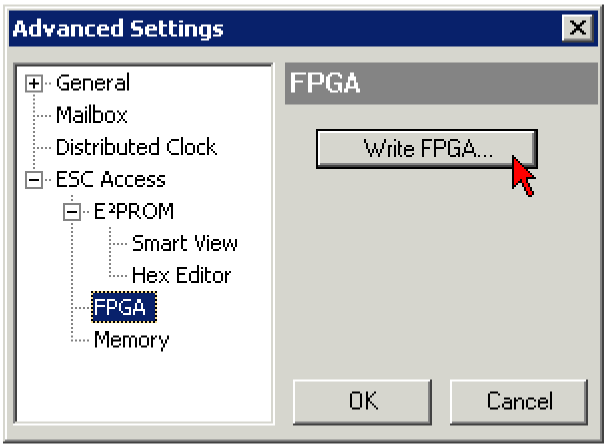
EtherCAT Address: □ 1005

Previous Port: Term 4 (EL5001) - B

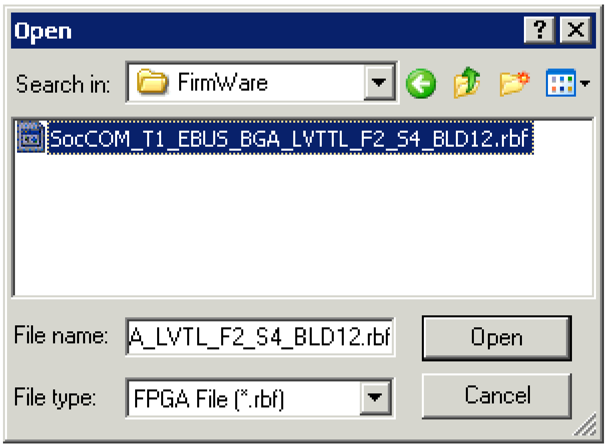
http://www.beckhoff.de/german/default.htm?EtherCAT/EL5001.htm

Name	Online	Type	Size
Status	0x41 (65)	BYTE	1.0
Value	0x00000000 (0)	UDINT	4.0
WcState	0	BOOL	0.1
State	0x0008 (8)	UINT	2.0
AdsAddr	AC 10 03 F3 03 01 ED 03	AMSADDRESS	8.0

- 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:



Search in: FirmWare

SocCOM_T1_EBUS_BGA_LVTTL_F2_S4_BLD12.rbf

File name: A_LVTL_F2_S4_BLD12.rbf

File type: FPGA File (*.rbf)

- 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

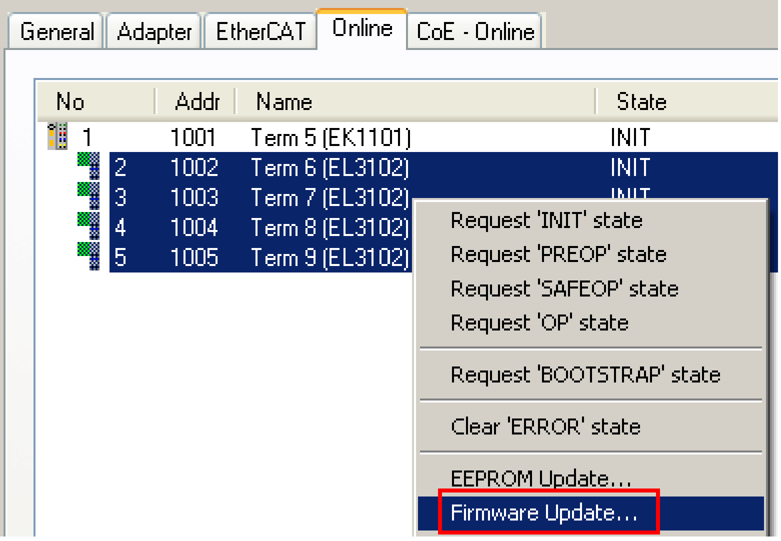
NOTE

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!

6.1.5 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.



No	Addr	Name	State
1	1001	Term 5 (EK1101)	INIT
2	1002	Term 6 (EL3102)	INIT
3	1003	Term 7 (EL3102)	INIT
4	1004	Term 8 (EL3102)	
5	1005	Term 9 (EL3102)	

Fig. 272: Multiple selection and firmware update

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

6.2 Firmware compatibility

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

Note

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

NOTE

Risk of damage to the device!

Pay attention to the instructions for firmware updates on the separate page [▶ 249]. If a device is placed in BOOTSTRAP mode for a firmware update, it does not check when downloading whether the new firmware is suitable. This can result in damage to the device! Therefore, always make sure that the firmware is suitable for the hardware version!

EL2596			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Release date**
01 - 03*	02	EL2596-0000-0017	2020/01
	03	EL2596-0000-0017	2020/03
	04*	EL2596-0000-0018	2020/07
		EL2596-0000-0019	2023/03

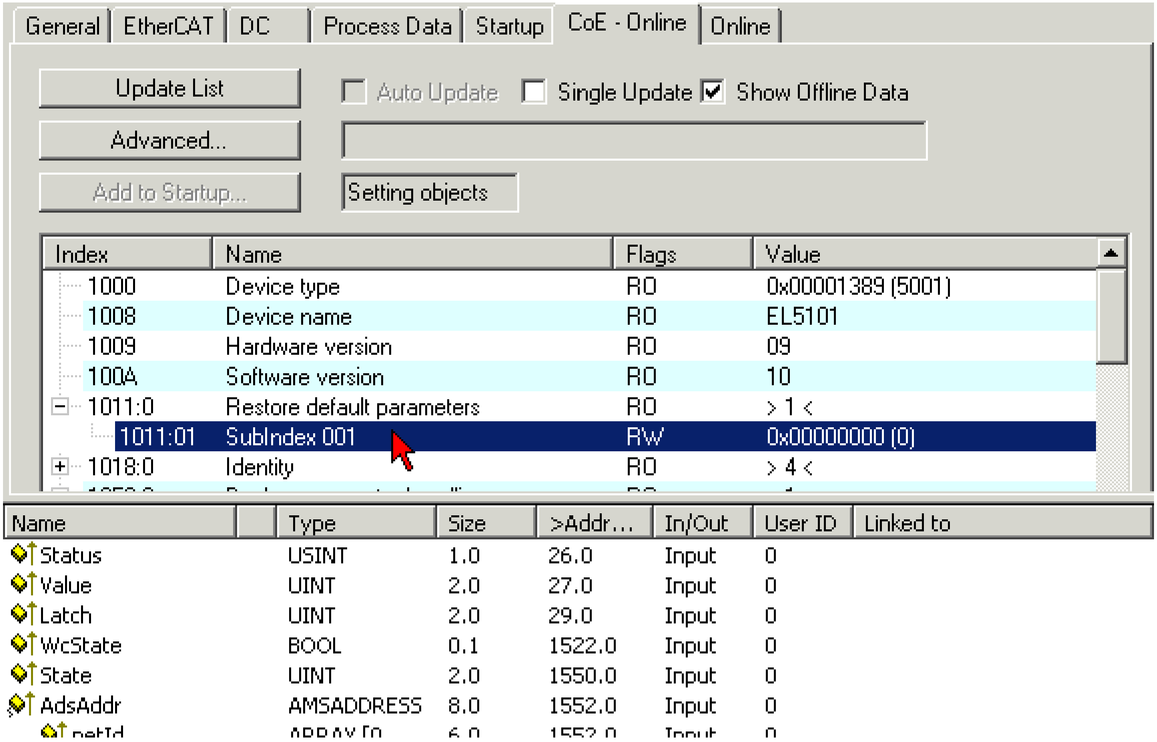
EL2596-0010			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Release date**
01*	01*	EL2596-0010-0016	2021/02

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

When updating the firmware of the EL2596, the settings and the user's calibration values are reset to the default values. The values of the vendor calibration are retained in case of a firmware update.

6.3 Restoring the delivery state

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



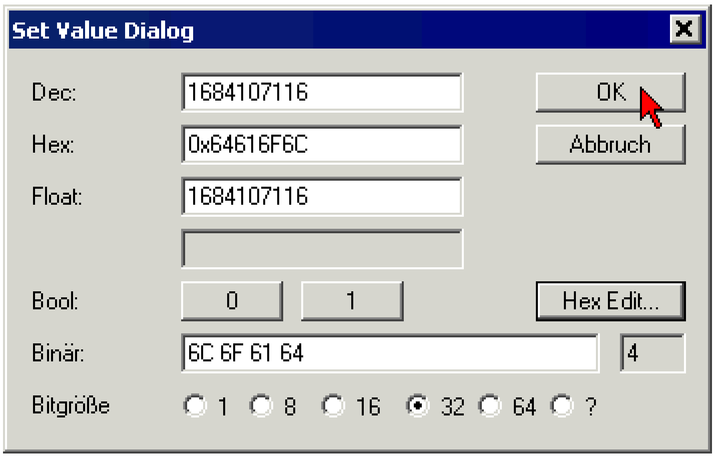
☐ Auto Update ☐ Single Update ☑ Show Offline Data

Index	Name	Flags	Value
1000	Device type	RO	0x00001389 (5001)
1008	Device name	RO	EL5101
1009	Hardware version	RO	09
100A	Software version	RO	10
1011:0	Restore default parameters	RO	> 1 <
1011:01	SubIndex 001	RW	0x00000000 (0)
1018:0	Identity	RO	> 4 <

Name		Type	Size	>Addr...	In/Out	User ID	Linked to
Status		USINT	1.0	26.0	Input	0	
Value		UINT	2.0	27.0	Input	0	
Latch		UINT	2.0	29.0	Input	0	
WcState		BOOL	0.1	1522.0	Input	0	
State		UINT	2.0	1550.0	Input	0	
AdsAddr		AMSADDRESS	8.0	1552.0	Input	0	
netId		ARRAY [0	6.0	1552.0	Input	0	

Fig. 273: 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.



Dec: 1684107116

Hex: 0x64616F6C

Float: 1684107116

Bool: 0 1

Binär: 6C 6F 61 64 4

Bitgröße ○ 1 ○ 8 ○ 16 ● 32 ○ 64 ○ ?

Fig. 274: 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: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Support

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

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

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

Service

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

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

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

Headquarters Germany

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Germany

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

More Information:
www.beckhoff.com/EL2596.htm

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