BECKHOFF New Automation Technology

Documentation | EN

EL7411

BLDC motor terminal with incremental encoder

2023-03-28 | Version: 1.4

Table of contents

1 Foreword

1.1 Notes on the documentation

Intended audience

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

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

Disclaimer

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

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

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

Trademarks

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

Patent Pending

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

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

Copyright

1.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Description of instructions

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

⚠ DANGER

Serious risk of injury!

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

⚠ WARNING

Risk of injury!

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

⚠ CAUTION

Personal injuries!

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

NOTE

Damage to environment/equipment or data loss

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

Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3 Documentation issue status

Version	Comment
1.4	• English translation updated
1.3	• Chapter "Commissioning" updated
1.2	• New chapters: ◦ Homing [▶ 129] ◦ Touch Probe [▶ 152] ◦ Commissioning with status word and control word [▶ 137] ◦ Commissioning with Drive Motion Control [▶ 141] ◦ Process data [▶ 156] • Structure update
1.1	• First release.
1.0	• First preliminary version.

1.4 Version identification of EtherCAT devices

1.4.1 General notes on marking

Designation

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

- family key
- type
- version
- revision

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

Notes

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

1.4.2 Version identification of EL terminals

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

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

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

1.4.3 Beckhoff Identification Code (BIC)

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

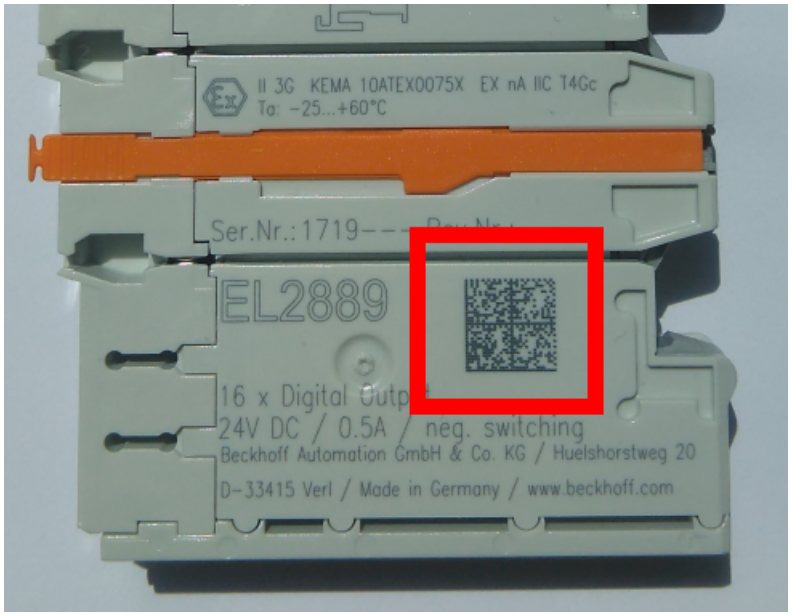


Fig. 2: BIC as data matrix code (DMC, code scheme ECC200)

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

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

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

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

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

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

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

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

Structure of the BIC

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

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

Accordingly as DMC:

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

BTN

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

NOTE

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

1.4.4 Electronic access to the BIC (eBIC)

Electronic BIC (eBIC)

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

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

K-bus devices (IP20, IP67)

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

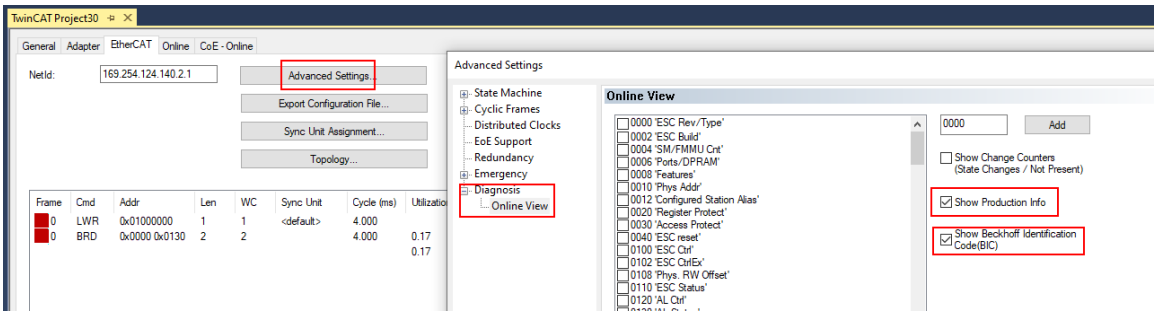
EtherCAT devices (IP20, IP67)

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

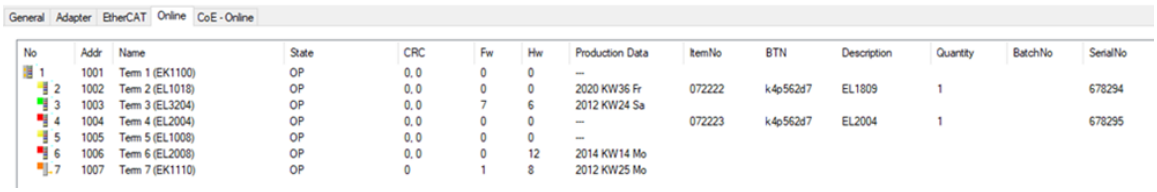
The eBIC is also stored in the ESI-EEPROM. The eBIC was introduced into the Beckhoff I/O production (terminals, box modules) from 2020; widespread implementation is expected in 2021.

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

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



  - The BTN and its contents are then displayed:



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

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

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

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

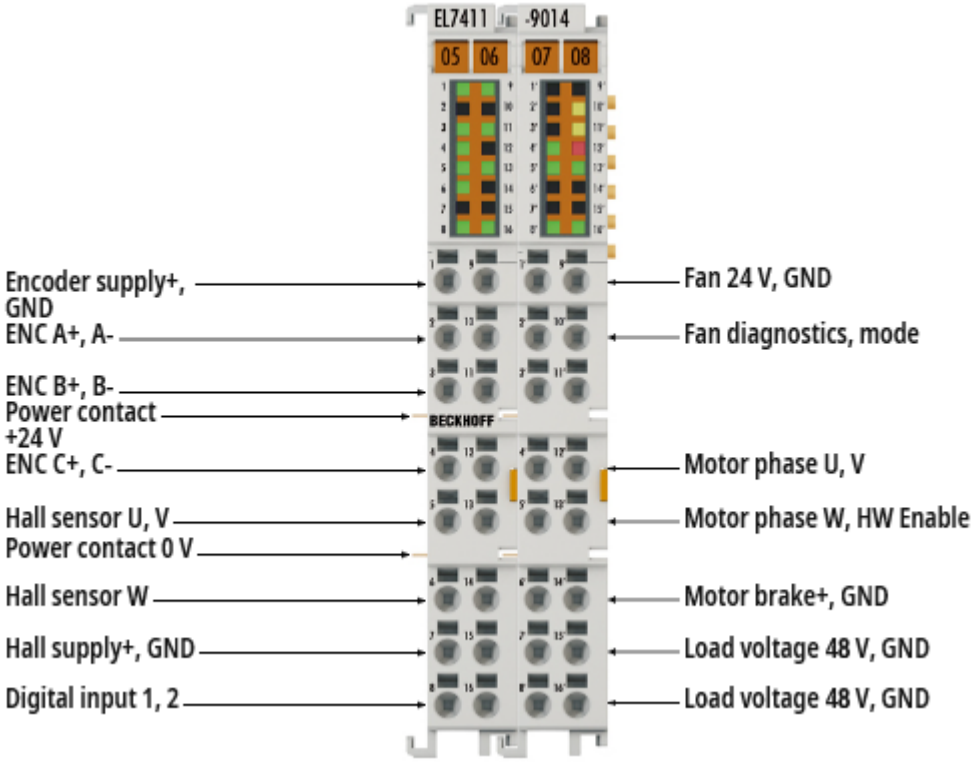
- Note: in the case of electronic further processing, the BTN is to be handled as a string(8); the identifier "SBTN" is not part of the BTN.
- Technical background
  The new BIC information is additionally written as a category in the ESI-EEPROM during the device production. The structure of the ESI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

Profibus/Profinet/DeviceNet... Devices

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

2 Product overview

2.1 Introduction



EL7411 | BLDC motor terminal with incremental encoder, 48 V_{DC}, 4.5 A (I_{rms})

The EL7411 BLDC motor terminal provides high control performance in a very compact design for the medium output range of BLDC motors. Due to the fast control technology and the connection of an incremental encoder, both very high speed profiles and dynamic positioning tasks can be implemented.

Maximum operational reliability is provided through the monitoring of numerous parameters, such as overvoltage, undervoltage, overcurrent, terminal temperature or motor load via the calculation of an I^2T model.

The terminal's output current can be increased in conjunction with the ZB8610 fan cartridge.

2.2 Technical data

All values are typical values over the entire temperature range, unless stated otherwise.

Inputs and outputs	
Inputs	2 end positions 1 encoder 1 fan status 3 Hall sensors
Outputs	1 x BLDC motor 1 x motor brake 1 x fan supply 1 x fan mode 1 x sensor supply 1 x encoder supply

E-bus	
Electrical isolation	500 V (E-bus / I/O)
Distributed Clocks	yes
Current consumption via E-bus	190 mA

Supply voltages	
Electronics supply voltage	24 V_{DC} via the power contacts
Current consumption from the power contacts	50 mA + holding current for the motor brake
Current load of the power contacts	max. 10 A
Load voltage	8 … 48 V_{DC} (must be supplied externally)
Reverse polarity protection	• 24 V supply voltage: yes, through the body diode of the surge protection device • 48 V supply voltage: yes, through the body diode of the surge protection device
Required fuse protection (to be performed by the user)	• 24 V power supply: 10 A • 48 V power supply: 10 A

Motor output stage	
Motor type	BLDC motor
Minimum winding inductance	200 µH
Number of channels	1
Output current (rms)	4.5 A
Output current (rms) with ZB8610 fan cartridge	7.0 A up to 55 °C ambient temperature 8.0 A up to 45 °C ambient temperature
Peak current (rms)	max. 9.0 A for 1 s
Peak current (rms) with ZB8610 fan cartridge	max. 16.0 A for 1 s up to 55 °C ambient temperature
Rotary field frequency	0...599 Hz
PWM clock frequency	16 kHz
Current controller frequency	32 kHz

Output for the motor brake	
Output voltage	24 V_{DC}
Output current	max. 0.5 A

Encoder	
encoder type	Incremental encoders
Signal types	See chapter Configuration of the incremental encoder [▶ 105].
Maximum input frequency	See chapter Configuration of the incremental encoder [▶ 105].
Supply voltage output for encoder	2…24 V_{DC}, adjustable. Factory setting: 5 V_{DC}. • resolution: 20 mV • accuracy: ± 10 % max. 300 mA

Hall sensors	
Signal type	Open Collector
Supply voltage output for Hall sensors	2…24 V_{DC}, adjustable. Factory setting: 5 V_{DC}. • resolution: 20 mV • accuracy: ± 10 % max. 300 mA

Digital inputs (Touch probe)	
Signal level high	≥ 5 V
Signal level low	≤ 2 V
Input current	5 … 6 mA

Firmware functionality	
NoCoEStorage function	yes

Environmental conditions	
Ambient temperature during operation	0 … +55 °C
Ambient temperature during storage	-25 … +85 °C
Relative humidity	95 % no condensation
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

Housing data	
Design	compact HD (High Density) housing with signal LEDs
Weight	approx. 95 g
Installation position	variable. Recommended: standard installation position [▶ 41]
Material	polycarbonate
Dimensions (W x H x D)	24 mm x 100 mm x 68 mm
Assembly	On 35 mm DIN rail, according to EN 60715. With lock.

Approvals and conformity	
Approvals	CE, EAC, UKCA, UL [▶ 52]

2.3 Technology

The EL7411 integrates a full-fledged amplifier for brushless DC motors/electric commutation motors, or BLDC motor/EC motor for short). Although the name would suggest otherwise, this motor does not resemble a conventional single-phase DC motor, but rather a three-phase permanently excited synchronous or servo motor. Both are three-phase motors whose windings are operated with a 120° phase shift. The differences in the interior structure are the slightly deviating windings. A servomotor has a sinusoidal counter-electromotive force, while that of a BLDC motor is trapezoidal. The differences in the design are much more visible. BLDC motors often have a much lower axial height, allowing them to be used in machines where installation space is limited. In addition, they often only have wires fed to the outside and no connector for the connection of a motor cable. Also, these motors normally have no high-resolution feedback installed, but are mostly commutated and operated with integrated Hall sensors or an incremental encoder mounted on the motor shaft. Due to the points listed, BLDC motors are in most cases cheaper than servomotors.

The BLDC motor and the EL7411 amplifier output stage together form the drive. The BLDC motor is operated in a closed control loop, with position, velocity or torque control.

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.

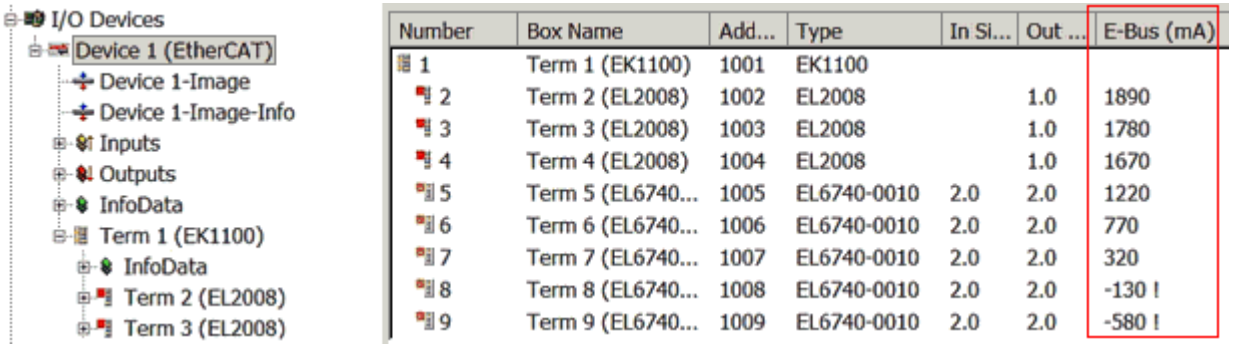


Fig. 4: System manager current calculation

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:

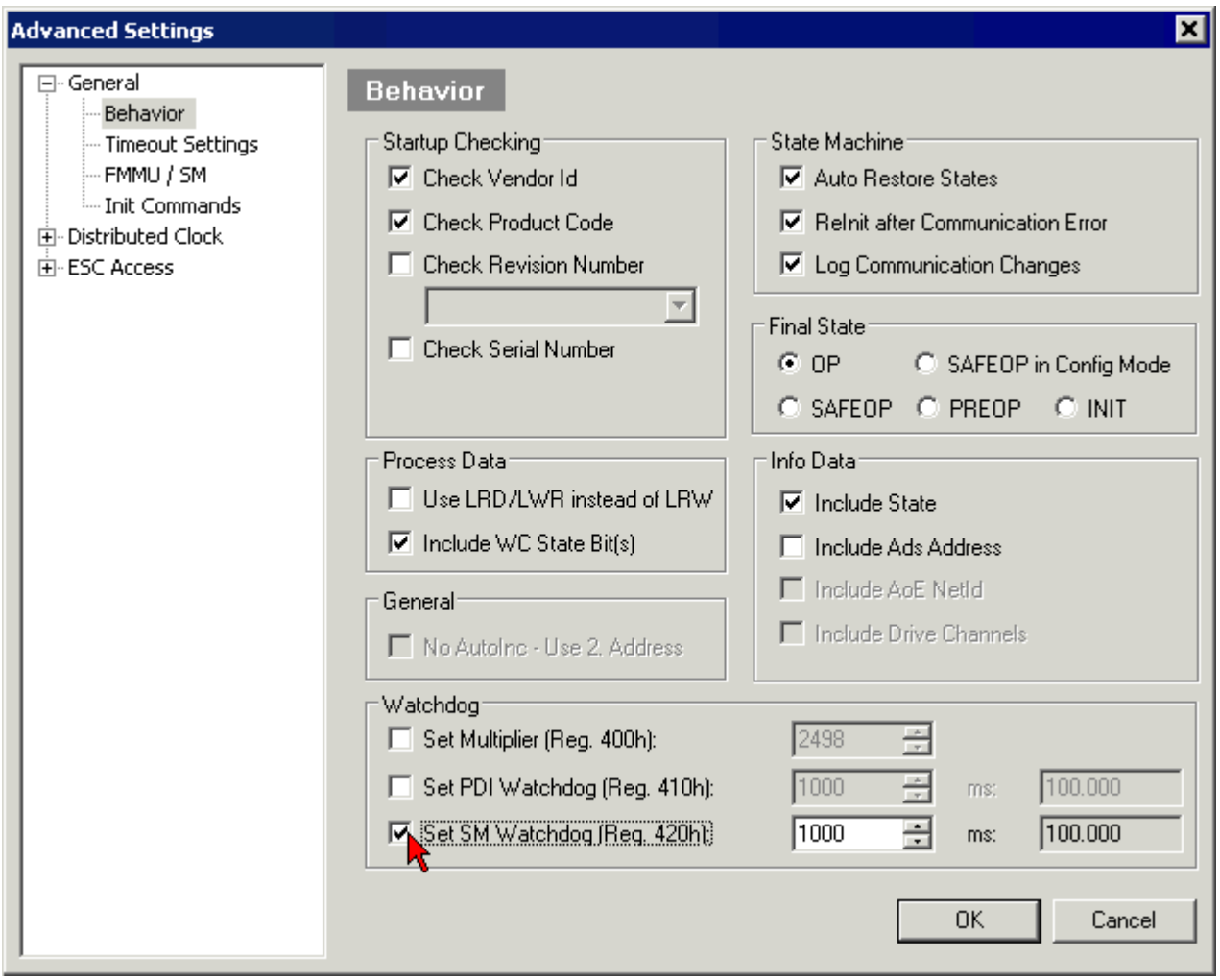


Fig. 5: eEtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

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.

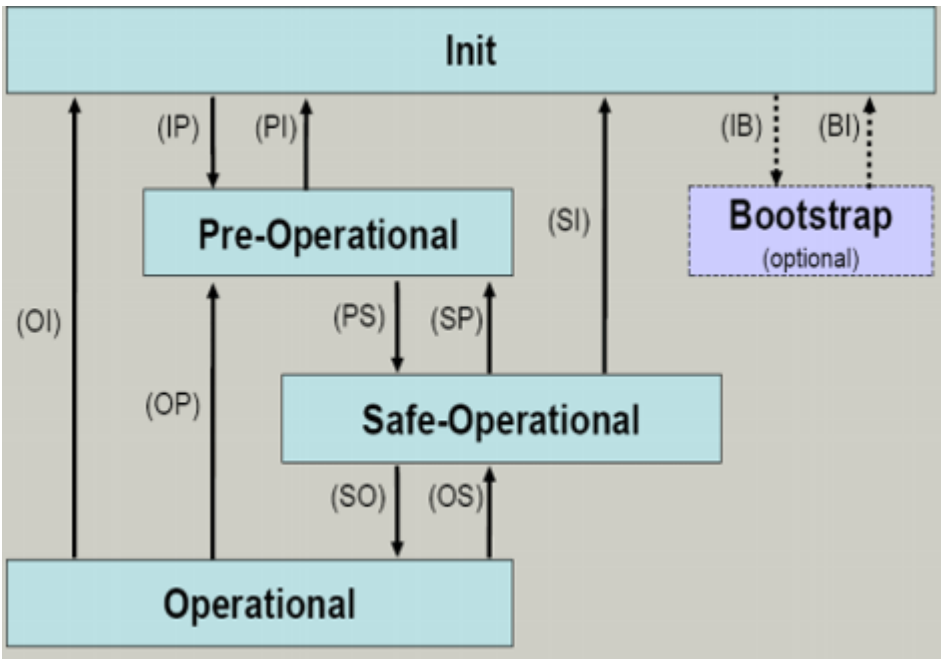


Fig. 6: States of the EtherCAT State Machine

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

Operational (Op)

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

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

Boot

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

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

3.5 CoE Interface

General description

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

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

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

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

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

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

The relevant ranges for EtherCAT fieldbus users are:

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

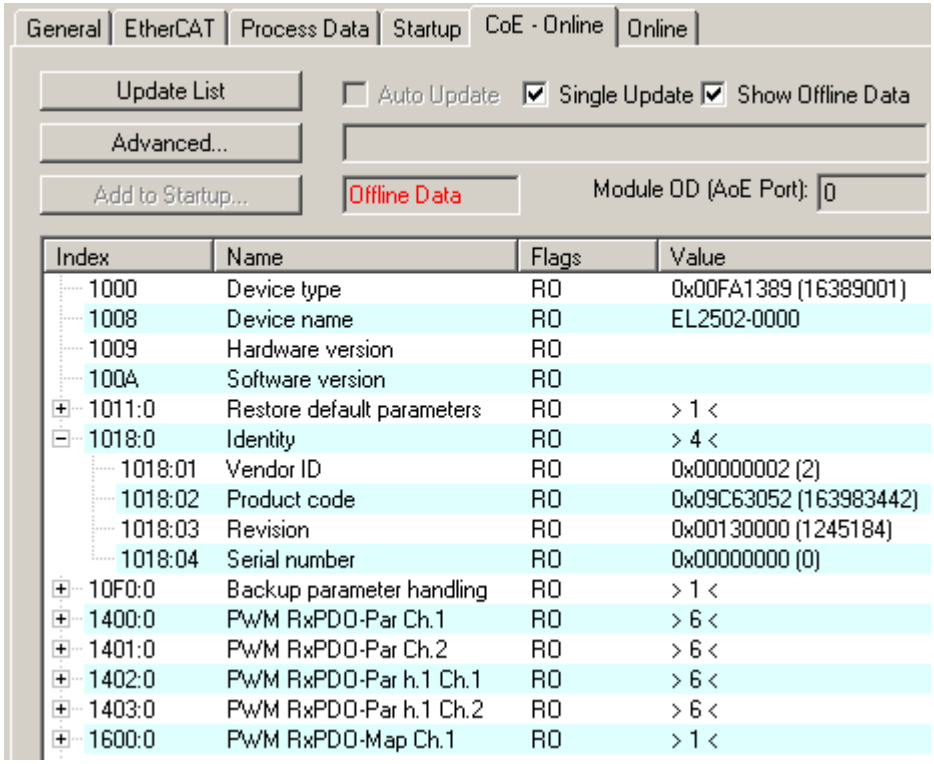
Other important ranges are:

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

Availability

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

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



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

Fig. 7: “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.

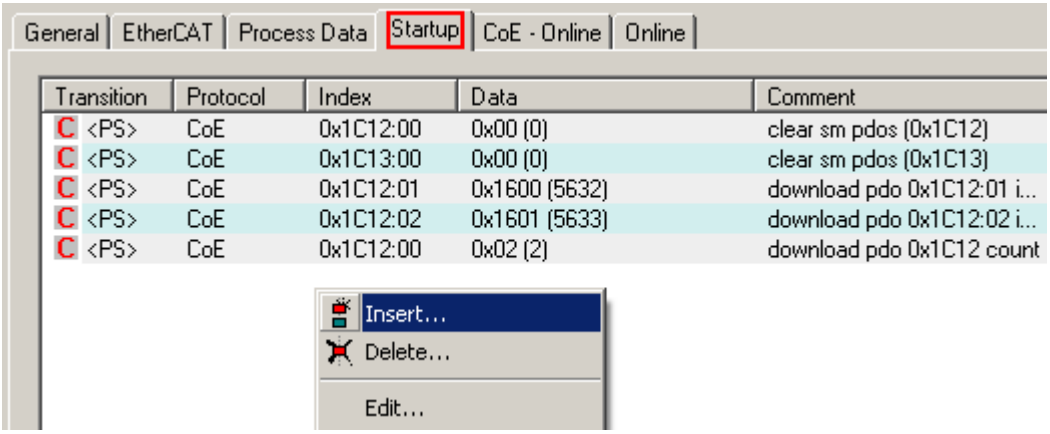


Fig. 8: Startup list in the TwinCAT System Manager

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

Online/offline list

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

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

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

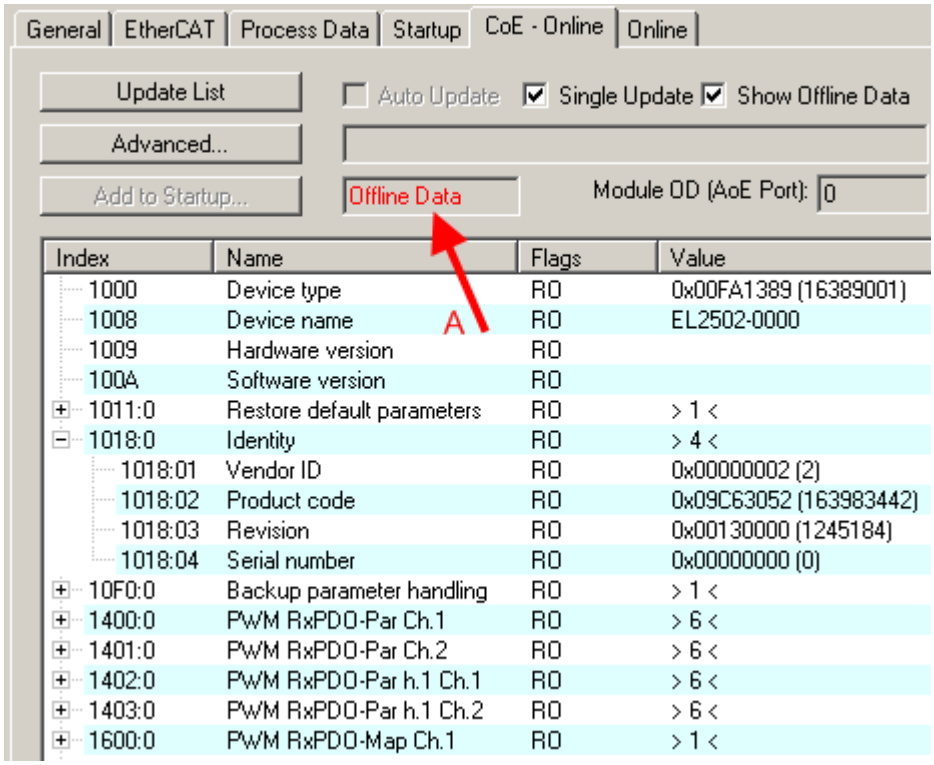


Fig. 9: Offline list

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

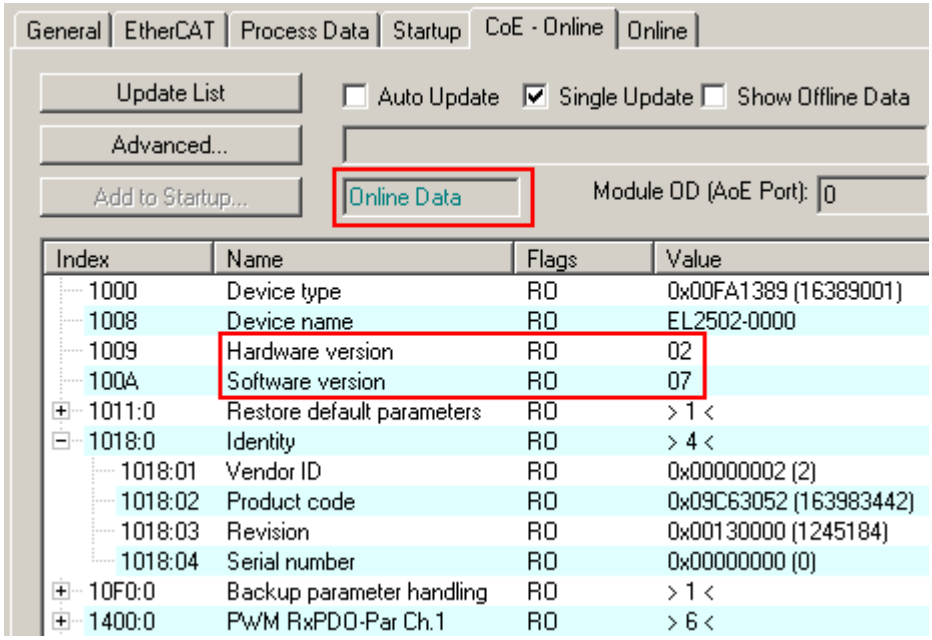


Fig. 10: Online list

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 Installation

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.

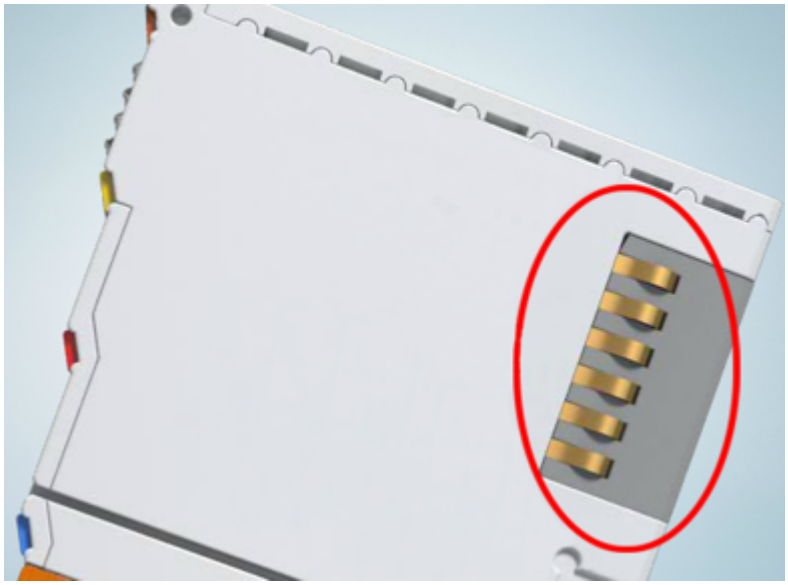


Fig. 11: Spring contacts of the Beckhoff I/O components

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

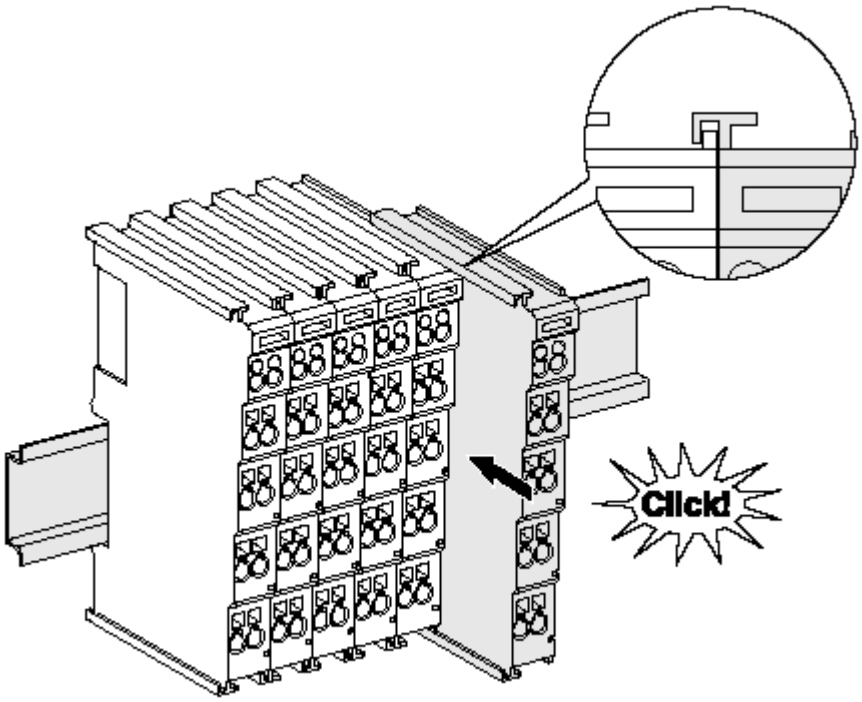


Fig. 12: Attaching on mounting rail

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

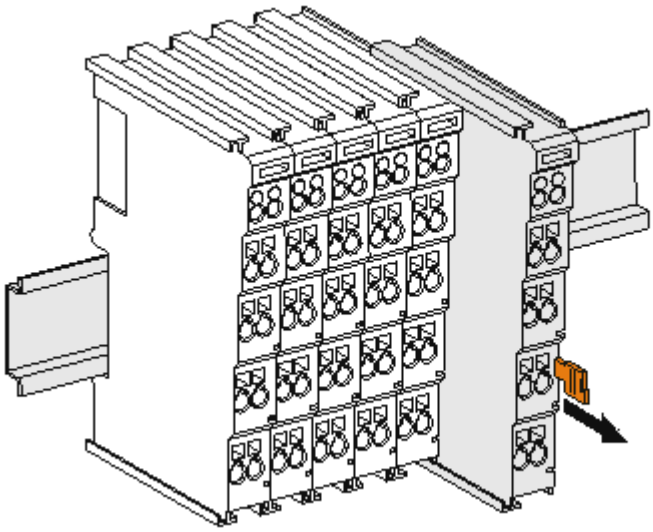


Fig. 13: Disassembling of terminal

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.

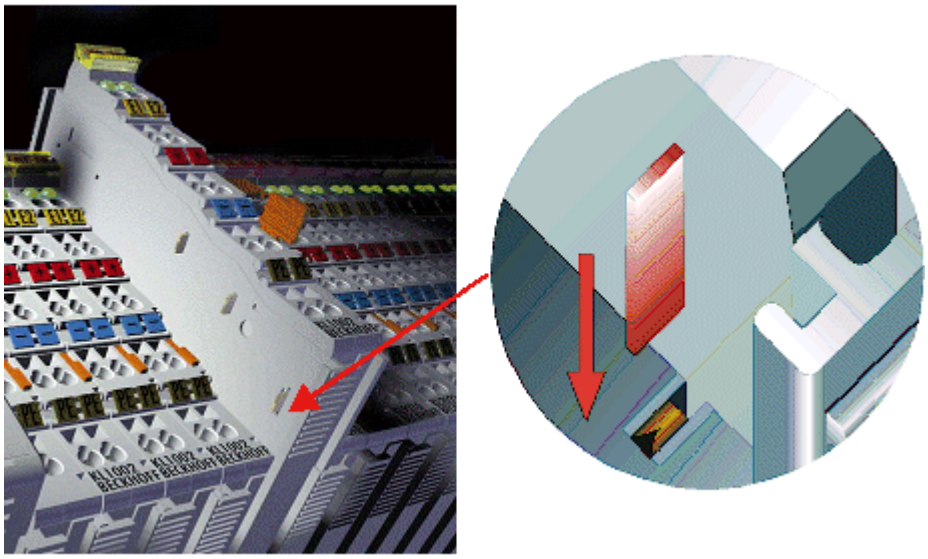


Fig. 14: Power contact on left side

NOTE

Possible damage of the device

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

⚠ WARNING

Risk of electric shock!

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

4.3 Installation instructions for enhanced mechanical load capacity

⚠ WARNING

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

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

Additional checks

The terminals have undergone the following additional tests:

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

Additional installation instructions

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

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

4.4 Connection 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. 15: 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)

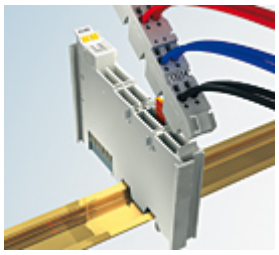


Fig. 16: Pluggable wiring

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

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

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

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

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

High Density Terminals (HD Terminals)

Fig. 17: 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 [▶ 38] below!

Wiring

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

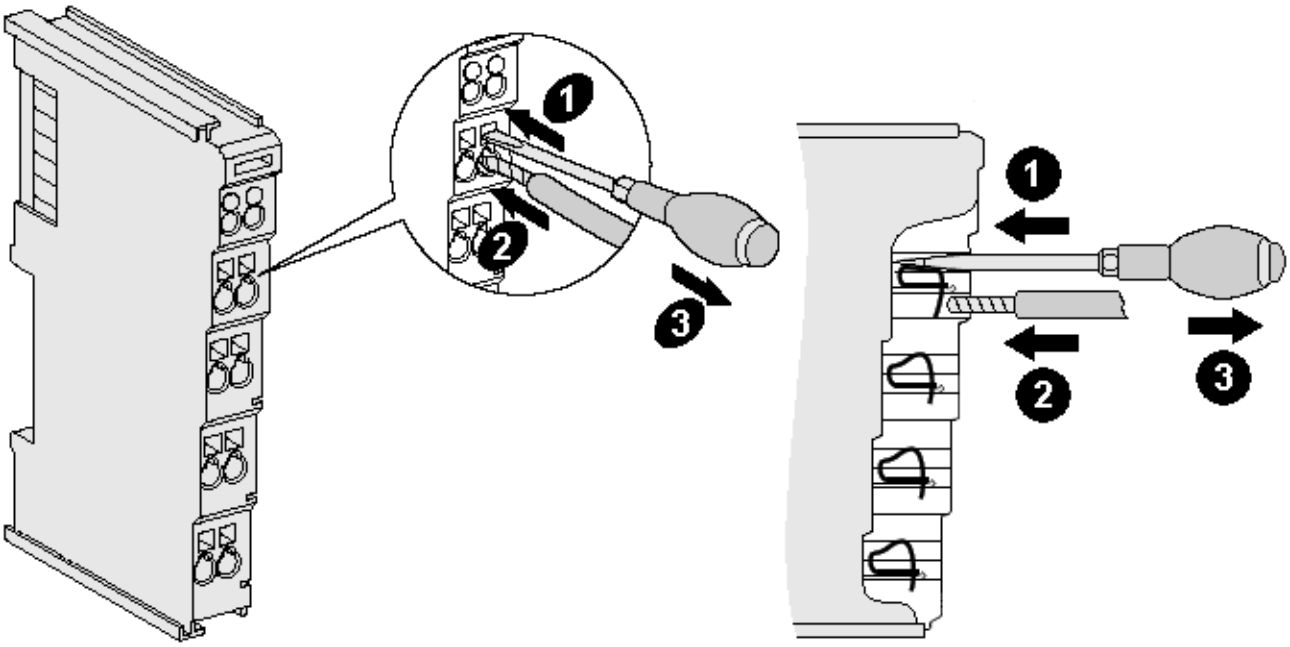


Fig. 18: Connecting a cable on a terminal point

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

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

See the following table for the suitable wire size width.

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

High Density Terminals ELx8xx, KLx8xx (HD)

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

Terminal housing	High Density Housing
Wire size width (single core wires)	0.08 ... 1.5 mm²
Wire size width (fine-wire conductors)	0.25 ... 1.5 mm²
Wire size width (conductors with a wire end sleeve)	0.14 ... 0.75 mm²
Wire size width (ultrasonically “bonded" conductors)	only 1.5 mm² (see notice [▶ 37]!)
Wire stripping length	8 ... 9 mm

Shielding

Shielding

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

4.5 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)

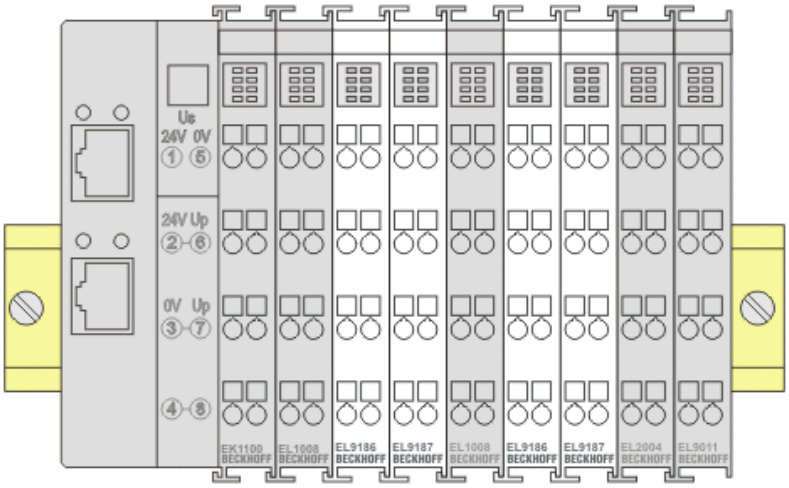


Fig. 19: Correct positioning

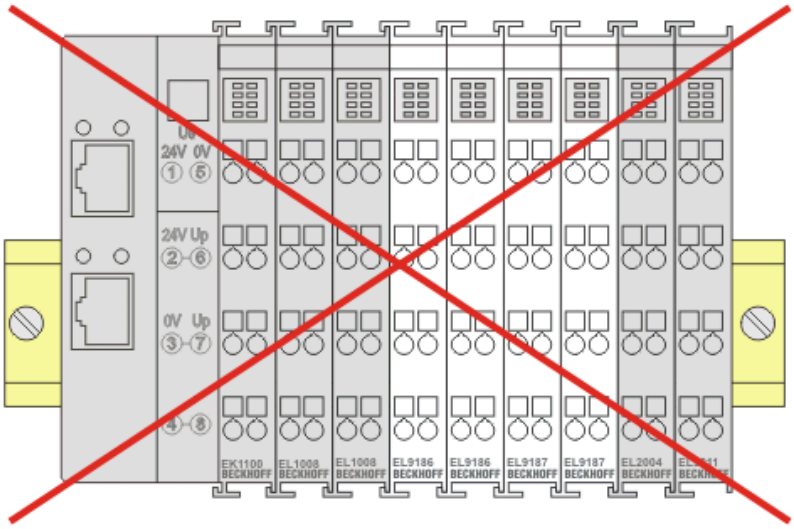


Fig. 20: Incorrect positioning

4.6 Installation position for operation with or without fan

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 for operation without fan

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 of installation position for operating without fan*").

The terminals are ventilated from below, which enables optimum cooling of the electronics through convection.

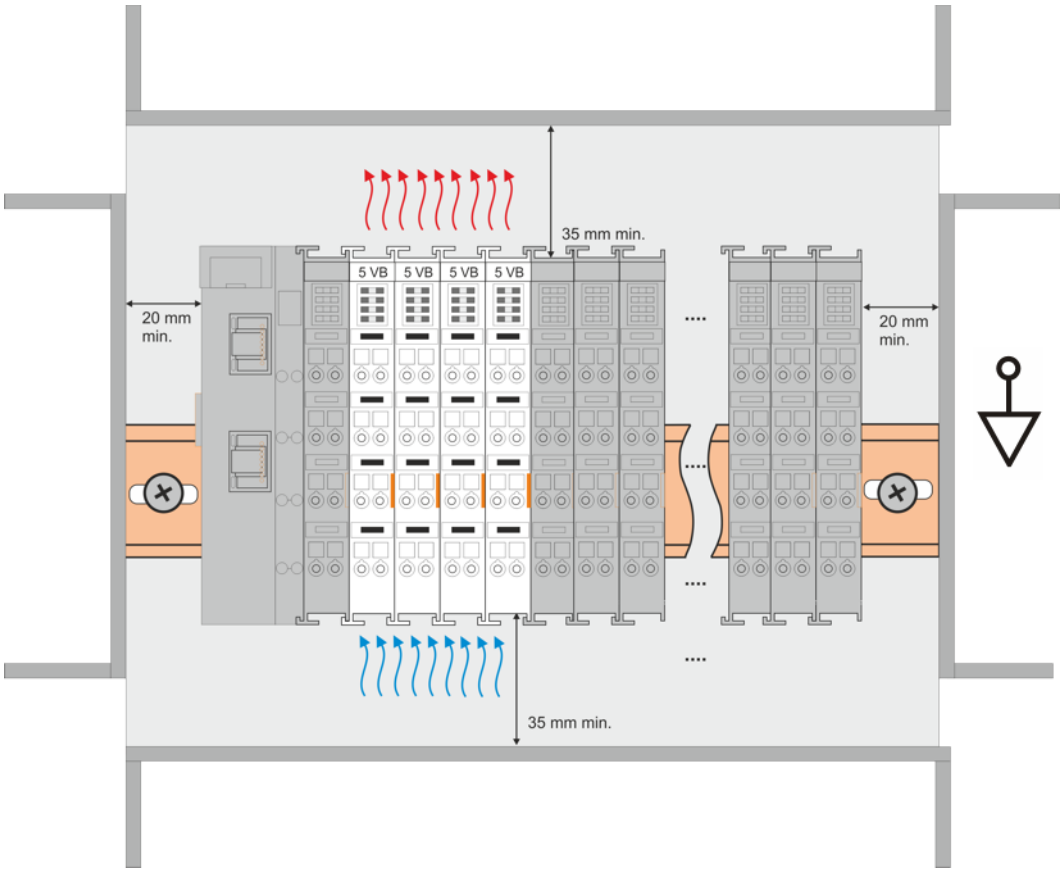


Fig. 21: Recommended distances of installation position for operating without fan

Compliance with the distances shown in Fig. "*Recommended distances of installation position for operating without fan*" is recommended.

For further information regarding the operation without fan refer to the Technical Data of the terminal.

Standard installation position for operation with fan

The standard installation position for operation with fan 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 installation position for operation with fan*).
The terminals are ventilated fan supported (e.g. with ZB8610 fan cartridge) from below.

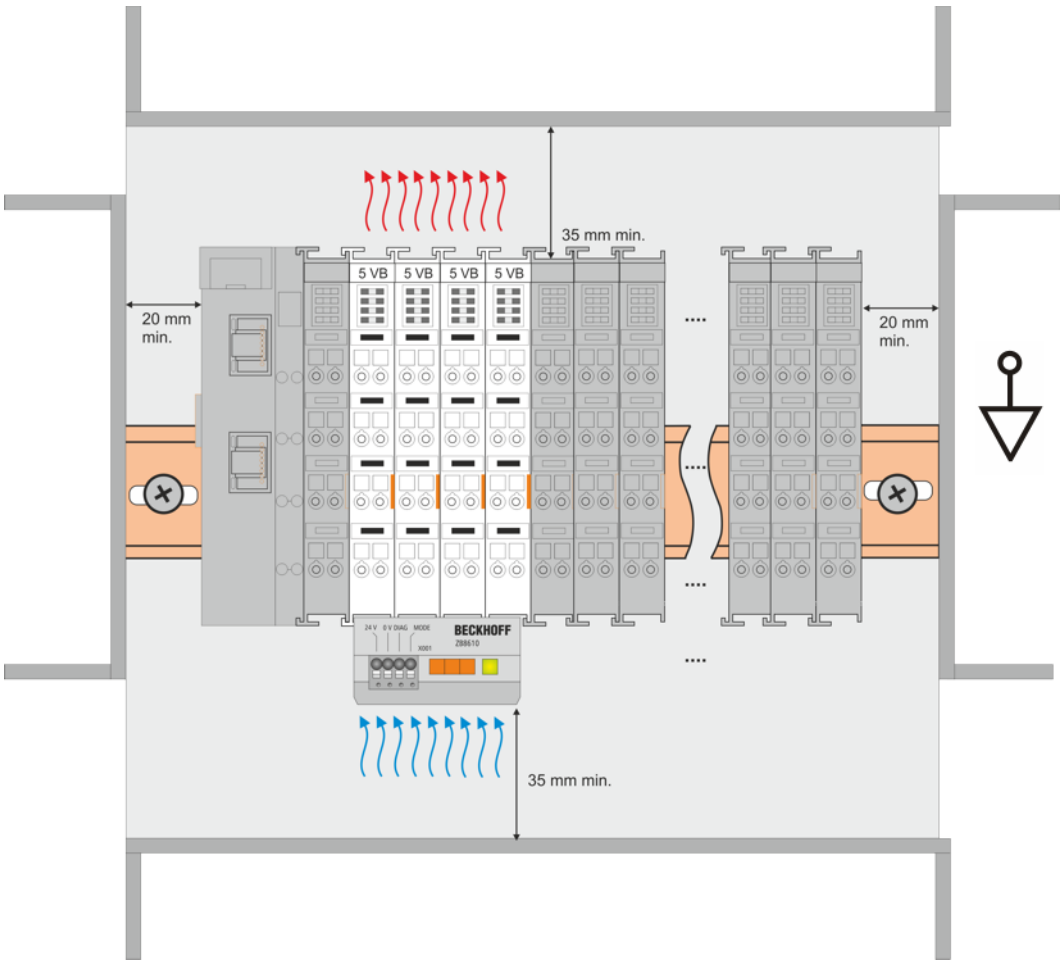


Fig. 22: Recommended distances for installation position for operation with fan

Other installation positions

Due to the enforced effect of the fan on the ventilation of the terminals, other installation positions (see Fig. "*Other installation positions, example 1 + 2*") may be permitted where appropriate.

See corresponding notes in the Technical Data of the terminal.

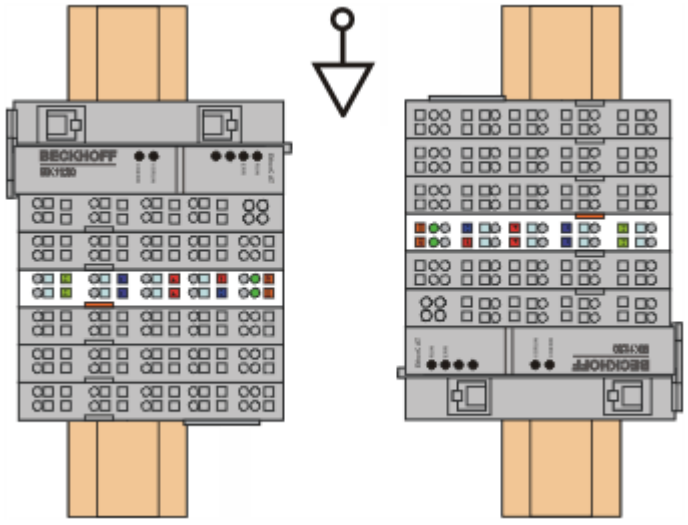


Fig. 23: Other installation positions, example 1

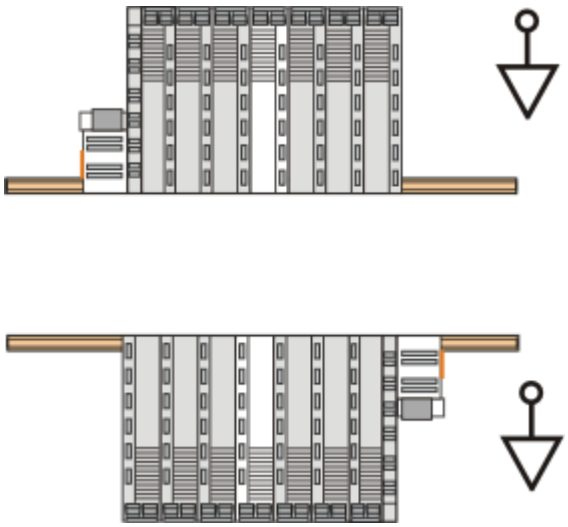


Fig. 24: Other installation positions, example 2

4.7 Shielding concept

Together with the shield busbar, the prefabricated cables from Beckhoff Automation offer optimum protection against electromagnetic interference.
It is highly recommended to apply the shield as close as possible to the terminal, in order to minimize operational disturbances.

Connection of the motor cable to the shield busbar

Fasten the shield busbar supports 1 to the DIN rail 2. The mounting rail 2 must be in contact with the metallic rear wall of the control cabinet over a wide area. Install the shield busbar 3 as shown below.
As an alternative, a shield busbar clamp 3a can be screwed directly to the metallic rear wall of the control cabinet (fig. “shield busbar clamp”)

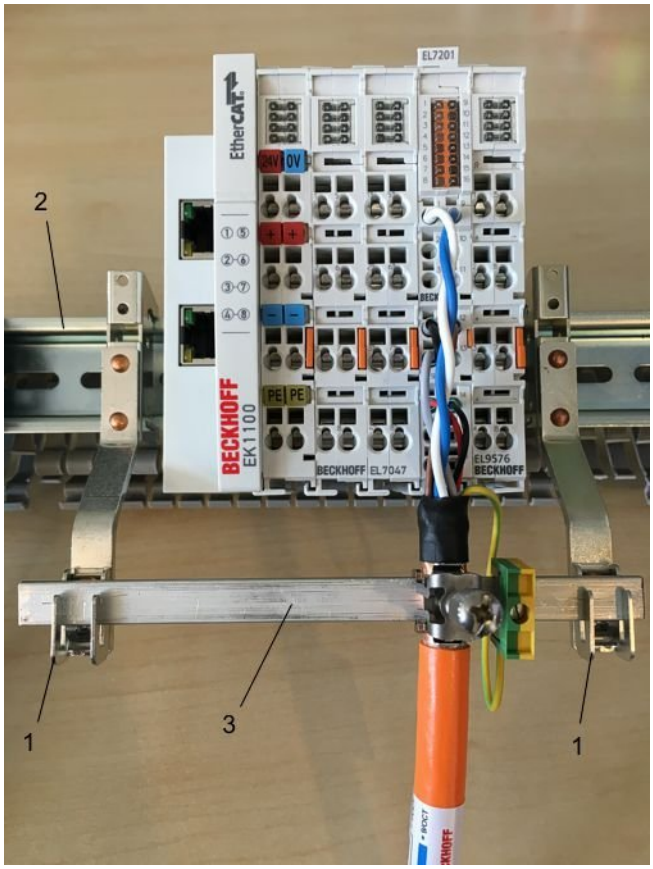


Fig. 25: Shield busbar

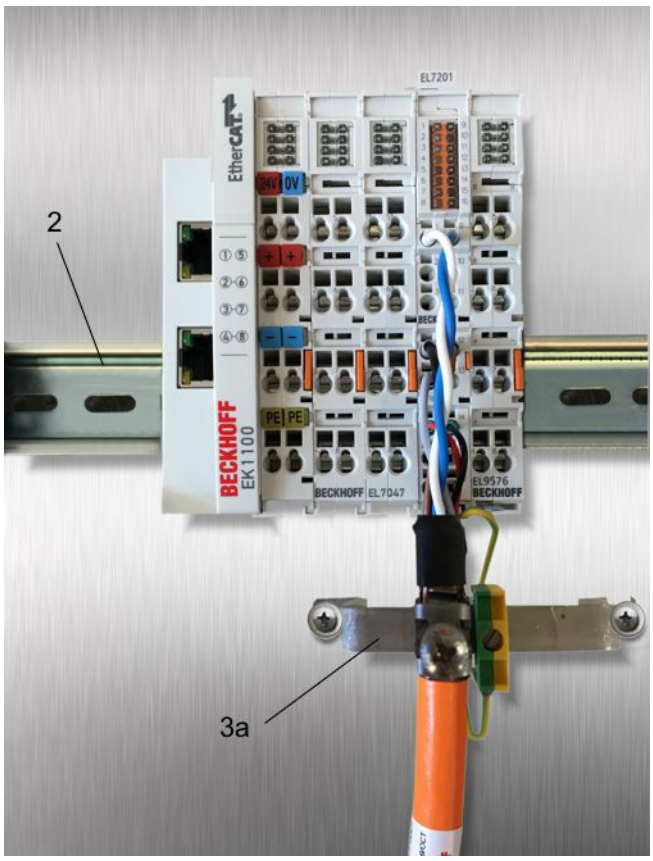


Fig. 26: Shield busbar clamp

Connect the cores 4 of the motor cable 5, then attach the copper-sheathed end 6 of the motor cable 5 with the shield clamp 7 to the shield busbar 3 or shield busbar clamp 3a. Tighten the screw 8 to the stop. Fasten the PE clamp 9 to the shield busbar 3 or shield busbar clamp 3a. Clamp the PE core 10 of the motor cable 5 under the PE clamp 9.

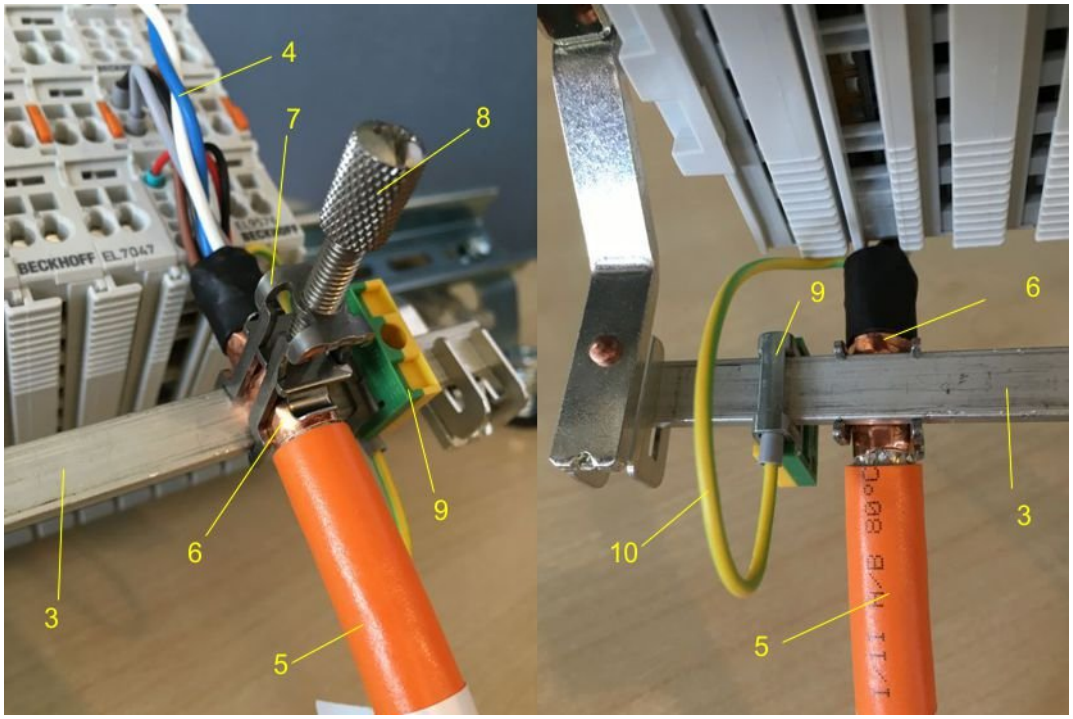


Fig. 27: Shield connection

Connection of the feedback cable to the motor

Feedback cables

Use shielded feedback cables. For differential signal types, it is recommended to twist the respective cores.

When the feedback connector is screwed to the motor, the shield connection of the feedback cable is made via the metallic connector fastening.
On the terminal side the shield can also be connected. Wire the cores of the feedback cable and fasten the copper-sheathed end of the feedback cable to the shield busbar 3 or shield busbar bracket 3a using the shield clamp 7. The motor cable and the feedback cable can be connected to the shield clamp 7 with the screw 8.

4.8 Notes on current measurements using Hall sensors

The device described in this documentation features one or several integrated Hall sensor for the purpose of current measurements.

During this process, the Hall sensor monitors the magnetic field generated by a current flowing through a conductor.

In order to prevent compromising the measurement we recommend screening exterior magnetic fields from the device, or to keep such fields at an adequate distance.

Fig. 28: Note

Background

A current-carrying conductor generates a magnetic field around it according to

$B = \mu_0 * I / (2\pi * d)$

with

B [Tesla] magnetic field

$\mu 0 = 4*\pi*10^{-7}$ [H/m] (assumption: no magnetic shielding)

I [A] current

d [m] distance to conductor

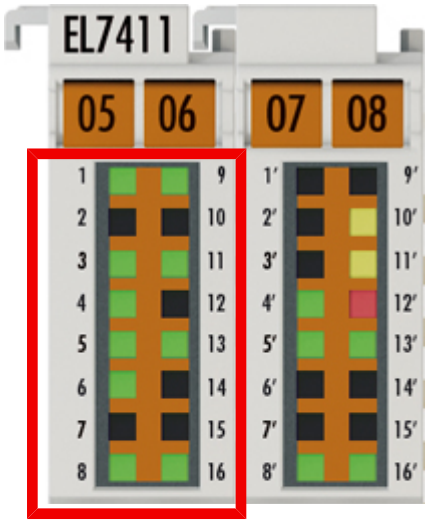
Interference from external magnetic fields

The magnetic field strength should not exceed a permitted level all around the device.
In practice this equates to a recommended minimum distance between a conductor and the device surface as follows:

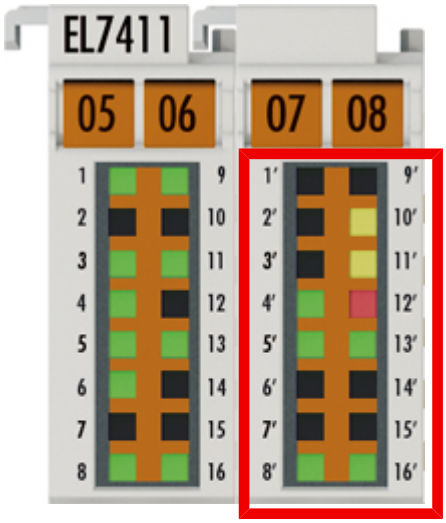
- Current 10 A: 12 mm
- Current 20 A: 25 mm
- Current 40 A: 50 mm

Unless specified otherwise in the device documentation, stringing together modules (e.g. terminal blocks based on a 12 mm grid) of same type (e.g. EL2212-0000) is permitted.

4.9 LEDs



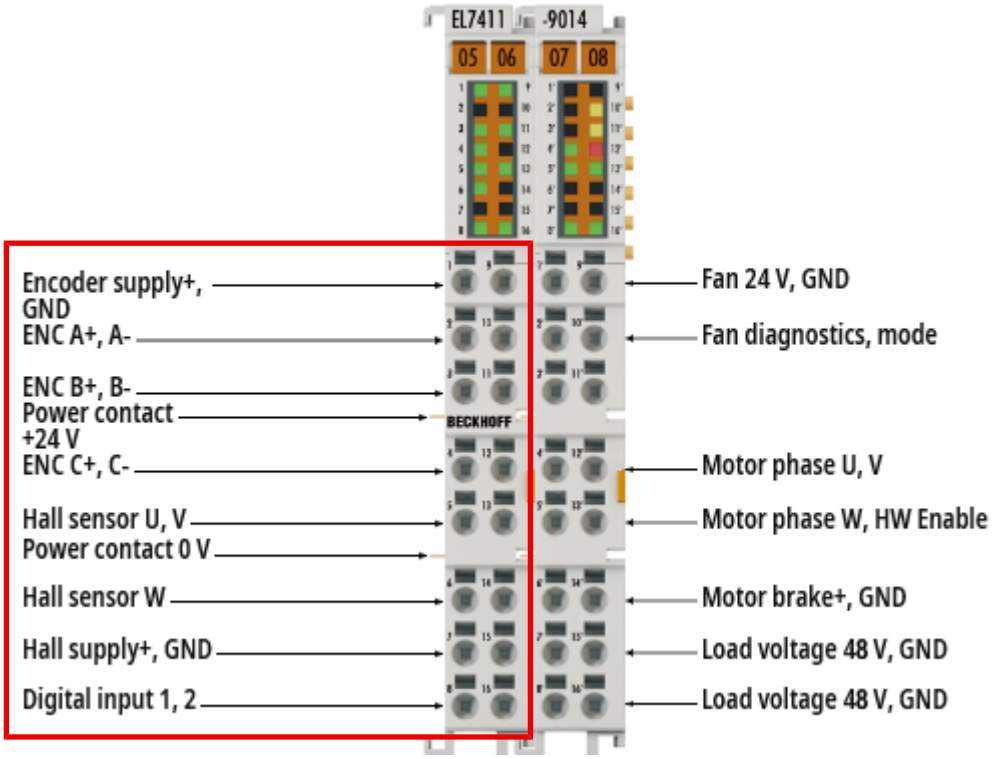
LED Number	Name	Color	Meaning	
1	Run	green	RUN: Status of the terminal in the EtherCAT network	
			off	Status "Init"
			flashes	Status "Pre-Operational"
			Single flash	Status "Safe-Operational"
			lit	Status "Operational"
2	-	-	-	
3	ENC A	green	on	A signal is present at encoder input A.
4	ENC C	green	on	A signal is present at encoder input C.
5	Hall U	green	on	Hall sensor U
6	Hall W	green	on	Hall sensor W
7	-	-	-	
8	Input 1	green	on	A high level is present at "Input 1".
9	-	-	-	
10	-	-	-	
11	ENC B	green	on	A signal is present at encoder input B.
12	-	-	-	
13	Hall V	green	on	Hall sensor V
14	-	-	-	
15	-	-	-	
16	Input 2	green	on	A high level is present at "Input 2".



LED Number	Name	Color	Meaning	
1‘	-	-	-	
2‘	-	-	-	
3‘	-	-	-	
4‘	Enable	green	on	The axis is enabled.
5‘	Fan Diag	green	on	The fan module is connected and works without errors.
6‘	-	-	-	
7‘	-	-	-	
8‘	24V	green	on	The electronics supply voltage is present
9‘	-	-	-	
10‘	Limit	yellow	on	Limit
11‘	Warning	yellow	on	Warning
12‘	Error	red	on	Error
13‘	Mode	green	on	Fan cartridge Signal "Mode"
14‘	-	-	-	
15‘	-	-	-	
16‘	48V	green	on	The load voltage is present

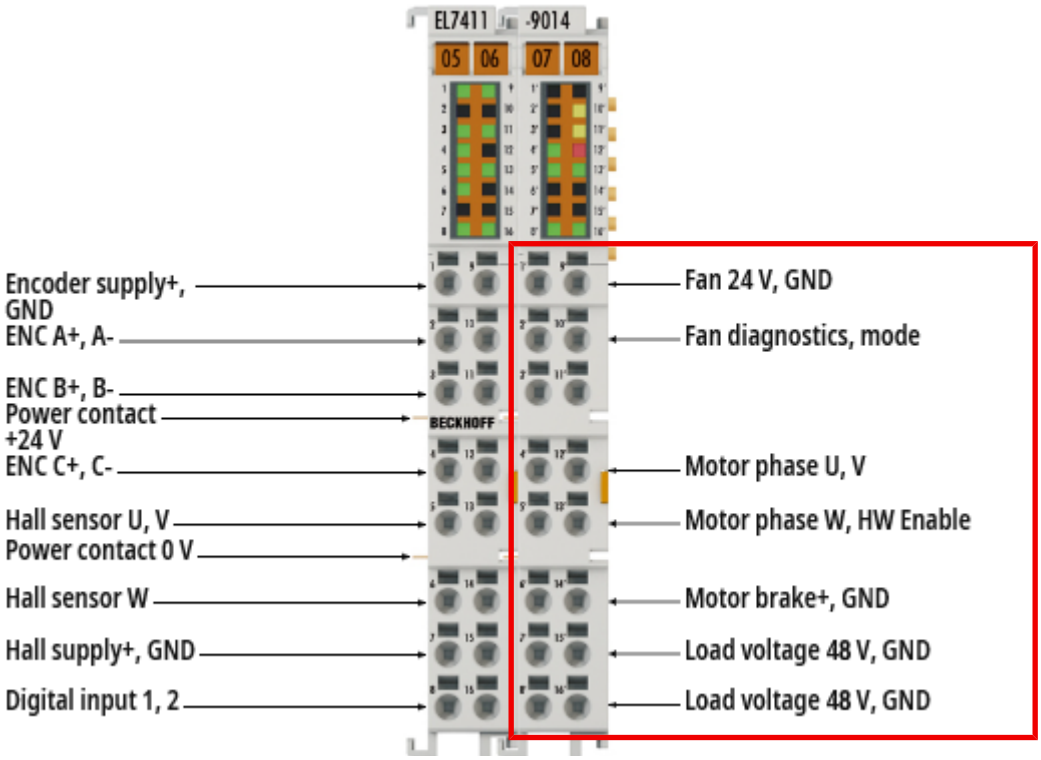
4.10 Connection

Terminal point 1...16



Terminal point	Name	Comment
1	Encoder supply +	Supply voltage output for the encoder. The output voltage is adjustable.
2	ENC A+	Input for the encoder signal A+
3	ENC B+	Input for the encoder signal B+
4	ENC C+	Input for the encoder signal C+
5	Hall sensor U	Input for Hall sensor U
6	Hall sensor W	Input for Hall sensor W
7	Hall sensor supply +	Supply voltage output for the Hall sensors. The output voltage is adjustable.
8	Input 1	Digital input 24 V
9	Encoder supply GND	GND for the encoder
10	ENC A-	Input for the encoder signal A-
11	ENC B-	Input for the encoder signal B-
12	ENC C-	Input for the encoder signal C-
13	Hall sensor V	Input for Hall sensor V
14	-	-
15	Hall sensor supply GND	GND for the Hall sensors
16	Input 2	Digital input 24 V

Terminal point 1'... 16'



NOTE

The load voltage is not short-circuit proof

Cable fire and defect possible.

- Use an overcurrent protection device for the load voltage.
- Dimension the overcurrent protection device so that the maximum current is limited to three times the value of the rated current for a maximum of 1 second.

Terminal point	Name	Comment
1‘	Fan supply +	24 V_{DC} supply voltage output for the ZB8610 fan cartridge (optional accessory).
2‘	Fan diagnostics	Input for the "Diag" signal of the ZB8610 fan cartridge.
3‘	-	-
4‘	Motor phase U	Motor phase U
5‘	Motor phase W	Motor phase W
6‘	Motor brake +	Digital output for the motor brake
7‘	Load voltage 48 V_{DC}	Inputs for the load voltage (8... 48 V)
8‘	Load voltage 48 V_{DC}	
9‘	Fan supply GND	GND for the ZB8610 fan cartridge.
10‘	Fan mode	Output for the "Mode" signal of the ZB8610 fan cartridge.
11‘	-	-
12‘	Motor phase V	Motor phase V
13‘	HW Enable	Input to enable the output stage
14‘	Motor brake GND	GND for the motor brake
15‘	Load voltage GND	GND for the load voltage
16‘	Load voltage GND	GND for the load voltage

4.11 UL notice - Compact Motion

⚠ CAUTION

Application

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

⚠ CAUTION

Examination

For cULus examination, the Beckhoff I/O System has only been investigated for risk of fire and electrical shock (in accordance with UL508 and CSA C22.2 No. 142).

⚠ CAUTION

For devices with Ethernet connectors

Not for connection to telecommunication circuits.

⚠ CAUTION

Notes on motion devices

- *Motor overtemperature*
  Motor overtemperature sensing is not provided by the drive.
- *Application for compact motion devices*
  The modules are intended for use only within Beckhoff's Programmable Controller system Listed in File E172151.
- *Galvanic isolation from the supply*
  The modules are intended for operation within circuits not connected directly to the supply mains (galvanically isolated from the supply, i.e. on transformer secondary).
- *Requirement for environmental conditions*
  For use in Pollution Degree 2 Environment only.

Basic principles

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

4.12 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 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 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.

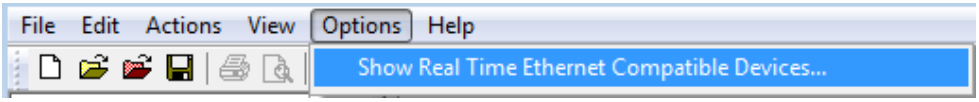


Fig. 29: System Manager “Options” (TwinCAT 2)

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

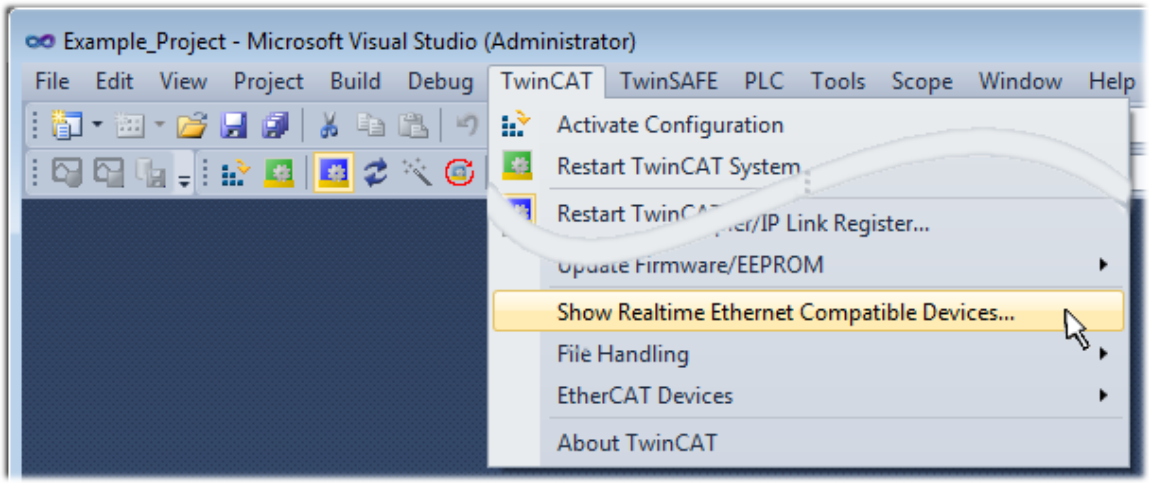


Fig. 30: Call up under VS Shell (TwinCAT 3)

B: Via TcRteInstall.exe in the TwinCAT directory

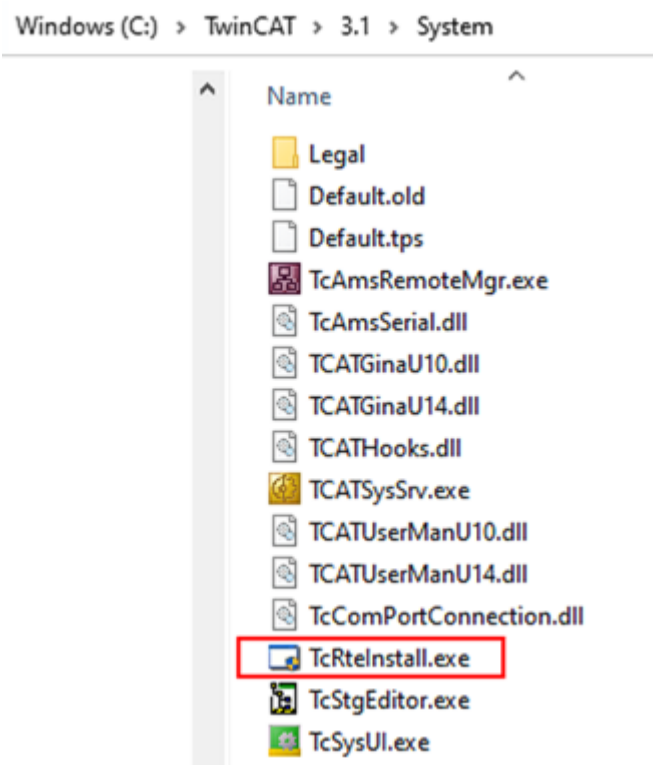


Fig. 31: TcRteInstall in the TwinCAT directory

In both cases, the following dialog appears:

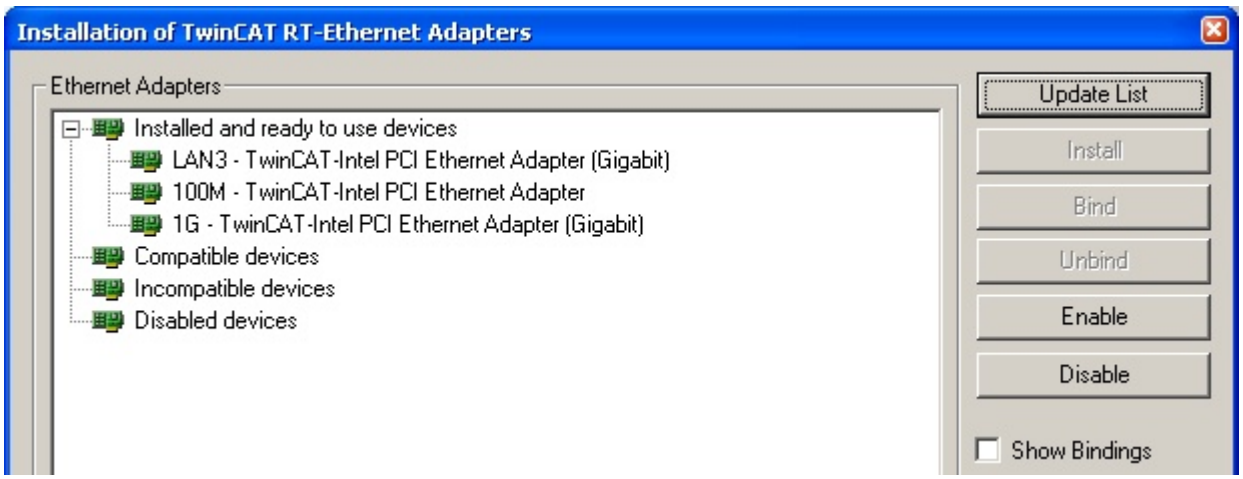


Fig. 32: Overview of network interfaces

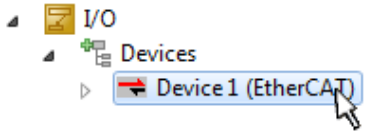
Interfaces listed under “Compatible devices” can be assigned a driver via the “Install” button. A driver should only be installed on compatible devices.

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

Fig. 33: EtherCAT device properties (TwinCAT 2): click on “Compatible Devices…” of tab “Adapter”

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



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

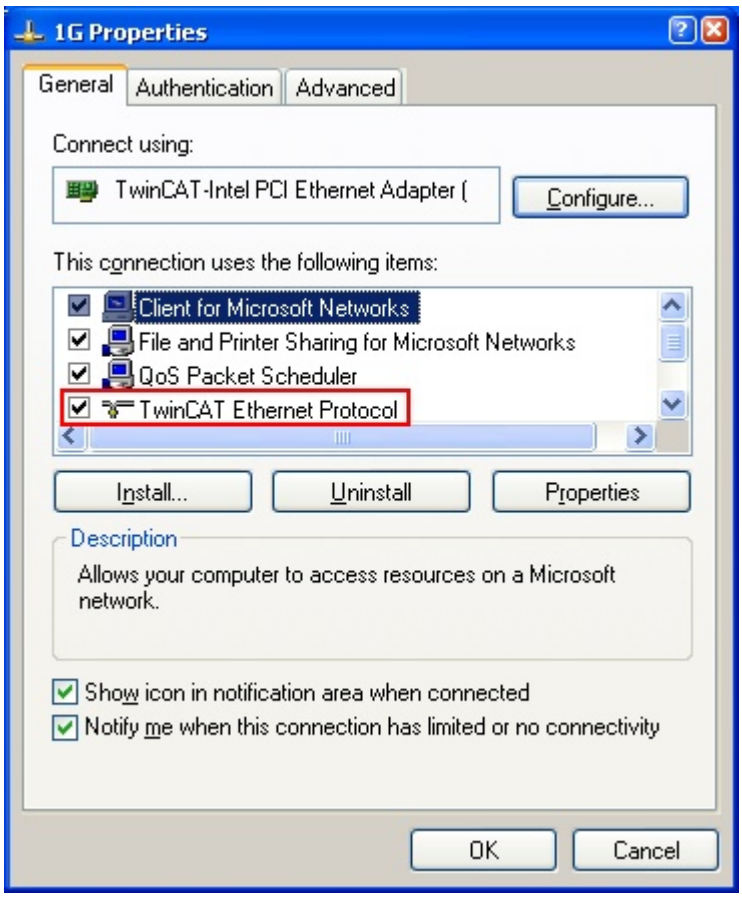


Fig. 34: Windows properties of the network interface

A correct setting of the driver could be:

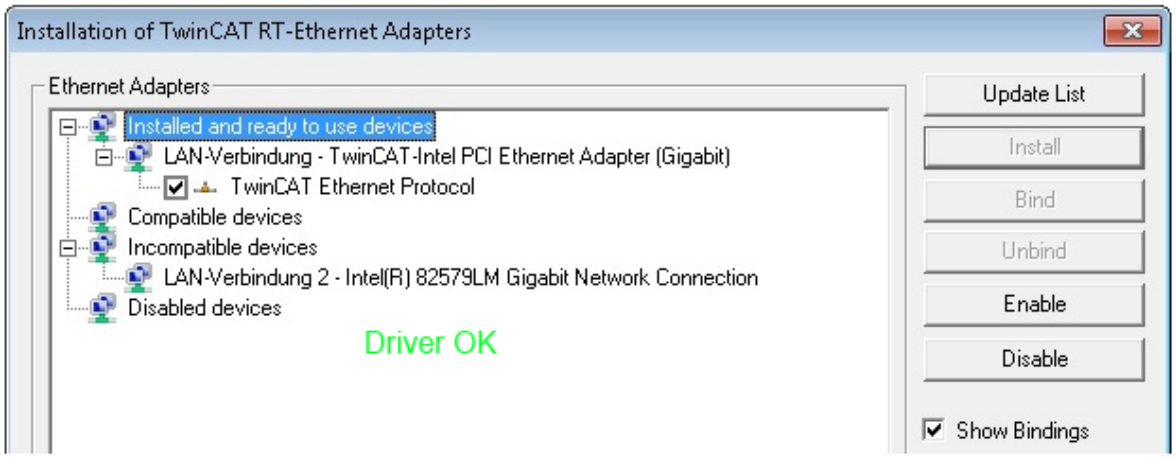


Fig. 35: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

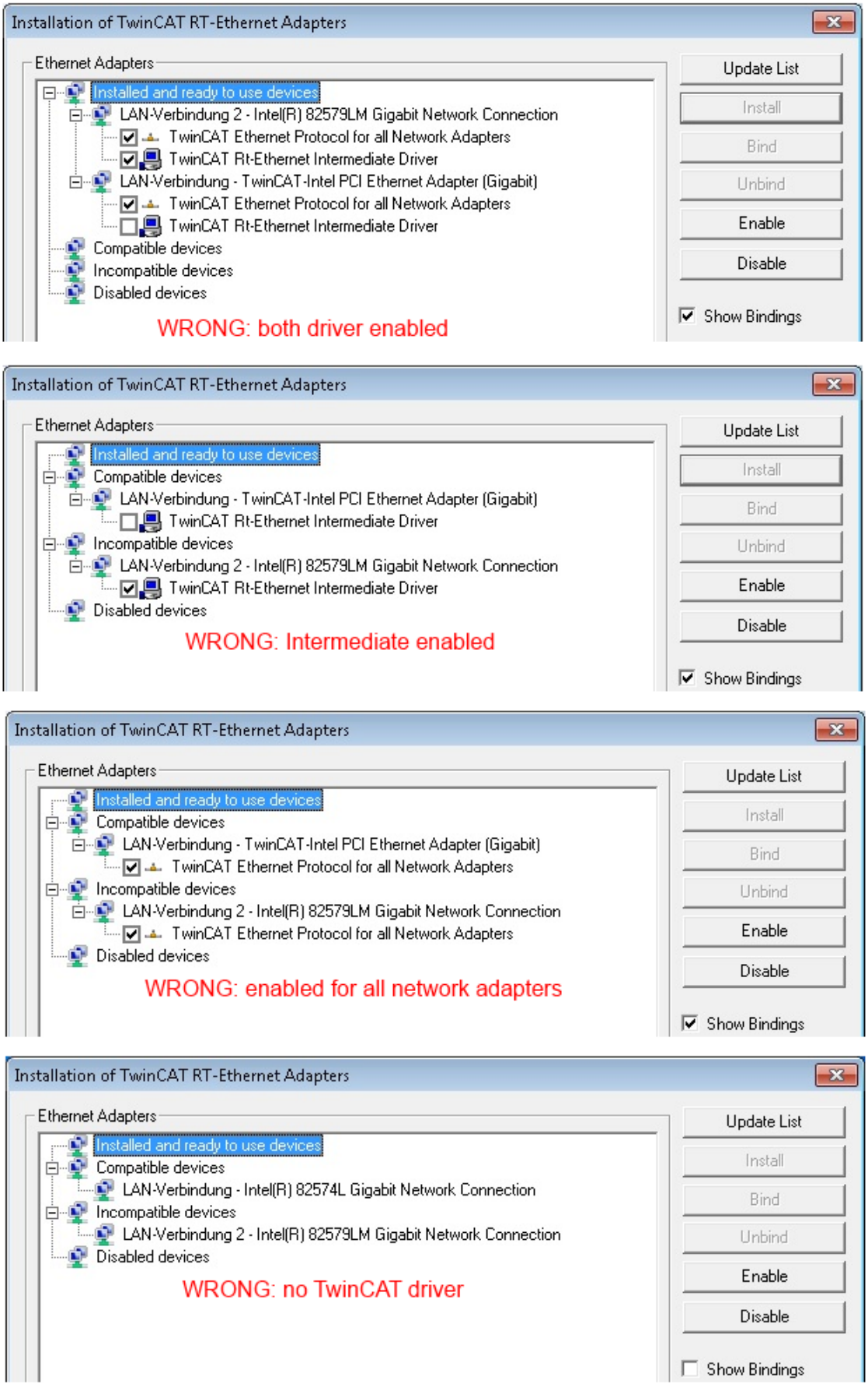


Fig. 36: Incorrect driver settings for the Ethernet port

IP address of the port used

IP address/DHCP

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

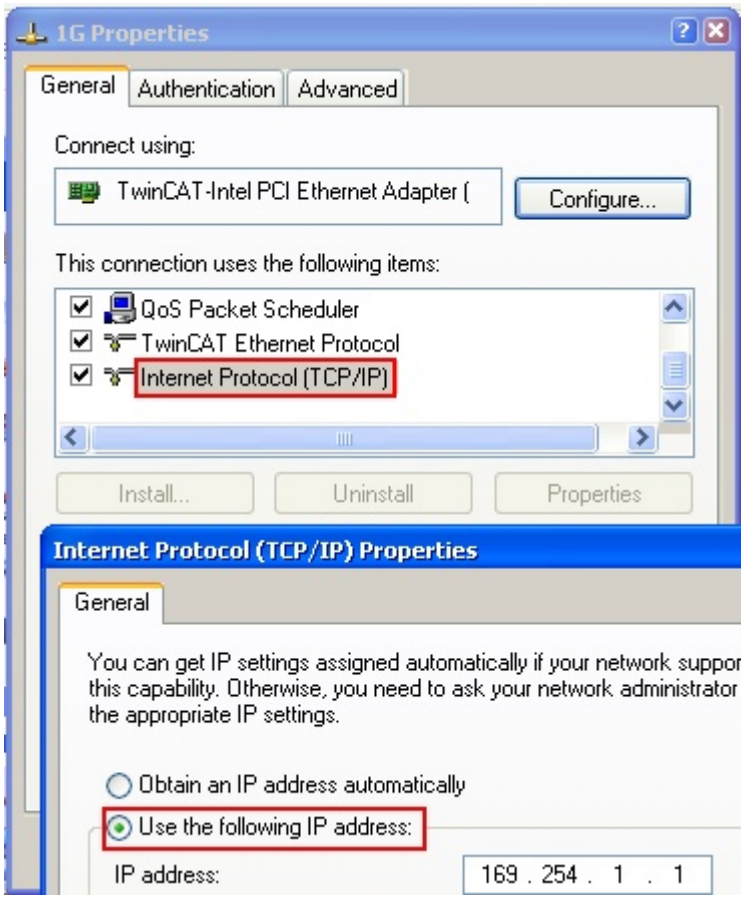


Fig. 37: TCP/IP setting for the Ethernet port

5.2 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 [▶ 64] 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”

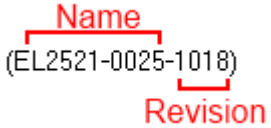


Fig. 38: Identifier structure

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

Online description

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

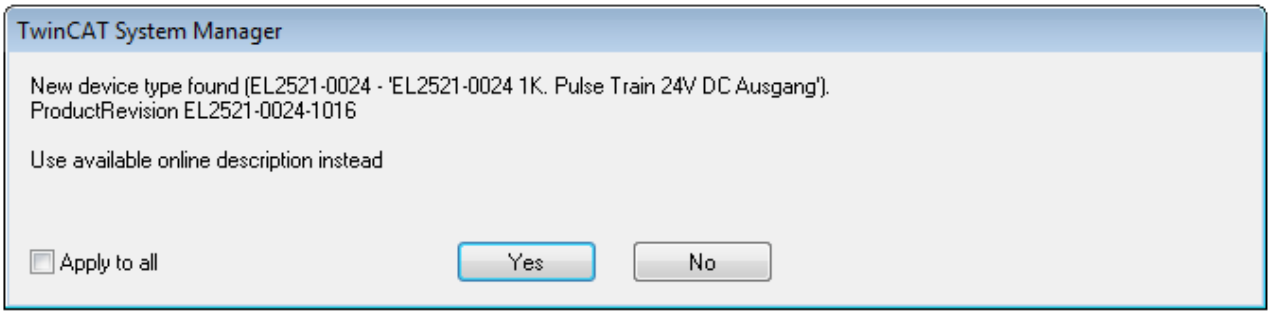


Fig. 39: OnlineDescription information window (TwinCAT 2)

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

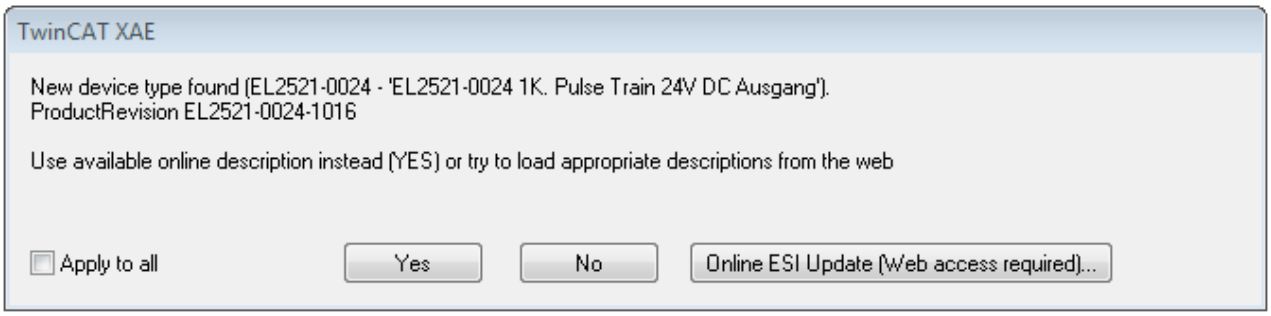


Fig. 40: Information window OnlineDescription (TwinCAT 3)

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 [▶ 65]".

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

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

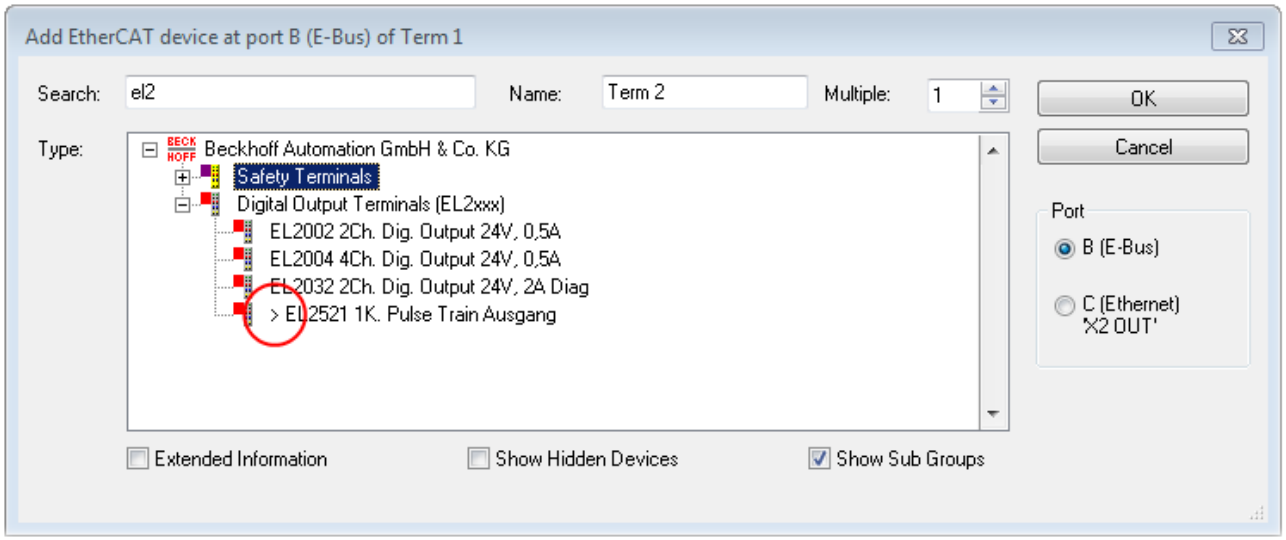


Fig. 42: Indication of an online recorded ESI of EL2521 as an example

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

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

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

OnlineDescription for TwinCAT 3.x

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

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

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

Faulty ESI file

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

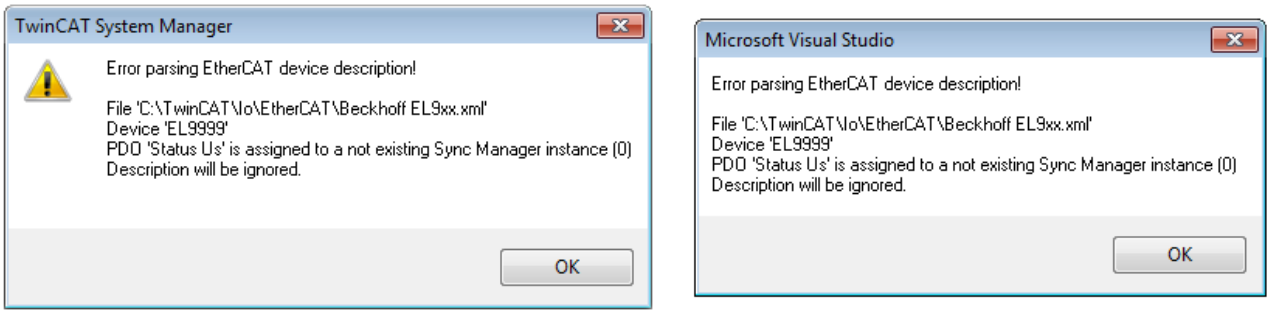


Fig. 43: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

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

5.3 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:

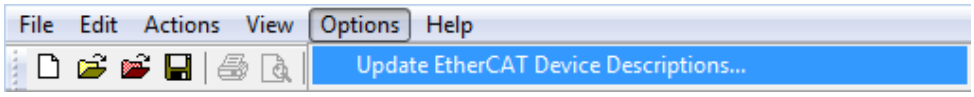


Fig. 44: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:

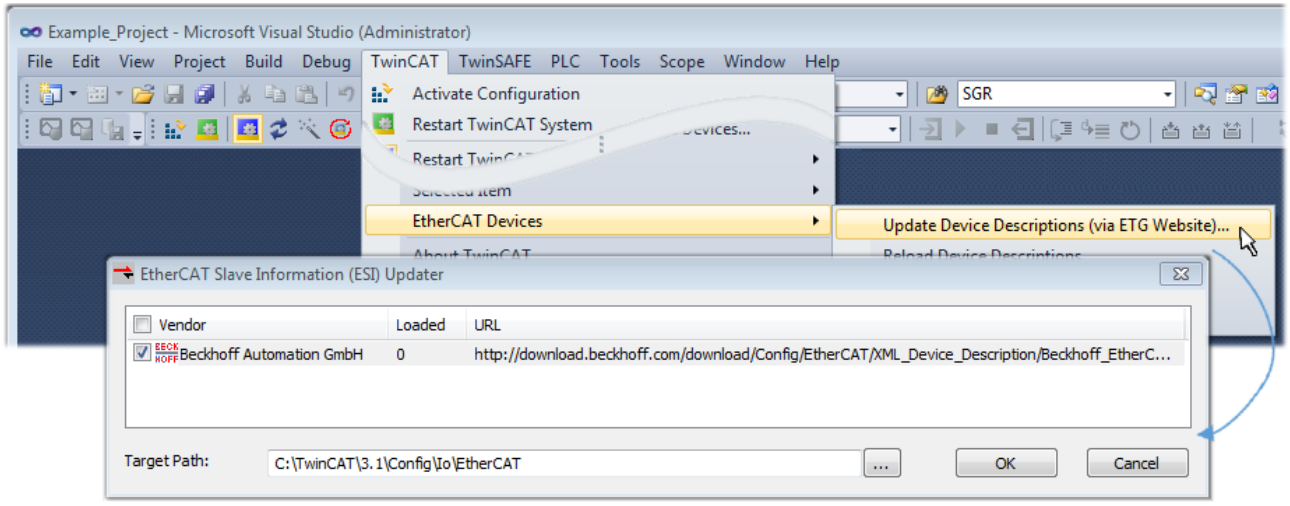


Fig. 45: Using the ESI Updater (TwinCAT 3)

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)…”.

5.4 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” [▶ 60].

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 [▶ 70] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 71]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 74]

The scan with existing configuration [▶ 75] can also be carried out for comparison.

5.5 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

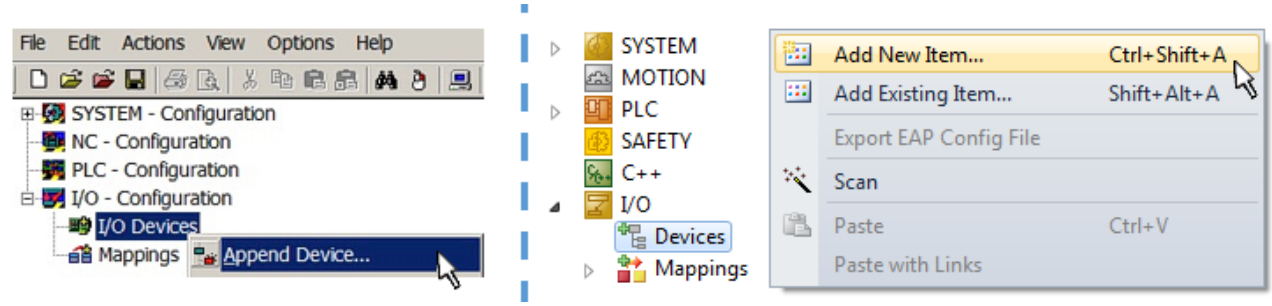


Fig. 46: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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”.

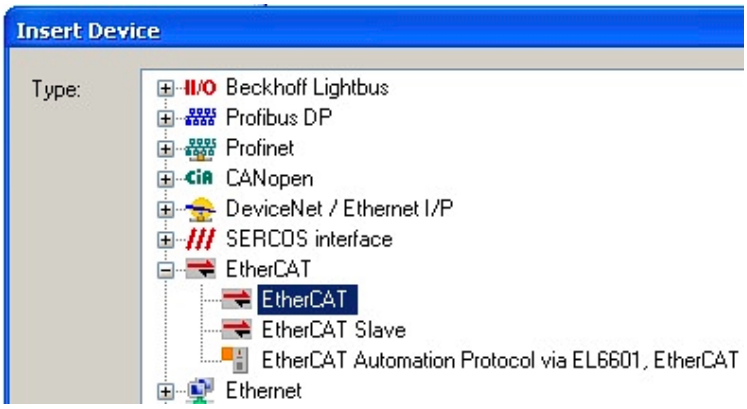


Fig. 47: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

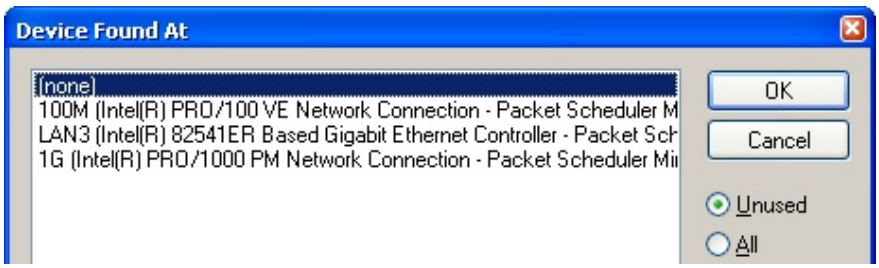


Fig. 48: Selecting the Ethernet port

This query may appear automatically when the EtherCAT device is created, or the assignment can be set/ modified later in the properties dialog; see Fig. “EtherCAT device properties (TwinCAT 2)”.

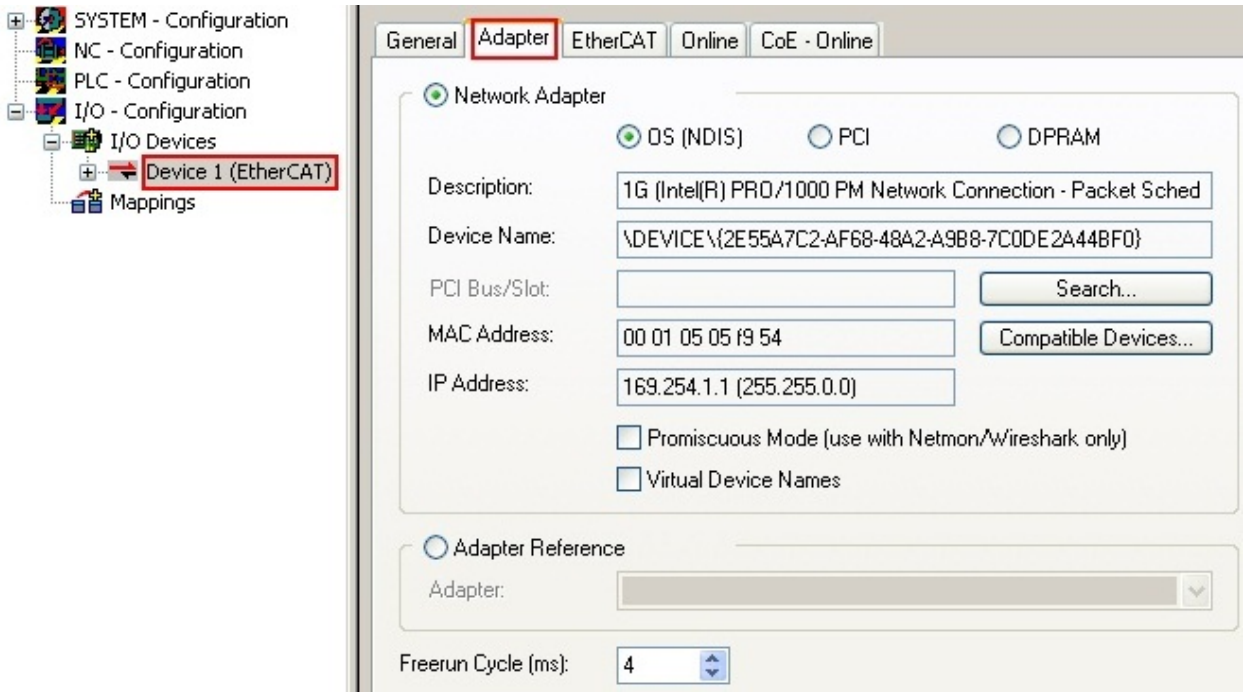
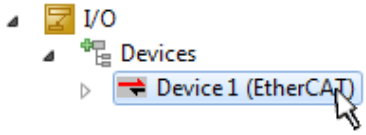


Fig. 49: EtherCAT device properties (TwinCAT 2)

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 [▶ 54].

Defining EtherCAT slaves

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

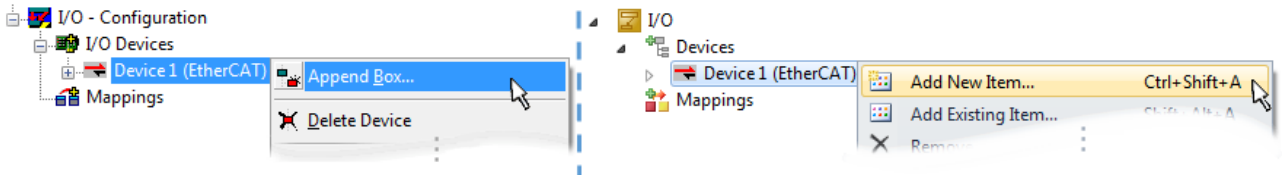


Fig. 50: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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

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

Overview of physical layer

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

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

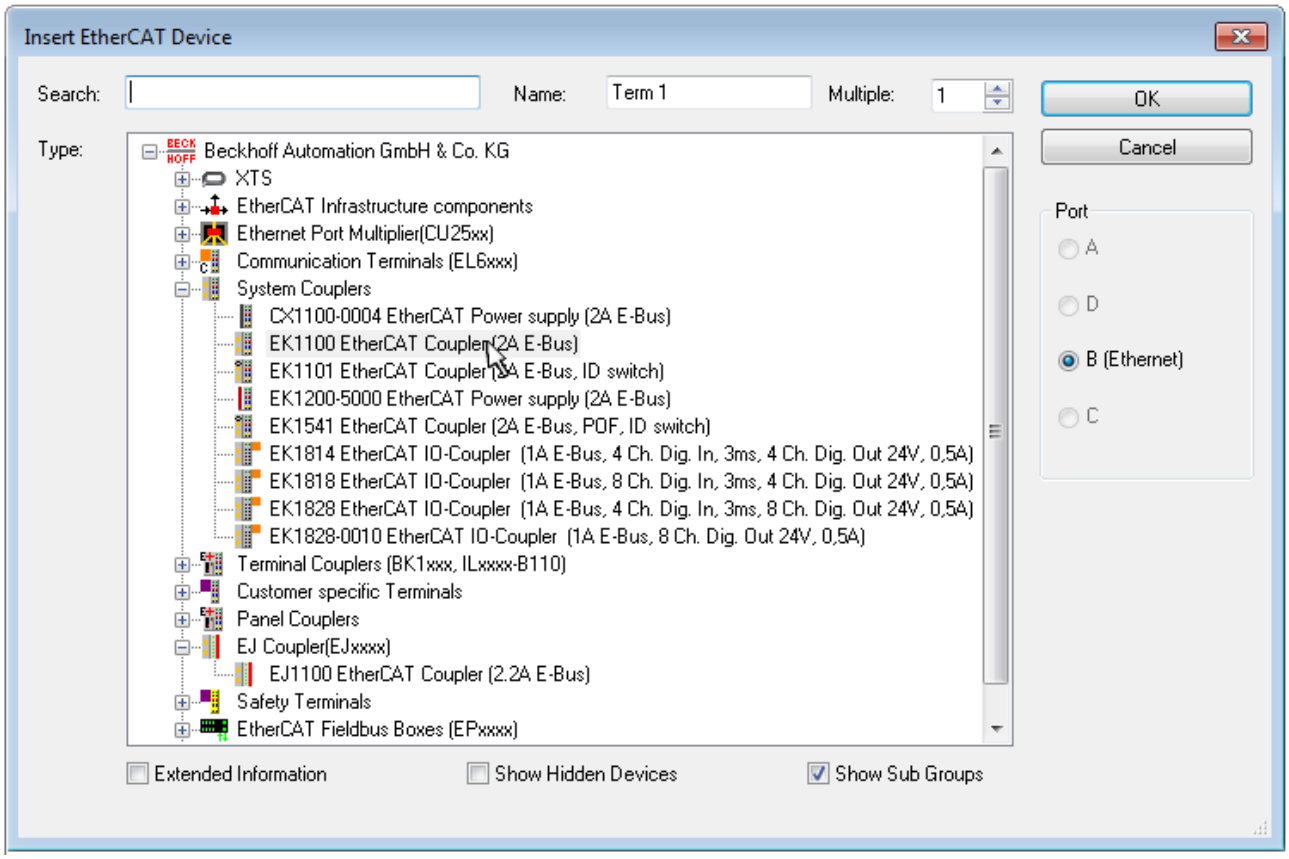


Fig. 51: Selection dialog for new EtherCAT device

By default, only the name/device type is used as selection criterion. For selecting a specific revision of the device, the revision can be displayed as “Extended Information”.

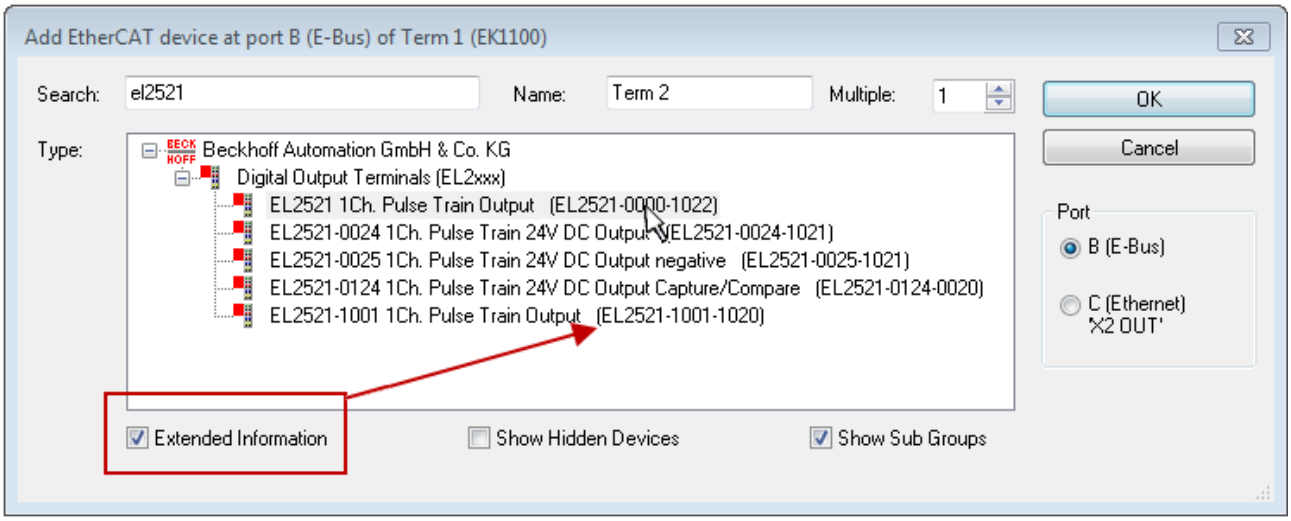


Fig. 52: Display of device revision

In many cases several device revisions were created for historic or functional reasons, e.g. through technological advancement. For simplification purposes (see Fig. “Selection dialog for new EtherCAT device”) only the last (i.e. highest) revision and therefore the latest state of production is displayed in the selection dialog for Beckhoff devices. To show all device revisions available in the system as ESI descriptions tick the “Show Hidden Devices” check box, see Fig. “Display of previous revisions”.

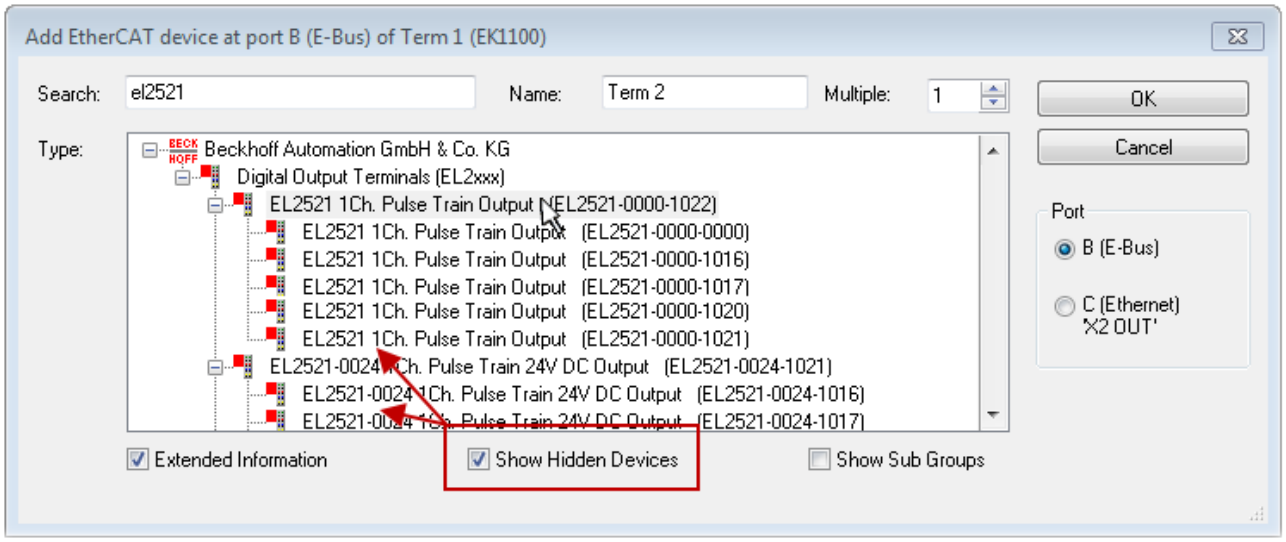


Fig. 53: Display of previous revisions

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.

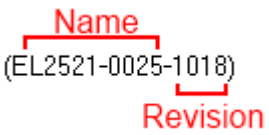


Fig. 54: Name/revision of the terminal

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, ...

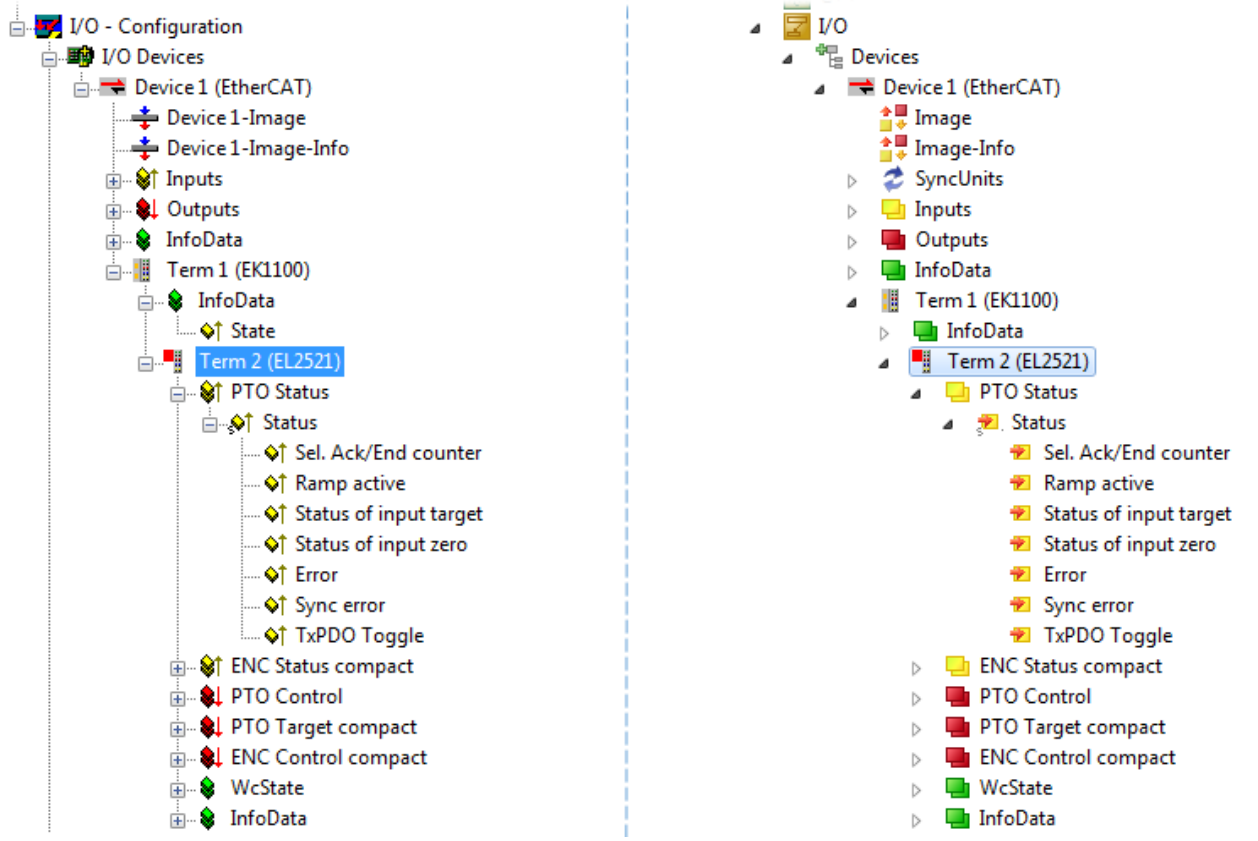


Fig. 55: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)

5.6 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.

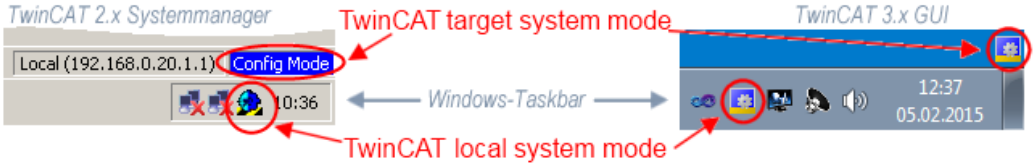


Fig. 56: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

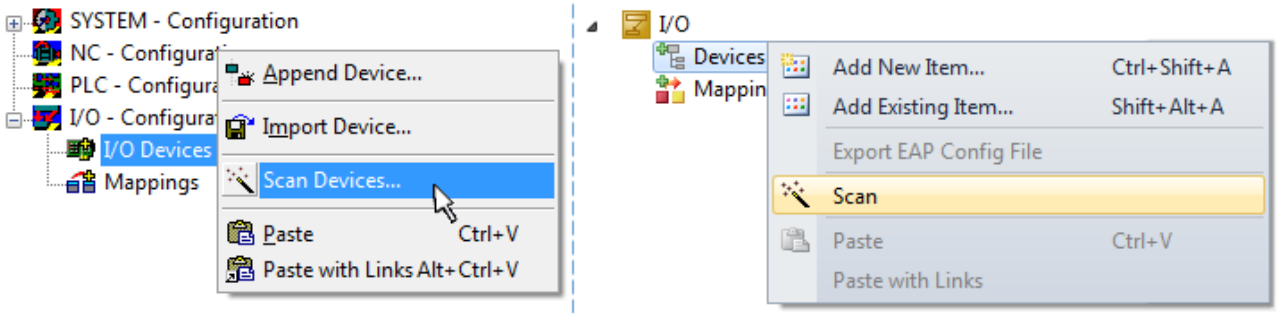


Fig. 57: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

This scan mode attempts to find not only EtherCAT devices (or Ethernet ports that are usable as such), but also NOVRAM, fieldbus cards, SMB etc. However, not all devices can be found automatically.

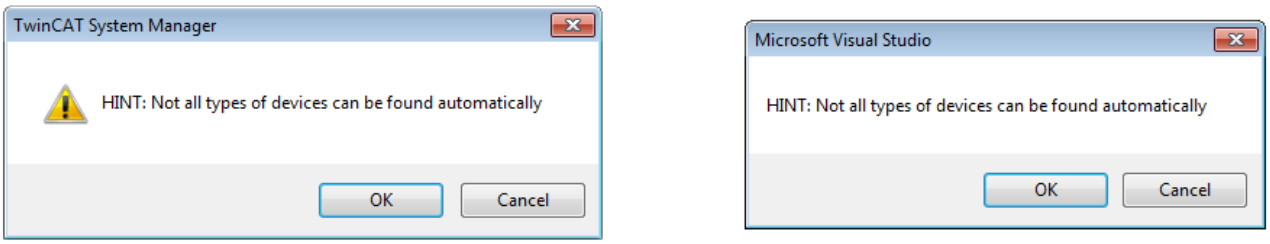


Fig. 58: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

Ethernet ports with installed TwinCAT real-time driver are shown as “RT Ethernet” devices. An EtherCAT frame is sent to these ports for testing purposes. If the scan agent detects from the response that an EtherCAT slave is connected, the port is immediately shown as an “EtherCAT Device” .

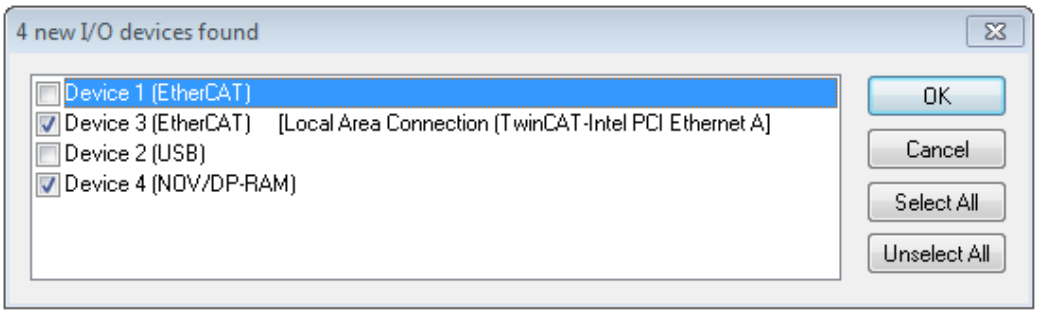


Fig. 59: Detected Ethernet devices

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 [▶ 54].

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.

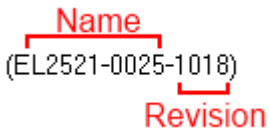


Fig. 60: Example default state

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 [▶ 75] with the defined initial configuration.Background: since Beckhoff occasionally increases the revision version of the delivered products for product maintenance reasons, a configuration can be created by such a scan which (with an identical machine construction) is identical according to the device list; however, the respective device revision may differ from the initial configuration.

Example:

Company A builds the prototype of a machine B, which is to be produced in series later on. To do this the prototype is built, a scan of the IO devices is performed in TwinCAT and the initial configuration “B.tsm” is created. The EL2521-0025 EtherCAT terminal with the revision 1018 is located somewhere. It is thus built into the TwinCAT configuration in this way:

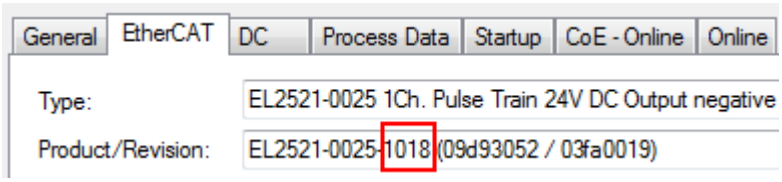


Fig. 61: Installing EthetCAT terminal with revision -1018

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 [▶ 75] against the initial configuration “B.tsm” in order to check the built machine.

However, if the series machine production department now doesn’t use “B.tsm”, but instead carries out a scan to create the productive configuration, the revision **-1019** is automatically detected and built into the configuration:

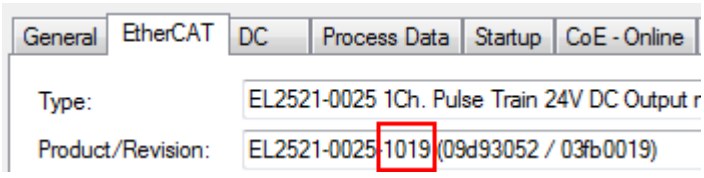


Fig. 62: Detection of EtherCAT terminal with revision -1019

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

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

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

Fig. 63: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

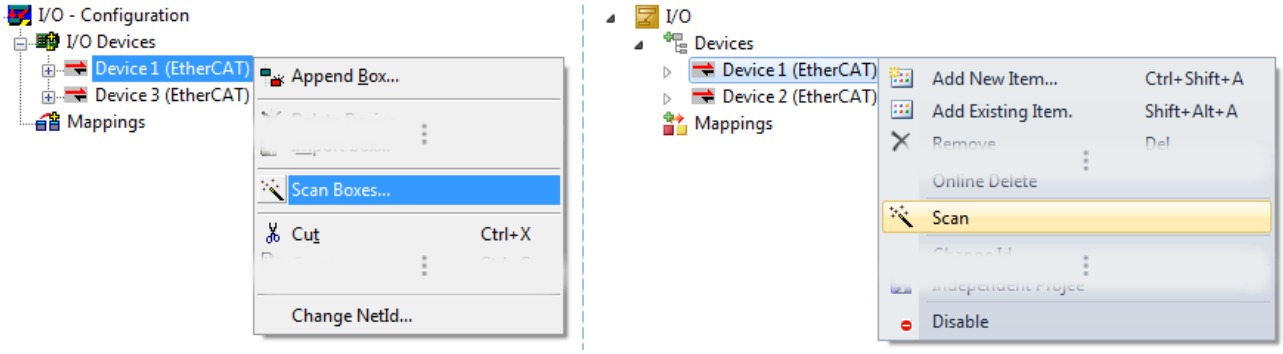


Fig. 64: Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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.

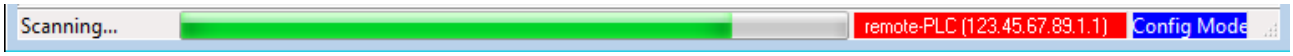


Fig. 65: *Scan progressexemplary by TwinCAT 2*

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

Fig. 66: 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).

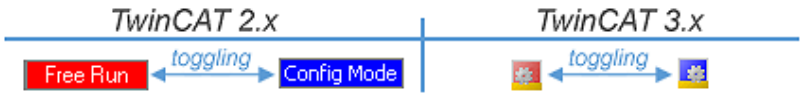


Fig. 67: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar

Fig. 68: TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3)

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

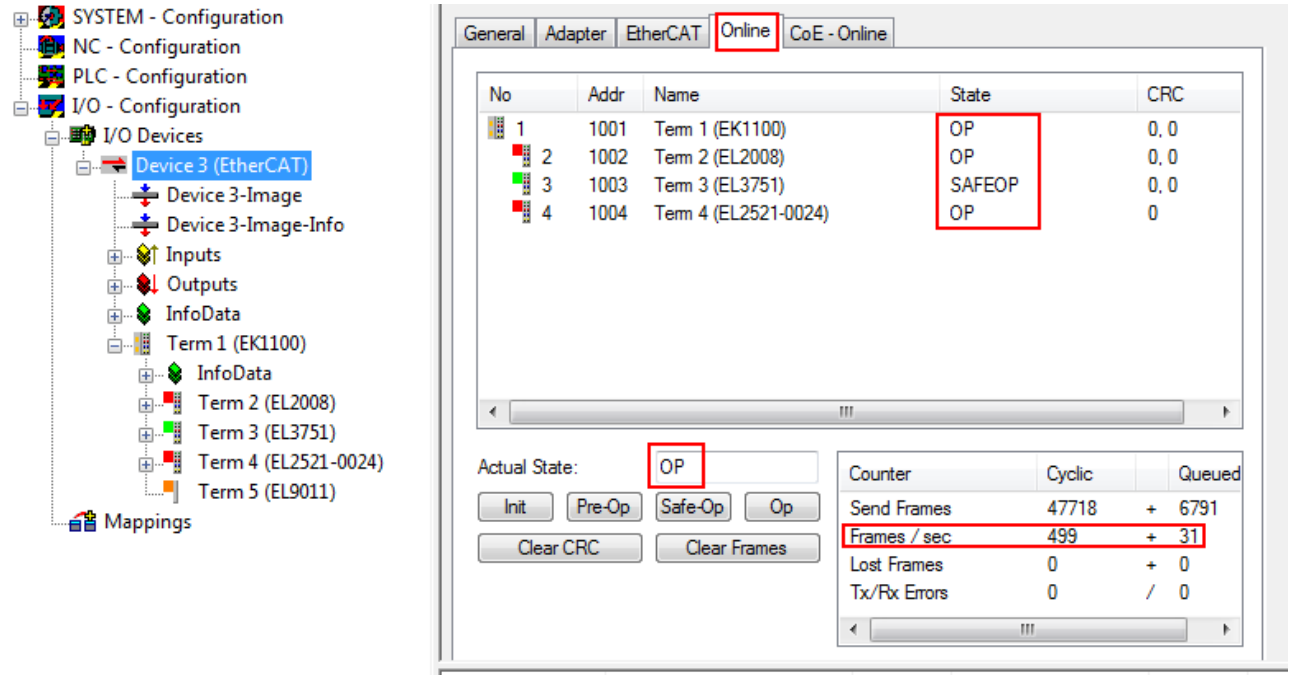


Fig. 69: Online display example

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 [▶ 65].

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.

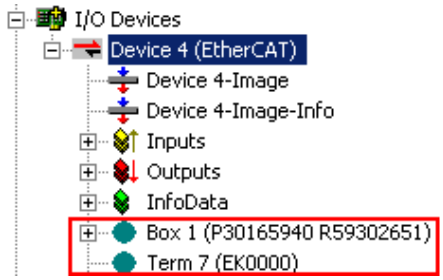


Fig. 70: Faulty identification

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

Scan over existing Configuration

NOTE

Change of the configuration after comparison

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

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

Fig. 71: Identical configuration (left: TwinCAT 2; right: TwinCAT 3)

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

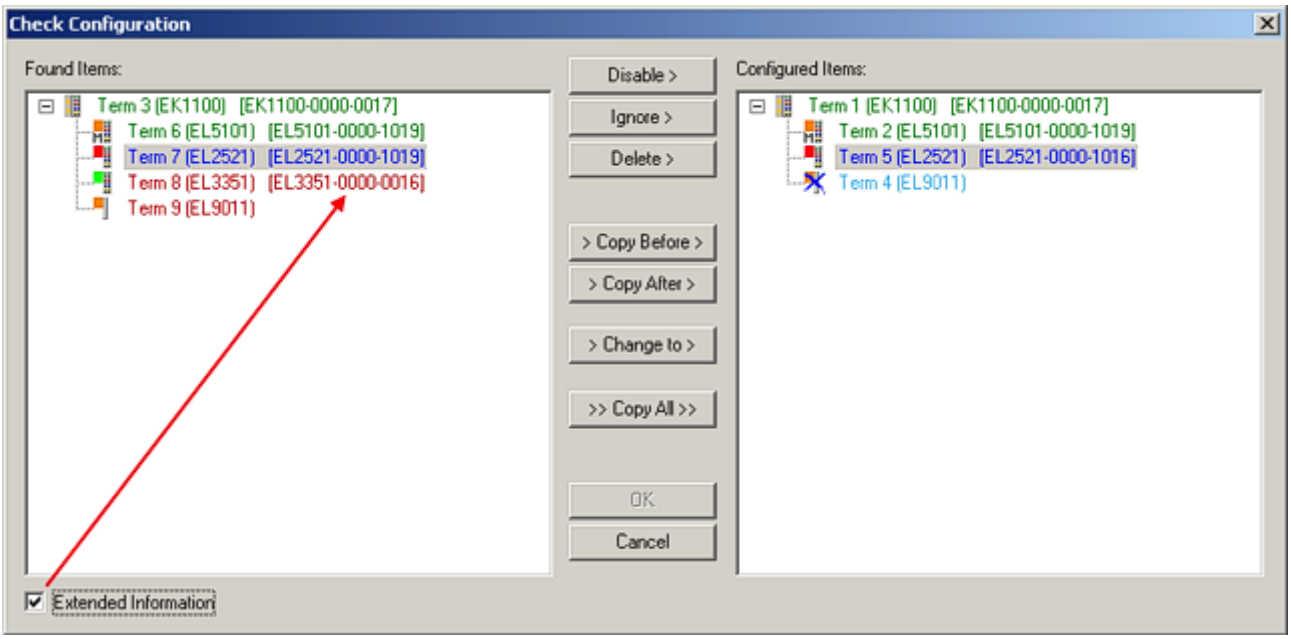


Fig. 72: Correction dialog

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.

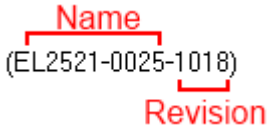


Fig. 73: Name/revision of the terminal

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, ...

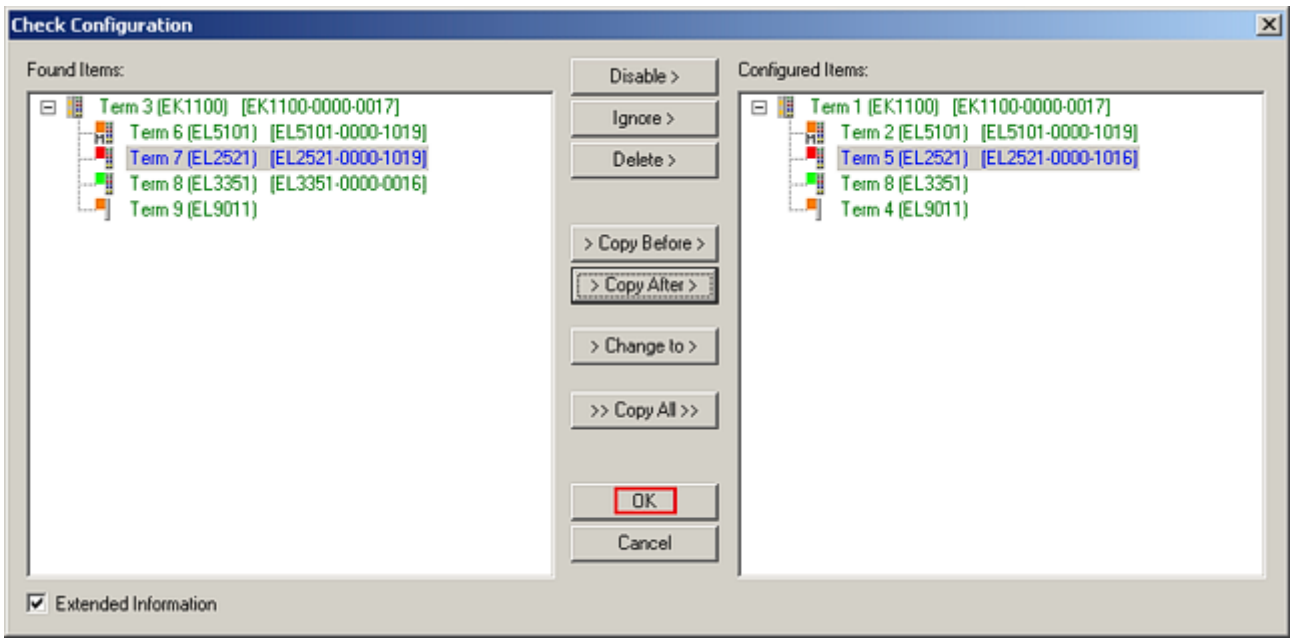


Fig. 74: Correction dialog with modifications

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.

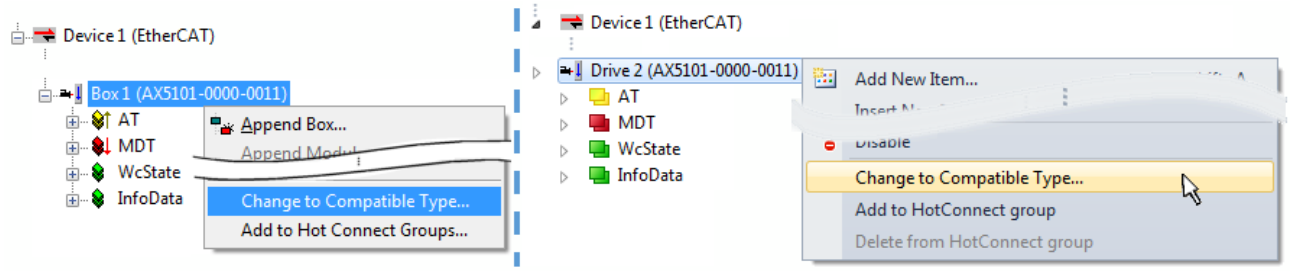


Fig. 75: Dialog “Change to Compatible Type…” (left: TwinCAT 2; right: TwinCAT 3)

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

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

- FMMU (additional ones are allowed)

- SyncManager (SM, additional ones are allowed)

- EoE (attributes MAC, IP)

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

- FoE

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

This function is preferably to be used on AX5000 devices.

Change to Alternative Type

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

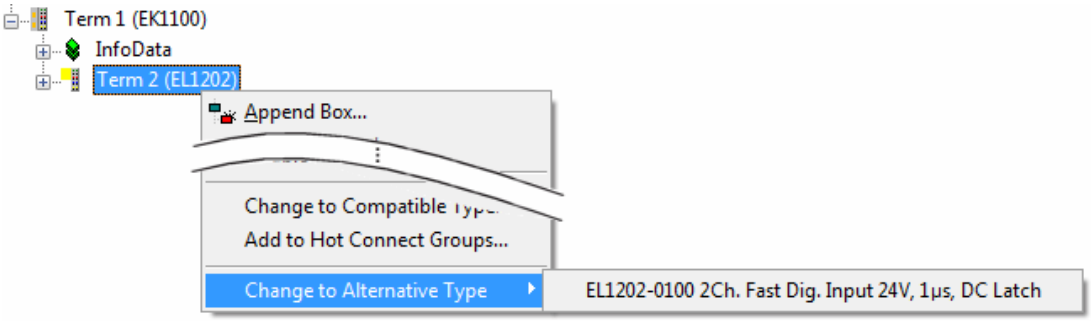


Fig. 76: TwinCAT 2 Dialog Change to Alternative Type

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).

5.7 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).

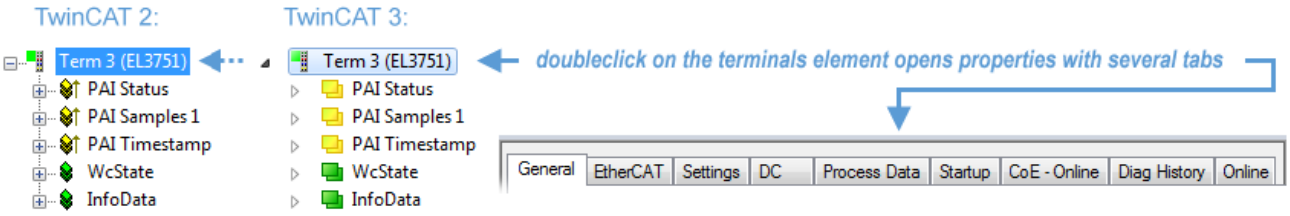


Fig. 77: Branch element as terminal EL3751

In the right-hand window of the TwinCAT System Manager (TwinCAT 2) or the Development Environment (TwinCAT 3), various tabs are now available for configuring the terminal. And yet the dimension of complexity of a subscriber determines which tabs are provided. Thus as illustrated in the example above the terminal EL3751 provides many setup options and also a respective number of tabs are available. On the contrary by the terminal EL1004 for example the tabs “General”, “EtherCAT”, “Process Data” and “Online“ are available only. Several terminals, as for instance the EL6695 provide special functions by a tab with its own terminal name, so “EL6695” in this case. A specific tab “Settings” by terminals with a wide range of setup options will be provided also (e.g. EL3751).

“General” tab

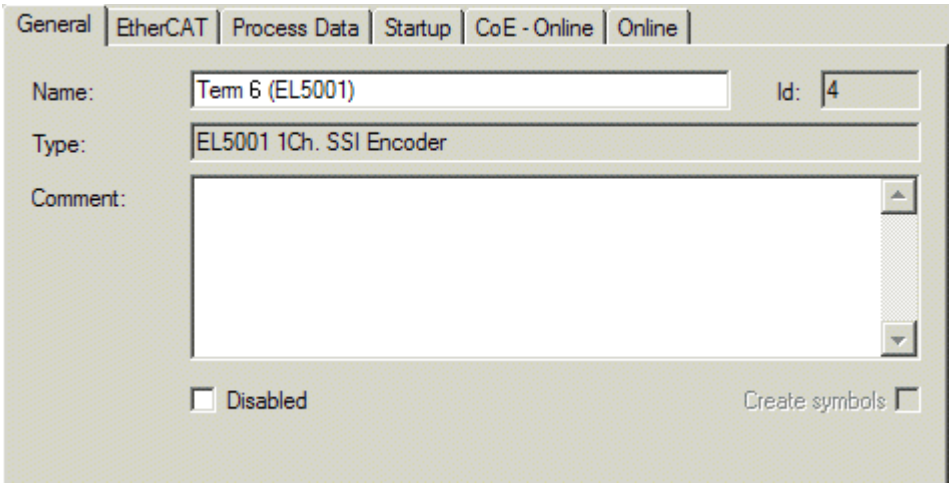


Fig. 78: “General” tab

Name	Name of the EtherCAT device
Id	Number of the EtherCAT device
Type	EtherCAT device type
Comment	Here you can add a comment (e.g. regarding the system).
Disabled	Here you can deactivate the EtherCAT device.
Create symbols	Access to this EtherCAT slave via ADS is only available if this control box is activated.

"EtherCAT" tab

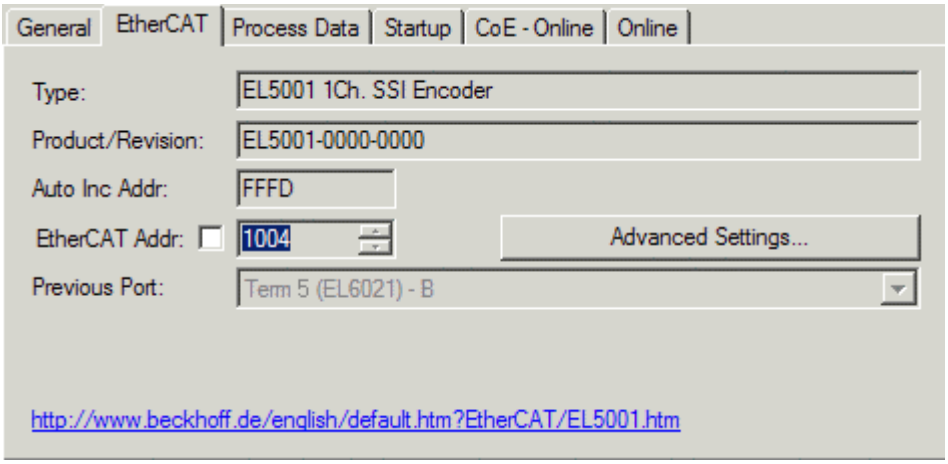


Fig. 79: "EtherCAT" tab

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

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

"Process Data" tab

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

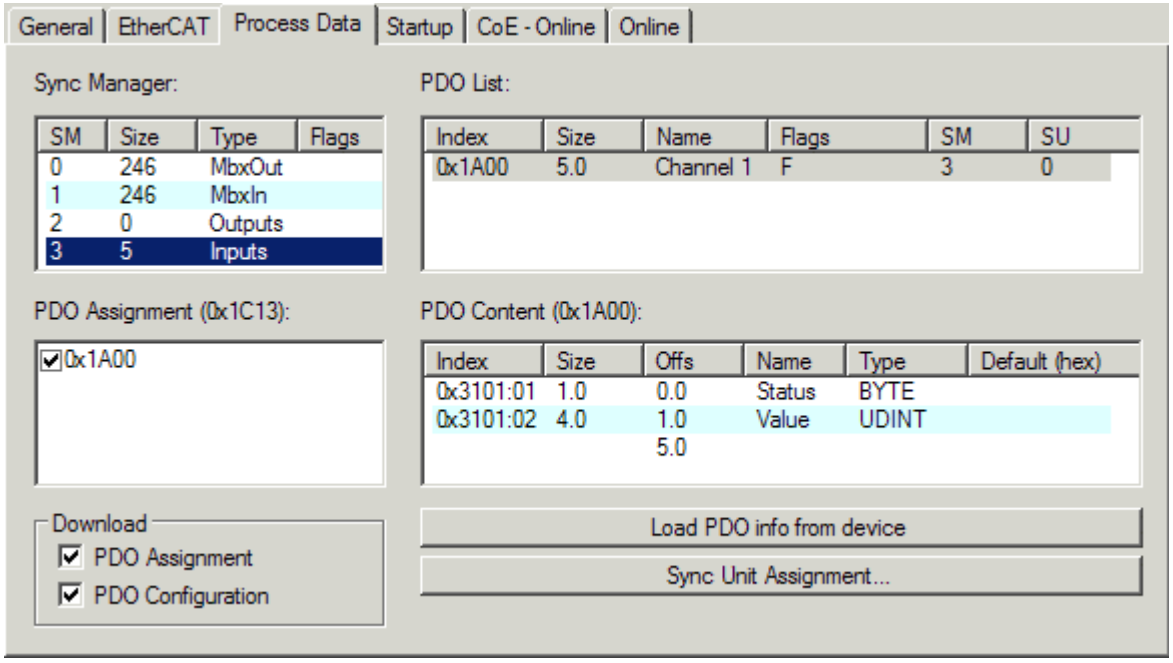


Fig. 80: “Process Data” tab

The process data (PDOs) transferred by an EtherCAT slave during each cycle are user data which the application expects to be updated cyclically or which are sent to the slave. To this end the EtherCAT master (Beckhoff TwinCAT) parameterizes each EtherCAT slave during the start-up phase to define which process data (size in bits/bytes, source location, transmission type) it wants to transfer to or from this slave. Incorrect configuration can prevent successful start-up of the slave.

For Beckhoff EtherCAT EL, ES, EM, EJ and EP slaves the following applies in general:

- The input/output process data supported by the device are defined by the manufacturer in the ESI/XML description. The TwinCAT EtherCAT Master uses the ESI description to configure the slave correctly.
- The process data can be modified in the System Manager. See the device documentation.
  Examples of modifications include: mask out a channel, displaying additional cyclic information, 16-bit display instead of 8-bit data size, etc.
- In so-called “intelligent” EtherCAT devices the process data information is also stored in the CoE directory. Any changes in the CoE directory that lead to different PDO settings prevent successful startup of the slave. It is not advisable to deviate from the designated process data, because the device firmware (if available) is adapted to these PDO combinations.

If the device documentation allows modification of process data, proceed as follows (see Figure *Configuring the process data*).

- A: select the device to configure
- B: in the “Process Data” tab select Input or Output under SyncManager (C)
- D: the PDOs can be selected or deselected
- H: the new process data are visible as linkable variables in the System Manager
  The new process data are active once the configuration has been activated and TwinCAT has been restarted (or the EtherCAT master has been restarted)
- E: if a slave supports this, Input and Output PDO can be modified simultaneously by selecting a so-called PDO record (“predefined PDO settings”).

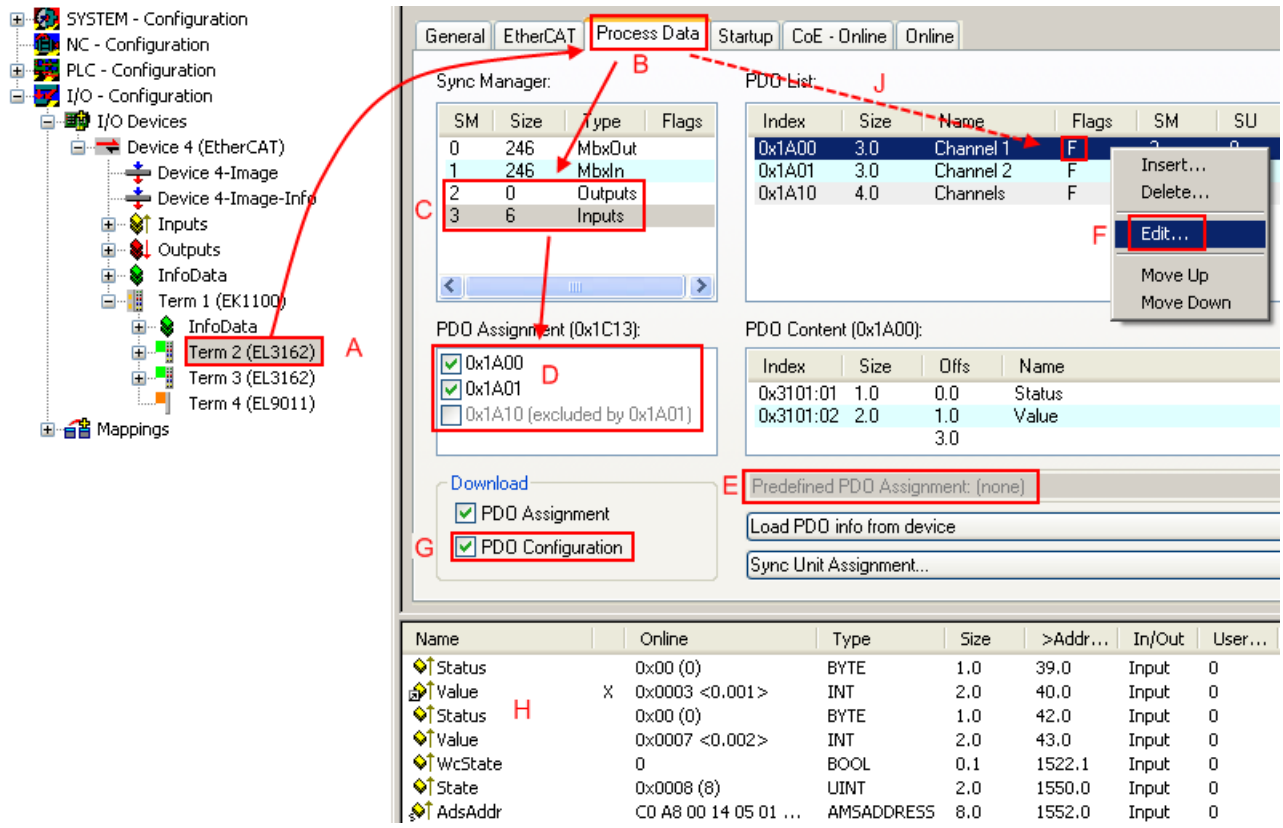


Fig. 81: Configuring the process data

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 [▶ 86] can be found at the end of this section.

“Startup” tab

The *Startup* tab is displayed if the EtherCAT slave has a mailbox and supports the *CANopen over EtherCAT* (CoE) or *Servo drive over EtherCAT* protocol. This tab indicates which download requests are sent to the mailbox during startup. It is also possible to add new mailbox requests to the list display. The download requests are sent to the slave in the same order as they are shown in the list.

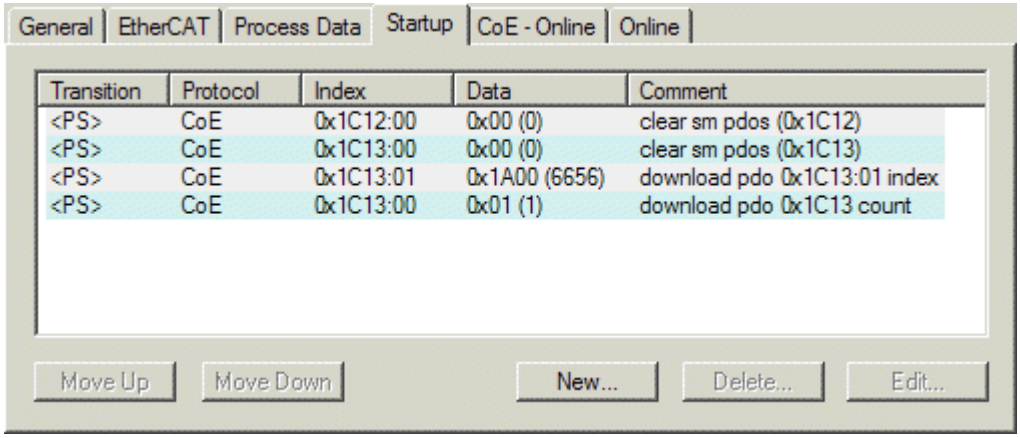


Fig. 82: “Startup” tab

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

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

“CoE - Online” tab

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

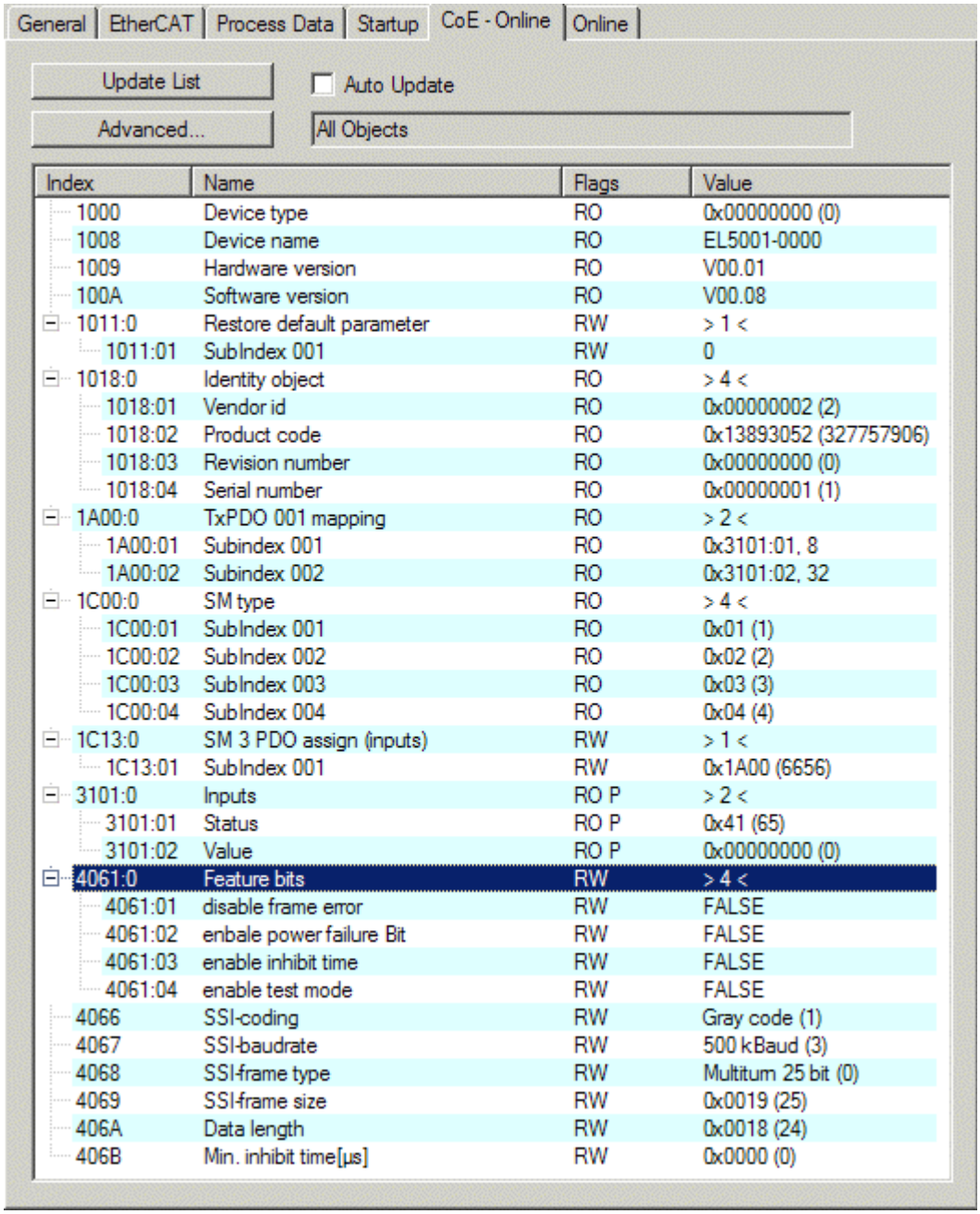


Fig. 83: “CoE - Online” tab

Object list display

Column	Description	
Index	Index and sub-index of the object	
Name	Name of the object	
Flags	RW	The object can be read, and data can be written to the object (read/write)
	RO	The object can be read, but no data can be written to the object (read only)
	P	An additional P identifies the object as a process data object.
Value	Value of the object	

Update List The *Update list* button updates all objects in the displayed list

Auto Update If this check box is selected, the content of the objects is updated automatically.

Advanced The *Advanced* button opens the *Advanced Settings* dialog. Here you can specify which objects are displayed in the list.

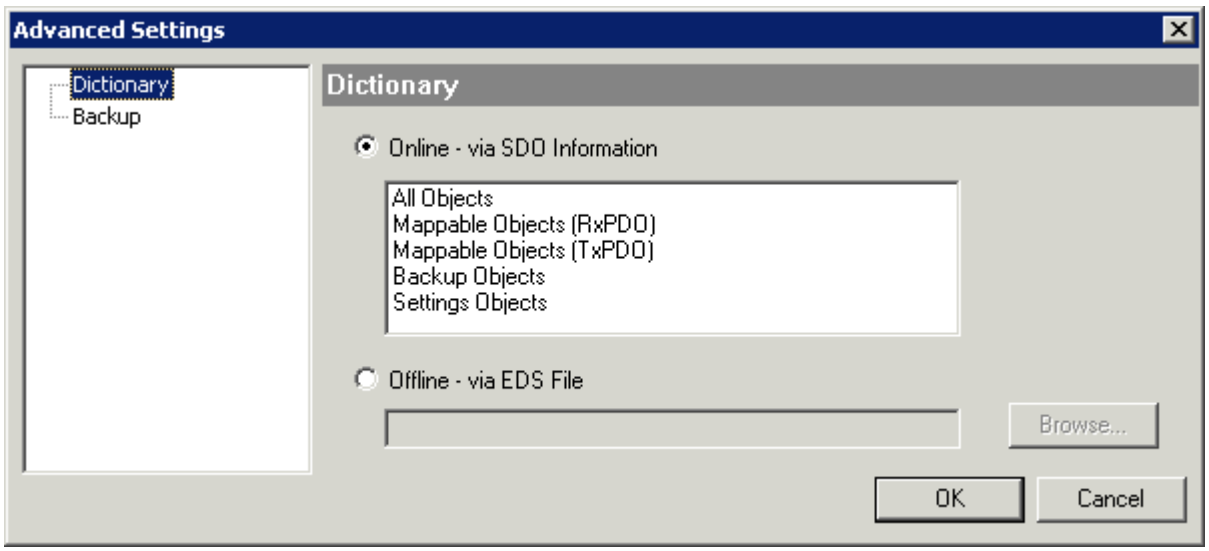


Fig. 84: Dialog “Advanced settings”

Online - via SDO Information If this option button is selected, the list of the objects included in the object list of the slave is uploaded from the slave via SDO information. The list below can be used to specify which object types are to be uploaded.

Offline - via EDS File If this option button is selected, the list of the objects included in the object list is read from an EDS file provided by the user.

“Online” tab

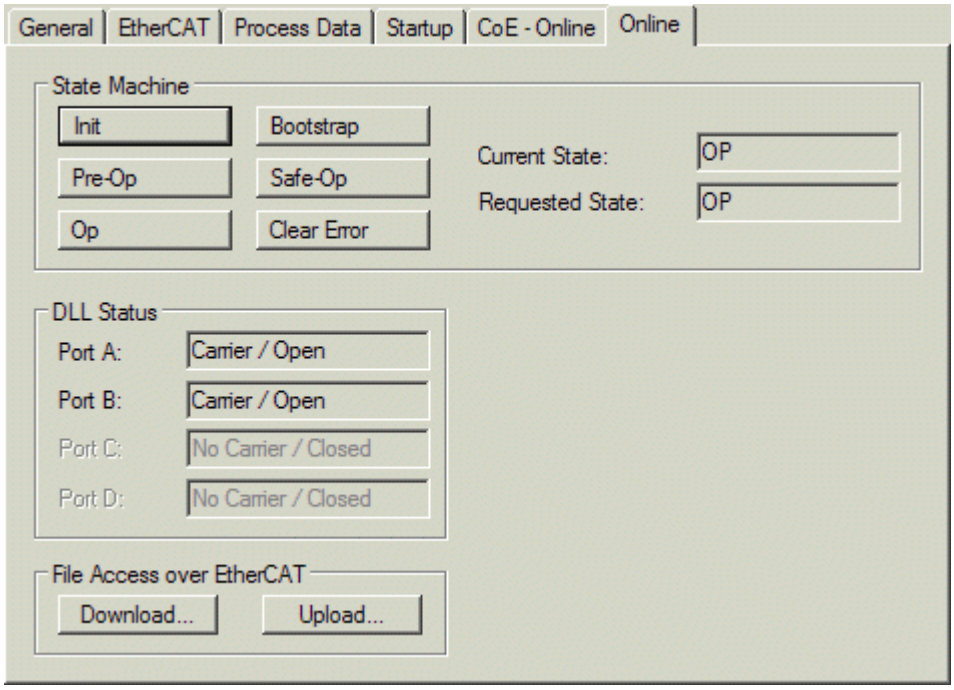


Fig. 85: “Online” tab

State Machine

Init	This button attempts to set the EtherCAT device to the *Init* state.
Pre-Op	This button attempts to set the EtherCAT device to the *pre-operational* state.
Op	This button attempts to set the EtherCAT device to the *operational* state.
Bootstrap	This button attempts to set the EtherCAT device to the *Bootstrap* state.
Safe-Op	This button attempts to set the EtherCAT device to the *safe-operational* state.
Clear Error	This button attempts to delete the fault display. If an EtherCAT slave fails during change of state it sets an error flag. Example: An EtherCAT slave is in PREOP state (pre-operational). The master now requests the SAFEOP state (safe-operational). If the slave fails during change of state it sets the error flag. The current state is now displayed as ERR PREOP. When the *Clear Error* button is pressed the error flag is cleared, and the current state is displayed as PREOP again.
Current State	Indicates the current state of the EtherCAT device.
Requested State	Indicates the state requested for the EtherCAT device.

DLL Status

Indicates the DLL status (data link layer status) of the individual ports of the EtherCAT slave. The DLL status can have four different states:

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

File Access over EtherCAT

Download	With this button a file can be written to the EtherCAT device.
Upload	With this button a file can be read from the EtherCAT device.

“DC” tab (Distributed Clocks)

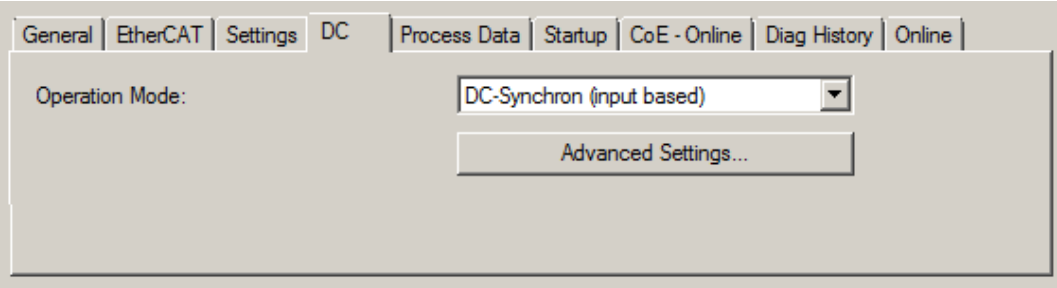


Fig. 86: “DC” tab (Distributed Clocks)

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

5.7.1 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 [▶ 84]),

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 [▶ 81] 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.

5.8 Import/Export of EtherCAT devices with SCI and XTI

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

5.8.1 Basic principles

An EtherCAT slave is basically parameterized through the following elements:

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

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

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

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

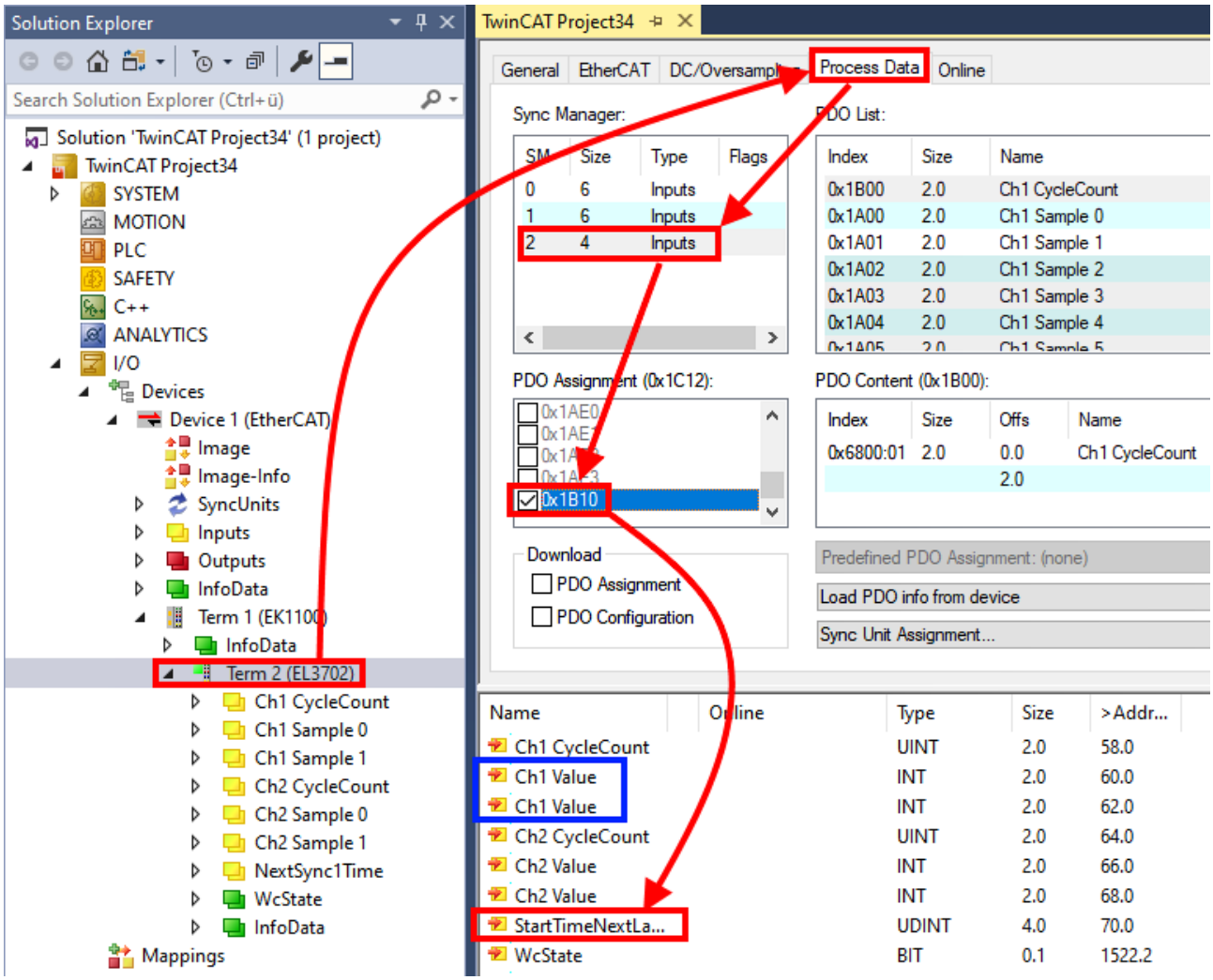
It may become necessary to reuse the modified slave in other projects in this way, without having to make equivalent configuration changes to the slave again. To accomplish this, proceed as follows:

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

TwinCAT offers two methods for this purpose:

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

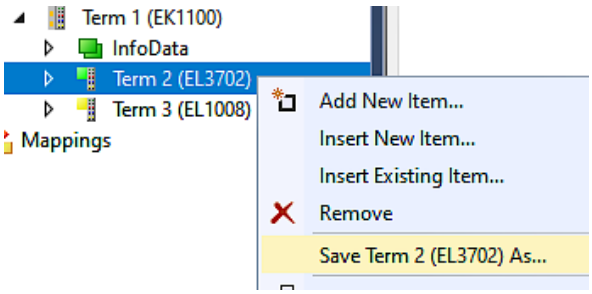
An example is provided below for illustration purposes: an EL3702 terminal with standard setting is switched to 2-fold oversampling (blue) and the optional PDO "StartTimeNextLatch" is added (red):



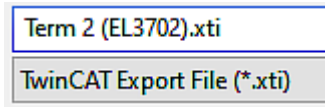
The two methods for exporting and importing the modified terminal referred to above are demonstrated below.

5.8.2 Procedure within TwinCAT with xti files

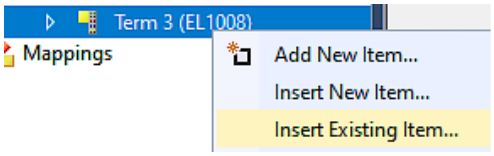
Each IO device can be exported/saved individually:



The xti file can be stored:



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



5.8.3 Procedure within and outside TwinCAT with sci file

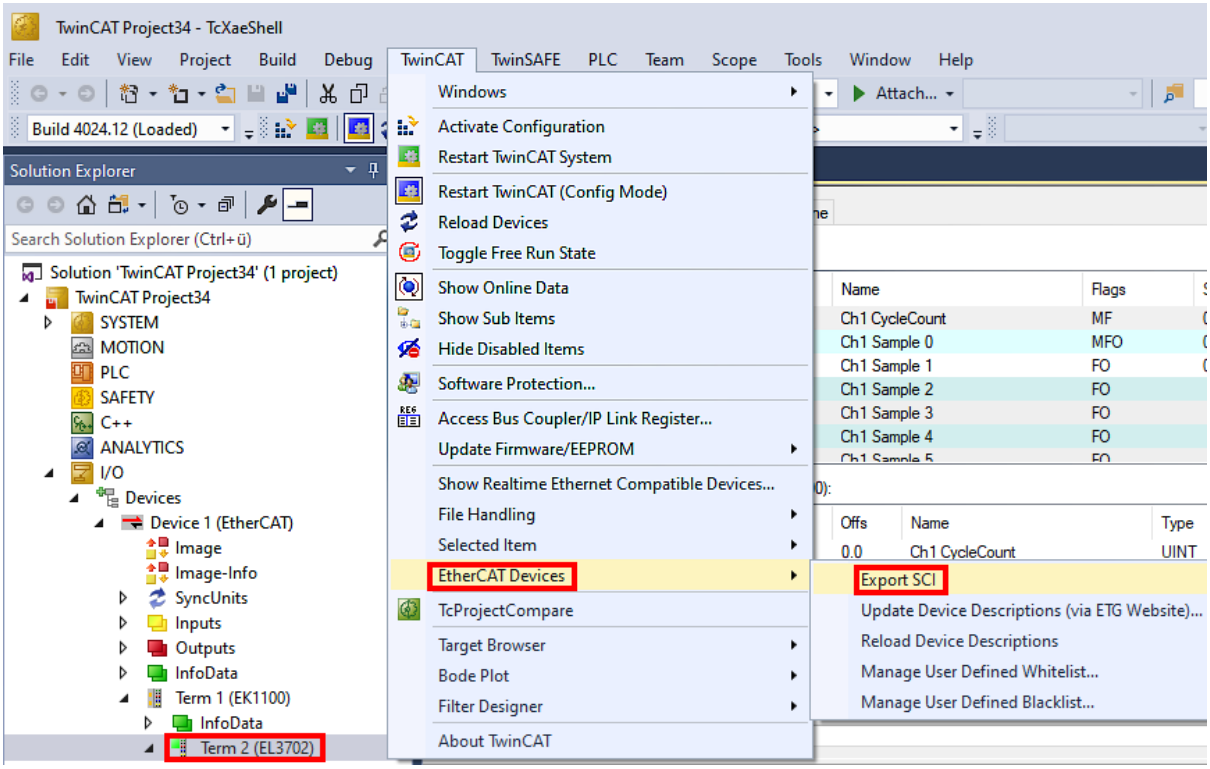
Note regarding availability (2021/01)

The SCI method is available from TwinCAT 3.1 build 4024.14.

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

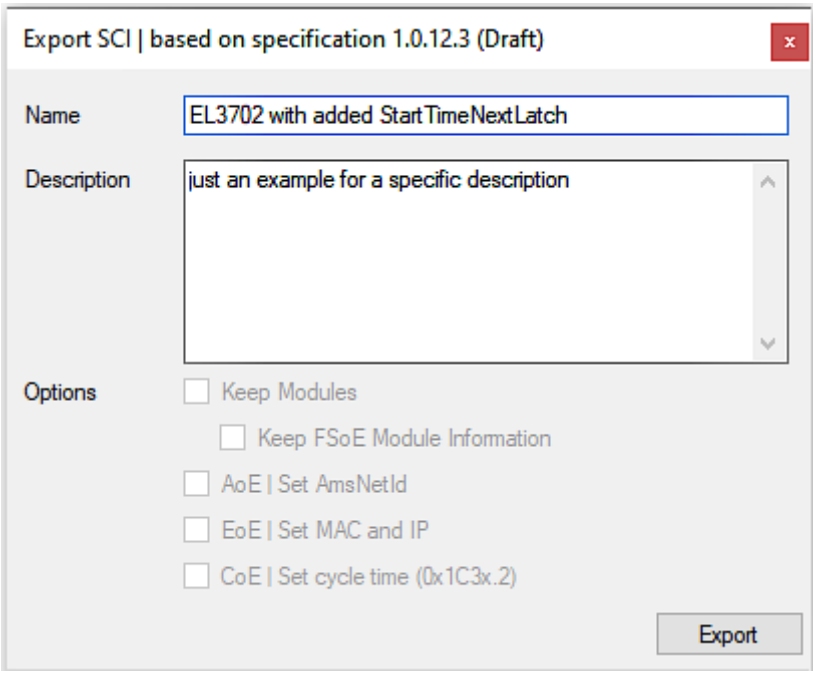
Export:

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



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

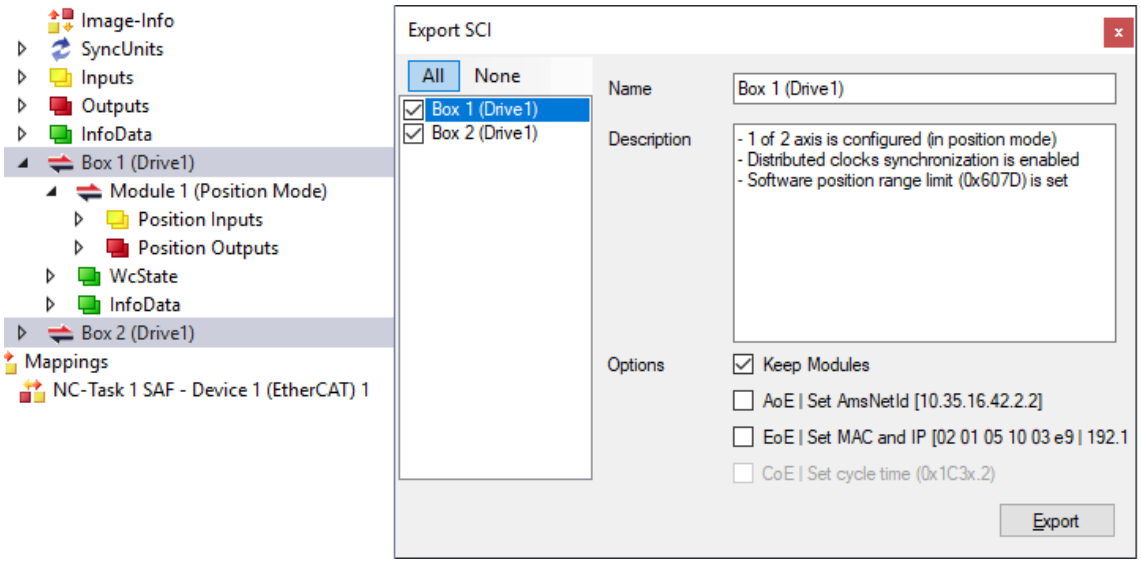
- A description may also be provided:



- Explanation of the dialog box:

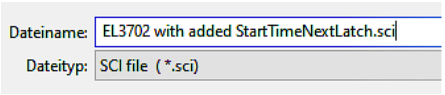
Name		Name of the SCI, assigned by the user.
Description		Description of the slave configuration for the use case, assigned by the user.
Options	Keep modules	If a slave supports modules/slots, the user can decide whether these are to be exported or whether the module and device data are to be combined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

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

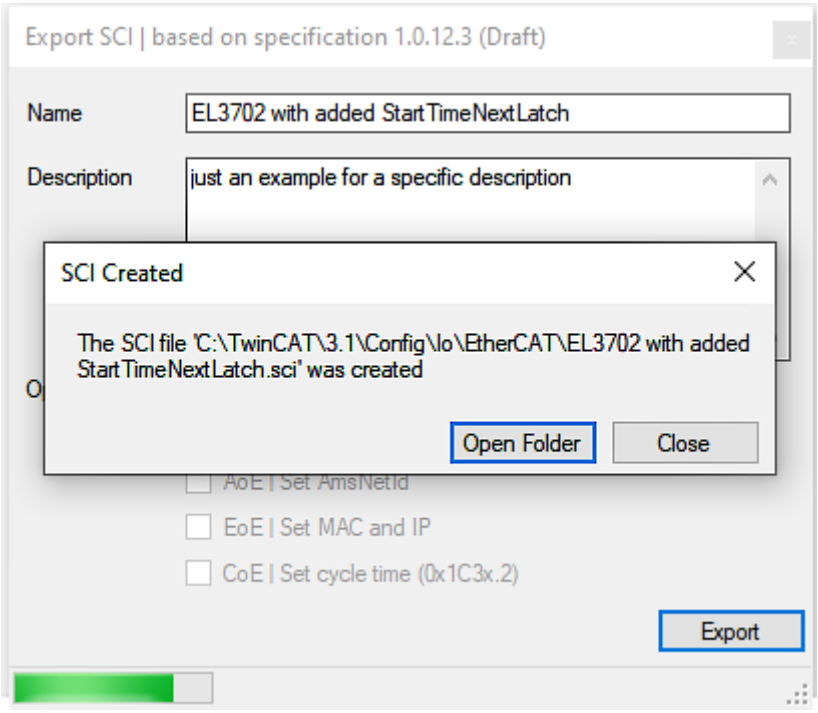


- Selection of the slaves to be exported:
  - All:
    All slaves are selected for export.

  - None:
    All slaves are deselected.
- The sci file can be saved locally:

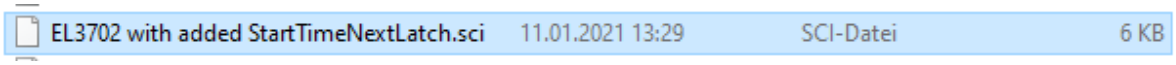


- The export takes place:

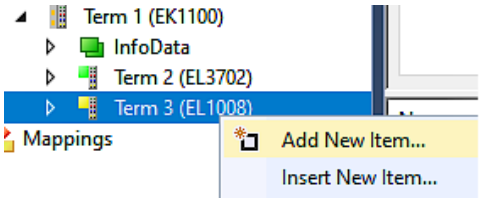


Import

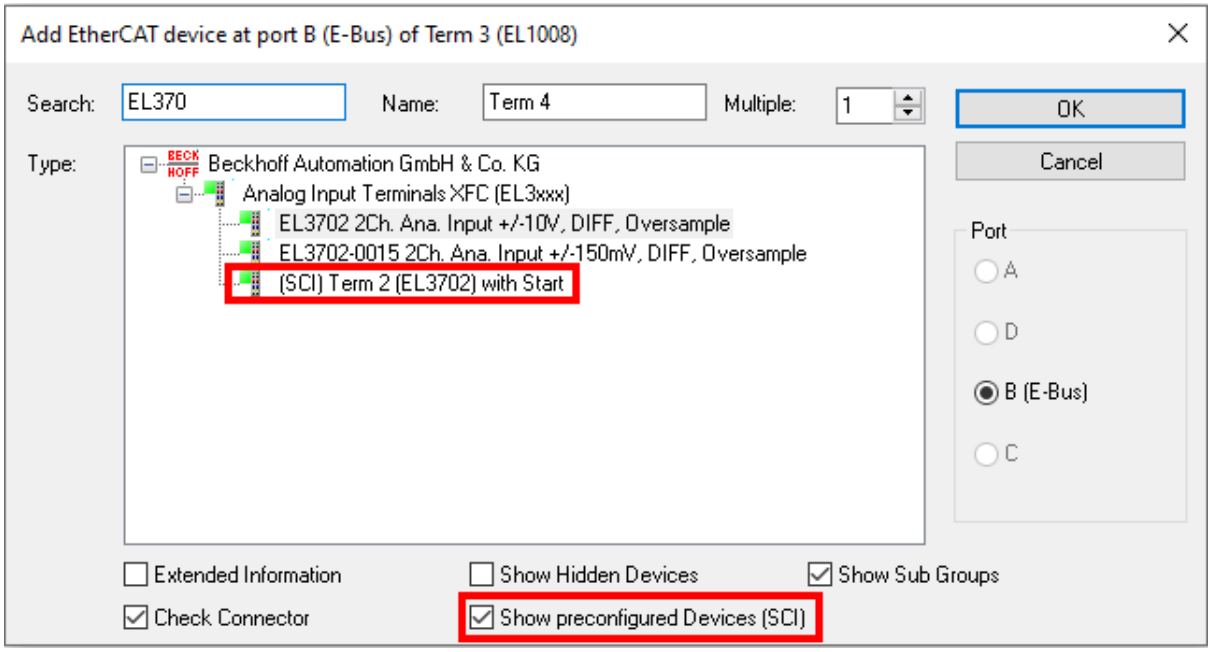
- An sci description can be inserted manually into the TwinCAT configuration like any normal Beckhoff device description.
- The sci file must be located in the TwinCAT ESI path, usually under:
  C:\TwinCAT\3.1\Config\Io\EtherCAT



- Open the selection dialog:

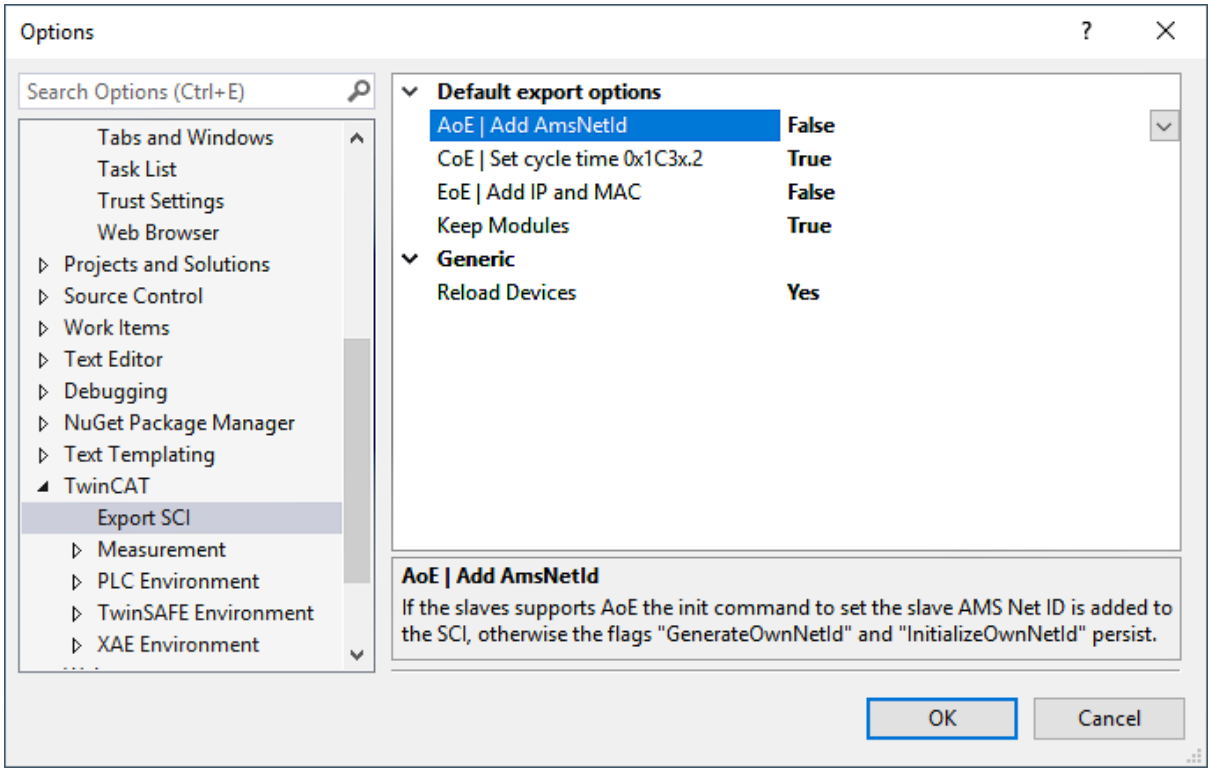


- Display SCI devices and select and insert the desired device:



Additional Notes

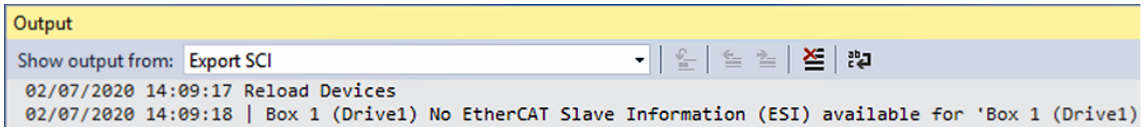
- Settings for the SCI function can be made via the general Options dialog (Tools → Options → TwinCAT → Export SCI):



Explanation of the settings:

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

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



6 Commissioning

NOTE

The internal memory may contain incorrectly set parameters

Risk of defect.

- Before commissioning, reset the terminal to the factory settings. See chapter Restoring the delivery state [▶ 226].

6.1 Wiring

NOTE

The load voltage is not short-circuit proof

Cable fire and defect possible.

- Use an overcurrent protection device for the load voltage.
- Dimension the overcurrent protection device so that the maximum current is limited to three times the value of the rated current for a maximum of 1 second.

Connect all existing components to the correct terminal points. See chapter Connection [▶ 50].

- Motor
  - Motor phases U, V, W
- Load voltage 8…48 V_{DC}
  - Use an overcurrent protection device for the load voltage. See above.
- Optional: feedback
  - Hall sensors U, V, W
  - Incremental encoders
- Optional: digital input signals at terminal points 8 and 16
  - e.g. limit switches

If necessary, supplement the hardware setup by the following components:

- The EL9576 brake chopper terminal and a ZB8110 brake resistor.
  The brake chopper terminal may be necessary if too much energy is fed back to the load voltage during braking procedures.
- The ZB8610 fan cartridge.
  A higher maximum output current is possible with the fan cartridge.

Enabling the output stage

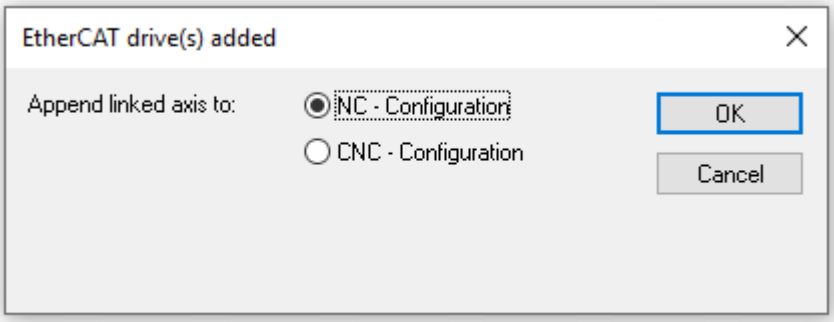
Apply 24 V to the input "HW Enable" in order to be able to enable the axis.

6.2 Integration in TwinCAT

Integrate the terminal in TwinCAT. You can find information on this in the following chapters:

- Distinction between Online and Offline [▶ 64]
- OFFLINE configuration creation [▶ 65]
- ONLINE configuration creation [▶ 70]

During the integration into TwinCAT, the following dialog box appears:



If you wish to use the TwinCAT NC functions, click OK (recommended).

This information is not binding. You can also link to an NC axis at a later time. See chapter "Integration in the NC configuration", section Manually adding an axis [▶ 114].

6.3 Selection of the operation mode

By selecting the operation mode, you determine the controlled variable and the controller structure.

If you want to operate the EL7411 with Drive Motion Control, select the operation mode "Drive Motion Control (DMC)". See chapter Commissioning with Drive Motion Control [▶ 141].

Otherwise, select the operation mode according to the desired controlled variable:

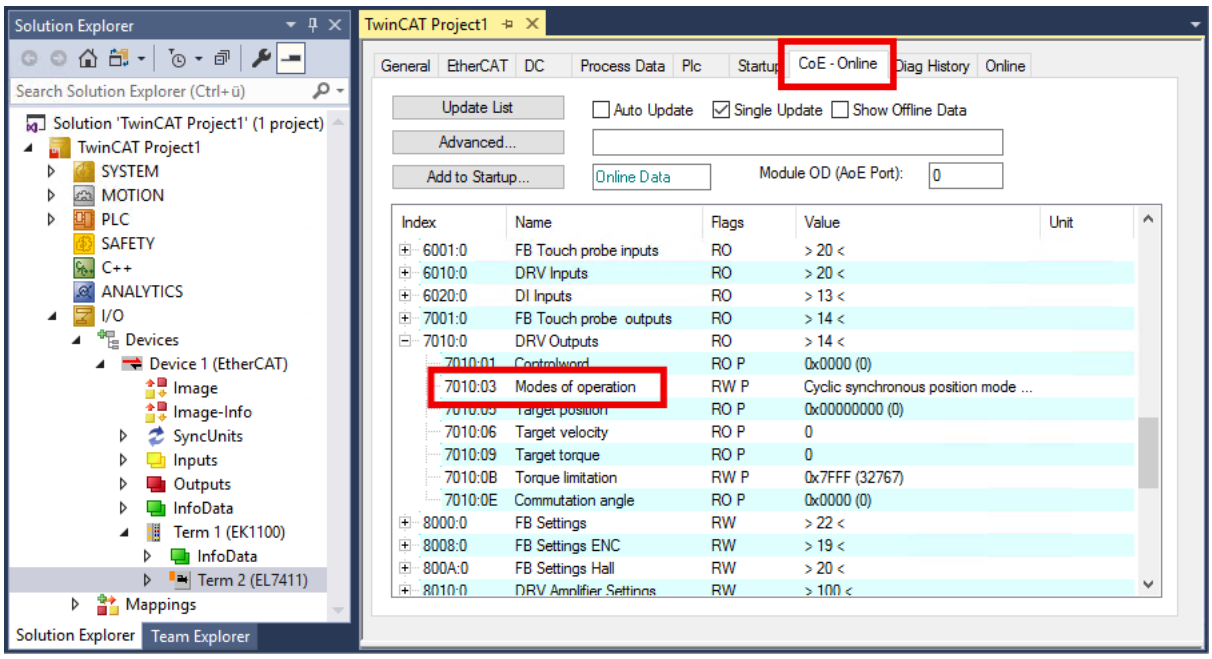
Controlled variable	Operation mode
Position	CSP [▶ 98] [1] (factory setting) (Cyclic Synchronous Position)
Velocity	CSV [▶ 100] (Cyclic Synchronous Velocity)
Torque	CST [▶ 100] (Cyclic Synchronous Torque)
Torque and commutation angle	CSTCA [▶ 101] (Cyclic Synchronous Torque with Commutation Angle)

[1] You can also control the position with the CSV operation mode. See chapter CSV (velocity control) [▶ 100]. The control performance is better with CSP, however.

Setting the operation mode

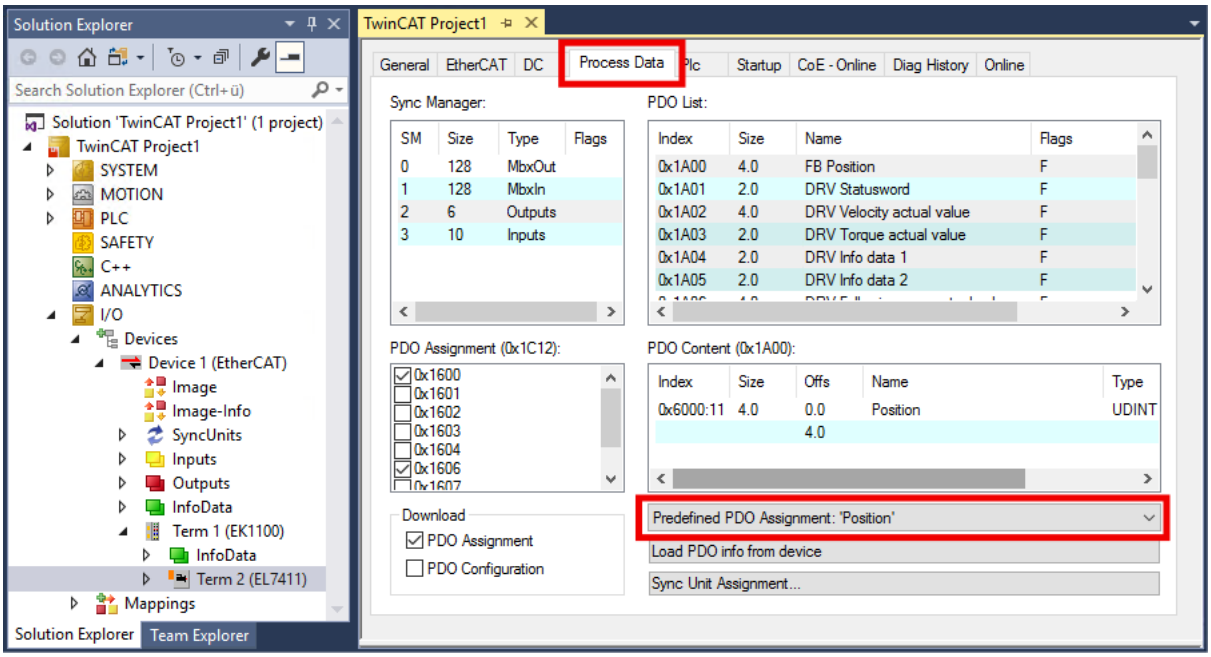
Proceed as follows to set the selected operation mode:

1. Click the "CoE - Online" tab.
2. Set the operation mode in parameter $7010:03_{hex}$ "Modes of operation".



3. Click the "Process Data" tab.

4. Click "Predefined PDO Assignment".



5. Select the correct entry according to the following table:

Operation mode	"Predefined PDO Assignment"
CSP	"Position"
CSV	"Velocity"
CST	"Torque"
CSTCA	"Torque" [1]

[1] Additional process data need to be activated for the CSTCA operation mode. See chapter CSTCA (torque control with commutation angle) [▶ 101]:

6.3.1 CSP (position control)

CSP is the abbreviation for "Cyclic synchronous position".

A defined target position can be set via the "Target position" variable.

Minimum cycle time

The cycle time in CSP modus must be 2^n * 125 µs (where n = 1 to 8),
i.e. 250 µs, 500 µs, 1 ms, 2 ms, 4 ms, 8 ms, 16 ms or 32 ms.

With the settings for the CSP operation mode, the terminal internally calculates the control loops for current, velocity and position. The NC calculates the setpoint for the position and transfers it to the terminal.

Following error monitor

Furthermore, there is an option in *CSP* mode to activate a following error monitor. The following error monitor is switched off on delivery. In all other modes this is not used and is ignored.

- The window of the following error monitor can be adjusted with the *Following error window* (Index 0x8010:50). The value set here – multiplied by the scaling factor – specifies by what position the actual position may differ from the set position, positively and negatively. The total accepted tolerance is thus twice as large as the position entered in the *Following error window* (see fig. *Following error window*).

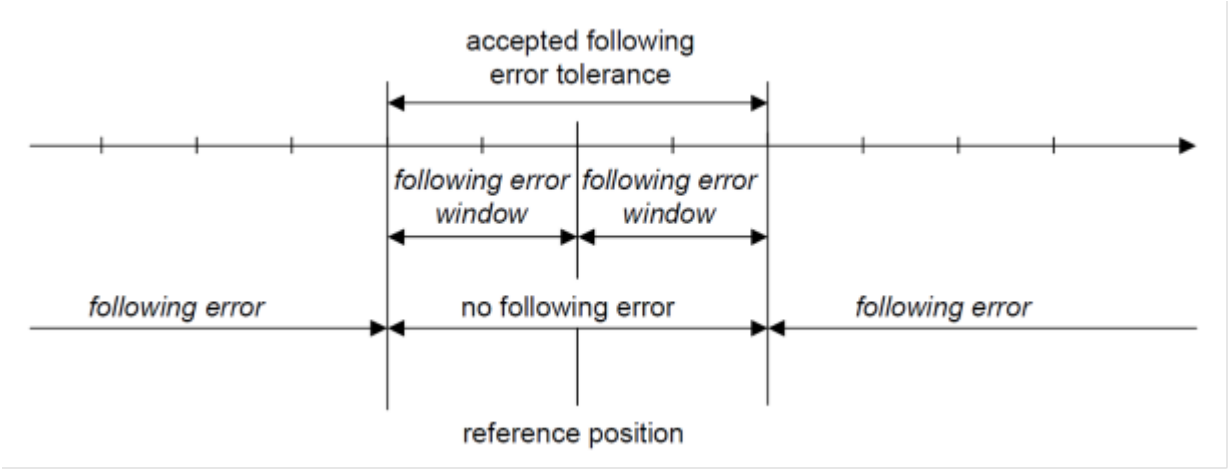


Fig. 87: Following error window

- The time (in ms) allowed for a following error exceedance can be set with the *Following error time out* (Index 0x8010:51). As soon as the target position is exceeded by more than the position entered in the *Following error window* for the time entered in the *Following error time out*, the terminal outputs an error and stops immediately.
- The current following error can be read in the *Following error actual value* (Index 0x6010:06).

The value 0xFFFFFF (- 1) in the *Following error window* means that the following error monitor is switched off and corresponds to the delivery status.
The *Following error time out* is 0x0000 (0) on delivery.

Example of motion command with following error monitoring

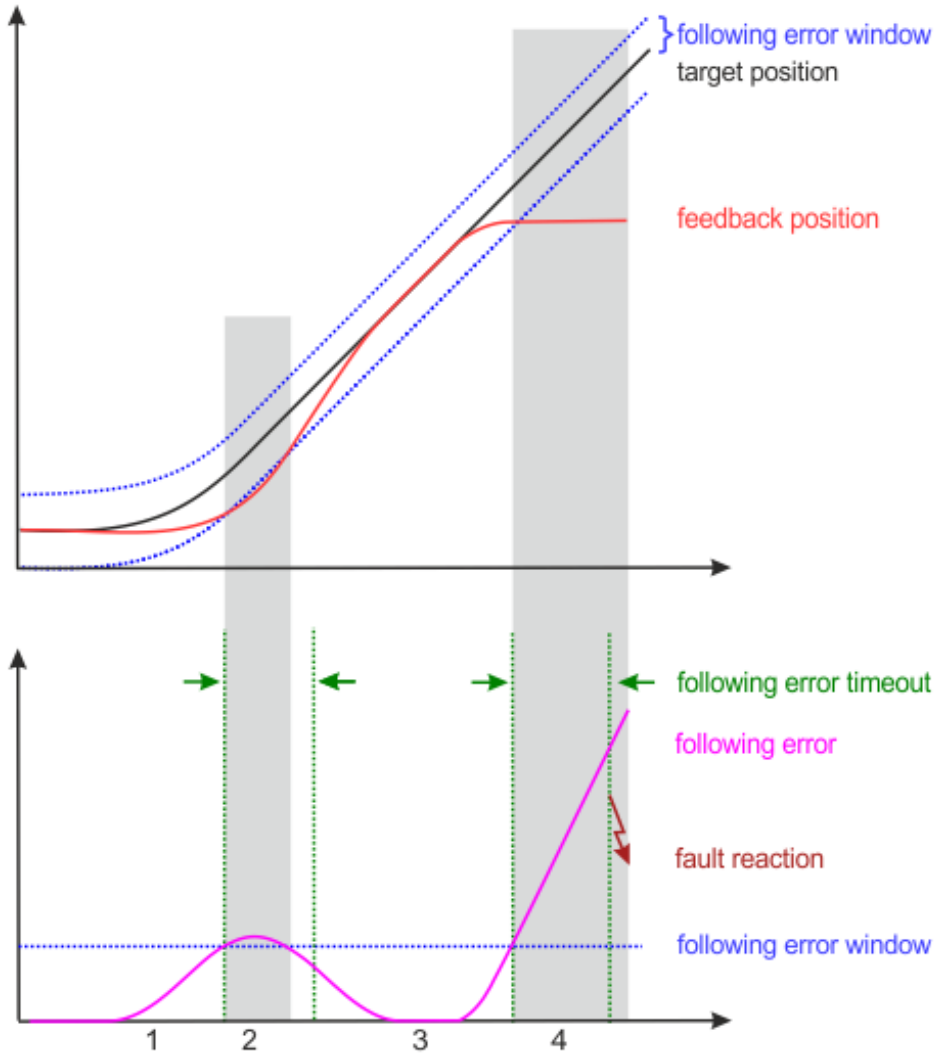


Fig. 88: Following error over time

1. When accelerating, the *following error* increases.
2. The set limit value for the following error (*following error window*) is exceeded.
   The duration of the exceedance of the *following error window* (shown in gray in the figure above) is shorter than the period specified in *following error timeout* (shown in green in the figure above).
   ⇨ No error is triggered.
3. The following error decreases to zero as soon as the *target position* is reached.
   In the event of a blockage of the axis (e.g. end stop), *target position* continues to run, while *feedback position* stops.
   ⇨ The following error increases.
4. The following error exceeds the limit value *Following error window* for a longer period than specified in *following error timeout*.
   ⇨ After expiration of *following error timeout* an error is triggered (*fault reaction*).

6.3.2 CSV (velocity control)

CSV is the abbreviation for "Cyclic synchronous velocity".

A defined velocity can be set via the "Target velocity" variable.

6.3.2.1 Position control

With the CSV operation mode you can also control the position by using TwinCAT NC as the position controller.
In the context of positioning tasks, however, the CSP [▶ 98] operation mode performs better as no bus dead times occur between the controllers (due to the communication between terminal and NC) and all controllers in the architecture are calculated in the same place.

6.3.3 CST (torque control)

CST is the abbreviation for "Cyclic synchronous torque".

A defined torque can be set via the "Target torque" variable. You cannot use TwinCAT NC to specify the torque.

6.3.4 CSTCA (torque control with commutation angle)

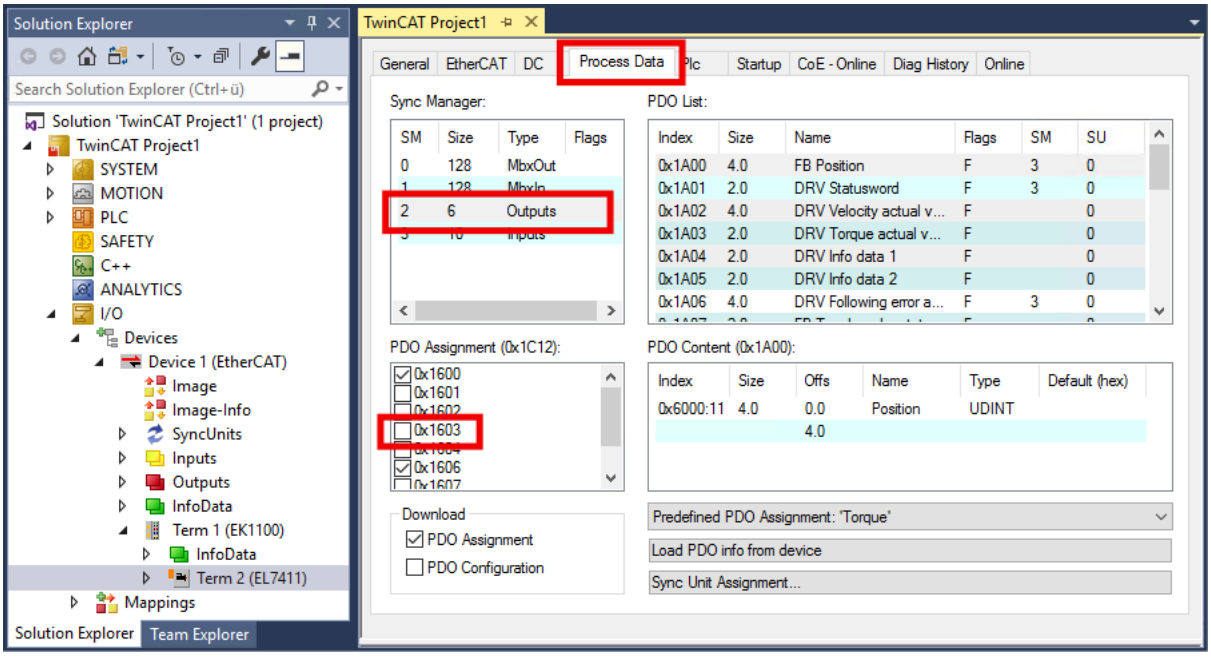
CSTCA is the abbreviation for "Cyclic synchronous torque with commutation angle".

This operation mode is a torque controller like CST [▶ 100]. In addition the user can specify the commutation angle.
You cannot use TwinCAT NC to specify speed and commutation angle.

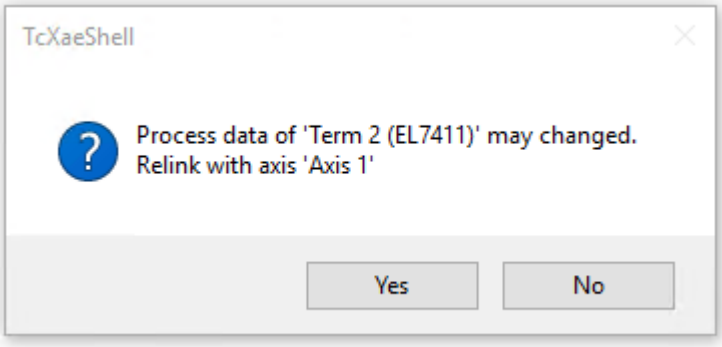
The variable "Commutation angle" can be used to set an angle which is to be maintained with a defined torque set in variable "Target torque".

The variable "Commutation angle" is located in the process data object "DRV Commutation angle", which is not activated in the factory settings.
Activate the process data object "DRV Commutation angle" as follows:

1. Click the "Process Data" tab.
2. In the "Sync Manager" box, click "Outputs".
3. Check 0x1603 in the box "PDO Assignment (0x1C12)".



⇨ If the terminal is linked with an NC axis, a dialog box appears:



In the dialog box, click "No".

6.4 Configuration of the hardware

6.4.1 Configuration of the feedback

The feedback serves to determine the relative position and velocity for the internal control. Determination of the absolute position is not possible with the feedback.

Operation without feedback is also possible. In this case, the counter-electromotive force of the motor is used as feedback. This operation mode is called "sensorless operation".

Select the feedback according to the requirements of the application.

Feedback	Recommended use cases
No feedback	• Constant high velocity • No positioning tasks
Hall sensors	• High velocity applications, e.g.: ◦ Fans ◦ pumps ◦ Conveyor belts • Coarse positioning tasks
Incremental encoder (factory setting)	Positioning tasks in which high synchronism is required
Incremental encoders and Hall sensors	

Configure the feedback as described in the following chapters.

Permissible combinations of feedback and operation mode

Operation mode	Feedback			
	An incremental encoder	An incremental encoder + Hall sensors	Hall sensors only	No feedback
CSTCA	Yes	Yes	No	No
CST	Yes	Yes [1)]	No	No
CSV	Yes	Yes	Yes	Yes
CSP	Yes	Yes	Yes	Yes

[1)] Enable with 0x8010:54 Feature bits = 0x100

6.4.1.1 Feedback systems

The following subchapters describe the configuration of the different feedback systems.

6.4.1.1.1 Operation without feedback

Configuration

1. Set parameter 8008:12 "Encoder type" to the value "disabled".
   ⇨ The input for the incremental encoder is disabled.
2. Set parameter 800A:14 "Hall sensor type" to the value "disabled".
   ⇨ The inputs for Hall sensors are disabled.
3. Set parameter 8010:64$_{hex}$ "Commutation type" to the value "Six-Step sensorless".

In sensorless operation, a clear switching torque is noticeable. From a certain velocity, control switches from open loop to closed loop. The velocity at which this change takes place depends on the nominal velocity and rated voltage. It can be calculated with the following equation.

$$v_{SensorlessControlThreshold} = 4\text{V} \times \frac{(8011:2E)\ RatedSpeed}{(8011:2F)\ NominalVoltage}$$

6.4.1.1.2 Operation with Hall sensors

Configuration

1. Set parameter 8010:64$_{hex}$ "Commutation type" to the value "Six-Step hall".
2. Set parameter 8008:12$_{hex}$ "Encoder type" to the value "disabled".
   ⇨ The input for the incremental encoder is disabled.
3. Configure the Hall sensors. See chapter Configuration of the Hall sensors [▶ 106].

6.4.1.1.3 Operation with an incremental encoder

When operating with an incremental encoder, a field-oriented control (FOC) is active.

Configuration

1. Set the parameter 8010:64$_{hex}$ "Commutation type" to the value "FOC with incremental encoder".
2. Set the parameter 800A:14 "Hall sensor type" to the value "disabled".
   ⇨ The input for Hall sensors is disabled.
3. Configure the incremental encoder. See chapter Configuration of the incremental encoder [▶ 104].

6.4.1.1.4 Operation with an incremental encoder and Hall sensors

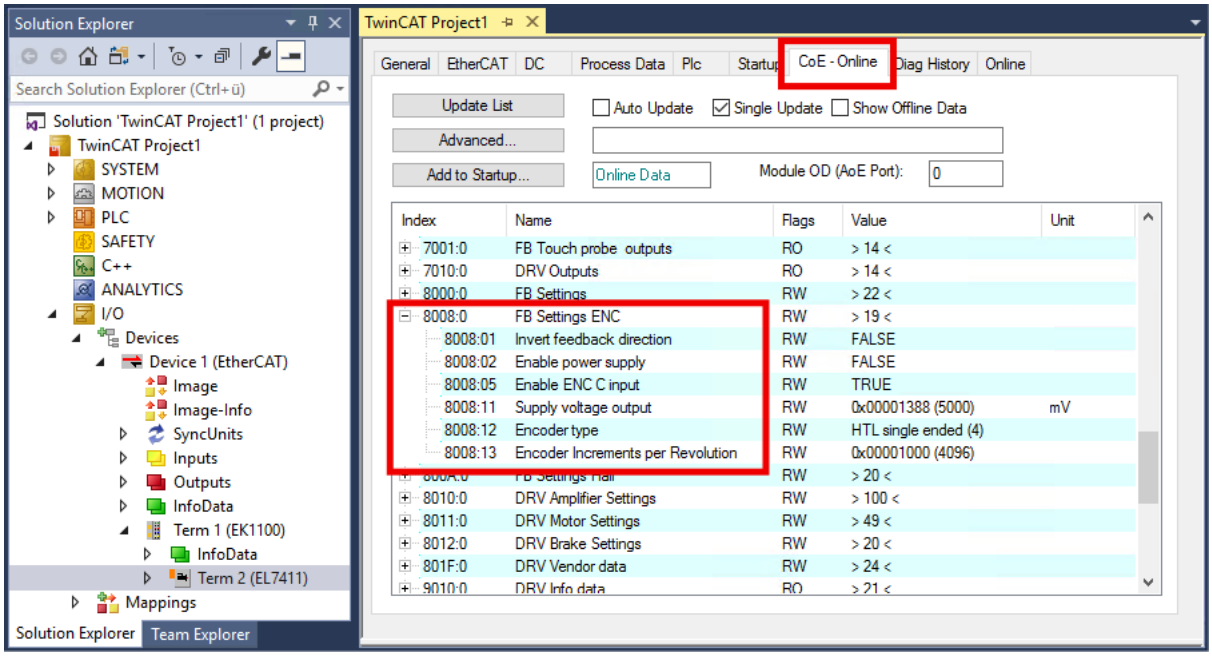
When operating with incremental encoder and Hall sensors, a field-oriented control (FOC) is active.

Configuration

1. Set the parameter 8010:64$_{hex}$ "Commutation type" to the value "FOC with incremental encoder and Hall".
2. Configure the incremental encoder. See chapter Configuration of the incremental encoder [▶ 104].
3. Configure the Hall sensors. See chapter Configuration of the Hall sensors [▶ 106].

6.4.1.2 Configuration of the incremental encoder

If you use an incremental encoder, configure it with the following CoE parameters:



Index	Name	Unit
8008:01$_{hex}$	Invert feedback direction	-
8008:02$_{hex}$	Enable power supply	-
8008:05$_{hex}$	Enable ENC C input	-
8008:11$_{hex}$	Supply voltage output [▶ 105]	mV
8008:12$_{hex}$	Encoder type [▶ 105]	-
8008:13$_{hex}$	Encoder Increments per Revolution [▶ 105]	inc/rev

8008:11 "Supply voltage output"

In this parameter, set the magnitude of the supply voltage for the encoder in millivolts. The value range is 2…24 V.

Switch on the supply voltage output by setting parameter 8008:02 "Enable power supply" to TRUE.

8008:12 "Encoder type"

In this parameter, set the signal type of the encoder.

An encoder must be connected if the value of this parameter is not "disabled". Otherwise, an error will be reported in the Diag History [▶ 163].

"HTL single ended" is set in the factory settings.

Signal type	Max. input frequency [1]	Current sink 2.5 mA	Signal level Low	Signal level High	Comment
disabled		-	-	-	Incremental encoder interface disabled Is disabled by DM2 for Six-Step Hall
RS422 differential	5 million inc/s	no	U_{diff} < -0.45 V	U_{diff} > +0.45 V	-
TTL single ended	1 million inc/s	yes	< 0.8 V	> 3.0 V	-
HTL differential	5 million inc/s	no	U_{diff} < -2 V	U_{diff} > +2 V	-
HTL single ended	5 million inc/s	yes	< 6 V	> 9.2 V	-
RS422 differential – high impedance input	5 million inc/s	no	U_{diff} < -0.45 V	U_{diff} > +0.45 V	For applications in which the encoder signals are too weak
TTL single ended – input filters disabled	5 million inc/s	yes	< 0.8 V	> 3.0 V	Higher signal frequency possible. More sensitive to interference.
Open collector	250,000 inc /s	no	< 0.8 V	> 2.0 V	Encoder switches to ground

[1] with 4-fold evaluation.

8008:13 "Encoder Increments per Revolution"

Multiply the number of increments of the encoder by a factor of four. Enter the result in this parameter.

If you are using a magnetic encoder with interpolation, calculate the number of increments according to the following equation:

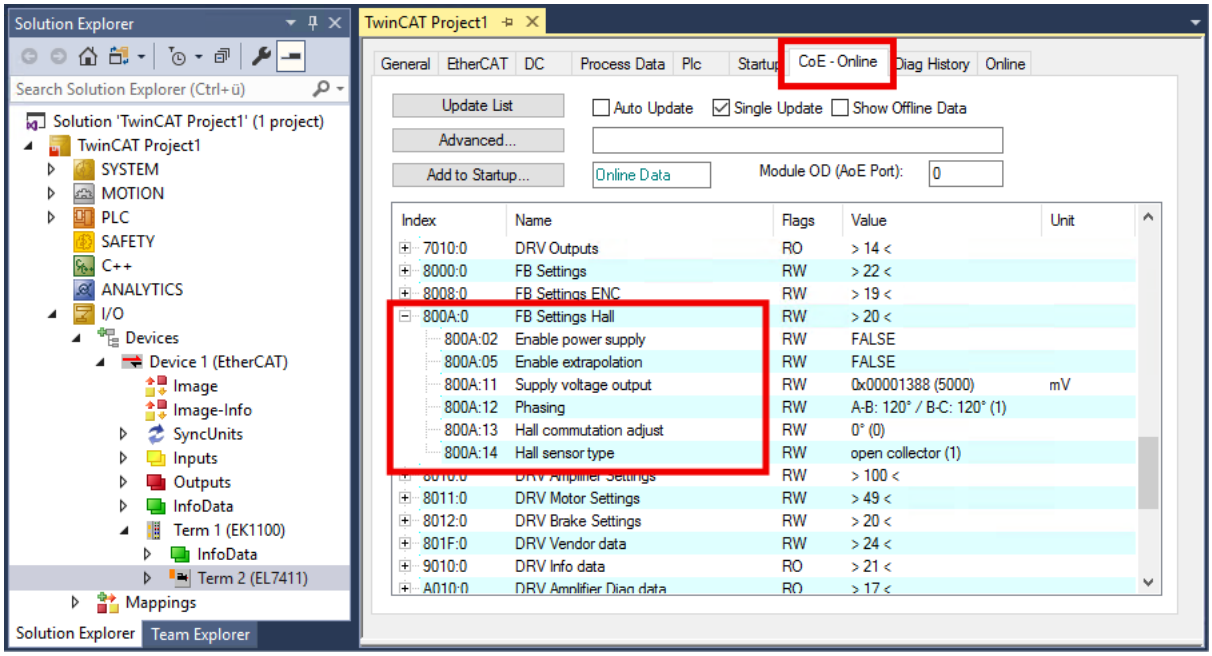
$$increments = poles \times resolution$$

Example:

$$increments = 50 \times 8192 \frac{inc}{rev} = 409600 \frac{inc}{rev}$$

6.4.1.3 Configuration of the Hall sensors

If you are using Hall sensors, configure them with the following CoE parameters:



Index	Name	Unit
800A:02	Enable power supply	-
800A:05	Enable extrapolation	-
800A:11	Supply voltage output [▶ 106]	-
800A:12	Phasing [▶ 106]	°
800A:13	Hall commutation adjust [▶ 106]	°
800A:14	Hall sensor type [▶ 106]	-

800A:11 Supply voltage output

In this parameter, set the magnitude of the supply voltage for the encoder in millivolts. The value range is 2…24 V.

Switch on the supply voltage output by setting parameter 800:02 "Enable power supply" to TRUE.

800A:12 Phasing

Use the Scan Feedback [▶ 111] function to automatically determine this parameter.

800A:13 Hall commutation adjust

Use the Scan Feedback [▶ 111] function to automatically determine this parameter.

800A:14 Hall sensor type

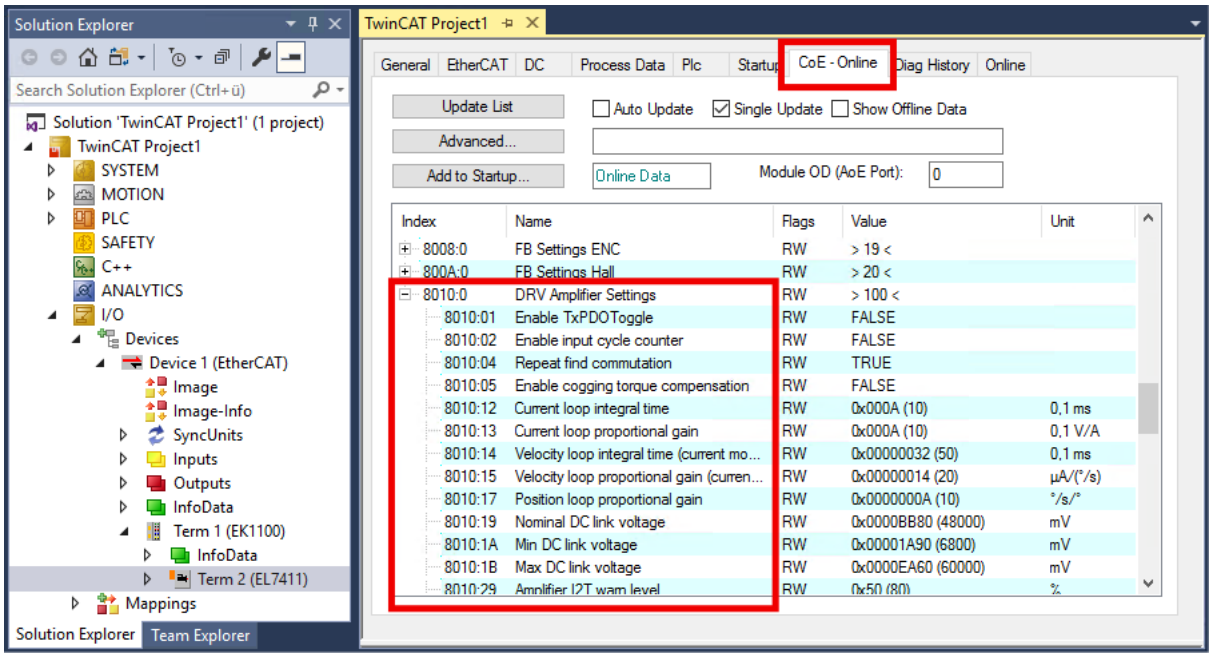
Signal type	Specification	Comment
(0) disabled	Hall sensor interface disabled	Is disabled by DM2 for FOC with incremental encoder
(1) Open collector	5 V referenced to the terminal's internal pull-up resistor	Hall sensor switches to ground. Digital Hall sensors must be used.

6.4.2 Configuration of the motor and amplifier

Before integrating the motor into the NC, it is important to make some settings in the CoE objects 8010:xx and 8011:xx. This information must be entered and, for the most part, taken from the data sheet or measured.

Contact the motor manufacturer if any information is missing from the data sheet.

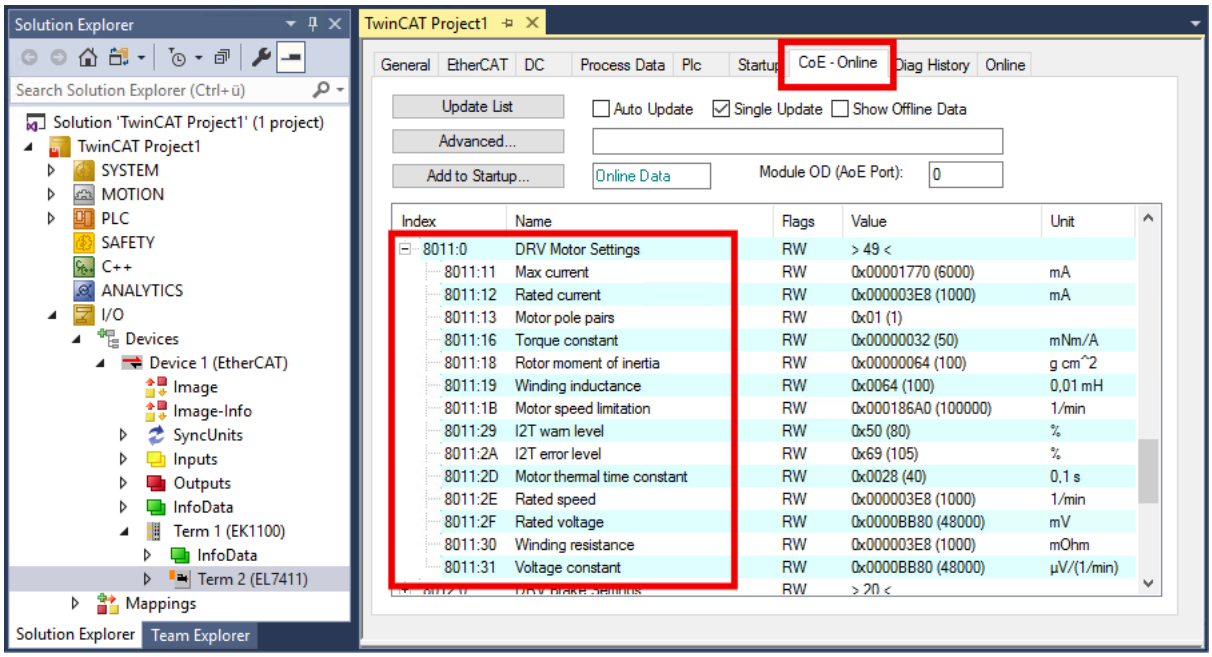
Index 8010: DRV Amplifier Settings



Index	Name	Description	Unit
8010:04	Repeat find commutation	This parameter is only relevant if you are using an incremental encoder. If this parameter is TRUE, the commutation angle is determined each time the axis is released. [1)] If this parameter is FALSE, the commutation angle is determined only if it is necessary, e.g. after a power cycle of the terminal.	-
8010:19	Nominal DC link voltage	Here, enter the magnitude of the load voltage in millivolts that you have connected at terminal points 7' and 8'. Value range: 8…48 V_{DC}	mV
8010:64	Commutation type	Set feedback. See chapter Configuration of the feedback [▶ 102].	-

[1)] There are scenarios in which commutation determination must be repeated as soon as the orientation of the feedback to the motor is no longer known. Possible causes are a power cycle of the encoder or an interim change of operation mode.

Index 8011: DRV Motor Settings



Index	Name	Description	Unit
8011:11	Max current	The maximum peak value of the winding current. This value is the maximum current with which the motor can be loaded for a short time.	mA
8011:12	Rated current	The nominal current of the motor. The nominal current is the maximum current with which the motor can be permanently loaded.	mA
8011:13	Motor pole pairs	Number of pole pairs. [1] If only the number of poles is specified in the data sheet, divide this value by two to get the number of pole pairs.	-
8011:16	Torque constant	Output torque per current intensity.	mNm/A
8011:18	Rotor moment of inertia	The moment of inertia from the point of view of the motor. Enter the entire moment of inertia of the rotor and the associated mechanism.	g cm²
8011:19	Winding inductance	The winding inductance. You can have this value measured automatically. See chapter Scan Motor [▶ 110].	mH
8011:2D	Motor thermal time constant	The thermal time constant of the winding. This value is relevant for the calculation of the I²T model.	0.1 s
8011:2E	Rated speed	The nominal speed of the motor at the nominal voltage. [1]	rpm
8011:2F	Rated voltage	The nominal voltage of the motor according to the manufacturer's specification.	mV
8011:30	Winding resistance	The winding resistance, measured between two motor phases. You can have this value measured automatically. See chapter Scan Motor [▶ 110].	mΩ
8011:31	Voltage constant	The voltage constant of the motor. [1] The counter-electromotive force of the motor depends on the speed. The voltage constant establishes a relationship between the two values. This value is important for the correct control of the motor.	µV / min^{-1}

[1] For linear axes, see chapter: Special case - linear axis [▶ 109].

6.4.2.1 Special case - linear axis

For linear axes, the following parameters differ from the parameters described in the chapter Configuration of the motor and amplifier [▶ 107]

8011:13 Motor pole pairs

Set this parameter to the value one.

8011:2E Rated speed

Convert the rated speed to an equivalent rated speed of rotation.

$$n_{rated} = \frac{v_{rated}}{2 \times pole\ pitch} \times 60 \frac{s}{min}$$

The pole pitch corresponds to the half pole pair spacing and is therefore multiplied by two in this equation.

Example:

- Rated speed v_{rated} = 544 mm/s
- Pole pitch = 16 mm

$$n_{rated} = \frac{544\ \frac{mm}{s}}{2 \times 16\ mm} \times 60 \frac{s}{min} = 1020\ rpm$$

8011:31 Voltage constant

Convert the voltage constant k_e as follows. Note the units.

$$k_e = \frac{k_{e,linear} \left[\frac{V}{m/s}\right]}{\frac{1000\ mm}{2 \times pole\ pitch\ [mm]} \times 60 \frac{s}{min}} \times 1000000 \frac{\mu V}{V}$$

Example:

- Pole pitch = 16 mm
- Voltage constant k_e = 6.8 V/(m/s)

$$k_e = \frac{6.8 \frac{V}{m/s}}{\frac{1000\ mm}{2 \times 16\ mm} \times 60 \frac{s}{min}} \times 1000000 \frac{\mu V}{V} = 3627 \frac{\mu V}{rpm}$$

6.4.3 Scanning the hardware

EL7411 can scan existing hardware automatically. Certain parameters of the hardware are determined and stored in the corresponding entries in the CoE.

Different procedure with TwinCAT 2

- See chapter Scanning the hardware with TwinCAT 2 [▶ 113].

6.4.3.1 Scan Motor

The following parameters are determined during the "Scan Motor" procedure:

- The winding inductance of the motor: parameter 8011:19 "Winding inductance".
- The winding resistance between two phases of the motor: parameter 8011:30 "Winding resistance".
- Initial values for the current controller, velocity controller and position controller.

Executing the scan

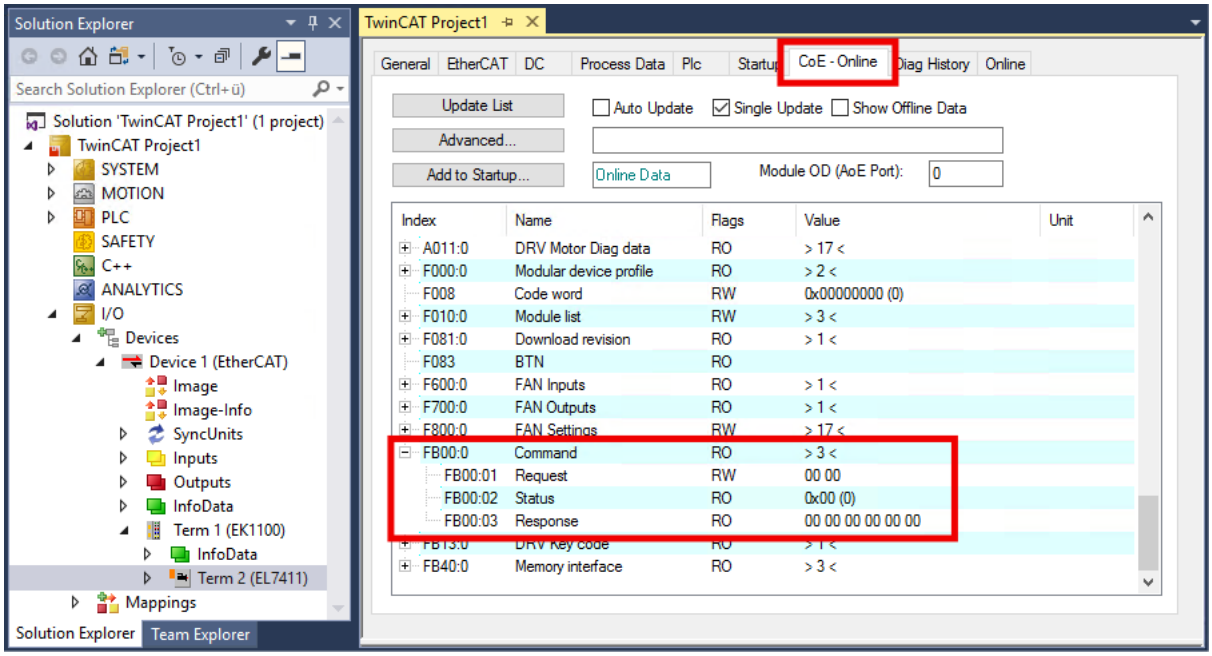
⚠ CAUTION

The motor shaft moves during the scan procedure

Injuries and damage to property are possible.

- Maintain a safe distance to the motor and the moving mechanism.
- Ensure that the motor shaft can move freely.

1. Ensure that all other motor parameters are set correctly. See chapter Configuration of the motor and amplifier [▶ 107].
2. Write the command 0x8007 to the register FB00:01 "Request".



⇨ The scan runs.

⇨ The value of the register FB00:02 "Status" shows the progress of the scan. The values 100_{dec}…199_{dec} correspond to 0... 99 %.

3. Wait until the register FB00:02 "Status" has one of the following values: 0, 1, 2, 3.

⇨ Value 0: The scan has been successfully completed.

⇨ Value 3: Error. See chapter Error diagnosis [▶ 113].

6.4.3.2 Scan Feedback

The following parameters are determined during the "Scan Feedback" procedure:

- The counting direction of the encoder: parameter 8008:01 "Invert feedback direction".
- The arrangement of the Hall sensors in the motor: parameter 800A:12 "Phasing".
- The commutation offset of the Hall sensors: parameter 800A:13 "Hall commutation adjust".

The following parameters are checked for plausibility during the scan procedure:

- 8008:13 "Encoder increments per Revolution"
- 8011:13 "Motor pole pairs"

Executing the scan

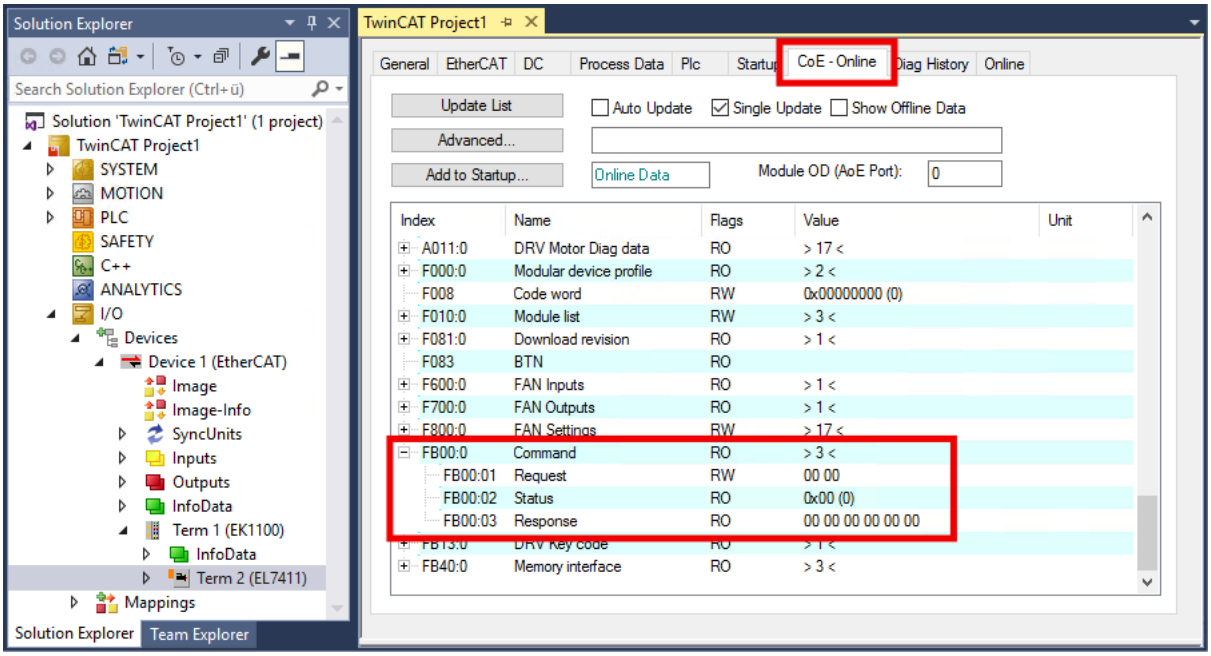
⚠ CAUTION

The motor shaft moves during the scan procedure

Injuries and damage to property are possible.

- Maintain a safe distance to the motor and the moving mechanism.
- Ensure that the motor shaft can move freely.

1. Ensure that all other feedback parameters are set correctly.
   See chapter Configuration of the feedback [▶ 102].
2. Ensure that all motor parameters and amplifier parameters are set correctly.
   See chapter Configuration of the motor and amplifier [▶ 107].
3. Write the command 0x8008 to the register FB00:01 "Request".



   ⇨ The scan runs.

   ⇨ The value of the register FB00:02 "Status" shows the progress of the scan.
   The values 100_{dec}…199_{dec} correspond to 0... 99 %.
4. Wait until the register FB00:02 "Status" has one of the following values: 0, 1, 2, 3.

   ⇨ Value 0: The scan has been successfully completed.

   ⇨ Value 3: Error. See chapter Error diagnosis [▶ 113].

6.4.3.3 Scan Motor Cogging

When moving the motor, position-dependent torque fluctuations occur due to the cogging torques between the magnets. This can be compensated by determining appropriate cogging coefficients in the control. The "Scan Motor Cogging" function determines and saves these coefficients (8010:61). It is a good idea to add the measured cogging coefficients to the start-up list after a successful scan.

Executing the scan

⚠ CAUTION

The motor shaft moves during the scan procedure

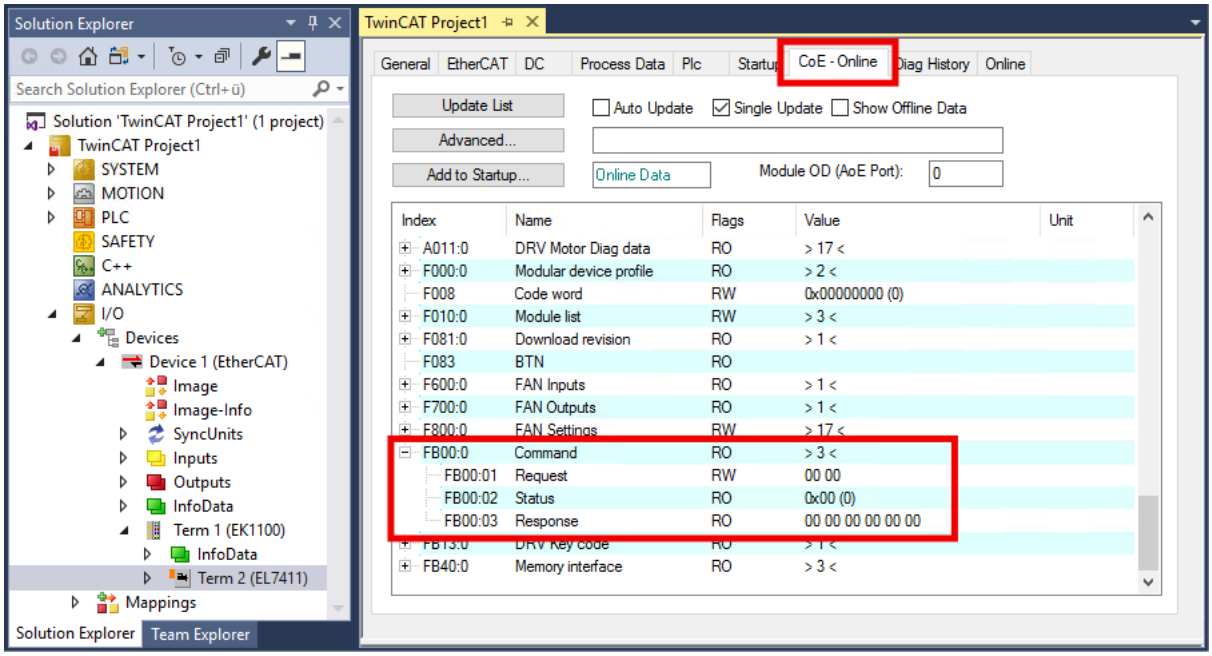
Injuries and damage to property are possible.

- Maintain a safe distance to the motor and the moving mechanism.
- Ensure that the motor shaft can move freely.

Requirements

- Stable operation in the CSP operation mode. See chapter Selection of the operation mode [▶ 96].
- An encoder is connected and configured.
- The encoder has at least 256 increments per revolution.

1. Remove any load from the motor shaft.
2. Write the command 0x8009 to the register FB00:01 "Request".



⇨ The scan runs.

⇨ The value of the FB00:02 register shows the progress of the scan.
The values 100_{dec}…199_{dec} correspond to 0... 99 %.

3. Wait until the register FB00:02 "Status" has one of the following values: 0, 1, 2, 3.

⇨ Value 0: The scan has been successfully completed.

⇨ Value 3: Error. See chapter Error diagnosis [▶ 113].

4. Check Diag History: the error 0x8420 means that the controller is insufficiently optimized.

You can enable cogging compensation in the CoE parameter 8010:05 "Enable cogging torque compensation".

6.4.3.4 Error diagnosis

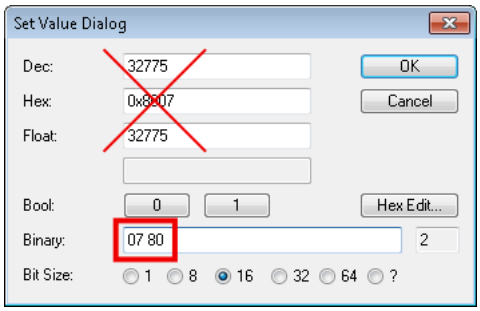
After a scan procedure is completed, an exit code is available on the tab FB00:02 "Status". If the exit code is 3, an error has occurred. Evaluate the tab FB00:03 "Response" using the following table:

Response	Meaning	Comment
xx 00 01 00 00 00	Invalidstartupstate	The axis is enabled. A scan procedure is only possible if the axis is not released.
xx 00 02 00 00 00	Timeout	Timeout during the scan procedure.
xx 00 03 00 00 00	Driveerror	An error has occurred. Check the Diag History [▶ 163].
xx 00 04 00 00 00	Invalid EtherCAT state	EL7411 is not in EtherCAT OP state.

An error does not need to be acknowledged. If a scan was aborted with an error message, you can simply start a new scan.

6.4.3.5 Scanning the hardware with TwinCAT 2

In TwinCAT 2 a decimal or hexadecimal input of the commands is not possible. Enter the commands in the "Binary" field.



Use the following values for the commands in the "Binary" field:

Command	Value "Binary"
Scan Motor	07 80
Scan Feedback	08 80
Scan Motor Cogging	09 80

6.5 Commissioning with the TwinCAT NC

6.5.1 Integration into the TwinCAT NC

You can only use the TwinCAT NC for the CSP and CSV operation modes.

Adding an axis automatically

- Once the terminals have been scanned successfully, TwinCAT detects the new axes automatically. The user is asked whether the detected axes should be added automatically (see Fig. *Axis detected*). If this is confirmed, all axes are automatically liked to the NC.

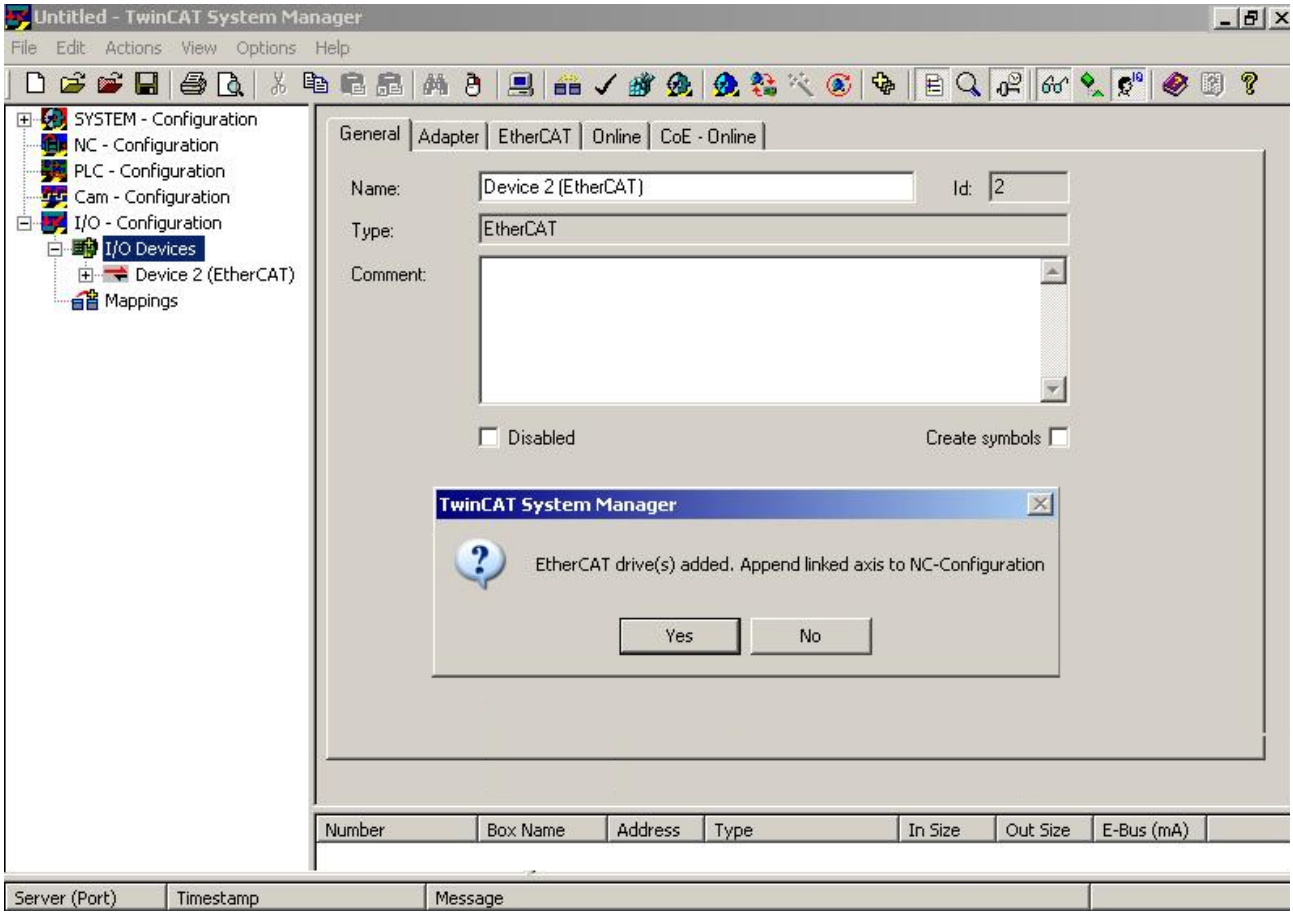


Fig. 89: Axis detected

- Several parameters have to be set before the motor can be started up. The values can be found in section "Configuration of the main parameters".
  Please set these parameters before continuing with the motor commissioning procedure.

Adding an axis manually

- First add a new task. Right-click on NC configuration and select "Append Task..." (see Fig. *Adding a new task*).
- Rename the task if required and confirm with OK.

Fig. 90: Adding a new task

- Right-click on *Axes*, then add a new axis (see Fig. *Adding a new axis*).

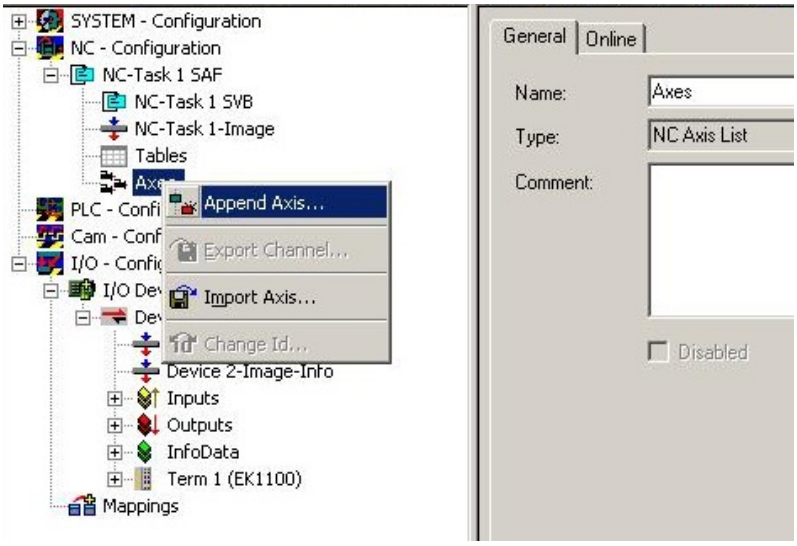


Fig. 91: Adding a new axis

- Select Continuous Axis type and confirm with OK (see Fig. *Selecting and confirming the axis type*).

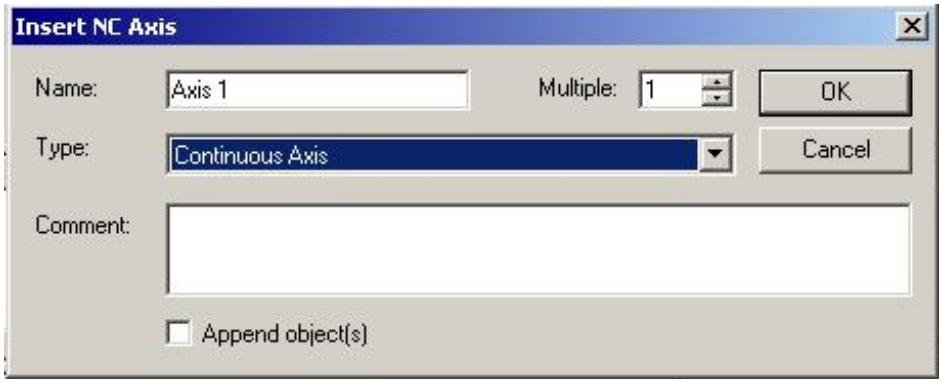


Fig. 92: Selecting and confirming the axis type

- Left-click your axis to select it. Under the S*ettings* tab select "Link To..." (see Fig. *Linking the axis with the terminal*).

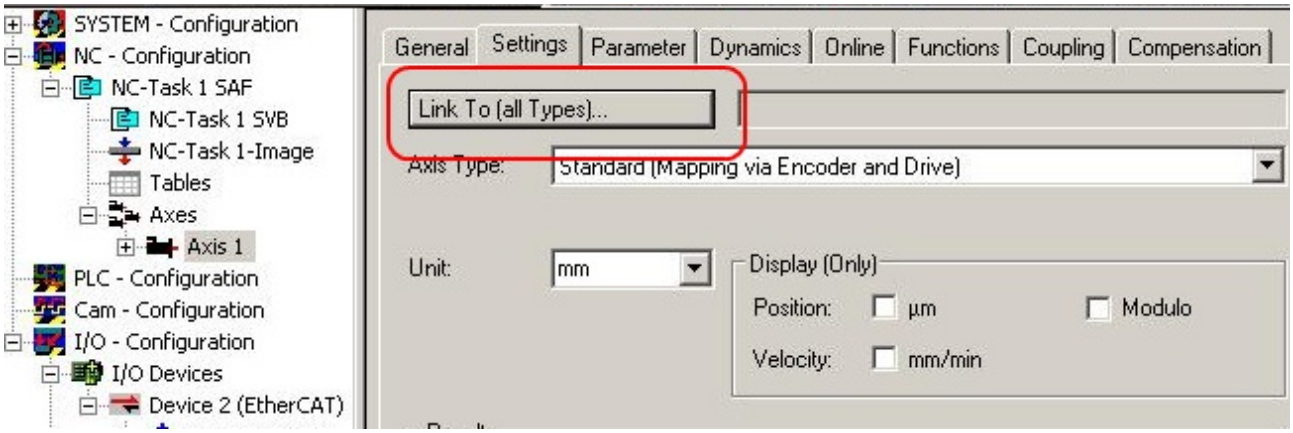


Fig. 93: Linking the axis with the terminal

- Select the required terminal (CANopen DS402, EtherCAT CoE) and confirm with OK.

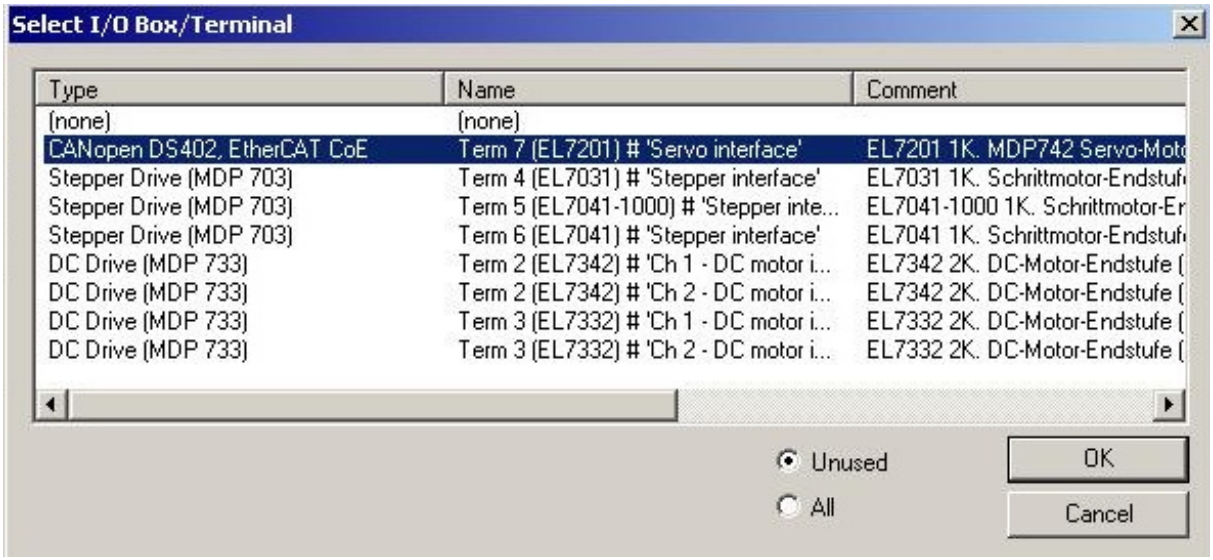


Fig. 94: Selecting the right terminal

- All main links between the NC configuration and the terminal are set automatically (see Fig. "*Automatic linking of all main variables*")

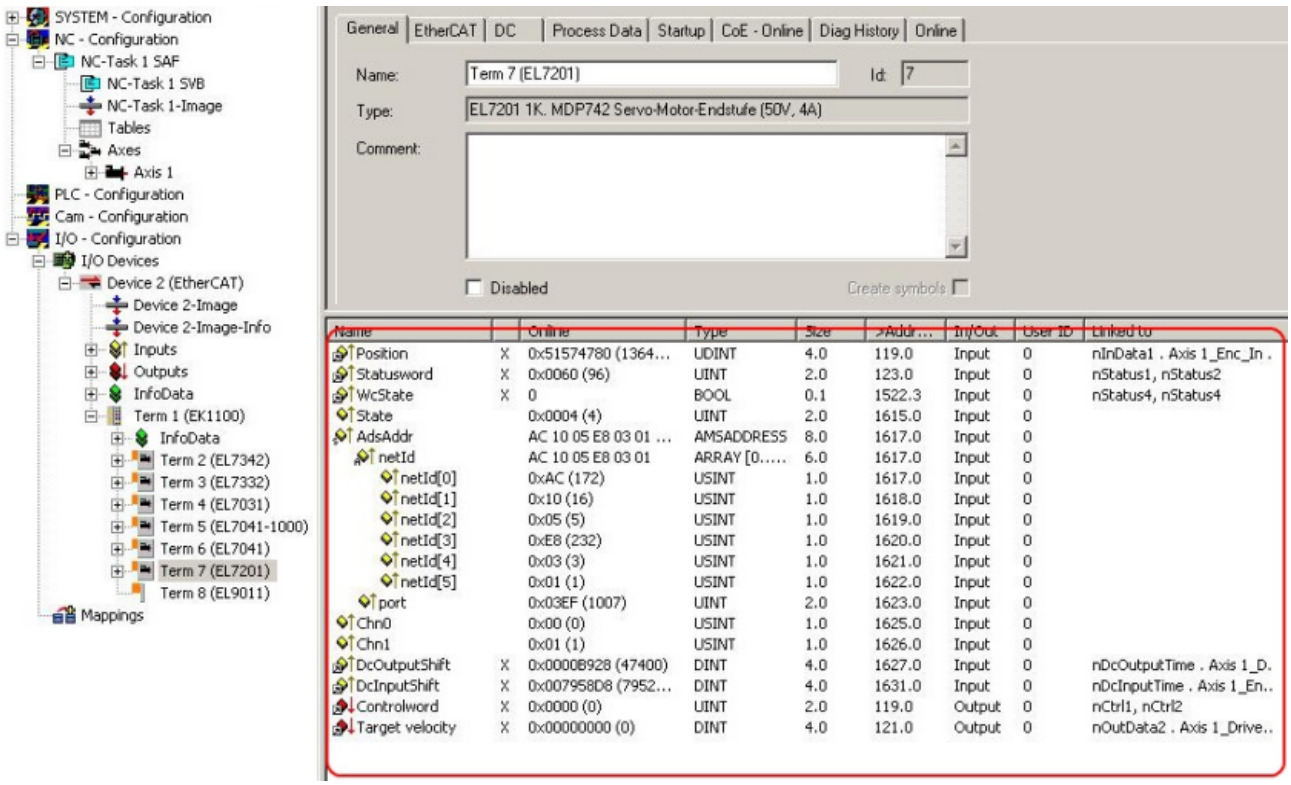


Fig. 95: Automatic linking of all main variables

- Several parameters have to be set before the motor can be started up. The values can be found in sections "CoE settings" and "NC settings".
  Please set these parameters before continuing with the motor commissioning procedure.

6.5.2 Configuration of the TwinCAT NC

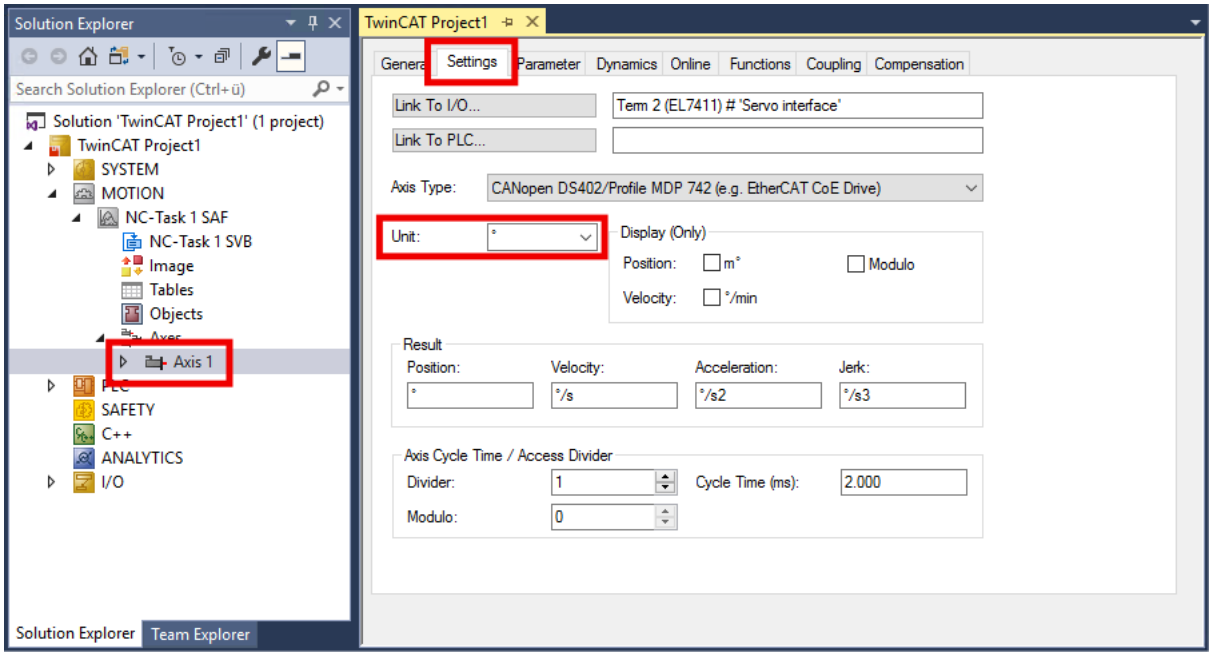
The TwinCAT NC can be configured by parameters. A complete description of the parameters of the TwinCAT NC can be found in the Documentation of the TwinCAT functions TF50x0 or on our website: https://www.beckhoff.de/tf5000.

Set the following parameters carefully:

Basic unit

This setting defines the units of the parameters of the axis.

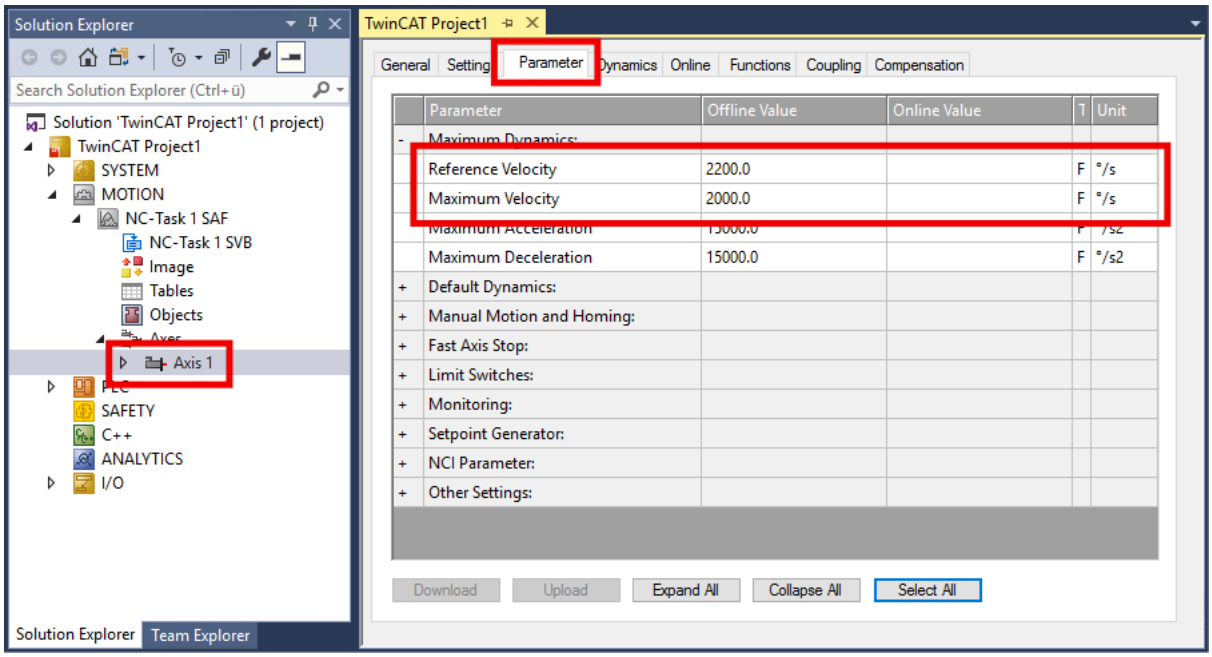
You can find this setting at:
NC Axis > "Settings" tab > drop-down menu "Unit".



- For rotary movements, select ° or "Degree".
- For linear movements, select mm or m.

Reference velocity and maximum velocity

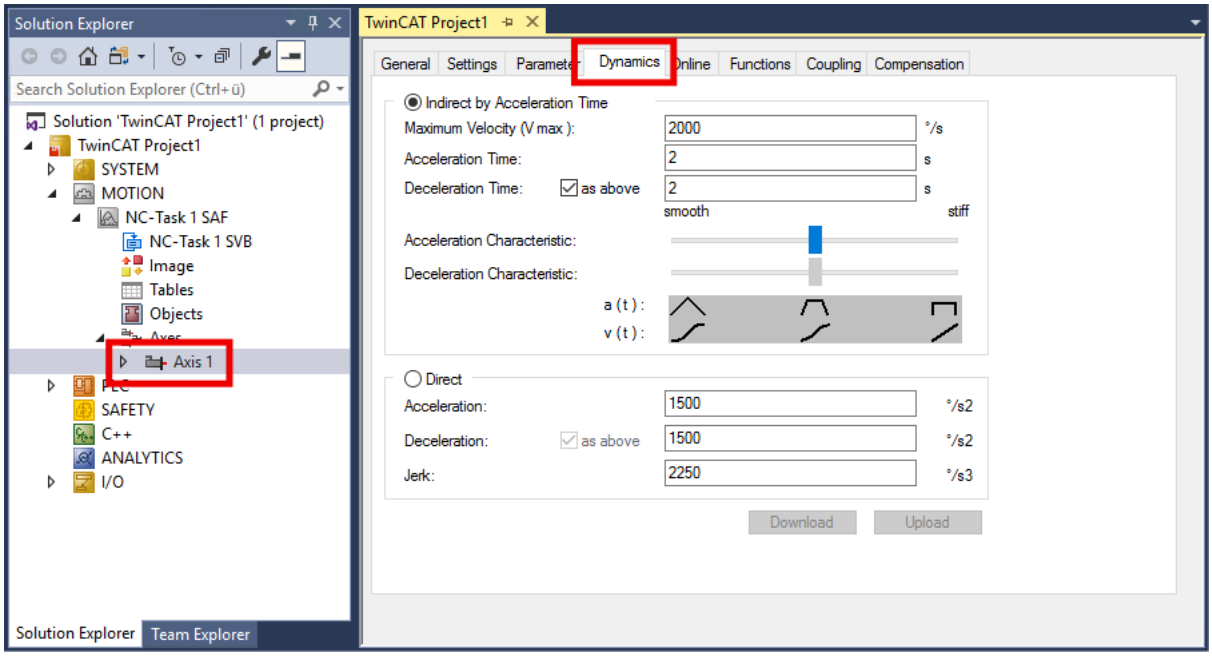
You can find these parameters at:
NC Axis > "Parameters" tab > "Reference Velocity" and "Maximum Velocity"



Recommendation: enter the rated speed or rated velocity for the "Reference Velocity".

Dynamics

You can find the dynamics parameters at:
NC Axis > "Dynamics" tab

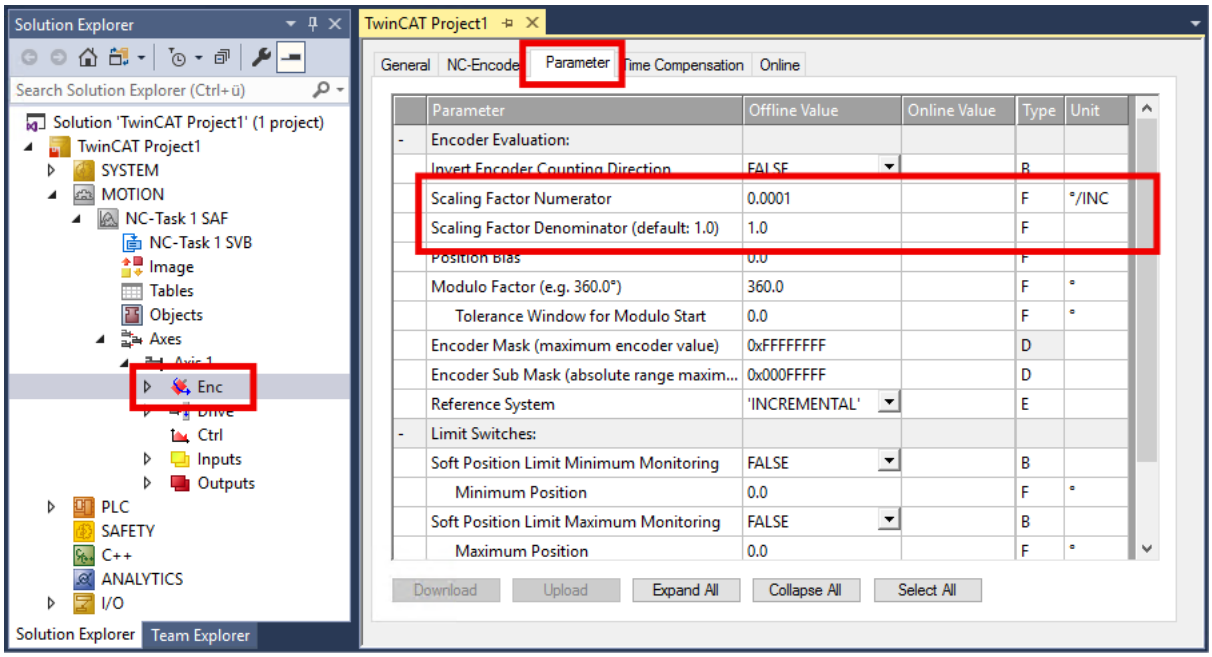


Set the acceleration time and deceleration time: "Acceleration Time" and "Deceleration Time".

Select the ramp curve using the sliders.

Encoder scaling factor for rotary axes

You can find these parameters at:
NC Axis > "Enc" > "Parameters" tab



The encoder scale factor is specified as a fraction. There is a parameter for the numerator and a parameter for the denominator.

- Numerator: "Scaling Factor Numerator"
- Denominator: "Scaling Factor Denominator"

The value for the encoder scaling factor depends on whether you are using an incremental encoder:

- For operation with an incremental encoder:
  - "Scaling Factor Numerator": 360°
  - "Scaling Factor Denominator": The resolution of the incremental encoder multiplied by a factor of four.
    Unit: Increments per revolution.
    Example with 1024 increments: 1024 x 4 = 4096
- For operation without incremental encoder:
  - "Scaling Factor Numerator": 360°
  - "Scaling Factor Denominator": 65536

Encoder scaling factor for linear axes

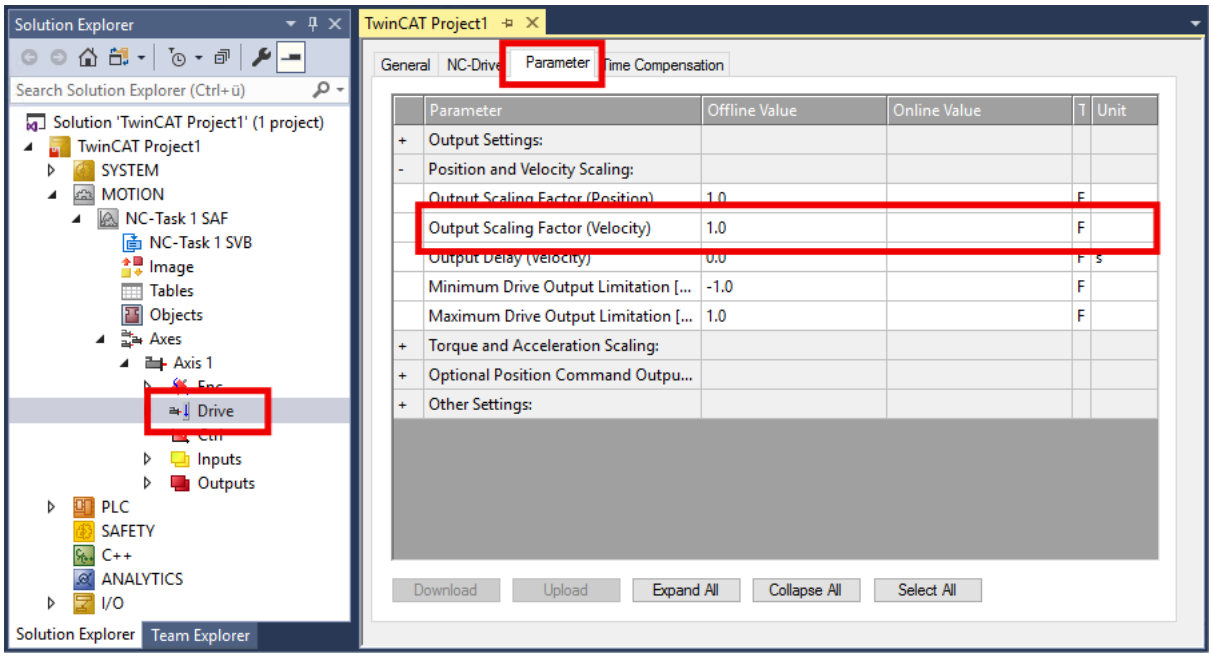
- "Scaling Factor Numerator": 1 mm
- "Scaling Factor Denominator":
  - Convert the resolution of the incremental encoder to Inc/mm.
  - Multiply the resolution by a factor of four.
  - Enter the result in the parameter "Scaling Factor Denominator".

Example:

The resolution of the incremental encoder is specified as 512 Inc / (2 mm). This corresponds to 256 Inc / mm.
For the "Scaling Factor Denominator", enter: 256 x 4 = 1024.

Output scaling for velocity

You can find this parameter at:
NC Axis > "Drive" > "Parameters" tab



The output scaling of the velocity is only relevant for the CSV operation mode.

The calculation depends on whether you are using an incremental encoder:

- For operation with an incremental encoder:

$$output\ scaling = \frac{360}{increments \times 4} \times 125$$

  Example: for an encoder with 1024 increments per revolution, the output scaling is 10.98632813.

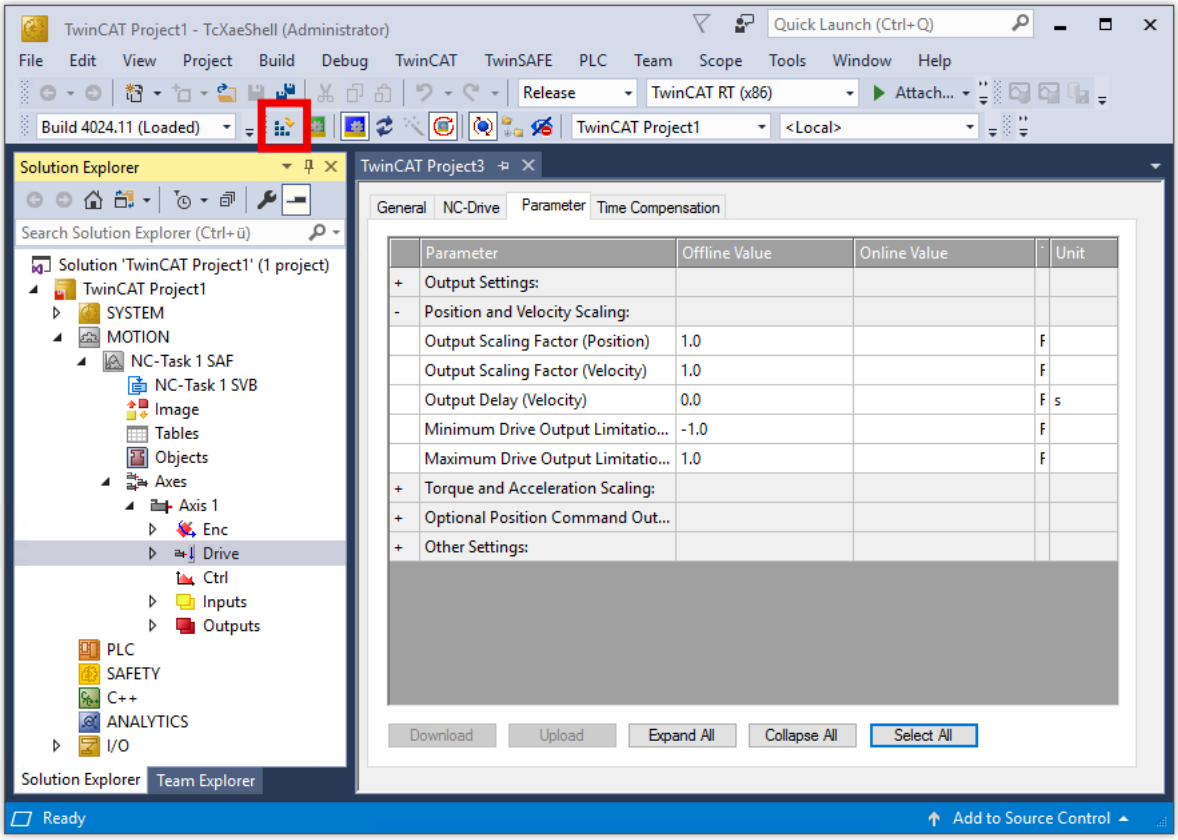
- For operation without incremental encoder:

$$output\ scaling = \frac{360}{65535} \times 125 = 0.6866455078$$

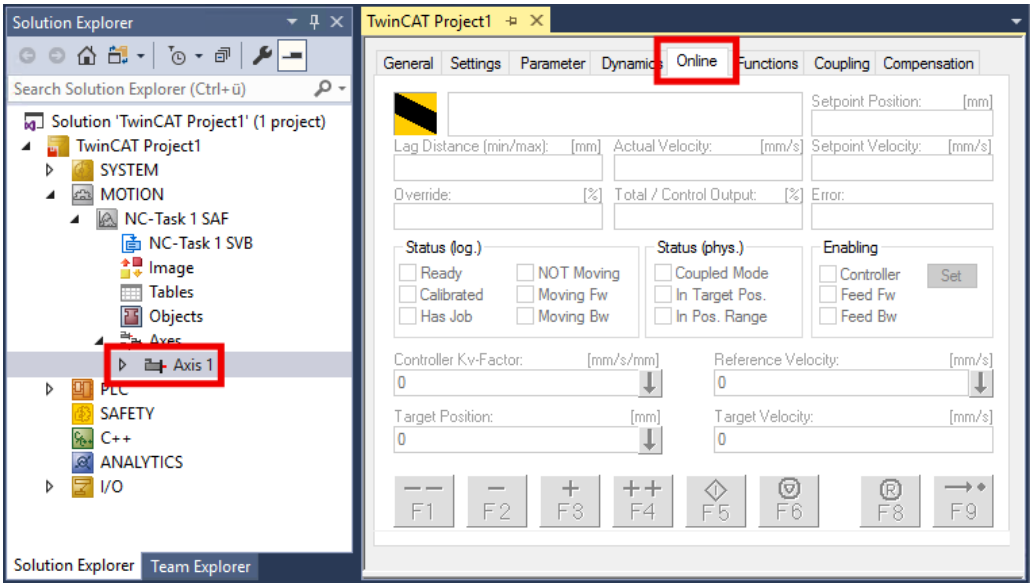
6.5.3 Execution of a test run

Step 1: Preparation

1. Activate configuration.



2. Click on the axis and select the "Online" tab.



3. Move the motor shaft by hand to check the configuration of the encoder.
   Does a relative movement of 360° actually correspond to a full revolution of the motor shaft?

   ⇨ If not: check the scaling factor. See chapter Configuration of the TwinCAT NC [▶ 118], section "Scaling Factor".

Step 2: Performing a test run

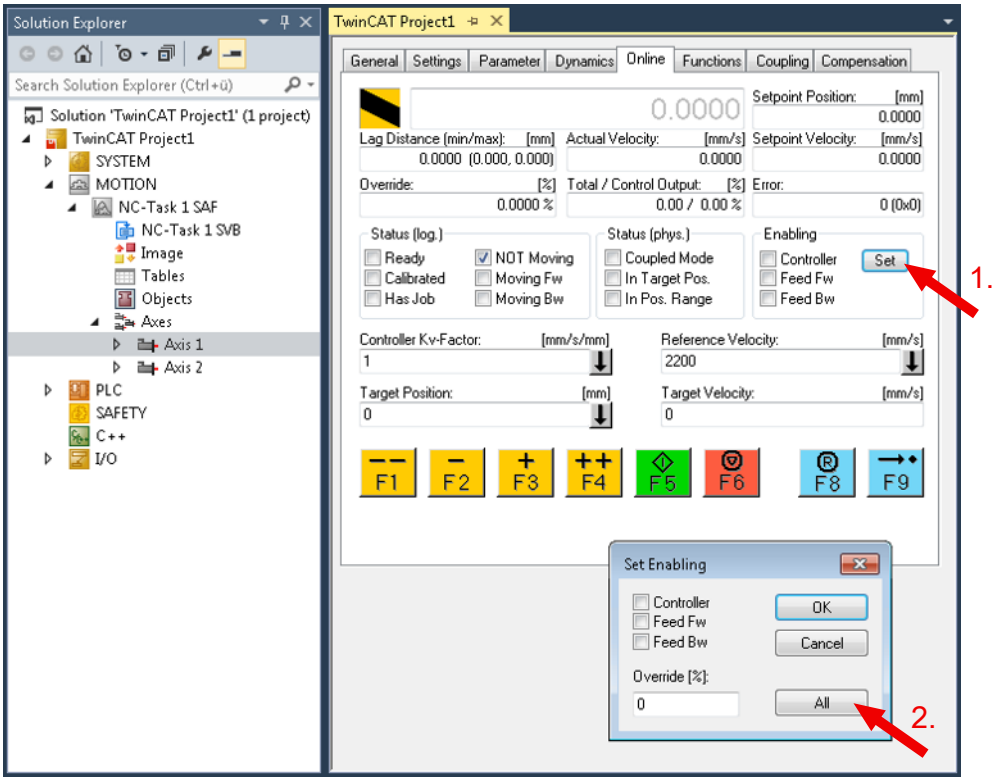
⚠ CAUTION

The motor may behave differently than expected

Injuries and damage to property are possible.

- Before the test run, ensure that no one is injured during any movements of the motor shaft and that no damage is caused.

1. Release the controller: click "Set" and then click "All" in the window that appears.



   ⇨ If the release fails: check whether 24 V are present at the input "HW Enable". See chapter Wiring [▶ 94].

2. Try different commands.
3. Check whether the motor follows the commands in a stable and reliable manner.

Step 3: Checking the result

Check the following points, for example:

- Were errors reported in the Diag History? See chapter Diag History [▶ 163].
- Is the motor turning in the expected direction?
- Does the speed match the preset?

6.5.4 Controller optimization

The parameters of the control loops are fundamentally preset when scanning the motor [▶ 110]. The parameters are optimized during the controller optimization.

Aims of the controller optimization:

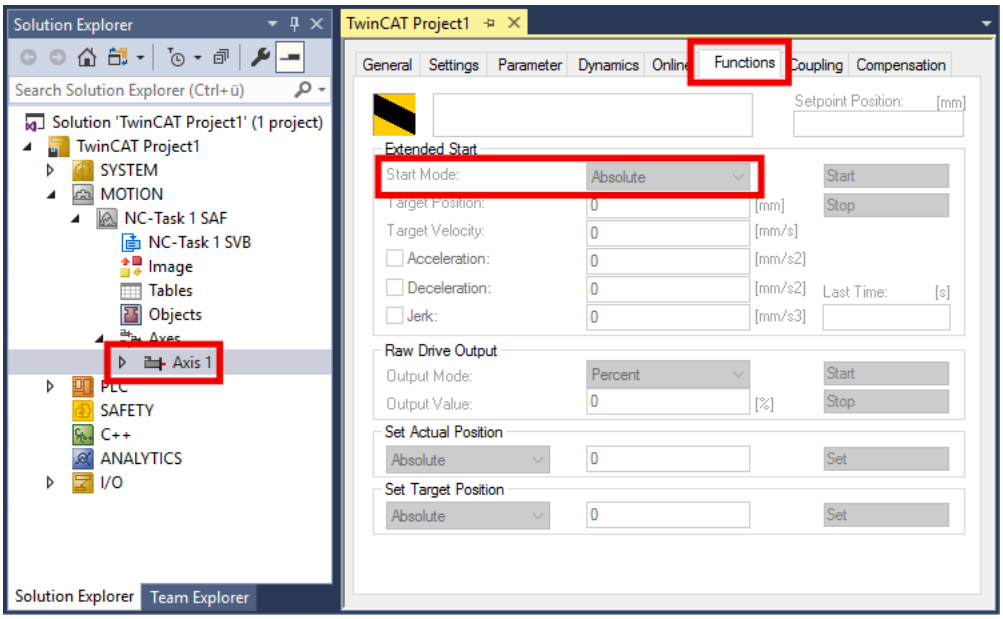
- Align the actual position with the target position: Minimize following errors.
- Aligning the actual velocity to the target velocity.
- Reduce overshoot and undershoot.

Observe the "Torque actual value" during controller optimization. This allows you to detect at an early stage whether the system is prone to oscillations.

Preparation

✓ Requirement: the test run [▶ 122] was successfully performed.

1. Establish load conditions as in the real application.
   Controller optimization without load is not useful, because the motor behavior is different in this case.
2. If necessary, increase the permissible following error. See chapter CSP (position control) [▶ 98].
3. Activate TwinCAT configuration.
4. Create a Scope project in TwinCAT.
5. Select the following variables from the "Target browser" to display them in the Scope:
   Position Lag
   0x1A03 Torque actual value
   From the TwinCAT NC:
   Position setpoint
   Position actual value
   Velocity setpoint
   Velocity actual value
6. In the Solution Explorer, click the axis.



7. Click the "Functions" tab.
8. In the drop-down menu "Start Mode": select "Reversing Sequence".
9. Adjust dynamics and velocity as needed in the real application.

The further procedure depends on which feedback is connected and configured:

- No feedback, sensorless operation [▶ 128]
- Hall sensors only [▶ 127]
- An incremental encoder [▶ 126] and optional Hall sensors

6.5.4.1 Feedback system: Incremental encoder

Step 1: Optimization of the current controller

The current controller is often sufficiently well adjusted by the Scan Motor [▶ 110] function.
If not, use conventional Ziegler/Nichols controller optimization methods.

Aim of current controller optimization: set the integral component as low as possible and set the proportional component as high as possible without achieving oscillating behavior.

- The integral component: parameter 8010:12 "Current loop integral time".
- The proportional component: parameter 8010:13 "Current loop proportional gain".

Step 2: Optimization of the velocity controller

✓ Requirement: the current controller is optimized.

1. Set the CoE parameter 8010:17 "Position loop proportional gain" to zero.
   ⇨ The position controller is disabled.
   ⇨ Interference of the position controller with the velocity controller is prevented.
2. Gradually reduce the integral component: parameter 8010:14 "Velocity loop integral time (current mode)".
   At the same time, gradually increase the proportional component: parameter 8010:15 "Velocity loop proportional gain (current mode)".
   While doing so, observe the actual velocity value.
   Do not increase the proportional component any further if the actual velocity value begins to oscillate.
3. Reduce the integral and proportional components by 20%.
   The 20% serves as a control reserve for abrupt movements.

⇨ The velocity controller is optimized.

Step 3: Optimization of the position controller

✓ Requirement: the velocity controller is optimized.

1. Gradually increase the proportional component 8010:17 "Position loop proportional gain" until the controller starts to oscillate.
2. Reduce the proportional component by 20%.

⇨ The position controller is optimized.

6.5.4.2 Feedback system: Hall sensors only

Step 1: Optimization of the velocity controller

1. Set the CoE parameter 8010:17 "Position loop proportional gain" to zero.
   ⇨ The position controller is disabled.
   ⇨ Interference of the position controller with the velocity controller is prevented.
2. Gradually reduce the integral component 8010:5A "Velocity loop integral time (voltage mode)".
   At the same time, gradually increase the proportional component 8010:5B "Velocity loop proportional gain (voltage mode)".
   While doing so, observe the actual value of the velocity and the actual value and the setpoint of the position.
   ⇨ From a certain point, the actual value of the velocity begins to oscillate or the difference between the actual value and the setpoint of the position increases.
3. Reduce the integral and proportional components by 20%. The 20% serves as a control reserve for abrupt movements.
4. If the actual value of the velocity overshoots, reduce 8010:5C "Velocity loop voltage feed forward gain (voltage mode)".

⇨ The velocity controller is optimized.

Step 2: Optimization of the position controller

✓ Requirement: The velocity controller is optimized.

1. Gradually increase the proportional component 8010:17 "Position loop proportional gain" so that the position difference oscillates around the value zero and the setpoint specification of the position is followed appropriately.

⇨ The position controller is optimized.

6.5.4.3 No feedback system: Sensorless operation

Without a feedback system, the following error (following error = actual position - target position) is much harder to optimize than when operating with Hall sensors and/or incremental encoders.

Step 1: Optimization of the velocity controller

1. Ensure that the voltage constant of the motor 8011:31 "Voltage constant" is set correctly.
2. Set the parameter 8010:60 "Sensorless max. acceleration" to a lower value, otherwise the velocity jump can become too big. e.g. 2000 °/s^2.
3. Reduce the parameter 8010:5E "Sensorless offset voltage scaling" to approx. 50 ... 80%.
4. Set the parameter 8010:17 "Position loop proportional gain" to 0.
   ⇨ The position controller is disabled.
   ⇨ Interference of the position controller with the velocity controller is prevented.
5. Set the parameter 8010:5B "Velocity loop proportional gain (voltage mode)" to 0.
6. Set the parameter 8010:5C "Velocity loop voltage feed forward gain (voltage mode)" to 100%.
7. Set the parameter 8010:5F "Sensorless observer bandwidth" to a lower value. e.g. 50 Hz.
8. Gradually increase the parameter 8010:5F "Sensorless observer bandwidth" until oscillation occurs. Then reduce by 50%.
9. Configure the integral component of the velocity controller in parameter 8010:5A "Velocity loop integral time (voltage mode)" to be rather sluggish.
10. Gradually increase the proportional component 8010:5B "Velocity loop proportional gain (voltage mode)" until the actual velocity in the scope begins to oscillate.
11. Reduce the proportional component by 20%.
    The 20% serves as a control reserve for abrupt movements.
12. If the velocity overshoots: slightly reduce the parameter 8010:5C "Velocity loop voltage feed forward gain (voltage mode)".
13. If necessary, increase the parameter 8010:60 "Sensorless max. acceleration" back to the required dynamics.

⇨ The velocity controller is optimized.

Step 2: Optimization of the position controller

✓ Requirement: the velocity controller is optimized.

1. Gradually increase the proportional component 8010:17 "Position loop proportional gain" until oscillation occurs.
2. Reduce the proportional component by 20%.
   The 20% serves as a control reserve for abrupt movements.

⇨ The position controller is optimized.

6.5.5 Homing

This chapter describes how to perform homing with an NC axis.

Homing is started from the PLC with the "MC_Home" function block. "MC_Home" is located in the TwinCAT library Tc2_MC2.
Further information can be found in the documentation Tc2_MC2.

For homing, a referencing cam is required that generates a digital switching signal at a defined position on the travel path. During homing, this defined position is determined and stored as the reference position for absolute positioning.

The configuration procedure depends on how you transmit the switching signal to the TwinCAT NC. The following options are available:

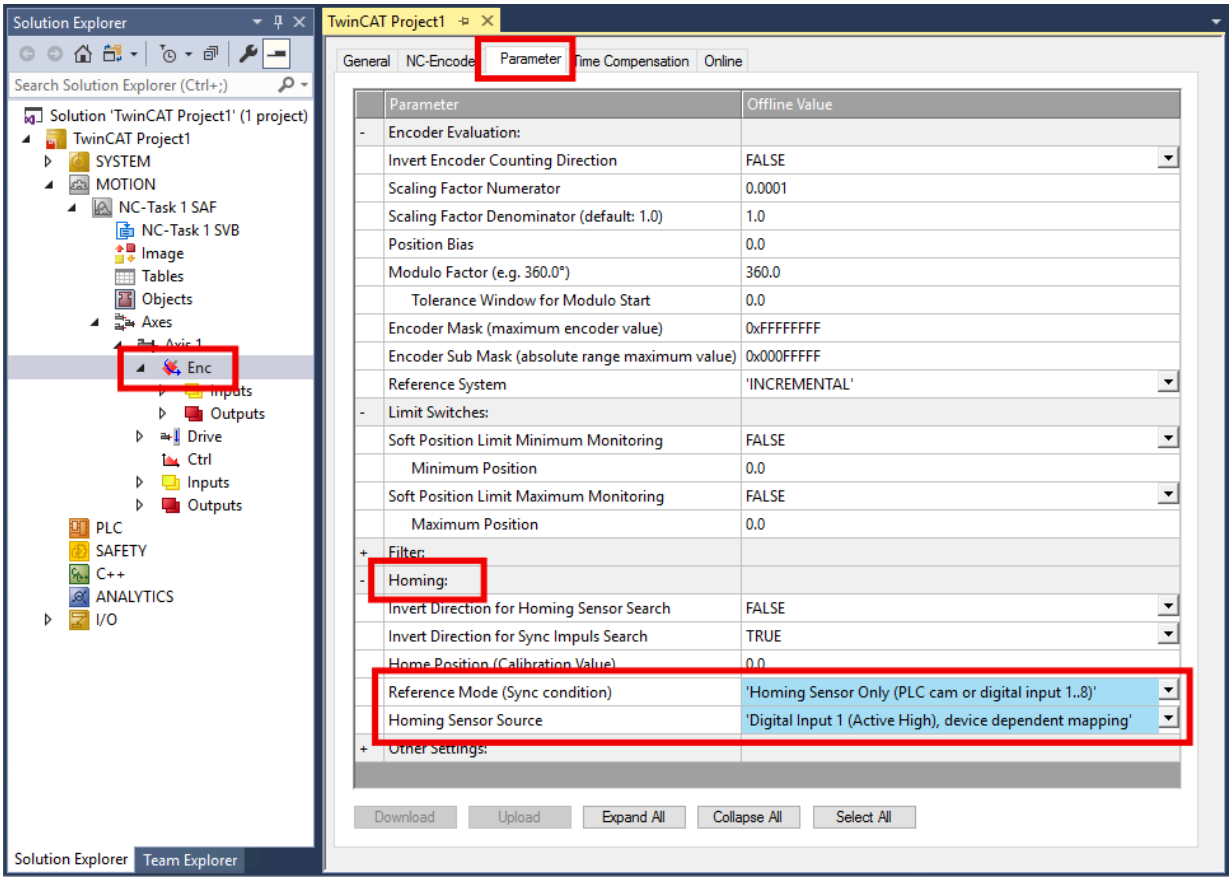
- Applying a 24 V switching signal to a digital input of EL7411. Configuration [▶ 129]
- Transmit any signal from the PLC.
  For this option, the referencing cam does not have to be physically present. It can also be simulated via a digital signal. Configuration [▶ 133]

In addition to one of these options, you can use the zero pulse on the C track of the encoder for referencing. Configuration [▶ 135]

6.5.5.1 Switching signal from a digital input of the EL7411

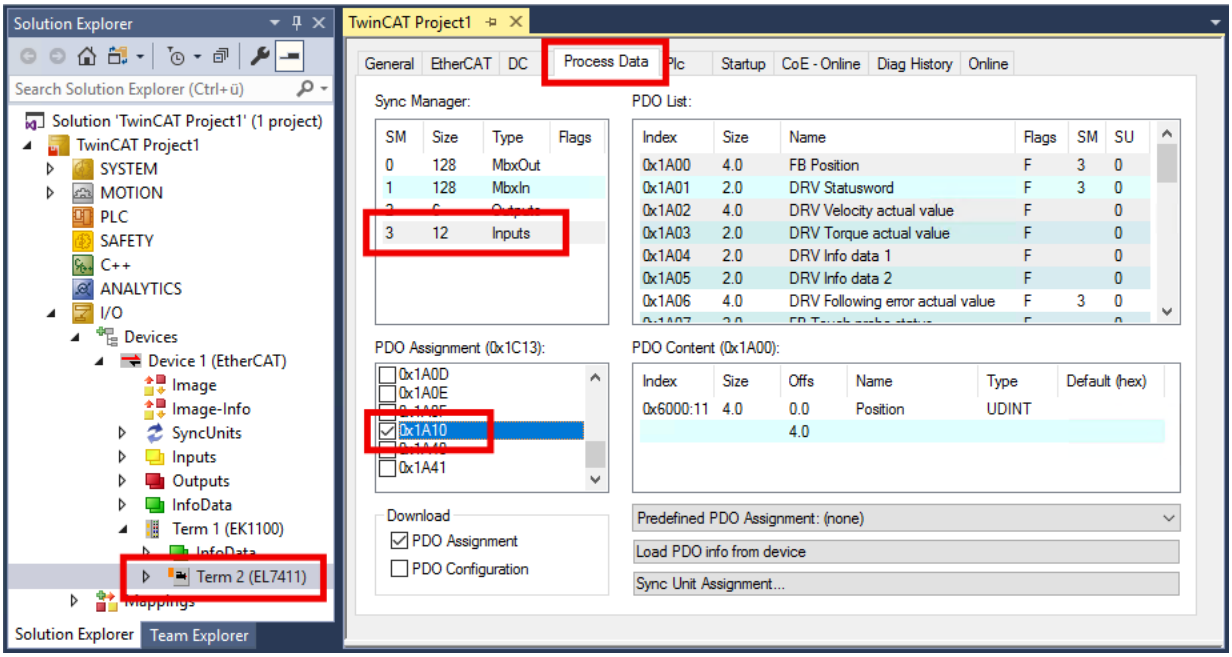
This chapter describes the configuration exemplarily for the digital input "Input 1".

1. In the Solution Explorer:
   click on NC axis > "ENC", open the tab "Parameters" and expand the section "Homing".

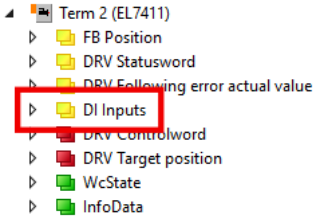


2. Set the parameter "Reference Mode" to "Homing Sensor Only (PLC cam or digital input 1..8)".
3. Set the parameter "Homing Sensor Source" to "Digital Input 1 (Active High), device depending mapping".

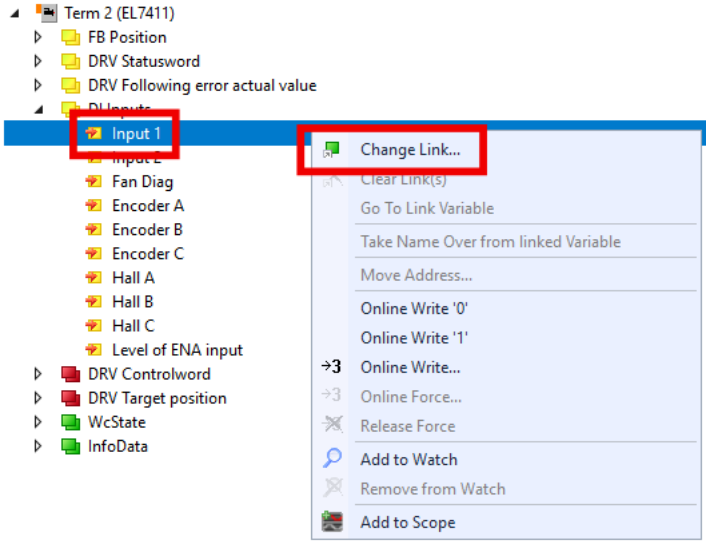
4. In the Solution Explorer:
   click on the EL7411, open the tab "Process Data", select the Sync Manager 3 "Inputs" and activate the PDO Assignment 0x1A10.



⇨ The process data object "DI Inputs" is activated.

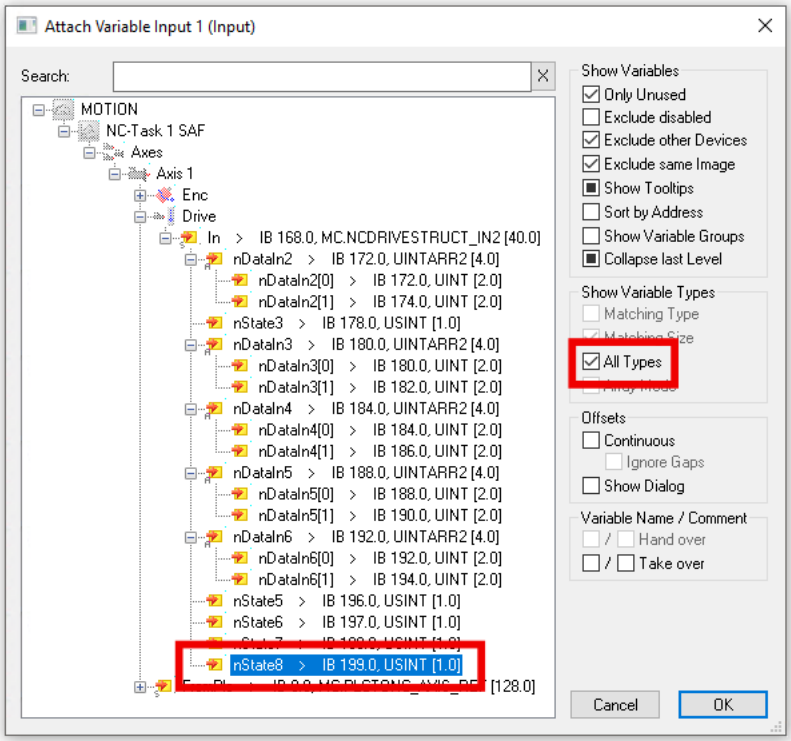


5. EL7411 > "DI Inputs" > "Input 1" right click, "Change Link" select.

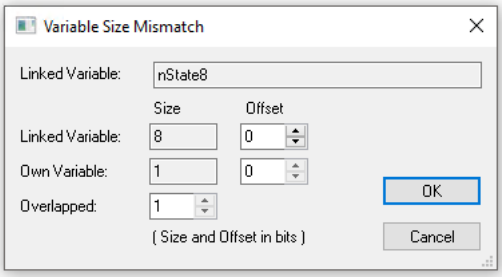


⇨ A dialog box appears.

6. click on "All Types", select the variable "nState8" and confirm with "OK".

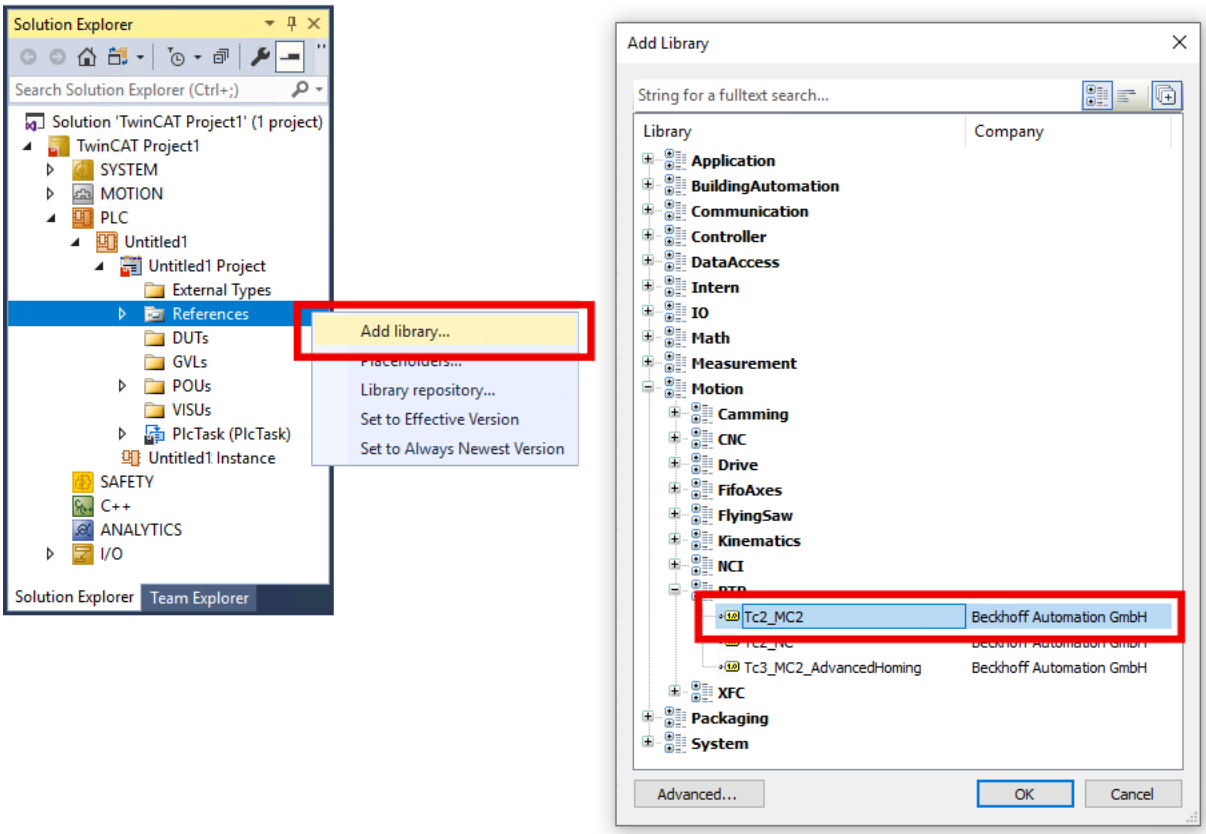


⇨ A dialog box appears.



7. Confirm with "OK".

8. Add the library "Tc2_MC2" to the PLC project:
   Navigate to PLC > "References", right click, "Add Library..."

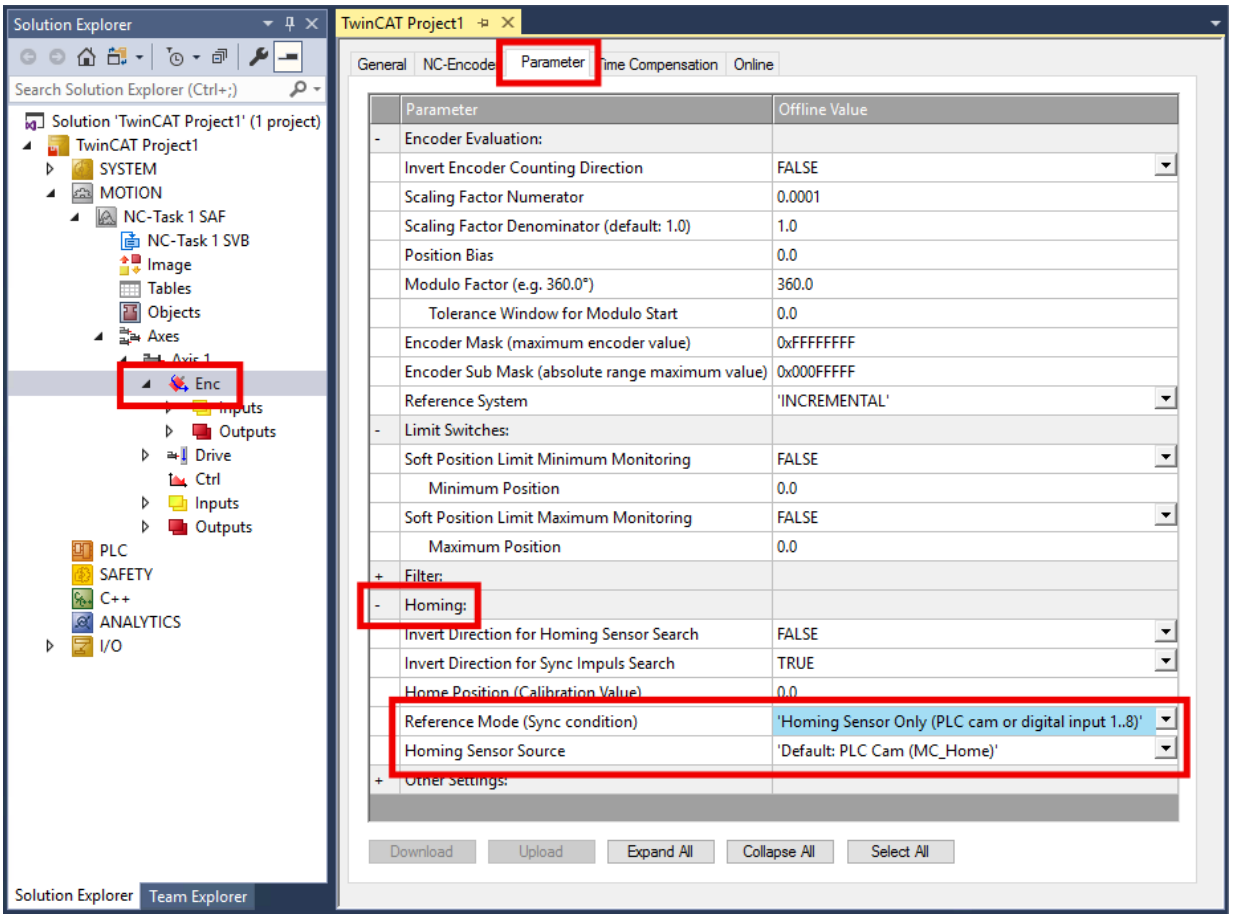


9. Insert an instance of the function block "MC_Home" from the library "Tc2_MC2" in the PLC.
   Do not connect the input "bCalibrationCam".

⇨ You can start homing with a positive edge at input "Execute".

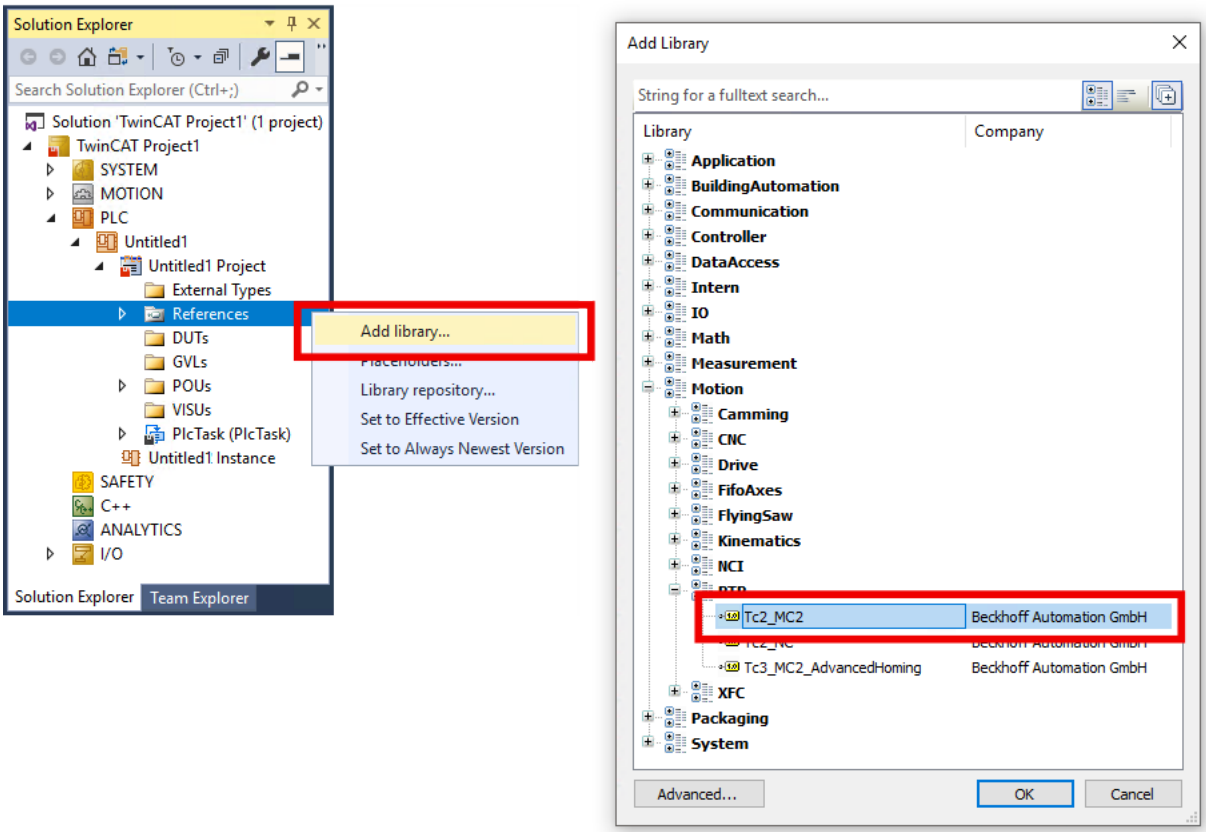
6.5.5.2 Switching signal from the PLC (PLC Camming)

1. In the Solution Explorer:
   click on NC axis > "ENC", open the tab "Parameters" and expand the section "Homing".



2. Set the parameter "Reference Mode (Sync condition)" to "Homing Sensor Only (PLC cam or digital input 1..8)".
3. Set the parameter "Homing Sensor Source" to "Default: PLC Cam (MC_Home)".

4. Insert the library "Tc2_MC2" into the PLC project:
   PLC > References, right click, "Add Library..."



5. Insert an instance of the function block "MC_Home" from "Tc2_MC2" in the PLC.
6. Apply the signal of the reference switch to the "bCalibrationCam" input of the function block.

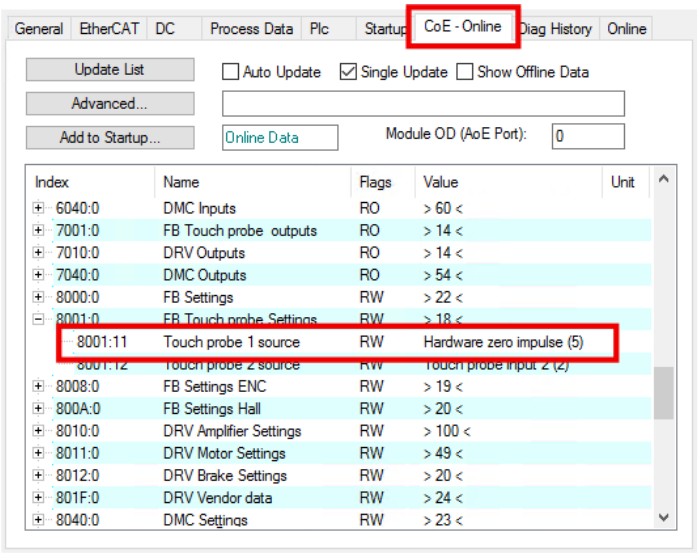
⇨ You can start homing with a positive edge at input "Execute".

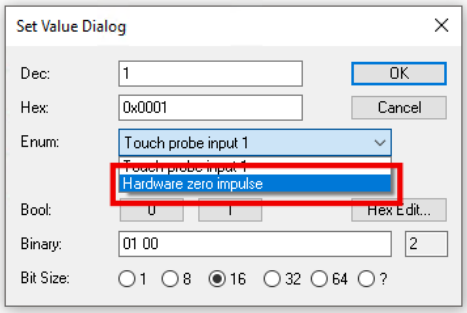
6.5.5.3 Zero pulse from the C track of the encoder

Referencing to the zero pulse of the encoder is only possible in addition to referencing to a referencing cam.

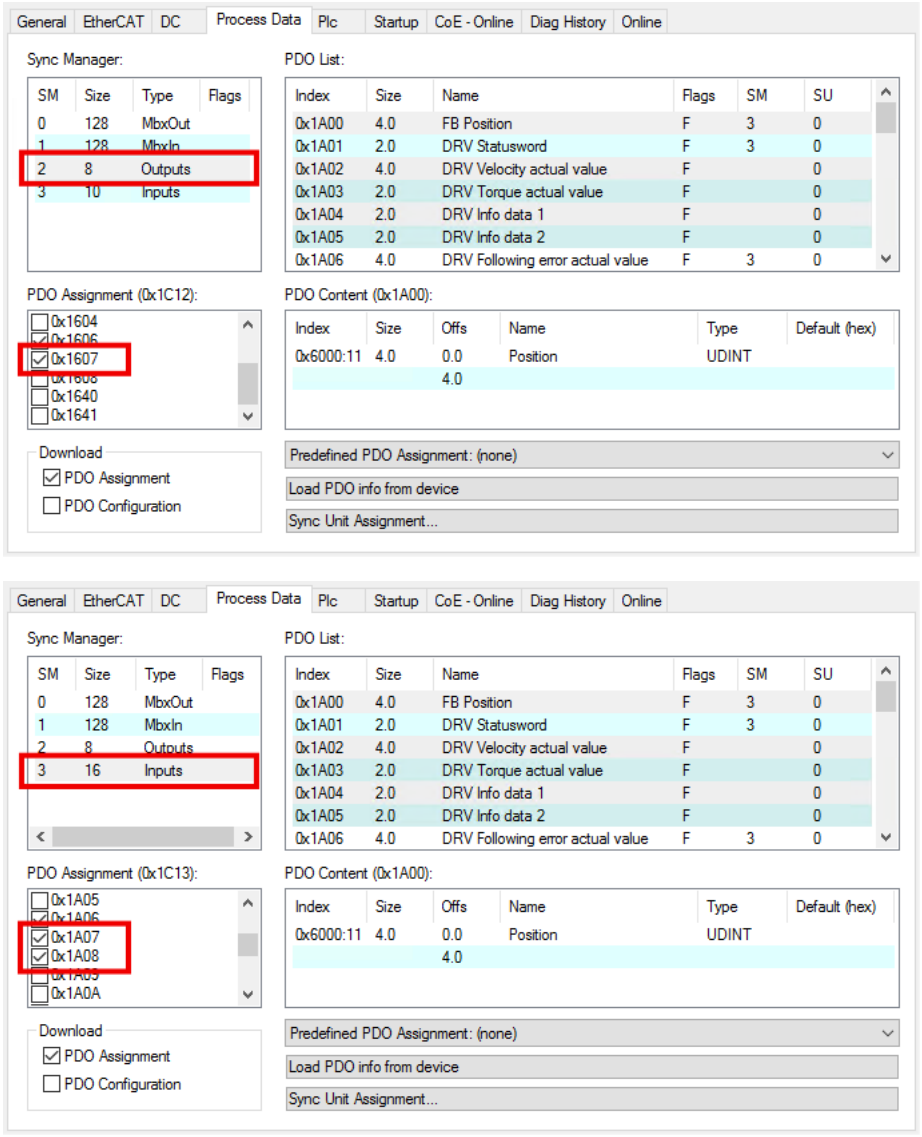
To ensure that the motor does not stop until the zero pulse is received after it has moved down from the referencing cam, configure the terminal as follows:

1. Configure a homing with referencing cams:
   Switching signal from a digital input of the EL7411 [▶ 129]
   or
   Switching signal from the PLC (PLC Camming) [▶ 133]
2. Set the CoE parameter 8001:11 "Touch probe 1 source" to "Hardware zero impulse".

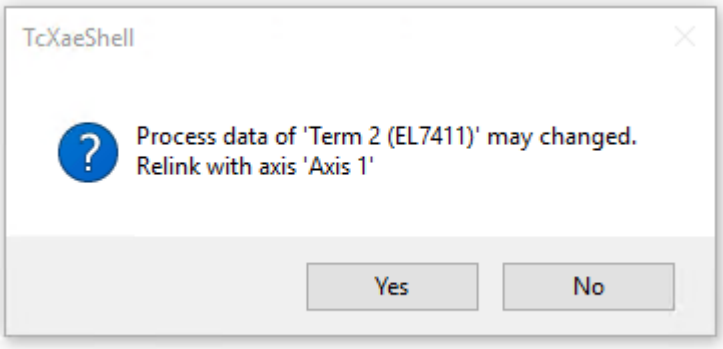




3. Activate the process data objects for the function "Touch Probe":
   0x1607, 0x1A07, 0x1A08



⇨ A dialog box appears after each activation of a process data object.



4. Confirm each dialog box with "Yes".

6.6 Commissioning with status word and control word

The operation modes CST, CSTCA, CSV and CSP can be used without TwinCAT NC.

Output stage enabled via the control word (index 0x7010:01)

The output stage has to be enabled for each operation mode. To do this, the values specified in the following table at *Enable output stage* must be entered via the PLC in the control word in the specified order (1. - 5.) (according to the definition for the state machine see Fig. DS402 State Machine [▶ 139]).

The bits of the high byte (CW.8 - CW.15) are reserved and not relevant for enabling the output stage.

Bit	Low-Byte							
	CW.7	CW.6	CW.5	CW.4	CW.3	CW.2	CW.1	CW.0
Name	Fault reset	Reserved	Reserved	Reserved	Enable operation	Reserved	Enable voltage	Switch on
Enable output stage:								
1. 0_{hex} (Start)	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
2. 80_{hex} (Fault reset)	**1**	x	x	x	x	x	x	x
3. 6_{hex} (Shutdown)	**0**	x	x	x	x	**1**	**1**	**0**
4. 7_{hex} (Switch on)	**0**	x	x	x	**0**	**1**	**1**	**1**
5. F_{hex} (Enable operation)	**0**	x	x	**0**	**1**	**1**	**1**	**1**
Disable voltage:								
Disable voltage	**0**	x	x	x	x	x	**0**	x

Fig. 96: Low byte of the control word (x: state of the bit is not relevant)

For the bits named "Reserved" further functions are defined according to the specifications for the state machine, which are not supported by the EL7411 (e. g. CW.2: "Quick stop (inverse)").

Checking the individual steps via the status word (0x6010:01)

The respective status messages are output in the status word.

Checking the status word mandatory

It is mandatory to check after each input in the control word whether the internal state machine has followed the requirements of the control word (see also Fig. DS402 State Machine [▶ 139]).

- Check the status word after each step (see following table) and wait for the status change if necessary!

To enable the output stage, check whether the corresponding status messages 1. - 5. (*Enable output stage*) of the following table are displayed.

Bit	High-Byte							
	SW.15	SW.14	SW.13	SW.12	SW.11	SW.10	SW.9	SW.8
Name	Reserved	Reserved	Reserved	Drive follows the command value	Reserved	Reserved	Reserved	Reserved
Enable output stage:								
1. Not ready to switch	x	x	x	**0**	x	x	x	x
2. Switch on disabled	x	x	x	**0**	x	x	x	x
3. Ready to switch on	x	x	x	**0**	x	x	x	x
4. Switched on	x	x	x	**0**	x	x	x	x
5. Operation enabled	x	x	x	**1**	x	x	x	x
Fault reaction:								
Fault reaction active	x	x	x	**0**	x	x	x	x
Fault	x	x	x	**0**	x	x	x	x

Bit	Low-Byte							
	SW.7	SW.6	SW.5	SW.4	SW.3	SW.2	SW.1	SW.0
Name	Reserved	Switch on disabled	Reserved	Reserved	Fault	Operation enabled	Switched on	Ready to switch on
Enable output stage:								
1. Not ready to switch	x	**0**	x	x	**0**	**0**	**0**	**0**
2. Switch on disabled	x	**1**	x	x	**0**	**0**	**0**	**0**
3. Ready to switch on	x	**0**	**1**	x	**0**	**0**	**0**	**1**
4. Switched on	x	**0**	**1**	x	**0**	**0**	**1**	**1**
5. Operation enabled	x	**0**	**1**	x	**0**	**1**	**1**	**1**
Fault reaction:								
Fault reaction active	x	**0**	x	x	**1**	**1**	**1**	**1**
Fault	x	**0**	x	x	**1**	**0**	**0**	**0**

Fig. 97: Status word (x: state of the bit is not relevant)

For the bits named "Reserved" further status messages are defined according to the specifications for the state machine, which are not supported by the EL7411 (e.g. SW.5: "Quick stop (inverse)").

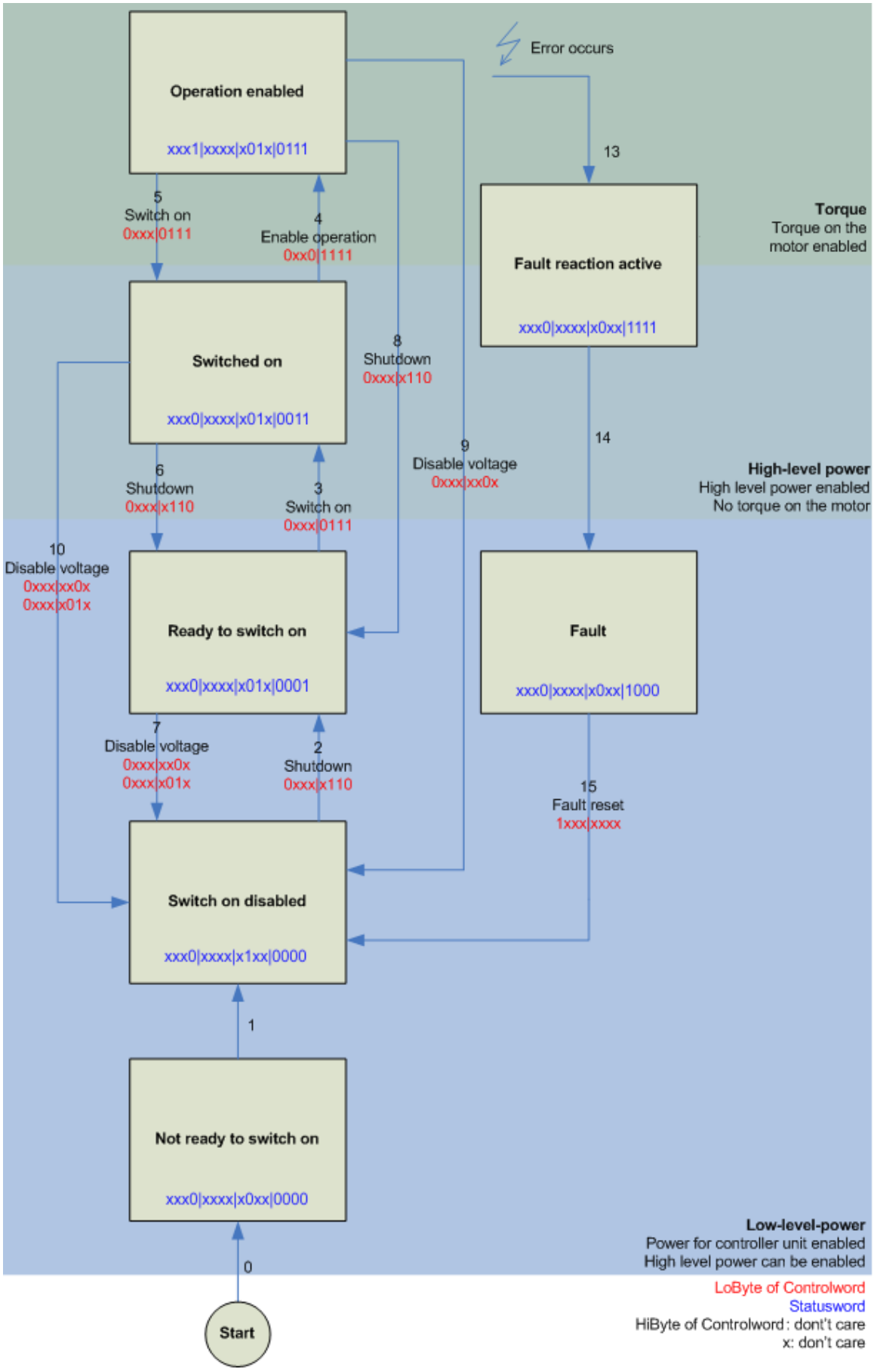


Fig. 98: DS402 State Machine

CST - cyclic synchronous torque

Select "Cyclic synchronous torque mode" in index 0x7010:03 "Modes of operation". In the respective process data, the Predefined PDO Assignment "Torque" should also be selected. Afterwards the configuration must be reloaded to accept the selection.
Under the index 0x6010:03 "Modes of operation display" it can be checked in which mode the EL7411 actually is.

Via the PLC a defined torque can be defined in the variable "Target torque" as a basis for the EL7411 control. The torque is specified in 1000th of the nominal current. A value of 1000_{dec}, for example, corresponds to the set index 0x8011:12 "Rated current". The value 1_{dec} corresponds to one 1000th of the nominal current.

CSTCA - cyclic synchronous torque with commutation angle

Select "Cyclic synchronous torque mode with commutation angle" in index 0x7010:03 "Modes of operation". In the respective process data, the Predefined PDO Assignment "Torque" should also be selected. Afterwards the process data 0x1603 "DRV Commutation angle" can be added and the configuration must be reloaded to accept the selection.
Under the index 0x6010:03 "Modes of operation display" it can be checked in which mode the EL7411 actually is.

Via the PLC a defined torque can be set in the "Target torque" variable as a basis for the EL7411 control. In the "Commutation angle" variable the angle to be maintained with the set torque can be specified. The torque is specified in 1000th of the nominal current. A value of 1000_{dec}, for example, corresponds to the set index 0x8011:12 "Rated current". The value 1_{dec} corresponds to one 1000th of the nominal current. The angle value must be converted, 65536_{dec} corresponds to an electrical angle of 360°.

CSV - cyclic synchronous velocity

Select "Cyclic synchronous velocity" in index 0x7010:03 "Modes of operation". In the respective process data, the Predefined PDO Assignment "Velocity" should also be selected. The configuration must then be reloaded to accept the selection.
Under the index 0x6010:03 "Modes of operation display" it can be checked in which mode the terminal actually is.

Via the PLC a defined speed can be set in the variable "Target velocity" 0x7010:06 as a basis for the terminal control. The constant value "Velocity encoder resolution" in CoE object 0x9010:14 corresponds to 1 revolution per second. If this value is entered under "Target velocity", the motor speed is 1 rpm. The velocity can be increased by entering a suitable multiple of the "Velocity encoder resolution" value under "Target velocity".

CSP - cyclic synchronous position

Select "Cyclic synchronous position" in index 0x7010:03 "Modes of operation". In the respective process data, the Predefined PDO Assignment "Position" should also be selected. The configuration must then be reloaded to accept the selection.
Under the index 0x6010:03 "Modes of operation display" it can be checked in which mode the servo terminal actually is.

Via the PLC a defined position can be set in the variable "Target position" 0x7010:05 to which the motor is to drive. The unit is increments. Depending on the feedback set, the number of increments per revolution is based, for example, on the number of pulses of the incremental encoder per revolution or the Hall sensor resolution of 65535 increments per revolution.

6.7 Commissioning with Drive Motion Control

With Drive Motion Control you can implement a travel distance control without TwinCAT NC.

The TwinCAT NC requires an EtherCAT master that supports Distributed Clocks. A possible use case for Drive Motion Control is therefore the operation of an EL7411 on a controller that does not support Distributed Clocks.

The documentation of the PLC library for Drive Motion Control can be found on the Beckhoff website: Tc3_DriveMotionControl.

6.7.1 Requirements

- TwinCAT 3.1, Build 4024.7 or higher
- EL7411-00x0 with firmware 06 and ESI revision 0019 or higher

6.7.2 Functionality

6.7.2.1 Supported functions

Administrative functions

- Axis functions
  - MC_Power
  - MC_Reset
  - MC_SetPosition
- Touch probe
  - MC_AbortTrigger
  - MC_TouchProbe

Motion functions

- Homing
  - MC_Home (Here the bCalibrationCam input of the Tc2_Mc2 library can not be used, but one of the digital inputs of the EL7411 must be used)
- Manual movement
  - MC_Jog
- Point to point movement
  - MC_Halt
  - MC_MoveAbsolute
  - MC_MoveModulo
  - MC_MoveRelative
  - MC_MoveVelocity
  - MC_Stop

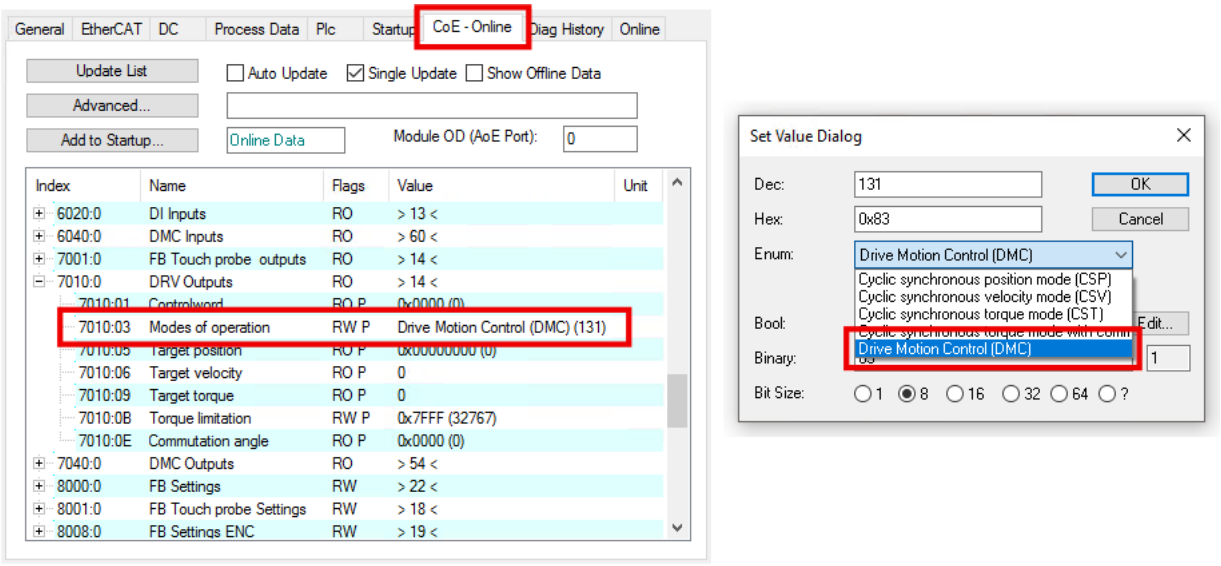
6.7.2.2 Functions that are not supported

All subsequently triggered functions with the aim of changing the target position or speed during an active travel command are not supported (buffer mode).

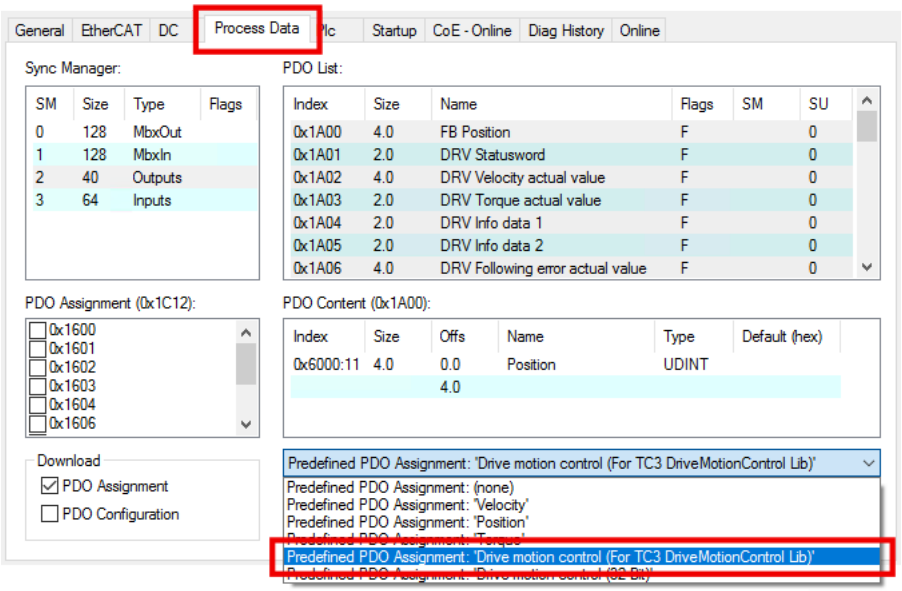
6.7.3 Commissioning in TwinCAT 3

With the following steps you configure an EL7411 in TwinCAT 3 for the operation with Drive Motion Control.

1. In the CoE parameter 0x7010:03 "Modes of operation" set the operation mode "Drive Motion Control (DMC)".



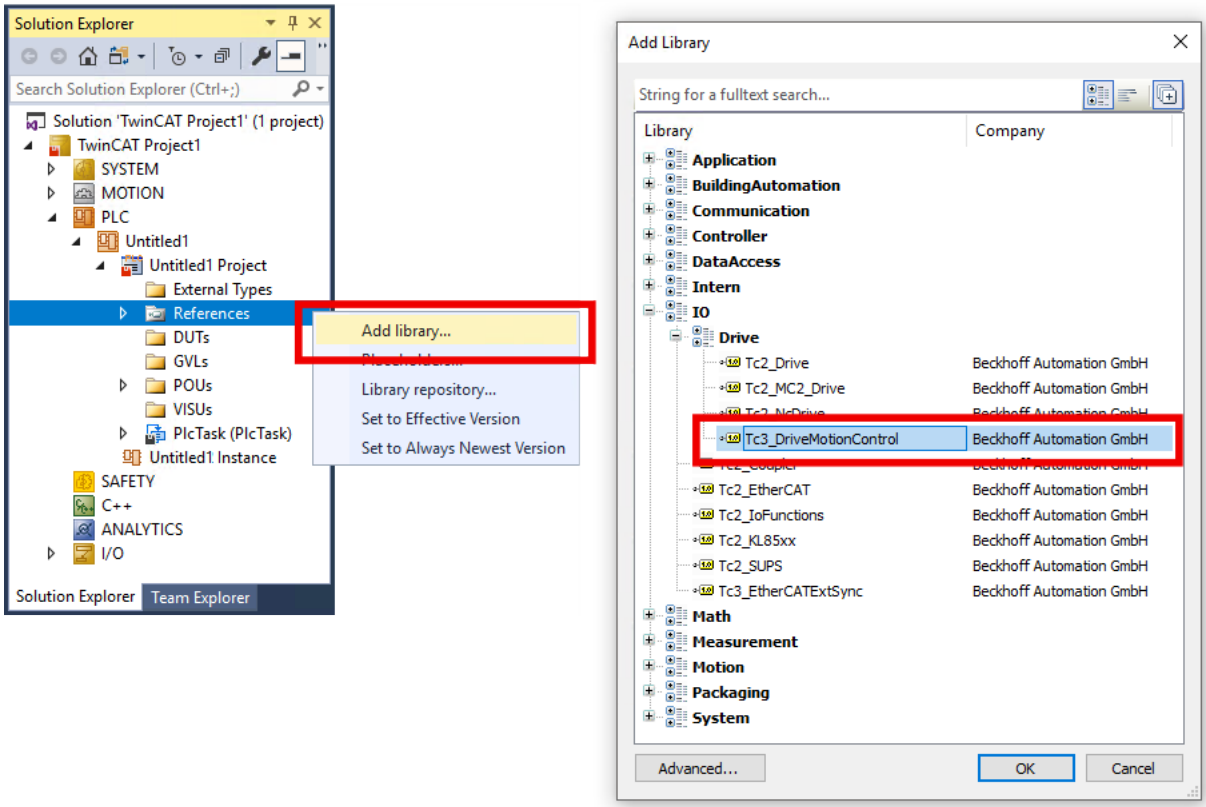
2. Activate the Predefined PDO Assignment "Drive motion control (For TC3 DriveMotionControl Lib)".



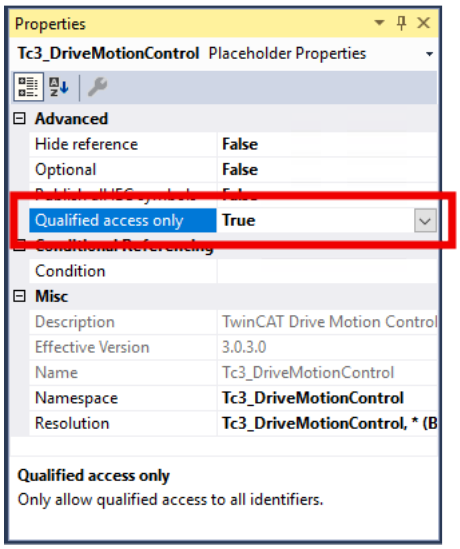
⇨ The process data for using Drive Motion Control is activated.

3. Create a PLC project if none has been created yet.

4. Add the library "Tc3_DriveMotionControl" to the PLC project.
   (If you want to use Drive Motion Control without the library "Tc3_DriveMotionControl", see chapter State machine [▶ 149].)



5. If the libraries "Tc3_DriveMotionControl" and "Tc2_Mc2" are used simultaneously in the current project:
   In the window "Properties" of one of the two libraries, set the property "Qualified access only" to "True".



   ⇨ The library with "Qualified access only" can now only be addressed in the PLC code via the corresponding namespace, e.g.: `Tc3_DriveMotionControl.MC_Stop()`
   This avoids name conflicts, e.g. when calling the function block `MC_Stop`, which has the same name in both libraries.

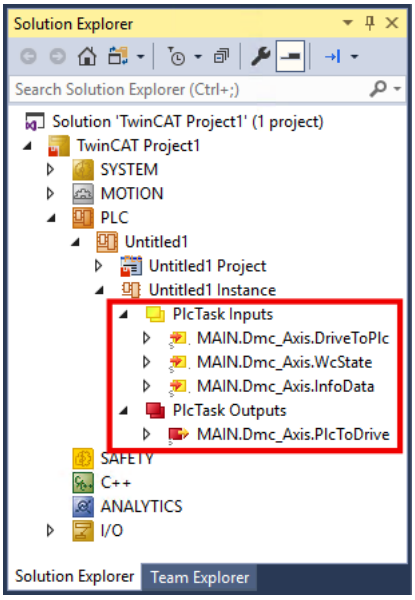
6. Declare a variable of type "AXIS_REF" in the PLC. Sample:

```
VAR
    DmcAxis:    AXIS_REF;
END_VAR
```

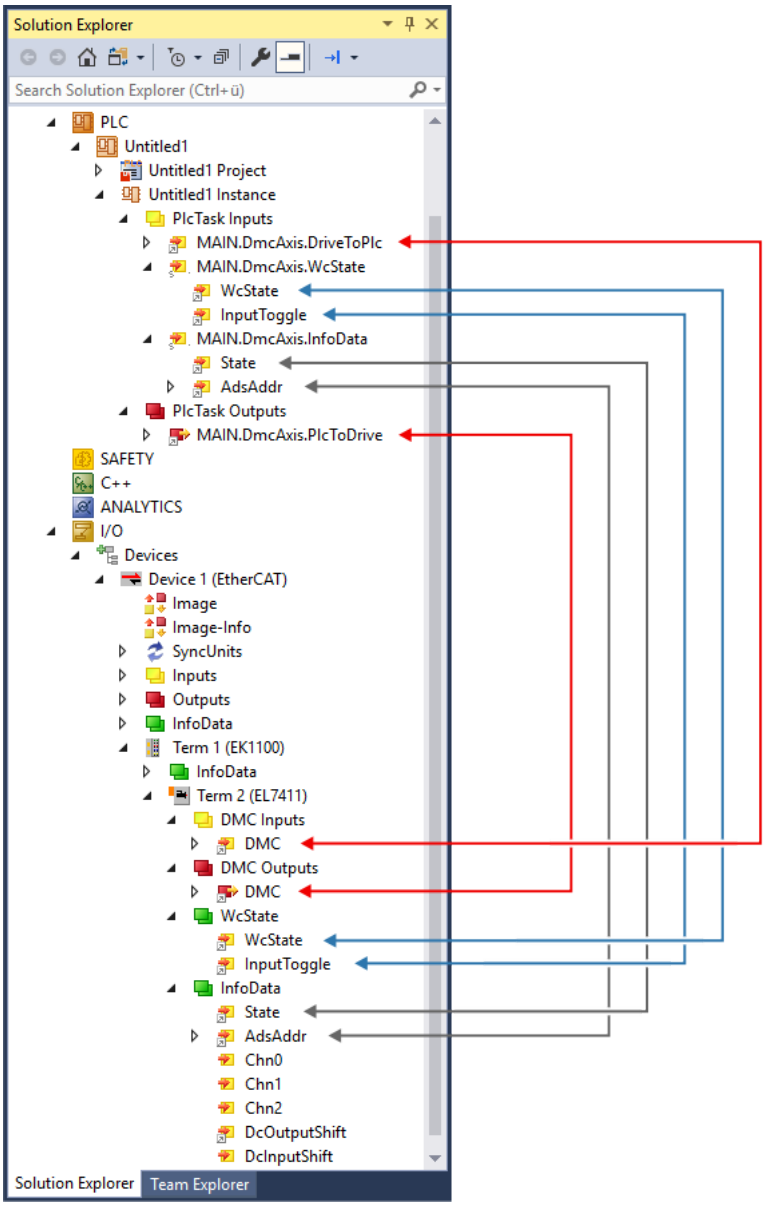
7. Click on "Build" > "Build Solution" in the menu bar.

⇨ The project is compiled.

⇨ The process image of the PLC task is generated.



8. In the Solution Explorer link the PLC variables with the process data of EL7411.



9. In the PLC code, call the function block `ReadStatus()` cyclically, ideally at the beginning of each PLC cycle.

6.7.3.1 Parameter

CoE parameters

The CoE parameters for configuring Drive Motion Control are located in the following CoE objects:

- 0x8040 "DMC Settings" [▶ 175]
- 0x8041 "DMC Features" [▶ 176]

Scaling factor and maximum velocity

Position values are defined as 64-bit variables at Drive Motion Control.
The 32 low-order bits resolve the single-turn increments. The possibly lower resolution of the feedback is extrapolated to the full 32 bits.
The 32 higher-order bits represent the multi-turn revolutions.

The "Feed constant" includes any gear ratios (gearbox, belt, etc.) and represents the output-side travel per motor revolution.

Therefore, the following exemplary formula without transmission ratio results for the scaling factor:

$$Encoder\ Scaling\ Factor = \frac{Feed\ constant}{32\ Bit} = \frac{360°}{32\ Bit} \approx 8{,}3819031715393066e-8$$

The maximum achievable speed of the motor depends on the DC link voltage. If lower voltages than specified in the data sheet are used, the nominal speed may have to be adjusted to the voltage. To specify the maximum speed of the motor in the CoE directory, the object 0x8011:1B Motor "Motor speed limitation" is also used. The DC link-dependent motor speed is specified here in 1/min. To adjust the speed of the scaling, this value is multiplied by the feed constant and normalized to the unit second. This results in the following formula for the exemplary calculation of the maximum speed:

$$MaxVelocity = \frac{Motor\ speed\ limitation\ \cdot Feed\ constant}{60\frac{s}{min}} = \frac{1000\frac{U}{min}\cdot 360°}{60\frac{s}{min}} = 6000\frac{°}{s}$$

The following example shows the implementation in a PLC program:

```
PROGRAM MAIN
VAR
   DmcAxis:   AXIS_REF
END_VAR


// Update the axis structure
DmcAxis.ReadStatus();

// Scaling factor without gear ratio, feed constant 360°
DmcAxis.Parameter.EncoderScalingFactor := 0.00000008381903171539306 6;

// Velocity scaling with 1000 rpm, feed constant 360°
DmcAxis.Parameter.MaxVelocity := 6000;
```

6.7.4 Commissioning with a third-party 32-bit controller

The "Tc3_DriveMotionControl" library cannot be used.

You can only execute travel commands by going through the state machine manually. See chapter State machine [▶ 149].

Since the terminal provides 64-bit process data by default, but this cannot be processed by some controllers, there is also the alternative option of mapping the process image with 32 bits. This can be set via the Predefined PDO Assignment "Drive motion control (32-bit)".
(To map the inputs and outputs manually, use the 0x1641 and 0x1A41 indices. See chapter Process data [▶ 158].)

All position related process data are 32-bit instead of 64-bit with this Predefined PDO Assignment. The 32 bits are divided into 20 bit single-turn and 12 bit multi-turn revolutions, independent of the resolution of the feedback.
The terminal still calculates internally with 64-bit data. Therefore e.g. 0x8040:08 "Calibration Position" must still be specified in 32-bit single-turn and 32-bit multi-turn instead of 20-bit single-turn and 12-bit multi-turn.

It should be noted that the EL7411 only supports incremental encoders and not absolute encoders. Therefore, the position data is not retained beyond a power cycle.

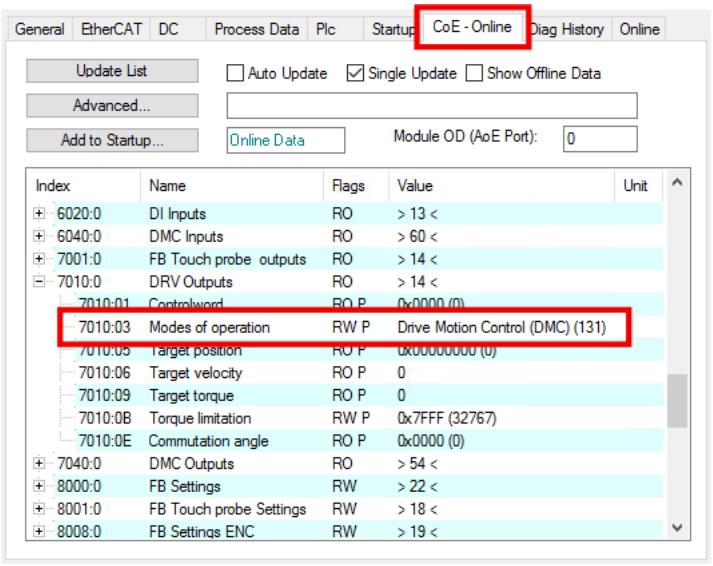
All non-position related process data remain unchanged in size. The address offsets of the individual process data are identical and padding bytes are inserted at the corresponding positions.

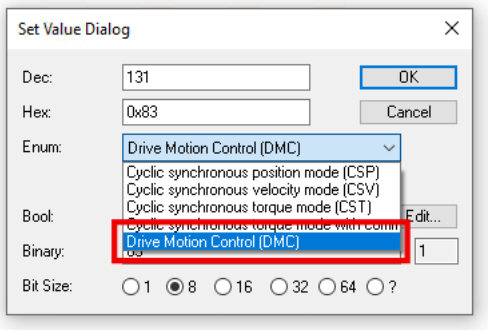
The speed related process data is scaled in 10,000ths of the parameter 0x8011:1B "Motor speed limitation".

The process data for acceleration and deceleration specify in ms how fast the motor should accelerate to the speed specified in 0x8011:1B "Motor speed limitation" or decelerate from speed to standstill. With a value of 2000 for the acceleration, the motor would need 2 s to reach the speed.

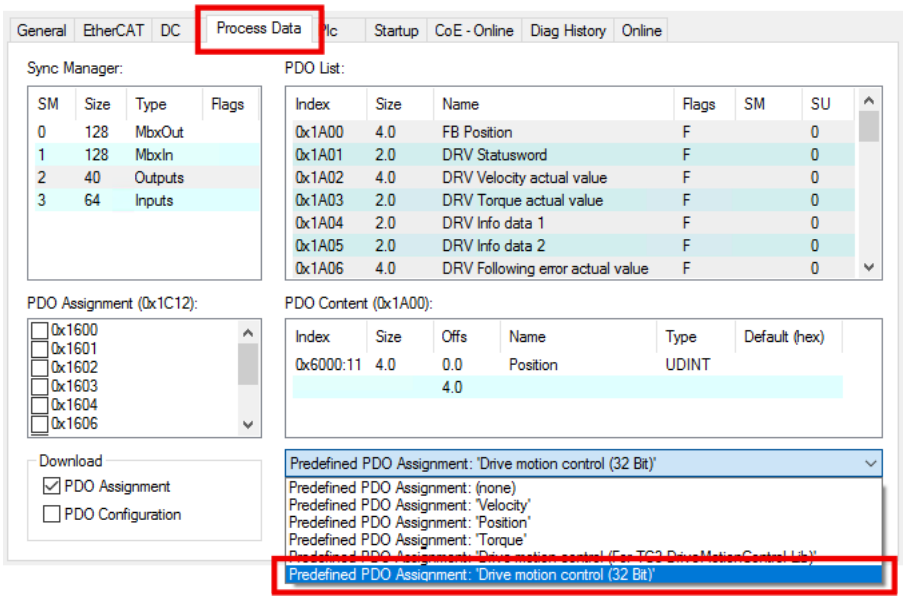
Configuration

1. In the CoE parameter 0x7010:03 "Modes of operation" set the operation mode "Drive Motion Control (DMC)".





2. Activate the Predefined PDO Assignment "Drive motion control (32 Bit)".



⇨ The process data for using Drive Motion Control is activated.

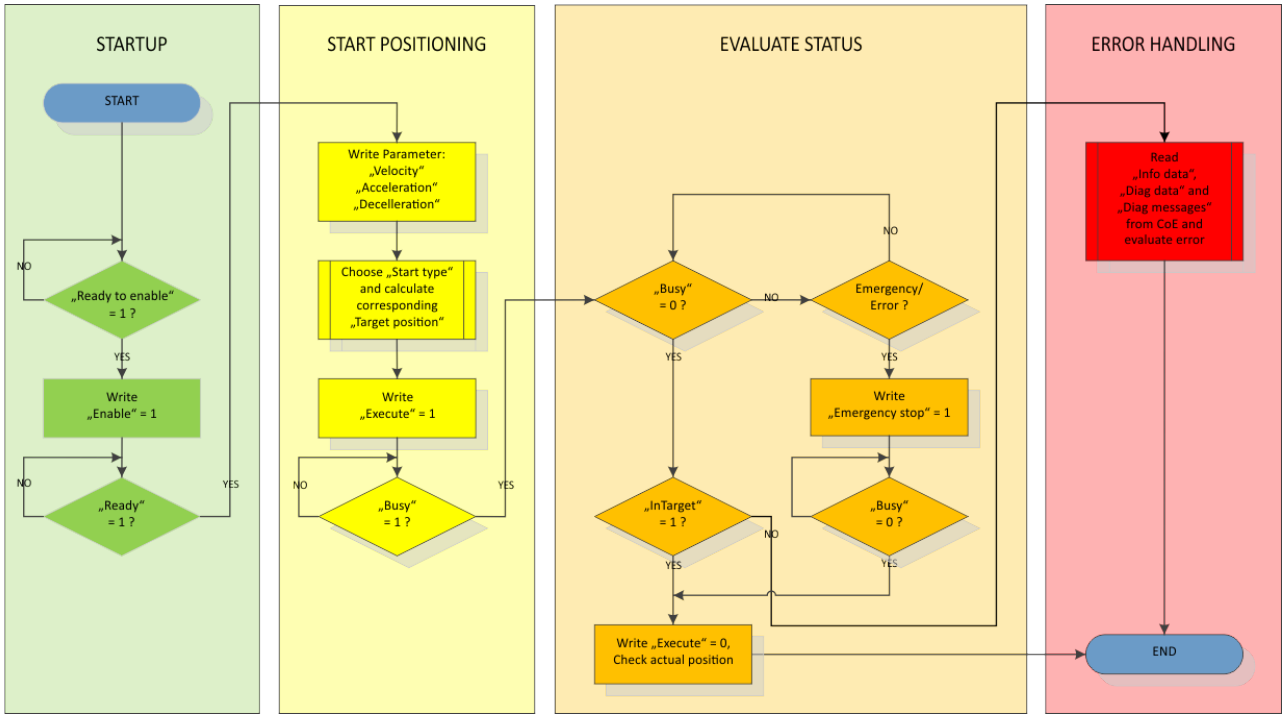
CoE parameters

The CoE parameters for configuring Drive Motion Control are located in the following CoE objects:

- 0x8040 "DMC Settings" [▶ 175]
- 0x8041 "DMC Features" [▶ 176]

6.7.5 State machine

It is also possible to start travel commands without the function blocks of the library "Tc3_DriveMotionControl". This is based on the following status machine:



The variables for control and evaluation are located in the process data objects "DMC Inputs" and "DMC Outputs":

- Term 2 (EL7411)
  - DMC Inputs
    - DMC
      - FeedbackStatus
        - Latch extern valid
        - Set counter done
        - Status of extern latch
      - DriveStatus
        - Ready to enable
        - Ready
        - Warning
        - Error
        - Moving positive
        - Moving negative
        - Digital input 1
        - Digital input 2
      - PositioningStatus
        - Busy
        - In-Target
        - Warning
        - Error
        - Calibrated
        - Accelerate
        - Decelerate
        - Ready to execute
      - Set position
      - Set velocity
      - Actual drive time
      - Actual position lag
      - Actual velocity
      - Actual position
      - Error id
      - Input cycle counter
      - Channel id
      - Latch value
      - Cyclic info data 1
      - Cyclic info data 2

- DMC Outputs
  - DMC
    - FeedbackControl
      - Enable latch extern on positive edge
      - Set counter
      - Enable latch extern on negative edge
    - DriveControl
      - Enable
      - Reset
    - PositioningControl
      - Execute
      - Emergency stop
    - Set counter value
    - Target position
    - Target velocity
    - Start type
    - Target acceleration
    - Target deceleration

6.7.6 Differences compared with Tc2_Mc2

Tc2_Mc2 is the PLC library used for the operation of EL7411 with TwinCAT NC.

The function blocks of the Tc3_DriveMotionControl library have a similar structure to that of Tc2_Mc2. However, Tc3_DriveMotionControl differs from Tc2_Mc2 in the following points:

- It is mandatory to specify values for the accelerations because there are no default values.
- "After-triggering functions" are not supported. Therefore there is no "BufferMode".
- MC_Home has no input "bCalibrationCam". The settings for homing are located in the CoE object 0x8041 "DMC Features".

6.7.7 Start types

Start type	Code	Description
ABSOLUTE	0x0001	Absolute positioning to a specified target position
RELATIVE	0x0002	Relative positioning to a calculated target position; a specified position difference is added to the current position
ENDLESS_PLUS	0x0003	Endless travel in the positive direction of rotation (direct specification of a speed)
ENDLESS_MINUS	0x0004	Endless travel in the negative direction of rotation (direct specification of a speed)
MODULO_SHORT	0x0105	Modulo positioning along the shortest path to the modulo position (positive or negative), calculated by the "Modulo factor"
MODULO_PLUS	0x0205	Modulo positioning in the positive direction of rotation to the calculated modulo position
MODULO_MINUS	0x0305	Modulo positioning in the negative direction of rotation to the calculated modulo position
CALI_PLC_CAM	0x6000	Start a calibration with cam (digital inputs)
CALI_ON_BLOCK	0x6200	Start a calibration "on Block"
CALI_SET_POS	0x6E00	Set as calibrated, do not change the position
CALI_CLEAR_POS	0x6F00	Clear calibration bit

6.7.8 Error messages

Error Code	Message
0x4420	Cogging compensation not supported
0x8450	Invalid Start Type 0x%x, "%x" replaced by the unsupported start type from the PDO
0x8451	Invalid limit switch level
0x8452	Drive error during positioning
0x8453	Latch unit will be used by multiple modules
0x8454	Drive not in control
0x8455	Invalid value for "Target acceleration"
0x8456	Invalid value for "Target deceleration"
0x8457	Invalid value for "Target velocity"
0x8458	Invalid value for "Target position"
0x8459	Emergency stop active
0x845A	Target position exceeds Modulofactor
0x845B	Drive must be disabled
0x845C	No feedback found
0x845D	Modulo factor invalid
0x845E	Invalid target position window

6.8 Touch Probe

Functional description

The "Touch Probe" function implemented in the EL7411 provides the user with the possibility to save the current position of the connected motor at a defined point in time. In the tab Process data [▶ 156] the necessary inputs and outputs can be added.

The EL7411 has two digital inputs that can be used for the "Touch Probe" function. Each Touch Probe input can only detect edges of one direction at any time (rising or falling). It is not possible to react to both edges at the same time. However, there are no dependencies between the inputs (i.e. TP1 rising edge and TP2 falling edge is allowed).

The abbreviation TP1 stands for Touch Probe 1 and is linked to input 1 (terminal point 8). The abbreviation TP2 stands for Touch Probe 2 and is linked to input 2 (terminal 16) of the terminal. TP1 is used here as an example for the description of the function. The C-track of the incremental encoder can be selected as Touch Probe-trigger via the CoE parameter 0x8001:11 or 0x8001:12 "Touch probe x source" with the value "Hardware zero impulse (5)". However, this is not possible with the "Drive Motion Control" operation mode.

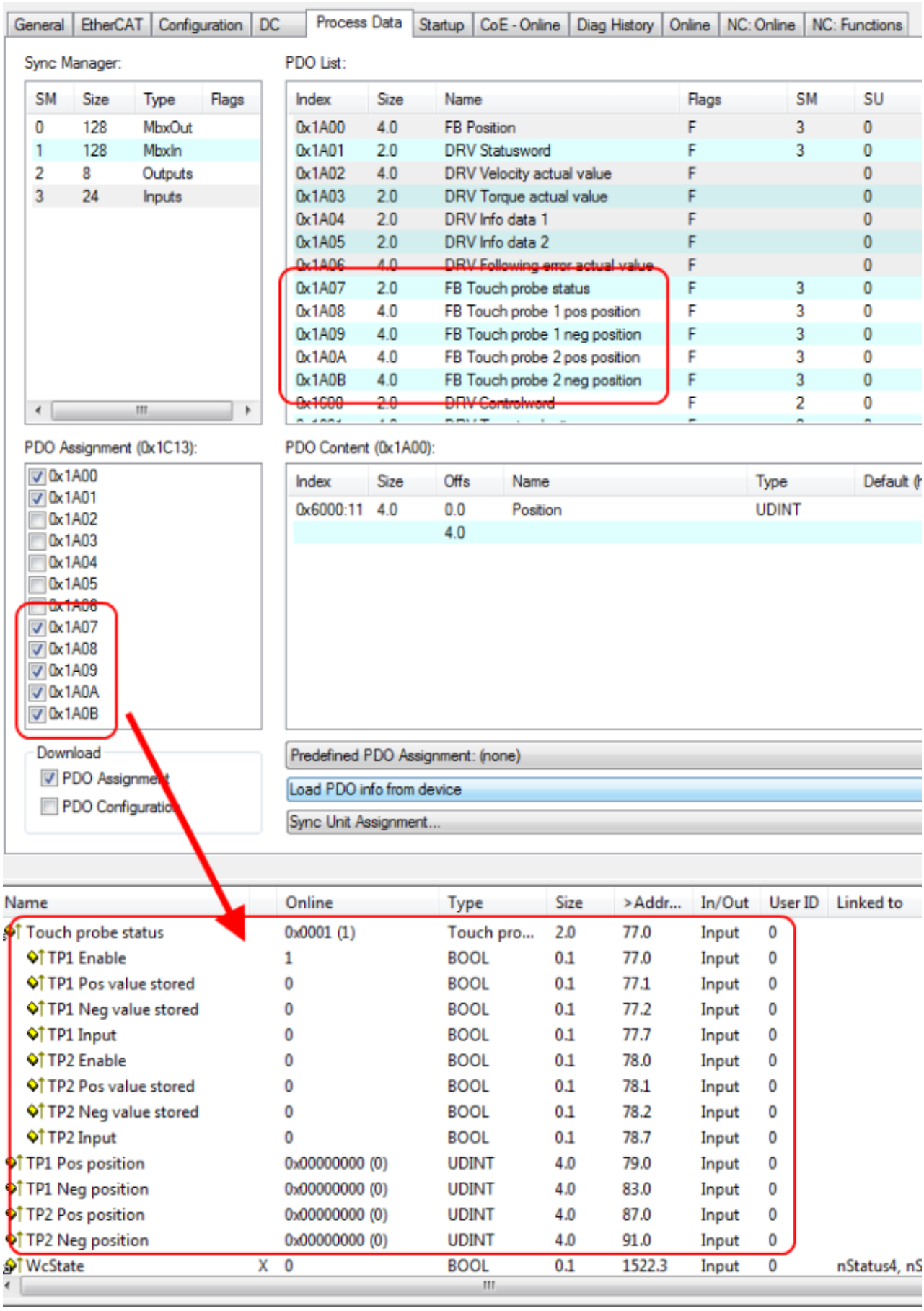


Fig. 99: Touch Probe inputs

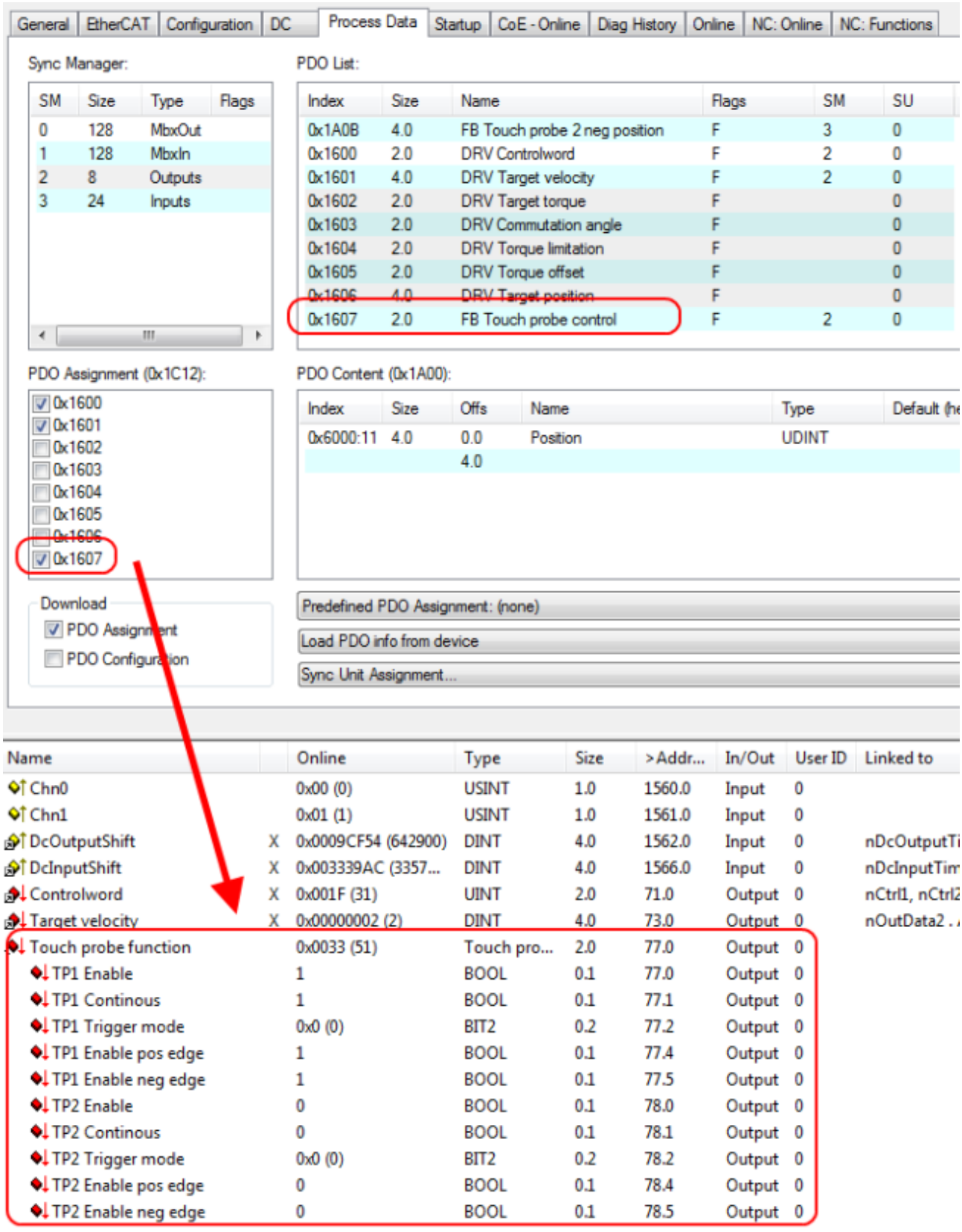


Fig. 100: Touch Probe outputs

Step-by-step

- "TP1 Enable" must be set to true in order to generally activate the Touch Probe function.
- Subsequently, you must decide whether the position is to be saved on a positive edge at input 1 ("TP1 Enable pos edge" = true) or on a negative edge ("TP1 Enable neg edge" = true).

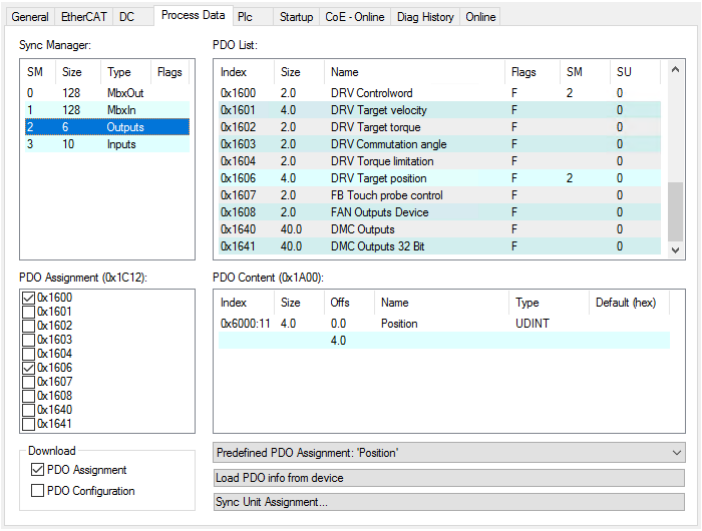
- With "TP1 Continuous" is decided whether only at the first event the position should be stored ("TP1 Continuous" = false) or whether this should happen at every event ("TP1 Continuous" = true).
  - For example, if "TP1 Continuous" and "TP1 Enablepos edge" are set, the position is saved on each rising edge at input 1 of the terminal.
  - If "TP1 Enable neg edge" is set and "TP1 Continuous" is not set, the position will only be saved on the first negative edge at input 1 of the terminal. If you wish to repeat this procedure, you must first deactivate "TP1 Enable" and then activate it again. Then the position is saved again on the first negative edge.
- "TP1 Trigger mode" has no function in the case of the EL7411.
- The saved position of the positive edge can be read in the inputs of the process data under "TP1 Pos position", that of the negative edge under "TP1 Neg position".
- The variables under "Touch probe status" are for the diagnosis.
- The Touch Probe inputs must be addressed with a 1-wire +24 V signal.

6.9 Process data

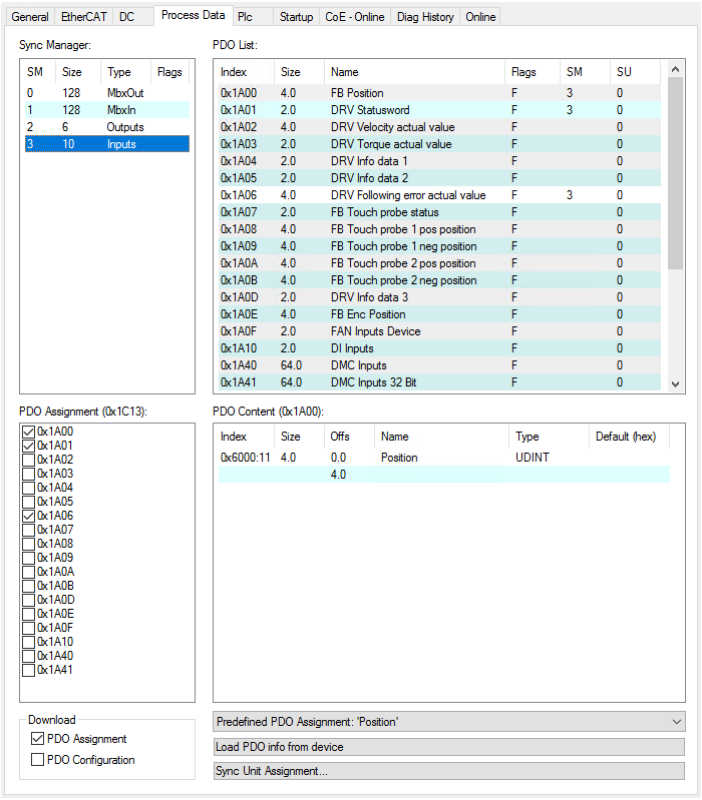
Sync Manager (SM)

The extent of the process data that is made available can be changed through the "Process Data" tab.:

- Sync Manager SM2 "Outputs"



- Sync Manager SM3 "Inputs"



PDO Assignment

For the configuration of the process data mark in the upper left field "Sync Manager" the desired Sync Manager SM 2 or SM 3. In the field below "PDO Assignment" you can switch on or off the process data assigned to the selected Sync Manager. Restarting the EtherCAT system, or reloading the configuration in Config mode, causes the EtherCAT communication to restart, and the process data is transferred from the terminal.

SM2, PDO Assignment 0x1C12				
Index	**Size (byte.bit)**	**Name**	**PDO Content**	
0x1600 (default)	2.0	DRV Controlword	Index 0x7010:01 [▶ 184]	
0x1601	4.0	DRV Target velocity	Index 0x7010:06 [▶ 184]	
0x1602	2.0	DRV Target torque	Index 0x7010:09 [▶ 184]	
0x1603	2.0	DRV Commutation angle	Index 0x7010:0E [▶ 184]	
0x1604	2.0	DRV Torque limitation	Index 0x7010:0B [▶ 184]	
0x1606 (default)	4.0	DRV Target position	Index 0x7010:05 [▶ 184]	
0x1607	2.0	FB Touch probe control	Index 0x7001:0 [▶ 183]	
			Index 0x7001:01 [▶ 183]	TP1 Enable
			Index 0x7001:02 [▶ 183]	TP1 Continous
			Index 0x7001:03 [▶ 183]	TP1 Trigger mode
			Index 0x7001:05 [▶ 183]	TP1 Enable pos edge
			Index 0x7001:06 [▶ 183]	TP1 Enable neg edge
			Index 0x7001:09 [▶ 183]	TP2 Enable
			Index 0x7001:0A [▶ 183]	TP2 Continous
			Index 0x7001:0B [▶ 183]	TP2 Trigger mode
			Index 0x7001:0D [▶ 183]	TP2 Enable pos edge
			Index 0x7001:0E [▶ 183]	TP2 Enable neg edge
0x1608	2.0	FAN Outputs Device	Index 0xF700:01 [▶ 186]	
0x1640 [1)]	40.0	DMC Outputs	Index 0x7040 [▶ 185]	
			Index 0x7040:02 [▶ 185]	DMC__FeedbackControl__Enable latch extern on positive edge
			Index 0x7040:03 [▶ 185]	DMC__FeedbackControl__Set counter
			Index 0x7040:04 [▶ 185]	DMC__FeedbackControl__Enable latch extern on negative edge
			Index 0x7040:11 [▶ 185]	DMC__DriveControl__Enable
			Index 0x7040:12 [▶ 185]	DMC__DriveControl__Reset
			Index 0x7040:21 [▶ 185]	DMC__PositioningControl__Execute
			Index 0x7040:22 [▶ 185]	DMC__PositioningControl__Emergency stop
			Index 0x7040:31 [▶ 185]	DMC__Set counter value
			Index 0x7040:32 [▶ 185]	DMC__Target position
			Index 0x7040:33 [▶ 185]	DMC__Target velocity
			Index 0x7040:34 [▶ 185]	DMC__Start type
			Index 0x7040:35 [▶ 185]	DMC__Target acceleration
			Index 0x7040:36 [▶ 185]	DMC__Target deceleration

SM2, PDO Assignment 0x1C12				
Index	**Size (byte.bit)**	**Name**	**PDO Content**	
0x1641 [1]	40.0	DMC Outputs 32 Bit	Index 0x7040 [▶ 185]	
			Index 0x7040:02 [▶ 185]	DMC__FeedbackControl__Enable latch extern on positive edge
			Index 0x7040:03 [▶ 185]	DMC__FeedbackControl__Set counter
			Index 0x7040:04 [▶ 185]	DMC__FeedbackControl__Enable latch extern on negative edge
			Index 0x7040:11 [▶ 185]	DMC__DriveControl__Enable
			Index 0x7040:12 [▶ 185]	DMC__DriveControl__Reset
			Index 0x7040:21 [▶ 185]	DMC__PositioningControl__Execute
			Index 0x7040:22 [▶ 185]	DMC__PositioningControl__Emergency stop
			Index 0x7040:31 [▶ 185]	DMC__Set counter value
			Index 0x7040:32 [▶ 185]	DMC__Target position
			Index 0x7040:33 [▶ 185]	DMC__Target velocity
			Index 0x7040:34 [▶ 185]	DMC__Start type
			Index 0x7040:35 [▶ 185]	DMC__Target acceleration
			Index 0x7040:36 [▶ 185]	DMC__Target deceleration

[1] These process data objects are only available from firmware 06 and ESI 0019.

SM3, PDO Assignment 0x1C13				
Index	**Size (byte.bit)**	**Name**	**PDO Content**	
0x1A00 (*default*)	4.0	FB Position	Index 0x6000:11 [▶ 178]	
0x1A01 (*default*)	2.0	DRV Statusword	Index 0x6010:01 [▶ 179]	
0x1A02	4.0	DRV Velocity actual value	Index 0x6010:07 [▶ 179]	
0x1A03	2.0	DRV Torque actual value	Index 0x6010:08 [▶ 179]	
0x1A04	2.0	DRV Info data 1	Index 0x6010:12 [▶ 179]	
0x1A05	2.0	DRV Info data 2	Index 0x6010:13 [▶ 179]	
0x1A06 (*default*)	4.0	DRV Following error actual value	Index 0x6010:06 [▶ 179]	
0x1A07	2.0	FB Touch probe status	Index 0x6001:0 [▶ 178]	
			Index 0x6001:01 [▶ 178]	TP1 Enable
			Index 0x6001:02 [▶ 178]	TP1 Pos value stored
			Index 0x6001:03 [▶ 178]	TP1 Neg value stored
			Index 0x6001:08 [▶ 178]	TP1 Input
			Index 0x6001:09 [▶ 178]	TP2 Enable
			Index 0x6001:0A [▶ 178]	TP2 Pos value stored
			Index 0x6001:0B [▶ 178]	TP2 Neg value stored
			Index 0x6001:10 [▶ 178]	TP2 Input
0x1A08	4.0	FB Touch probe 1 pos position	Index 0x6001:11 [▶ 178]	
0x1A09	4.0	FB Touch probe 1 neg position	Index 0x6001:12 [▶ 178]	
0x1A0A	4.0	FB Touch probe 2 pos position	Index 0x6001:13 [▶ 178]	
0x1A0B	4.0	FB Touch probe 2 neg position	Index 0x6001:14 [▶ 178]	
0x1A0D	2.0	DRV Info data 3	Index 0x6010:14 [▶ 179]	
0x1A0E	4.0	FB Enc Position	Index 0x6000:12 [▶ 178]	
0x1A0F	2.0	FAN Inputs Device	Index 0xF600:01 [▶ 182]	
0x1A10	2.0	DI Inputs	Index 0x6020:0 [▶ 180]	
			Index 0x6020:01 [▶ 180]	Input 1
			Index 0x6020:02 [▶ 180]	Input 2
			Index 0x6020:03 [▶ 180]	Fan Diag
			Index 0x6020:05 [▶ 180]	Encoder A
			Index 0x6020:06 [▶ 180]	Encoder B
			Index 0x6020:07 [▶ 180]	Encoder C
			Index 0x6020:09 [▶ 180]	Hall A
			Index 0x6020:0A [▶ 180]	Hall B
			Index 0x6020:0B [▶ 180]	Hall C
			Index 0x6020:0D [▶ 180]	Level of ENA input
0x1A40 [1)]	64.0	DMC Inputs	Index 0x6040:0 [▶ 181]	
			Index 0x6040:02 [▶ 181]	DMC__FeedbackStatus__Latch extern valid
			Index 0x6040:03 [▶ 181]	DMC__FeedbackStatus__Set counter done
			Index 0x6040:0D [▶ 181]	DMC__FeedbackStatus__Status of extern latch
			Index 0x6040:11 [▶ 181]	DMC__DriveStatus__Ready to enable
			Index 0x6040:12 [▶ 181]	DMC__DriveStatus__Ready
			Index 0x6040:13 [▶ 181]	DMC__DriveStatus__Warning
			Index 0x6040:14 [▶ 181]	DMC__DriveStatus__Error
			Index 0x6040:15 [▶ 181]	DMC__DriveStatus__Moving positive
			Index 0x6040:16 [▶ 181]	DMC__DriveStatus__Moving negative
			Index 0x6040:1C [▶ 181]	DMC__DriveStatus__Digital input 1

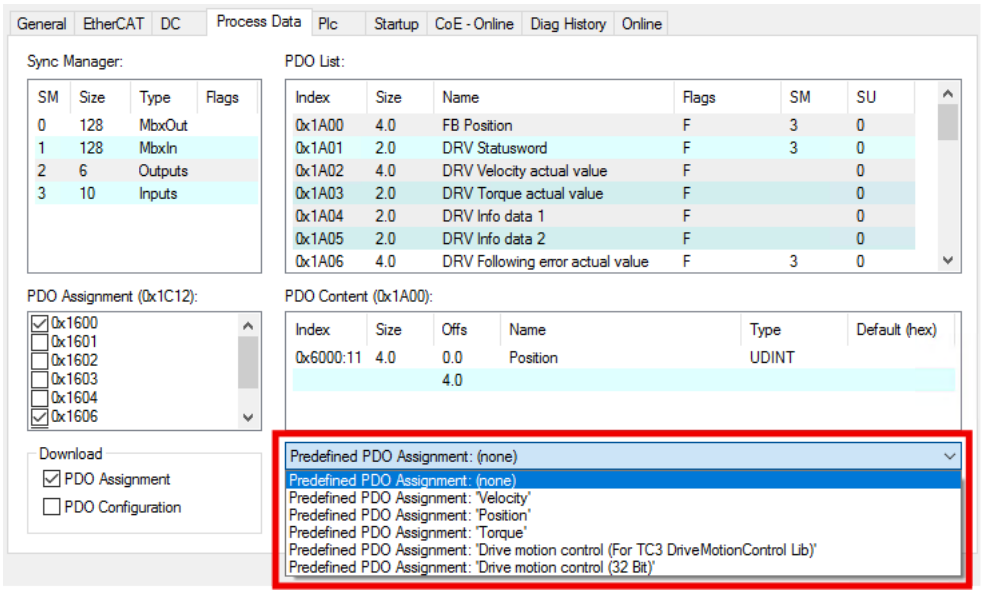
SM3, PDO Assignment 0x1C13				
Index	**Size (byte.bit)**	**Name**	**PDO Content**	
			Index 0x6040:1D [▶ 181]	DMC__DriveStatus__Digital input 2
			Index 0x6040:21 [▶ 181]	DMC__PositioningStatus__Busy
			Index 0x6040:22 [▶ 181]	DMC__PositioningStatus__In-Target
			Index 0x6040:23 [▶ 181]	DMC__PositioningStatus__Warning
			Index 0x6040:24 [▶ 181]	DMC__PositioningStatus__Error
			Index 0x6040:25 [▶ 181]	DMC__PositioningStatus__Calibrated
			Index 0x6040:26 [▶ 181]	DMC__PositioningStatus__Accelerate
			Index 0x6040:27 [▶ 181]	DMC__PositioningStatus__Decelerate
			Index 0x6040:28 [▶ 181]	DMC__PositioningStatus__Ready to execute
			Index 0x6040:31 [▶ 181]	DMC__Set position
			Index 0x6040:32 [▶ 181]	DMC__Set velocity
			Index 0x6040:33 [▶ 181]	DMC__Actual drive time
			Index 0x6040:34 [▶ 181]	DMC__Actual position lag
			Index 0x6040:35 [▶ 181]	DMC__Actual velocity
			Index 0x6040:36 [▶ 181]	DMC__Actual position
			Index 0x6040:37 [▶ 181]	DMC__Error id
			Index 0x6040:38 [▶ 181]	DMC__Input cycle counter
			Index 0x6040:39 [▶ 181]	DMC__Channel id
			Index 0x6040:3A [▶ 181]	DMC__Latch value
			Index 0x6040:3B [▶ 181]	DMC__Cyclic info data 1
			Index 0x6040:3C [▶ 181]	DMC__Cyclic info data 2
0x1A41 [1)]	64.0	DMC Inputs 32 Bit	Index 0x6040:0 [▶ 181]	
			Index 0x6040:02 [▶ 181]	DMC__FeedbackStatus__Latch extern valid
			Index 0x6040:03 [▶ 181]	DMC__FeedbackStatus__Set counter done
			Index 0x6040:0D [▶ 181]	DMC__FeedbackStatus__Status of extern latch
			Index 0x6040:11 [▶ 181]	DMC__DriveStatus__Ready to enable
			Index 0x6040:12 [▶ 181]	DMC__DriveStatus__Ready
			Index 0x6040:13 [▶ 181]	DMC__DriveStatus__Warning
			Index 0x6040:14 [▶ 181]	DMC__DriveStatus__Error
			Index 0x6040:15 [▶ 181]	DMC__DriveStatus__Moving positive
			Index 0x6040:16 [▶ 181]	DMC__DriveStatus__Moving negative
			Index 0x6040:1C [▶ 181]	DMC__DriveStatus__Digital input 1
			Index 0x6040:1D [▶ 181]	DMC__DriveStatus__Digital input 2
			Index 0x6040:21 [▶ 181]	DMC__PositioningStatus__Busy
			Index 0x6040:22 [▶ 181]	DMC__PositioningStatus__In-Target
			Index 0x6040:23 [▶ 181]	DMC__PositioningStatus__Warning
			Index 0x6040:24 [▶ 181]	DMC__PositioningStatus__Error

SM3, PDO Assignment 0x1C13				
Index	**Size (byte.bit)**	**Name**	**PDO Content**	
			Index 0x6040:25 [▶ 181]	DMC__PositioningStatus__Calibrated
			Index 0x6040:26 [▶ 181]	DMC__PositioningStatus__Accelerate
			Index 0x6040:27 [▶ 181]	DMC__PositioningStatus__Decelerate
			Index 0x6040:28 [▶ 181]	DMC__PositioningStatus__Ready to execute
			Index 0x6040:31 [▶ 181]	DMC__Set position
			Index 0x6040:32 [▶ 181]	DMC__Set velocity
			Index 0x6040:33 [▶ 181]	DMC__Actual drive time
			Index 0x6040:34 [▶ 181]	DMC__Actual position lag
			Index 0x6040:35 [▶ 181]	DMC__Actual velocity
			Index 0x6040:36 [▶ 181]	DMC__Actual position
			Index 0x6040:37 [▶ 181]	DMC__Error id
			Index 0x6040:38 [▶ 181]	DMC__Input cycle counter
			Index 0x6040:39 [▶ 181]	DMC__Channel id
			Index 0x6040:3A [▶ 181]	DMC__Latch value
			Index 0x6040:3B [▶ 181]	DMC__Cyclic info data 1
			Index 0x6040:3C [▶ 181]	DMC__Cyclic info data 2

[1)] These process data objects are only available from firmware 06 and ESI 0019.

Predefined PDO Assignment

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



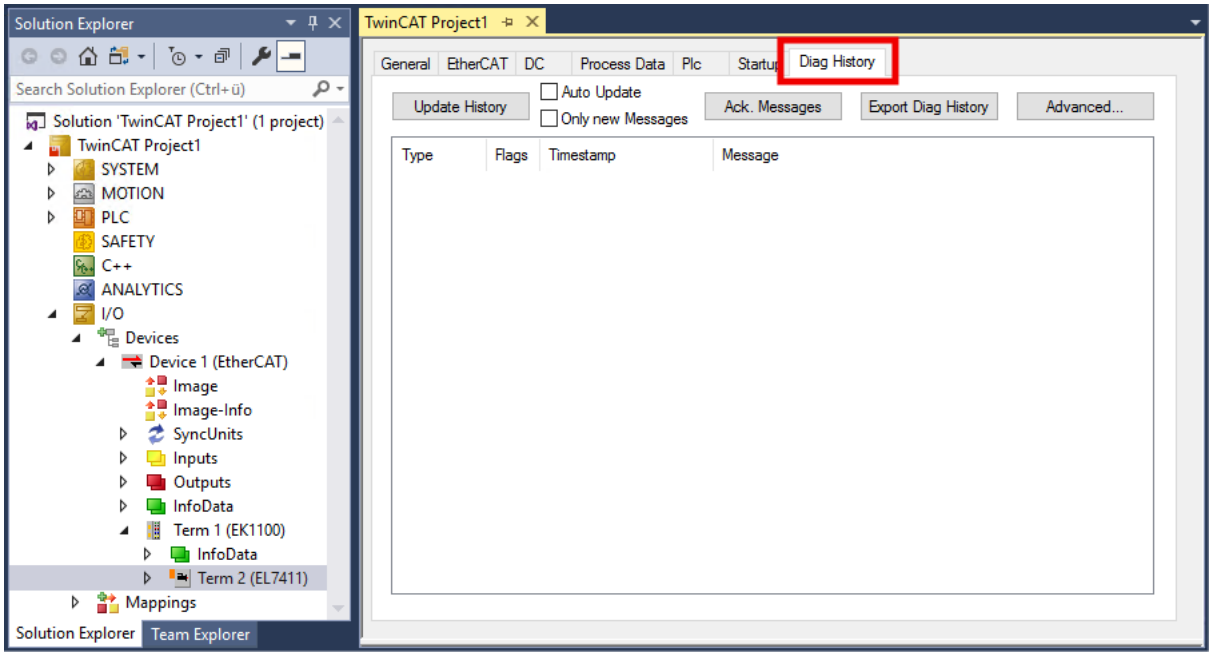
The following PDO assignments are available:

Name	SM2, PDO assignment	SM3, PDO assignment
Velocity	0x1600 [▶ 194] (DRV Controlword) 0x1601 [▶ 194] (DRV Target velocity)	0x1A00 [▶ 199] (FB Position) 0x1A01 [▶ 199] (DRV Statusword)
Position (*default*)	0x1600 [▶ 194] (DRV Controlword) 0x1606 [▶ 195] (DRV Target position)	0x1A00 [▶ 199] (FB Position) 0x1A01 [▶ 199] (DRV Statusword) 0x1A06 [▶ 200] (DRV Following error actual value)
Torque	0x1600 [▶ 194] (DRV Controlword) 0x1602 [▶ 194] (DRV Target torque)	0x1A00 [▶ 199] (FB Position) 0x1A01 [▶ 199] (DRV Statusword)
Drive motion control (For TC3 DriveMotionControl Lib)	0x1640 [▶ 197] (DMC Outputs)	0x1A40 [▶ 203] (DMC Inputs)
Drive motion control (32 bit)	0x1641 [▶ 198] (DMC Outputs 32 Bit)	0x1A41 [▶ 206] (DMC Inputs 32 Bit)

7 Error diagnosis

7.1 Diag History

The Diag History displays status messages from EtherCAT devices. The status messages are used for diagnosis and troubleshooting.



The following table shows all possible status messages from the EL7411:

Hex	English	German
0x1201	Communication re-established	Verbindung zur Feldseite wiederhergestellt
0x4101	Terminal-Overtemperature	Klemmen-Übertemperatur
0x4102	PDO-configuration is incompatible to the selected mode of operation	PDO-Konfiguration ist zur gewählten Betriebsart nicht kompatibel
0x4107	Undervoltage Up	Unterspannung Up
0x4109	Overvoltage Up	Überspannung Up
0x410A	Fan	Lüfter
0x410B	Error detected, but disabled by suppression mask	Ausmaskierter Fehler erkannt
0x4301	Feedback-Warning	Feedback-Warnung
0x4411	DC-Link undervoltage	Unterspannung DC-Zwischenkreis
0x4412	DC-Link overvoltage	Überspannung DC-Zwischenkreis
0x4413	I2T Amplifier overload	I2T-Modell Verstärker Überlast
0x4414	I2T Motor overload	I2T-Modell Motor Überlast
0x4415	Speed limitation active	Geschwindigkeit wird begrenzt
0x4418	Limit: Current	Limit: Strom wird limitiert
0x4419	Limit: Amplifier I2T-model exceeds 100%	Limit: Verstärker I2T-Modell übersteigt 100%
0x441A	Limit: Motor I2T-model exceeds 100%	Limit: Motor I2T-Modell übersteigt 100%
0x441B	Limit: Velocity limitation	Limit: Drehzahl wird limitiert
0x441C	Voltage on Enable-Input missing	Spannung am Enable-Eingang fehlt
0x441D	Internal hardware error	Interner Hardwarefehler
0x441E	Invalid configuration of touchprobe inputs	Ungültige Konfiguration der Touchprobe-Eingänge

Hex	English	German
0x8002	Communication aborted	Kommunikation abgebrochen
0x8102	Invalid combination of Inputs and Outputs PDOs	Ungültige Kombination von In- und Output PDOs
0x8104	Terminal-Overtemperature	Klemmen-Übertemperatur
0x8105	PD-Watchdog	PD-Watchdog
0x810A	Fan	Lüfter
0x810B	Undervoltage Up	Unterspannung Up
0x810C	Overvoltage Up	Überspannung Up
0x8135	Cycletime has to be a multiple of 125 µs	Zykluszeit muss ein vielfaches von 125 µs sein
0x8144	Hardware fault (%d)	Hardwarefehler (%d)
0x817F	Error: 0x%X, 0x%X, 0x%X	Fehler: 0x%X, 0x%X, 0x%X
0x8201	No communication to field-side (Auxiliary voltage missing)	Keine Kommunikation zur Feldseite (Hilfsspannung fehlt)
0x8302	Feedback-Error	Feedback-Fehler
0x8303	Encoder supply error	Encoder Netzteil Fehler
0x830D	Encoder Termination overload	Überlastung der Encoderterminierung
0x830E	Overvoltage on encoder track %s	Überspannung auf Encoderspur %s
0x830F	Weak signals on encoder track %s	Schwache Pegel auf Encoderspur %s
0x8340	Hallsensor supply error	Hallsensor Netzteil Fehler
0x8341	Hallsensor-Error	Hallsensor-Fehler
0x8342	Misalignment of hall sensors (offset: %d°)	Fehlerhafte Ausrichtung der Hallsensoren (Abweichung %d°)
0x8400	Encoder disabled	Encoder deaktiviert
0x8404	Overcurrent	Überstrom auf Phase U, V oder W
0x8406	Undervoltage DC-Link	Unterspannung DC-Zwischenkreis
0x8407	Overvoltage DC-Link	Überspannung DC-Zwischenkreis
0x8408	I2T-Model Amplifier overload	I2T-Modell Verstärker Überlast
0x8409	I2T-Model motor overload	I2T-Modell Motor Überlast
0x840B	Commutation error	Kommutierungsfehler
0x840C	Motor not connected	Kabelbruch Motoranschluss
0x840F	An Encoder has to be configured in FOC mode	FOC-Modus aktiviert, aber kein Encoder konfiguriert
0x8417	Maximum rotating field velocity exceeded	Maximale Drehfeldfrequenz überschritten
0x841C	Enable input was disabled while the axis was enabled	Enable wurde bei aktiver Achse abgeschaltet
0x841D	Internal hardware error	Interner Hardwarefehler
0x841E	Number of encoder increments or number of pole pairs incorrect	Anzahl der Encoderinkremente oder Polpaarzahl falsch konfiguriert
0x841F	Torque limitation too low	Drehmomentbegrenzung zu gering
0x8420	Teach-In Process (%d) failed	Teach-In Prozess %d fehlgeschlagen
0x8421	Teach-In Process Timeout (Enable, DC-Link, ...)	Teach-In Prozess Timeout (Enable, Zwischenkreis, ...)
0x8441	Maximum following error distance exceeded	Maximaler Schleppfehlerabstand überschritten
0x8442	Encoder-Resolution insufficient	Encoder-Auflösung nicht ausreichend
0x8443	Combination of Mode of Operation and Commutation Type is invalid	Kombination aus Mode of Operation und Commutation Type ist nicht zulässig
0x8601	Supply voltage too low	Versorgungsspannung zu klein
0x8602	Supply voltage too high	Versorgungsspannung zu gross

8 CoE parameters

8.1 Restore object

Index 1011 Restore default parameters

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

8.2 Configuration data

Index 8000 FB Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8000:0	FB Settings	Observer settings	UINT8	RO	0x16 (22_{dec})
8000:11	Device type	Description of available feedback profiles (incremental encoder and Hall sensor)	UINT32	RW	0x00000605 (1541_{dec})
8000:14	Observer bandwidth	Bandwidth of the speed observer [Hz] Typical value range: 200..500 Hz	UINT16	RW	0x00C8 (200_{dec})
8000:15	Observer feed-forward	Load ratio [%] 100% = load-free 50% = mass moments of inertia of input and output are equal Pre-control for velocity measurement.	UINT8	RW	0x00 (0_{dec})
8000:16	Sub-increment bits	Bitwise shift to left from target and actual position and following error	UINT8	RW	0x00 (0_{dec})

Index 8001 FB Touch probe Settings

Index (hex)	Name	Meaning		Data type	Flags	Default
8001:0	FB Touch probe Settings			UINT8	RO	0x12 (18_{dec})
8001:11	Touch probe 1 source	permitted values:		INT16	RW	0x0001 (1_{dec})
		1	Touch probe input 1			
		2	Hardware zero impulse			
8001:12	Touch probe 2 source	permitted values:		INT16	RW	0x0001 (1_{dec})
		1	Touch probe input 1			
		2	Hardware zero impulse			

Index 8008 FB Settings ENC

Index (hex)	Name	Meaning		Data type	Flags	Default
8008:0	FB Settings ENC			UINT8	RO	0x13 (19_{dec})
8008:01	Invert feedback direction	Inversion of the encoder counting direction. Adjusted by the Scan Feedback feature.		BOOLEAN	RW	0x00 (0_{dec})
8008:02	Enable power supply	Enablement of the encoder supply voltage		BOOLEAN	RW	0x00 (0_{dec})
8008:05	Enable ENC C input	Enablement of the C-input of the terminal for the evaluation of the C-track on the incremental encoder		BOOLEAN	RW	0x01 (1_{dec})
8008:11	Supply voltage output	Voltage level of the encoder supply [mV] Typical value range: 2..24 V		UINT32	RW	0x00001388 (5000_{dec})
8008:12	Encoder type	Definition of encoder signal types		UINT16	RW	0x0004 (4_{dec})
		0	disabled			
		1	RS422 differential			
		2	TTL single ended			
		3	HTL differential			
		4	HTL single ended			
		5	RS422 differential – high impedance input			
		6	TTL single ended – input filters disabled			
		7	open collector			
8008:13	Encoder Increments per Revolution	Resolution of the encoder after 4-fold evaluation		UINT32	RW	0x00001000 (4096_{dec})

Index 800A FB Settings Hall

Index (hex)	Name	Meaning		Data type	Flags	Default
800A:0	FB Settings Hall	Settings for the Hall sensors		UINT8	RO	0x14 (20_{dec})
800A:02	Enable power supply	Enablement of the Hall sensor supply voltage		BOOLEAN	RW	0x00 (0_{dec})
800A:05	Enable extrapolation	Extrapolation of the measured values of the Hall sensor. This does not improve the physical resolution.		BOOLEAN	RW	0x00 (0_{dec})
800A:11	Supply voltage output	Voltage level of the Hall sensor supply [mV] Typical value range: 2…24 V		UINT32	RW	0x00001388 (5000_{dec})
800A:12	phasing	Arrangement of the Hall sensors. Determined by the Scan Feedback function. See chapter Scan Feedback [▶ 111].		UINT8	RW	0x01 (1_{dec})
		0	A-B: 60° / B-C: 60°			
		1	A-B: 120° / B-C: 120°			
		2	A-B: 240° / B-C: 240°			
		3	A-B: 300° / B-C: 300°			
		4	A-B: 60° / B-C: 240°			
		5	A-B: 120° / B-C: 300°			
		6	A-B: 240° / B-C: 60°			
		7	A-B: 300° / B-C: 120°			
800A:13	Hall commutation adjust	Commutation offset of the Hall sensors in 60° increments. Determined by the Scan Feedback function. See chapter Scan Feedback [▶ 111].		UINT8	RW	0x00 (0_{dec})
		0	0°			
		1	60°			
		2	120°			
		3	180°			
		4	240°			
		5	300°			
800A:14	Hall sensor type	Enable or disable the open-collector Hall sensor inputs		UINT8	RW	0x01 (1_{dec})
		0	disabled			
		1	open collector			

Index 8010 DRV Amplifier Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8010:0	DRV Amplifier Settings	Amplifier settings	UINT8	RO	0x64 (100_{dec})
8010:01	Enable TxPDOToggle	Enable or disable the TxPDO toggle in bit 10 of the status word	BOOLEAN	RW	0x00 (0_{dec})
8010:02	Enable input cycle counter	1: enabled Two-bit counter that is incremented with each process data cycle up to a maximum value of 3, after which it starts again at 0. The low bit is represented in bit 10 and the high bit in bit 14 of the Status word.	BOOLEAN	RW	0x00 (0_{dec})
8010:04	Repeat find commutation	Repeat the commutation angle determination. (Effective for all FOC operation modes)	BOOLEAN	RW	0x01 (1_{dec})
8010:05	Enable cogging torque compensation	Enable or disable the cogging torque compensation (for FOC operation modes only).	BOOLEAN	RW	0x00 (0_{dec})
8010:12	Current loop integral time	Integral component of current controller [0.1 ms]	UINT16	RW	0x000A (10_{dec})
8010:13	Current loop proportional gain	Proportional component of current controller [0.1 V/A]	UINT16	RW	0x000A (10_{dec})
8010:14	Velocity loop integral time (current mode)	Integral component of velocity controller [0.1 ms] (For operation modes with FOC. See chapter Selection of the operation mode [▶ 96])	UINT32	RW	0x00000032 (50_{dec})
8010:15	Velocity loop proportional gain (current mode)	Proportional component of velocity controller [µA/(°/s)] (For operation modes with FOC. See chapter Selection of the operation mode [▶ 96])	UINT32	RW	0x00000014 (20_{dec})
8010:17	Position loop proportional gain	Proportional component position controller. Unit: (°/s) / °	UINT32	RW	0x0000000A (10_{dec})
8010:19	Nominal DC link voltage	Supplied DC link voltage [mV]	UINT32	RW	0x0000BB80 (48000_{dec})
8010:1A	Min DC link voltage	Minimum DC link voltage [mV]	UINT32	RW	0x00001A90 (6800_{dec})
8010:1B	Max DC link voltage	Maximum DC link voltage [mV]	UINT32	RW	0x0000EA60 (60000_{dec})
8010:29	Amplifier I2T warn level	I²T model warning threshold [%]	UINT8	RW	0x50 (80_{dec})
8010:2A	Amplifier I2T error level	I²T model error threshold [%]	UINT8	RW	0x69 (105_{dec})
8010:2B	Amplifier Temperature warn level	Terminal overtemperature warning threshold [0.1 °C]	UINT16	RW	0x0320 (800_{dec})
8010:2C	Amplifier Temperature error level	Terminal overtemperature error threshold [0.1 °C]	UINT16	RW	0x03E8 (1000_{dec})
8010:31	Velocity limitation	Limitation of the velocity setpoint [rpm] (effective only in CSV and CSP mode)	UINT32	RW	0x000186A0 (100000_{dec})

Index	Name	Meaning		Data type	Flags	Default
8010:33	Stand still window	Standstill window [rpm] Velocity range for which the axis is considered to be at a standstill.		UINT16	RW	0x0000 (0_{dec})
8010:39	Select info data 1	Selection of "Info data 1" Optional display of additional information in the cyclic process data. The following parameters are available:		UINT8	RW	0x02 (2_{dec})
		2	DC link voltage [mV]			
		4	PCB temperature - internal terminal temperature [0.1 °C]			
		7	I2T Motor [%]			
		8	I2T Amplifier [%]			
		10	Digital inputs Bit0: Digital Input 1 Level Bit1: Digital Input 2 Level Bit2: Fan Status (EL7411 only) Bit3: unused Bit4: Encoder A Bit5: Encoder B Bit6: Encoder C Bit7: unused Bit8: Hall Sensor U Bit 9: Hall Sensor V Bit 10: Hall Sensor W Bit 11: unused Bit 12: Hardware Enable Input Level (required for the STO functionality)			
		12	Phase Voltage U [mV]			
		13	Phase Voltage V [mV]			
		14	Phase Voltage W [mV]			

Index	Name	Meaning		Data type	Flags	Default
8010:3A	Select info data 2	Selection of "Info data 2" Optional display of additional information in the cyclic process data. The following parameters are available:		UINT8	RW	0x04 (4_{dec})
		2	DC link voltage [mV]			
		4	PCB temperature - internal terminal temperature [0.1 °C]			
		7	I²T Motor [%]			
		8	I²T Amplifier [%]			
		10	Digital inputs Bit0: Digital Input 1 Level Bit1: Digital Input 2 Level Bit2: Fan Status (EL7411 only) Bit3: unused Bit4: Encoder A Bit5: Encoder B Bit6: Encoder C Bit7: unused Bit8: Hall Sensor U Bit 9: Hall Sensor V Bit 10: Hall Sensor W Bit 11: unused Bit 12: Hardware Enable Input Level (required for the STO functionality)			
		12	Phase Voltage U			
		13	Phase Voltage V			
		14	Phase Voltage W			
8010:50	Following error window	Following error monitor: Following error window Unit: Reference to increments of the raw position data. Can be scaled in the PLC if necessary. 0xFFFFFFFF (-1dec) = following error monitor off Any other value = following error monitoring on [Inc] Applies in conjunction with following error timeout 8010:51.		UINT32	RW	0xFFFFFFFF (-1_{dec})
8010:51	Following error time out	Following error monitor: Timeout [ms] If the following error is larger than the following error window for a time that exceeds the timeout, this leads to an error reaction.		UINT16	RW	0x0000 (0_{dec})
8010:54	Feature bits	Reserved		UINT32	RW	0x00000000 (0_{dec})
8010:57	Position loop velocity feed forward gain	Velocity pre-control of the position controller [%].		UINT8	RW	0x64 (100_{dec})

Index	Name	Meaning	Data type	Flags	Default
8010:58	Select info data 3	Selection of "Info data 3" Optional display of additional information in the cyclic process data. The following parameters are available: 2 DC link voltage (mV) 4 PCB temperature (0.1 °C) 7 I2T Motor 8 I2T Amplifier 10 Digital inputs Bit0: Digital Input 1 Level Bit1: Digital Input 2 Level Bit2: Fan Status (EL7411 only) Bit3: unused Bit4: Encoder A Bit5: Encoder B Bit6: Encoder C Bit7: unused Bit8: Hall Sensor U Bit 9: Hall Sensor V Bit 10: Hall Sensor W Bit 11: unused Bit 12: Hardware Enable Input Level (required for the STO functionality) 12 Phase Voltage U 13 Phase Voltage V 14 Phase Voltage W	UINT8	RW	0x07 (7_{dec})
8010:59	Error suppression mask	Error suppression mask Bit 0: Suppresses detection of a cable breakage of the motor phases Bit 1: Suppresses commutation angle monitoring	UINT32	RW	0x00000000 (0_{dec})
8010:5A	Velocity loop integral time (voltage mode)	Integral component of velocity controller [0.1 ms] (for operation modes with six-step. See chapter Selection of the operation mode [▶ 96])	UINT32	RW	0x000001F4 (500_{dec})
8010:5B	Velocity loop proportional gain (voltage mode)	Proportional component of velocity controller [µV/(°/s)] (for operation modes with six-step. See chapter Selection of the operation mode [▶ 96])	UINT32	RW	0x00000064 (100_{dec})
8010:5C	Velocity loop voltage feed forward gain (voltage mode)	Velocity pre-control of the velocity controller [%] (For operation modes with six-step. See chapter Selection of the operation mode [▶ 96])	UINT8	RW	0x32 (50_{dec})
8010:5E	Sensorless offset voltage scaling	Configures the voltage output at standstill [%].	UINT16	RW	0x0032 (50_{dec})
8010:5F	Sensorless observer bandwidth	Bandwidth of the position/velocity observer in sensorless mode [Hz].	UINT16	RW	0x0064 (100_{dec})

Index	Name	Meaning	Data type	Flags	Default
8010:60	Sensorless max acceleration	Maximum acceleration in sensorless operation [°/s²].	UINT32	RW	0x000186A0 (100000_{dec})
8010:61	Cogging torque compensation	Contains the coefficients of the cogging torque compensation. These are determined by the Cogging Scan function. See chapter Scan Motor Cogging [▶ 112].	OCTET-STRING[16]	RW	{0}
8010:62	Position loop deadband window	Deadband window of the position controller [Inc] For position deviations smaller than the defined range applies (if set velocity is in the "Standstill window" (see parameter: 8010:33) 0 - 75% position controller switched off 75 - 100% linear transition of the position controller gain	UINT32	RW	0x00000000 (0_{dec})
8010:63	Find commutation time	Defines the time for determining the commutation angle [0.1 s] This process is divided into two phases. In total, it takes twice the time specified here. Required time depends on the application.	UINT16	RW	0x0009 (9_{dec})
8010:64	Commutation type	Method for determining the commutation angle: 1 FOC with incremental encoder 2 Six-Step with Hall 3 Six-step sensorless 4 FOC with incremental encoder and Hall	UINT8	RW	0x01 (1_{dec})

Index 8011 DRV Motor Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8011:0	DRV Motor Settings	Motor settings	UINT8	RO	0x31 (49_{dec})
8011:11	Max current	Peak current of the motor [mA]. Limitation by maximum output current of the EL7411. The motor current values are to be specified as peak value.	UINT32	RW	0x00001770 (6000_{dec})
8011:12	Rated current	Nominal current of the motor [mA], corresponds to the maximum continuous motor output current. The motor current values are to be specified as peak value. Target Torque, Torque actual value and Torque limitation are scaled relative to the rated current in per mill.	UINT32	RW	0x000003E8 (1000_{dec})
8011:13	Motor pole pairs	Number of pole pairs	UINT8	RW	0x01 (1_{dec})
8011:16	Torque constant	Torque constant [mNm/A]	UINT32	RW	0x00000032 (50_{dec})
8011:18	Rotor moment of inertia	Mass moment of inertia of the motor including mechanism [g cm²]	UINT32	RW	0x00000064 (100_{dec})
8011:19	Winding inductance	Winding inductance [0.01 mH]	UINT16	RW	0x0064 (100_{dec})
8011:1B	Motor speed limitation	Motor speed limitation [rpm]	UINT32	RW	0x000186A0 (100000_{dec})
8011:29	I2T warn level	I²T motor warning threshold [%]	UINT8	RW	0x50 (80_{dec})
8011:2A	I2T error level	I²T motor error threshold [%]	UINT8	RW	0x69 (105_{dec})
8011:2D	Motor thermal time constant	Motor thermal time constant of the winding [0.1 s]	UINT16	RW	0x0028 (40_{dec})
8011:2E	Rated speed	Nominal speed [rpm]	UINT32	RW	0x000003E8 (1000_{dec})
8011:2F	Rated voltage	Nominal motor voltage [mV]	UINT32	RW	0x0000BB80 (48000_{dec})
8011:30	Winding resistance	Winding resistance, phase - phase [mΩ]	UINT32	RW	0x000003E8 (1000_{dec})
8011:31	Voltage constant	Voltage constant [µV/min^(-1)] Specifies the voltage induced by the motor as a generator (counter-electromotive force).	UINT32	RW	0x0000BB80 (48000_{dec})

Index 8012 DRV brake settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8012:0	DRV brake settings	Brake settings	UINT8	RO	0x14 (20_{dec})
8012:01	Manual override (release)	Manual release of the motor brake. Intended for commissioning purposes.	BOOLEAN	RW	0x00 (0_{dec})
8012:11	Release delay	Time the holding brake requires for opening (releasing) after the current was applied. [ms]	UINT16	RW	0x0000 (0_{dec})
8012:12	Application delay	Time the holding brake requires for closing (holding) after the current was switched off. [ms]	UINT16	RW	0x0000 (0_{dec})
8012:13	Emergency application timeout	Time that the amplifier waits until the speed reaches the standstill window after the stop request. The holding brake is triggered irrespective of the speed if the set waiting time is exceeded. [ms] Note: This parameter must be adjusted to at least the longest time for the axis to run down. In the case of suspended axes, this parameter should be set to a very short time in order to prevent the axis/load from sagging.	UINT16	RW	0x0000 (0_{dec})
8012:14	Brake moment of inertia	Moment of inertia of the motor brake [g cm²]	UINT16	RW	0x0000 (0_{dec})

Index 8040 DMC Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8040:0	DMC Settings	Settings for Drive Motion Control.	UINT8	RW	0x17 (23_{dec})
8040:07	Emergency deceleration	Deceleration for the emergency stop ramp. Indication in milliseconds from nominal motor speed to standstill.	UINT16	RW	0x64 (100_{dec})
8040:08	Calibration position	If homing is successful, the "Actual position" is set to this value. It is basically displayed unscaled in 64 bits. Scaling: The 32 low-order bits refer to single-turn revolution, the 32 high-order bits to multi-turn revolutions.	UINT64	RW	0x00 (0_{dec})
8040:09	Calibration velocity (towards plc cam)	Speed for driving on the referencing cams. Is specified in 10,000ths of 0x8011:1B "Motor speed limitation".	UINT16	RW	0x64 (100_{dec})
8040:0A	Calibration velocity (off plc cam)	Speed to drive down from the referencing cam. Is specified in 10,000ths of 0x8011:1B "Motor speed limitation".	UINT16	RW	0x0A (10_{dec})
8040:0E	Modulo factor	Modulo factor for modulo travel commands. The default value of 2^32 corresponds to a single-turn revolution and thus e.g. 360°.	UINT64	RW	0x100000000 (4294967296_{dec})
8040:12	Block calibration torque limit	Torque limitation for block calibration. The value is specified in 1000ths of 0x8011:12 "Rated current"	UINT16	RW	0x64 (100_{dec})
8040:13	Block calibration stop distance	Specifies the distance by which to move out of the block after calibration. Scaling see 0x8040:08.	UINT64	RW	0x100000000 (4294967296_{dec})
8040:14	Block calibration lag threshold	Maximum permissible position lag during block calibration. Scaling see 0x8040:08.	UINT64	RW	0x100000000 (4294967296_{dec})
8040:15	Target position window	General position target window for travel commands for reaching the InTarget state. Scaling see 0x8040:08. The "Target position window" is valid in connection with 0x8040:16 "Target position monitor time".	UINT64	RW	0x16C16C1 (23860929_{dec})
8040:16	Target position monitor time	The actual position must be within the position target window 0x8040:15 for the specified time to reach the InTarget state. Unit: ms.	UINT16	RW	0x14 (20_{dec})
8040:17	Target position timeout	Specifies the time for the timer to start when the setpoint generator reaches the target position. If the InTarget condition (see 0x8040:15 and 0x8040:16) is not reached within this time, the travel command is aborted. The function block for the travel command returns an error. Unit: ms.	UINT16	RW	0x1770 (6000_{dec})

Index 8041 DMC Features

Index (hex)	Name	Meaning	Data type	Flags	Default
8041:0	DMC Features	Drive Motion Control functions	UINT8	RW	0x1B (27_{dec})
8041:13	Invert calibration cam search direction	For the default homing sequence the direction for referencing cam search can be inverted. • FALSE: the cam is looked for in the direction of positive movement. • TRUE: the cam is looked for in the direction of negative movement.	BOOLEAN	RW	0x00 (0_{dec})
8041:14	Invert sync impulse search direction	For the default homing sequence the direction for sync pulse search can be inverted. • FALSE: sync pulse is looked for in the direction of positive movement. • TRUE: sync pulse is looked for in the direction of negative movement.	BOOLEAN	RW	0x01 (1_{dec})
8041:19	Calibration cam source	Signal source of the referencing cam: • 0: Input 1 • 1: Input 2	UINT8	RW	0x00 (0_{dec})
8041:1A	Calibration cam active level	Signal level of the referencing cam that is evaluated as "active".	UINT8	RW	0x00 (0_{dec})
8041:1B	Latch source	Signal source of the latch signal: • 0: Input 1 • 1: Input 2	UINT8	RW	0x00 (0_{dec})

Index F800 FAN Settings

Index (hex)	Name	Meaning		Data type	Flags	Default
F800:0	FAN Settings	Fan settings		UINT8	RO	0x11 (17_{dec})
F800:01	Select function of Fan mode/ Standard output	Select fan output function:		BIT4	RW	0x01 (1_{dec})
		0	Fan mode			
		1	Digital output			
F800:05	Select function of fan diag/Standard input	Select fan input function:		BIT4	RW	0x01 (1_{dec})
		0	Fan diag			
		1	Digital input			
F800:11	Fan behaviour	Fan behavior		UINT8	RW	0x0D (13_{dec})
		0	Controlled by fan temperature sensor			
		1	Always on - low speed			
		4	Always on - mid speed			
		9	Always on - full speed			
		10	Controlled by terminal temperature			
		13	No fan installed			

8.3 Configuration data vendor-specific

Index 801F DRV Vendor data

Index (hex)	Name	Meaning	Data type	Flags	Default
801F:0	DRV vendor data	Vendor-specific data	UINT8	RO	0x18 (24_{dec})
801F:11	Amplifier peak current	Amplifier peak current [mA]	UINT32	RW	0x000031B8 (12728_{dec})
801F:12	Amplifier rated current	Amplifier rated current [mA]	UINT32	RW	0x000018DC (6364_{dec})
801F:13	Amplifier thermal time constant	Thermal time constant of the amplifier [0.1 s]	UINT16	RW	0x0023 (35_{dec})
801F:14	Amplifier overcurrent threshold	Threshold value for short-circuit detection [mA]	UINT32	RW	0x000057E3 (22499_{dec})
801F:15	Max rotary field frequency	Maximum electric rotor rotation frequency [Hz]	UINT16	RW	0x0257 (599_{dec})
801F:16	Amplifier peak current with fan	Amplifier peak current in operation with fan [mA]	UINT32	RW	0x0000501A (20506_{dec})
801F:17	Amplifier rated current with fan	Amplifier rated current in operation with fan [mA]	UINT32	RW	0x00002C31 (11313_{dec})
801F:18	Vendor feature bits	Reserved	UINT32	RW	0x00000000 (0_{dec})

Index FB13 DRV Key code

Index (hex)	Name	Meaning	Data type	Flags	Default
FB13:0	DRV Key code		UINT8	RO	0x01 (1_{dec})
FB13:01	Code		OCTET-STRING[32]	RW	{0}

8.4 Command object

Index FB00 command

Index (hex)	Name	Meaning	Data type	Flags	Default
FB00:0	Command	Command register Used for the scanning functions. See chapter Scanning the hardware [▶ 110].	UINT8	RO	0x03 (3_{dec})
FB00:01	Request	Request	OCTET-STRING[2]	RW	{0}
FB00:02	Status	Status	UINT8	RO	0x00 (0_{dec})
FB00:03	Response	Response	OCTET-STRING[6]	RO	{0}

8.5 Input data

Index 6000 FB Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
6000:0	FB Inputs	Maximum subindex	UINT8	RO	0x12 (18_{dec})
6000:0E	TxPDO State	True: the position data is invalid False: the position data is valid	BOOLEAN	RO	0x00 (0_{dec})
6000:0F	Input cycle counter	Incremented with each process data cycle, switches to 0 after reaching the maximum value of 3.	BIT2	RO	0x00 (0_{dec})
6000:11	Position	Position	UINT32	RO	0x00000000 (0_{dec})
6000:12	Enc Position		UINT32	RO	0x00000000 (0_{dec})

Index 6001 FB Touch probe inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
6001:0	FB Touch probe inputs	Maximum subindex	UINT8	RO	0x14 (20_{dec})
6001:01	TP1 Enable	Touch probe 1 switched on	BOOLEAN	RO	0x00 (0_{dec})
6001:02	TP1 Pos value stored	Positive value of Touch probe 1 stored	BOOLEAN	RO	0x00 (0_{dec})
6001:03	TP1 Neg value stored	Negative value of Touch probe 1 stored	BOOLEAN	RO	0x00 (0_{dec})
6001:08	TP1 Input	Digital input Touch probe 1. The input must be addressed with a 1-wire +24 V signal.	BOOLEAN	RO	0x00 (0_{dec})
6001:09	TP2 Enable	Touch probe 2 switched on	BOOLEAN	RO	0x00 (0_{dec})
6001:0A	TP2 Pos value stored	Positive value of Touch probe 2 stored	BOOLEAN	RO	0x00 (0_{dec})
6001:0B	TP2 Neg value stored	Negative value of Touch probe 2 stored	BOOLEAN	RO	0x00 (0_{dec})
6001:10	TP2 Input	Digital input Touch probe 2. The input must be addressed with a 1-wire +24 V signal.	BOOLEAN	RO	0x00 (0_{dec})
6001:11	TP1 Pos position	Positive value of Touch probe 1. The given value must be multiplied by the corresponding scaling factor.	UINT32	RO	0x00000000 (0_{dec})
6001:12	TP1 Neg position	Negative value of Touch probe 1. The given value must be multiplied by the corresponding scaling factor.	UINT32	RO	0x00000000 (0_{dec})
6001:13	TP2 Pos position	Positive value of Touch probe 2. The given value must be multiplied by the corresponding scaling factor.	UINT32	RO	0x00000000 (0_{dec})
6001:14	TP2 Neg position	Negative value of Touch probe 2. The given value must be multiplied by the corresponding scaling factor.	UINT32	RO	0x00000000 (0_{dec})

Index 6010 DRV Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
6010:0	DRV Inputs		UINT8	RO	0x14 (20_{dec})
6010:01	Statusword	Bit 0: ready to switch on Bit 1: switched on Bit 2: operation enabled Bit 3: fault Bit 4: reserved Bit 5: reserved Bit 6: switch on disabled Bit 7: warning Bit 8 + 9: reserved Bit 10: TxPDOToggle Bit 11: internal limit active Bit 12: drive follows the command value Bit 13: input cycle counter Bit 14 - 15: reserved	UINT16	RO	0x0000 (0_{dec})
6010:03	Modes of operation display	permitted values: 8: Cyclic synchronous position mode (CSP) 9: Cyclic synchronous velocity mode (CSV) 10: Cyclic synchronous torque mode (CST) 11: Cyclic synchronous torque mode with commutation angle (CSTCA)	UINT8	RO	0x00 (0_{dec})
6010:06	Following error actual value	Following error. The given value must be multiplied by the corresponding scaling factor.	INT32	RO	0x00000000 (0_{dec})
6010:07	Velocity actual value		INT32	RO	0x00000000 (0_{dec})
6010:08	Torque actual value		INT16	RO	0x0000 (0_{dec})
6010:12	Info data 1	Synchronous information (selection via subindex 0x8010:39).	INT16	RO	0x0000 (0_{dec})
6010:13	Info data 2	Synchronous information (selection via subindex 0x8010:3A).	INT16	RO	0x0000 (0_{dec})
6010:14	Info data 3	Synchronous information.	INT16	RO	0x0000 (0_{dec})

Index 6020 DI Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
6020:0	DI Inputs		UINT8	RO	0x0D (13_{dec})
6020:01	Input 1		BOOLEAN	RO	0x00 (0_{dec})
6020:02	Input 2		BOOLEAN	RO	0x00 (0_{dec})
6020:03	Fan Diag		BOOLEAN	RO	0x00 (0_{dec})
6020:05	Encoder A		BOOLEAN	RO	0x00 (0_{dec})
6020:06	Encoder B		BOOLEAN	RO	0x00 (0_{dec})
6020:07	Encoder C		BOOLEAN	RO	0x00 (0_{dec})
6020:09	Hall A		BOOLEAN	RO	0x00 (0_{dec})
6020:0A	Hall B		BOOLEAN	RO	0x00 (0_{dec})
6020:0B	Hall C		BOOLEAN	RO	0x00 (0_{dec})
6020:0D	Level of ENA input		BOOLEAN	RO	0x00 (0_{dec})

Index 6040 DMC Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
6040:0	DMC Inputs		UINT8	RO	0x3C (60_{dec})
6040:02	DMC__Feedback Status__Latch extern valid	An edge was detected on the external input and latched.	BOOLEAN	RO	0x00 (0_{dec})
6040:03	DMC__Feedback Status__Set counter done	The setting of the feedback position was successful. This bit remains present until "Set counter" is released again	BOOLEAN	RO	0x00 (0_{dec})
6040:0D	DMC__Feedback Status__Status of extern latch	Status of the external latch input.	BOOLEAN	RO	0x00 (0_{dec})
6040:11	DMC__DriveStatus__Ready to enable	The drive hardware is ready for activation.	BOOLEAN	RO	0x00 (0_{dec})
6040:12	DMC__DriveStatus__Ready	The drive hardware is activated.	BOOLEAN	RO	0x00 (0_{dec})
6040:13	DMC__DriveStatus__Warning	A warning is pending in the drive.	BOOLEAN	RO	0x00 (0_{dec})
6040:14	DMC__DriveStatus__Error	An error is pending in the drive. The "Ready to enable" bit and the "Ready" bit are set to FALSE.	BOOLEAN	RO	0x00 (0_{dec})
6040:15	DMC__DriveStatus__Moving positive	The axis moves in positive direction.	BOOLEAN	RO	0x00 (0_{dec})
6040:16	DMC__DriveStatus__Moving negative	The axis moves in negative direction.	BOOLEAN	RO	0x00 (0_{dec})
6040:1C	DMC__DriveStatus__Digital input 1	Status of the first digital input.	BOOLEAN	RO	0x00 (0_{dec})
6040:1D	DMC__DriveStatus__Digital input 2	Status of the second digital input.	BOOLEAN	RO	0x00 (0_{dec})
6040:21	DMC__PositioningStatus__Busy	The positioning task is running.	BOOLEAN	RO	0x00 (0_{dec})
6040:22	DMC__PositioningStatus__In-Target	The axis is at the target position.	BOOLEAN	RO	0x00 (0_{dec})
6040:23	DMC__PositioningStatus__Warning	Warning.	BOOLEAN	RO	0x00 (0_{dec})
6040:24	DMC__PositioningStatus__Error	Error.	BOOLEAN	RO	0x00 (0_{dec})
6040:25	DMC__PositioningStatus__Calibrated	The axis is calibrated.	BOOLEAN	RO	0x00 (0_{dec})
6040:26	DMC__PositioningStatus__Accelerate	The axis accelerates.	BOOLEAN	RO	0x00 (0_{dec})

Index (hex)	Name	Meaning	Data type	Flags	Default
6040:27	DMC__PositioningStatus__Decelerate	The axis is decelerating.	BOOLEAN	RO	0x00 (0_{dec})
6040:28	DMC__PositioningStatus__Ready to execute	The path control is ready to accept a command. This bit is FALSE … • ... if the drive has a fault • ... if the drive is not activated • ... as long as the "PositioningControl__Execute" is pending.	BOOLEAN	RO	0x00 (0_{dec})
6040:31	DMC__Set Position	Current target position specified by the ramp generator in feedback increments.	INT64	RO	0x0000000000000000 (0_{dec})
6040:32	DMC__Set velocity	Current velocity specified by the ramp generator in 10000ths of the nominal motor speed	INT16	RO	0x0000 (0_{dec})
6040:33	DMC__Actual drive time	The time since the start of the travel command in ms. Stops when the target position is reached.	UINT32	RO	0x00000000 (0_{dec})
6040:34	DMC__Actual position lag	Lag error.	INT64	RO	0x0000000000000000 (0_{dec})
6040:35	DMC__Actual velocity	Current velocity in 10000ths of the nominal motor speed.	INT16	RO	0x0000 (0_{dec})
6040:36	DMC__Actual position	Current position from the feedback (incl. possible offsets due to homing, ...).	INT64	RO	0x0000000000000000 (0_{dec})
6040:37	DMC__Error id	Error Id (identical to Diag History).	UINT32	RO	0x00000000 (0_{dec})
6040:38	DMC__Input cycle counter	Incremented with each process data cycle.	UINT8	RO	0x00 (0_{dec})
6040:39	DMC__Channel id		UINT8	RO	0x00 (0_{dec})
6040:3A	DMC__Latch value	Feedback position at latch time.	INT64	RO	0x0000000000000000 (0_{dec})
6040:3B	DMC__Cyclic info data 1	Synchronous info data	INT16	RO	0x0000 (0_{dec})
63C040:	DMC__Cyclic info data 2	Synchronous info data	INT16	RO	0x0000 (0_{dec})

Index F600 FAN Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
F600:0	FAN Inputs	Fan input	UINT8	RO	0x01 (1_{dec})
F600:01	Fan diag/ Standard input	Diagnostic input for the fan. Can also be used as a standard digital input when operated without a fan. (Configured via F800:05)	BOOLEAN	RO	0x00 (0_{dec})

8.6 Output data

Index 7001 FB Touch probe outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
7001:0	FB Touch probe outputs	Maximum subindex	UINT8	RO	0x0E (14_{dec})
7001:01	TP1 Enable	Switch on Touch probe 1.	BOOLEAN	RO	0x00 (0_{dec})
7001:02	TP1 Continous	0: triggered only on the first event. 1: triggered on every event.	BOOLEAN	RO	0x00 (0_{dec})
7001:03	TP1 Trigger mode	No function.	BIT2	RO	0x00 (0_{dec})
7001:05	TP1 Enable pos edge	Trigger on positive edge	BOOLEAN	RO	0x00 (0_{dec})
7001:06	TP1 Enable neg edge	Trigger on negative edge	BOOLEAN	RO	0x00 (0_{dec})
7001:09	TP2 Enable	Switch on Touch probe 2.	BOOLEAN	RO	0x00 (0_{dec})
7001:0A	TP2 Continous	0: triggered only on the first event. 1: triggered on every event.	BOOLEAN	RO	0x00 (0_{dec})
7001:0B	TP2 Trigger mode	No function.	BIT2	RO	0x00 (0_{dec})
7001:0D	TP2 Enable pos edge	Trigger on positive edge	BOOLEAN	RO	0x00 (0_{dec})
7001:0E	TP2 Enable neg edge	Trigger on negative edge	BOOLEAN	RO	0x00 (0_{dec})

Index 7010 DRV Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
7010:0	DRV Outputs	Maximum subindex	UINT8	RO	0x0E (14_{dec})
7010:01	Controlword	Bit 0: switch on Bit 1: enable voltage Bit 2: reserved Bit 3: enable operation Bit 4 - 6: reserved Bit 7: fault reset Bit 8 - 15: reserved	UINT16	RO	0x0000 (0_{dec})
7010:03	Modes of operation	Selection of the operation mode [▶ 96]	UINT8	RW	0x08 (8_{dec})
		8 CSP			
		9 CSV			
		10 CST			
		11 CSTCA			
		131 Drive Motion Control (DMC)			
7010:05	Target position	Configured target position. The value must be multiplied by the corresponding scaling factor	UINT32	RO	0x00000000 (0_{dec})
7010:06	Target velocity	Configured target velocity. The velocity scaling can be found in object 0x9010:14 ("Velocity encoder resolution").	INT32	RO	0x00000000 (0_{dec})
7010:09	Target torque	Configured input value for torque monitoring. The value is specified in 1000ths of Rated current (0x8011:12).	INT16	RO	0x0000 (0_{dec})
7010:0B	Torque limitation	Torque threshold value for torque monitoring (bipolar limit) The value is specified in 1000ths of Rated current (0x8011:12).	UINT16	RW	0x7FFF (32767_{dec})
7010:0E	Commutation angle	Commutation angle for CSTCA operation mode. Unit: 360° / 2^{16}	UINT16	RO	0x0000 (0_{dec})

Index 7040 DMC Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
7040:0	DMC Outputs		UINT8	RO	0x36 (54_{dec})
7040:02	DMC__Feedback Control__Enable latch extern on positive edge	Latches to the positive edge of the external input.	BOOLEAN	RO	0x00 (0_{dec})
7040:03	DMC__Feedback Control__Set counter	With a rising edge "Actual position" is set to the value of "Set counter value".	BOOLEAN	RO	0x00 (0_{dec})
7040:04	DMC__Feedback Control__Enable latch extern on negative edge	Latches to the negative edge of the external input.	BOOLEAN	RO	0x00 (0_{dec})
7040:11	DMC__DriveControl__Enable	Activate drive.	BOOLEAN	RO	0x00 (0_{dec})
7040:12	DMC__DriveControl__Reset	Perform a reset of the drive hardware.	BOOLEAN	RO	0x00 (0_{dec})
7040:21	DMC__PositioningControl__Execute	Start travel command with a rising edge. The task runs as long as this bit is set or until the command is completed. If the level drops during travel, the axis is brought to a standstill with the deceleration specified for the task.	BOOLEAN	RO	0x00 (0_{dec})
7040:22	DMC__PositioningControl__Emergency stop	In the event of a rising edge, decelerate to a standstill with the emergency stop ramp.	BOOLEAN	RO	0x00 (0_{dec})
7040:31	DMC__Set counter value	See index 0x7040:03.	INT64	RO	0x0000000000000000 (0_{dec})
7040:32	DMC__Target position	Position specification in feedback increments.	INT64	RO	0x0000000000000000 (0_{dec})
7040:33	DMC__Target velocity	Maximum speed during the travel command in 10000ths of the nominal motor speed.	UINT16	RO	0x0000 (0_{dec})
7040:34	DMC__Start type	Type of positioning task: • 0x0001: Absolute • 0x0002: Relative • 0x0003: Endless + • 0x0004: Endless – • 0x0105: Modulo short • 0x0205: Modulo + • 0x0305: Modulo – • 0x6000: Cali PLC cam • 0x6200: Cali Block • 0x6E00: Cali set • 0x6F00: Cali clear	UINT16	RO	0x0000 (0_{dec})

Index (hex)	Name	Meaning	Data type	Flags	Default
7040:35	DMC__Target acceleration	Acceleration: time in ms from standstill to reaching the nominal motor speed.	UINT16	RO	0x0000 (0_{dec})
7040:36	DMC__Target deceleration	Deceleration: time in ms for the deceleration from the nominal motor speed to standstill.	UINT16	RO	0x0000 (0_{dec})

Index F700 FAN Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
F700:0	FAN Outputs	Fan output	UINT8	RO	0x01 (1_{dec})
F700:01	Fan mode/ Standard output	Output for fan mode	BOOLEAN	RO	0x00 (0_{dec})

8.7 Information data, diagnostic data

Index 9010 DRV Info data

Index (hex)	Name	Meaning	Data type	Flags	Default
9010:0	DRV Info data	Amplifier information data	UINT8	RO	0x15 (21_{dec})
9010:11	Amplifier temperature	Internal terminal temperature [0.1 °C]	INT16	RO	0x0000 (0_{dec})
9010:12	DC link voltage	DC link voltage [mV]	UINT32	RO	0x00000000 (0_{dec})
9010:13	Supported drive modes	Information about supported drive modes. (DS402: Object 0x6502) Only modes CSP, CSV, CST and CSTCA are supported. Bit 0: PP Bit 1: VL Bit 2: PV Bit 3: TQ Bit 4: R Bit 5: HM Bit 6: IP **Bit 7: CSP** **Bit 8: CSV** **Bit 9: CST** **Bit 10: CSTCA** Bit 11 - 15: reserved Bit 16-31: manufacturer-specific	UINT32	RO	0x00000000 (0_{dec})
9010:14	Velocity encoder resolution	Velocity encoder resolution [Inc/(U/s)]	UINT32	RO	0x00000000 (0_{dec})
9010:15	Position encoder resolution increments	Encoder increments per motor revolution (Compare 8008:13 Encoder increments per revolution for FOC. In the Six-Step operation modes: 65536)	UINT32	RO	0x00000000 (0_{dec})

Index A010 DRV Amplifier Diag data

Index (hex)	Name	Meaning	Data type	Flags	Default
A010:0	DRV Amplifier Diag data	Amplifier diagnosis data	UINT8	RO	0x11 (17_{dec})
A010:11	Amplifier I2T temperature	Amplifier I²T model utilization [%]	UINT8	RO	0x00 (0_{dec})

Index A011 DRV Motor Diag data

Index (hex)	Name	Meaning	Data type	Flags	Default
A011:0	DRV Motor Diag data	Motor diagnosis data	UINT8	RO	0x11 (17_{dec})
A011:11	Motor I2T temperature	Motor I²T model utilization [%]	UINT8	RO	0x00 (0_{dec})

Index FB40 Memory interface

Index (hex)	Name	Meaning	Data type	Flags	Default
FB40:0	Memory interface		UINT8	RO	0x03 (3_{dec})
FB40:01	Address		UINT32	RW	0x00000000 (0_{dec})
FB40:02	Length		UINT16	RW	0x0000 (0_{dec})
FB40:03	Data		OCTET-STRING[8]	RW	{0}

8.8 Standard objects

The standard objects have the same meaning for all EtherCAT slaves.

Index 1000 Device type

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

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EL7411

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	00

Index 100B Bootloader version

Index (hex)	Name	Meaning	Data type	Flags	Default
100B:0	Bootloader version		STRING	RO	N/A

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	0x1CF33052 (485699666_{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 (0dec)

Index 10E2 Manufacturer-specific Identification Code

Index (hex)	Name	Meaning	Data type	Flags	Default
10E2:0	Manufacturer-specific identification code		UINT8	RO	0x01 (1_{dec})
10E2:01	SubIndex 001		STRING	RO	

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		UINT8	RO	0x37 (55_{dec})
10F3:01	Maximum Messages		UINT8	RO	0x00 (0_{dec})
10F3:02	Newest Message		UINT8	RO	0x00 (0_{dec})
10F3:03	Newest Acknowledged Message		UINT8	RW	0x00 (0_{dec})
10F3:04	New Messages Available		BOOLEAN	RO	0x00 (0_{dec})
10F3:05	Flags		UINT16	RW	0x0000 (0_{dec})
10F3:06	Diagnosis Message 001		OCTET-STRING[28]	RO	{0}
10F3:07	Diagnosis Message 002		OCTET-STRING[28]	RO	{0}
10F3:08	Diagnosis Message 003		OCTET-STRING[28]	RO	{0}
10F3:09	Diagnosis Message 004		OCTET-STRING[28]	RO	{0}
10F3:0A	Diagnosis Message 005		OCTET-STRING[28]	RO	{0}
10F3:0B	Diagnosis Message 006		OCTET-STRING[28]	RO	{0}
10F3:0C	Diagnosis Message 007		OCTET-STRING[28]	RO	{0}
10F3:0D	Diagnosis Message 008		OCTET-STRING[28]	RO	{0}
10F3:0E	Diagnosis Message 009		OCTET-STRING[28]	RO	{0}
10F3:0F	Diagnosis Message 010		OCTET-STRING[28]	RO	{0}
10F3:10	Diagnosis Message 011		OCTET-STRING[28]	RO	{0}
10F3:11	Diagnosis Message 012		OCTET-STRING[28]	RO	{0}
10F3:12	Diagnosis Message 013		OCTET-STRING[28]	RO	{0}

Index (hex)	Name	Meaning	Data type	Flags	Default
10F3:13	Diagnosis Message 014		OCTET-STRING[28]	RO	{0}
10F3:14	Diagnosis Message 015		OCTET-STRING[28]	RO	{0}
10F3:15	Diagnosis Message 016		OCTET-STRING[28]	RO	{0}
10F3:16	Diagnosis Message 017		OCTET-STRING[28]	RO	{0}
10F3:17	Diagnosis Message 018		OCTET-STRING[28]	RO	{0}
10F3:18	Diagnosis Message 019		OCTET-STRING[28]	RO	{0}
10F3:19	Diagnosis Message 020		OCTET-STRING[28]	RO	{0}
10F3:1A	Diagnosis Message 021		OCTET-STRING[28]	RO	{0}
10F3:1B	Diagnosis Message 022		OCTET-STRING[28]	RO	{0}
10F3:1C	Diagnosis Message 023		OCTET-STRING[28]	RO	{0}
10F3:1D	Diagnosis Message 024		OCTET-STRING[28]	RO	{0}
10F3:1E	Diagnosis Message 025		OCTET-STRING[28]	RO	{0}
10F3:1F	Diagnosis Message 026		OCTET-STRING[28]	RO	{0}
10F3:20	Diagnosis Message 027		OCTET-STRING[28]	RO	{0}
10F3:21	Diagnosis Message 028		OCTET-STRING[28]	RO	{0}
10F3:22	Diagnosis Message 029		OCTET-STRING[28]	RO	{0}
10F3:23	Diagnosis Message 030		OCTET-STRING[28]	RO	{0}
10F3:24	Diagnosis Message 031		OCTET-STRING[28]	RO	{0}

Index (hex)	Name	Meaning	Data type	Flags	Default
10F3:25	Diagnosis Message 032		OCTET-STRING[28]	RO	{0}
10F3:26	Diagnosis Message 033		OCTET-STRING[28]	RO	{0}
10F3:27	Diagnosis Message 034		OCTET-STRING[28]	RO	{0}
10F3:28	Diagnosis Message 035		OCTET-STRING[28]	RO	{0}
10F3:29	Diagnosis Message 036		OCTET-STRING[28]	RO	{0}
10F3:2A	Diagnosis Message 037		OCTET-STRING[28]	RO	{0}
10F3:2B	Diagnosis Message 038		OCTET-STRING[28]	RO	{0}
10F3:2C	Diagnosis Message 039		OCTET-STRING[28]	RO	{0}
10F3:2D	Diagnosis Message 040		OCTET-STRING[28]	RO	{0}
10F3:2E	Diagnosis Message 041		OCTET-STRING[28]	RO	{0}
10F3:2F	Diagnosis Message 042		OCTET-STRING[28]	RO	{0}
10F3:30	Diagnosis Message 043		OCTET-STRING[28]	RO	{0}
10F3:31	Diagnosis Message 044		OCTET-STRING[28]	RO	{0}
10F3:32	Diagnosis Message 045		OCTET-STRING[28]	RO	{0}
10F3:33	Diagnosis Message 046		OCTET-STRING[28]	RO	{0}
10F3:34	Diagnosis Message 047		OCTET-STRING[28]	RO	{0}
10F3:35	Diagnosis Message 048		OCTET-STRING[28]	RO	{0}
10F3:36	Diagnosis Message 049		OCTET-STRING[28]	RO	{0}

Index (hex)	Name	Meaning	Data type	Flags	Default
10F3:37	Diagnosis Message 050		OCTET-STRING[28]	RO	{0}

Index 10F8 Actual Time Stamp

Index (hex)	Name	Meaning	Data type	Flags	Default
10F8:0	Actual Time Stamp		UINT64	RO	

Index 1600 DRV RxPDO-map control word

Index (hex)	Name	Meaning	Data type	Flags	Default
1600:0	DRV RxPDO-map control word	PDO Mapping RxPDO 1	UINT8	RO	0x01 (1_{dec})
1600:01	SubIndex 001	1. PDO mapping entry (object 0x7010 (DRV outputs), entry 0x01 (control word))	UINT32	RO	0x7010:01, 16

Index 1601 DRV RxPDO-map target velocity

Index (hex)	Name	Meaning	Data type	Flags	Default
1601:0	DRV RxPDO-map target velocity	PDO Mapping RxPDO 2	UINT8	RO	0x01 (1_{dec})
1601:01	SubIndex 001	1. PDO Mapping entry (object 0x7010 (DRV Outputs), entry 0x06 (Target velocity))	UINT32	RO	0x7010:06, 32

Index 1602 DRV RxPDO-Map Target torque

Index (hex)	Name	Meaning	Data type	Flags	Default
1602:0	DRV RxPDO-Map Target torque	PDO Mapping RxPDO 3	UINT8	RO	0x01 (1_{dec})
1602:01	SubIndex 001	1. PDO mapping entry (object 0x7010 (DRV outputs), entry 0x09 (target torque))	UINT32	RO	0x7010:09, 16

Index 1603 DRV RxPDO-map commutation angle

Index (hex)	Name	Meaning	Data type	Flags	Default
1603:0	DRV RxPDO-map commutation angle	PDO Mapping RxPDO 4	UINT8	RO	0x01 (1_{dec})
1603:01	SubIndex 001	1. PDO mapping entry (object 0x7010 (DRV outputs), entry 0x0E (commutation angle))	UINT32	RO	0x7010:0E, 16

Index 1604 DRV RxPDO-map torque limitation

Index (hex)	Name	Meaning	Data type	Flags	Default
1604:0	DRV RxPDO-map torque limitation	PDO Mapping RxPDO 5	UINT8	RO	0x01 (1_{dec})
1604:01	SubIndex 001	1. PDO mapping entry (object 0x7010 (DRV outputs), entry 0x0B (torque limitation))	UINT32	RO	0x7010:0B, 16

Index 1606 DRV RxPDO-Map Target position

Index (hex)	Name	Meaning	Data type	Flags	Default
1606:0	DRV RxPDO-Map Target position	PDO Mapping RxPDO 7	UINT8	RO	0x01 (1_{dec})
1606:01	SubIndex 001	1. PDO mapping entry (object 0x7010 (DRV outputs), entry 0x05 (target position))	UINT32	RO	0x7010:05, 32

Index 1607 FB RxPDO-Map Touch probe control

Index (hex)	Name	Meaning	Data type	Flags	Default
1607:0	FB RxPDO-Map Touch probe control	PDO Mapping RxPDO 8	UINT8	RO	0x0C (12_{dec})
1607:01	SubIndex 001	1. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x01 (TP1 Enable))	UINT32	RO	0x7001:01, 1
1607:02	SubIndex 002	2. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x02 (TP1 Continous))	UINT32	RO	0x7001:02, 1
1607:03	SubIndex 003	3. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x03 (TP1 Trigger mode))	UINT32	RO	0x7001:03, 2
1607:04	SubIndex 004	4. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x05 (TP1 Enable pos edge))	UINT32	RO	0x7001:05, 1
1607:05	SubIndex 005	5. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x06 (TP1 Enable neg edge))	UINT32	RO	0x7001:06, 1
1607:06	SubIndex 006	6. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1607:07	SubIndex 007	7. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x09 (TP2 Enable))	UINT32	RO	0x7001:09, 1
1607:08	SubIndex 008	8. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x0A (TP2 Continous))	UINT32	RO	0x7001:0A, 1
1607:09	SubIndex 009	9. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x0B (TP2 Trigger mode))	UINT32	RO	0x7001:0B, 2
1607:0A	SubIndex 010	10. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x0D (TP2 Enable pos edge))	UINT32	RO	0x7001:0D, 1
1607:0B	SubIndex 011	11. PDO Mapping entry (object 0x7001 (FB Touch probe outputs), entry 0x0E (TP2 Enable neg edge))	UINT32	RO	0x7001:0E, 1
1607:0C	SubIndex 012	12. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2

Index 1608 FAN RxPDO-Map Outputs Device

Index (hex)	Name	Meaning	Data type	Flags	Default
1608:0	FAN RxPDO-Map Outputs Device	PDO Mapping RxPDO 9	UINT8	RO	0x02 (2_{dec})
1608:01	SubIndex 001	1. PDO Mapping entry (object 0xF700 (FAN Outputs), entry 0x01 (Fan mode/Standard output))	UINT32	RO	0xF700:01, 1
1608:02	SubIndex 002	2. PDO Mapping entry (15 bits align)	UINT32	RO	0x0000:00, 15

Index 1640 DMC RxPDO-Map Outputs

Index	Name	Meaning	Data type	Flags	Default
1640:0	DMC RxPDO-Map Outputs	DMC RxPDO-Map Outputs	UINT8	RO	0x12 (18_{dec})
1640:01	SubIndex 001	1. PDO Mapping entry (1 bit align)	UINT32	RO	0x0000:00, 1
1640:02	SubIndex 002	2. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x02 (DMC_FeedbackControl__Enable latch extern on positive edge))	UINT32	RO	0x7040:02, 1
1640:03	SubIndex 003	3. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x03 (DMC_FeedbackControl__Set counter))	UINT32	RO	0x7040:03, 1
1640:04	SubIndex 004	4. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x04 (DMC__FeedbackControl__Enable latch extern on negative edge))	UINT32	RO	0x7040:04, 1
1640:05	SubIndex 005	5. PDO Mapping entry (12 bits align)	UINT32	RO	0x0000:00, 12
1640:06	SubIndex 006	6. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x11 (DMC__DriveControl__Enable))	UINT32	RO	0x7040:11, 1
1640:07	SubIndex 007	7. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x12 (DMC__DriveControl__Reset))	UINT32	RO	0x7040:12, 1
1640:08	SubIndex 008	8. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14
1640:09	SubIndex 009	9. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x21 (DMC__PositioningControl__Execute))	UINT32	RO	0x7040:21, 1
1640:0A	SubIndex 010	10. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x22 (DMC__PositioningControl__Emergency stop))	UINT32	RO	0x7040:22, 1
1640:0B	SubIndex 011	11. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14
1640:0C	SubIndex 012	12. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x31 (DMC__Set counter value))	UINT32	RO	0x7040:31, 64
1640:0D	SubIndex 013	13. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x32 (DMC__Target position))	UINT32	RO	0x7040:32, 64
1640:0E	SubIndex 014	14. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x33 (DMC__Target velocity))	UINT32	RO	0x7040:33, 16
1640:0F	SubIndex 015	15. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x34 (DMC__Start type))	UINT32	RO	0x7040:34, 16
1640:10	SubIndex 016	16. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x35 (DMC__Target acceleration))	UINT32	RO	0x7040:35, 16
1640:11	SubIndex 017	17. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x36 (DMC__Target deceleration))	UINT32	RO	0x7040:36, 16
1640:12	SubIndex 018	18. PDO Mapping entry (80 bits align)	UINT32	RO	0x0000:00, 80

Index 1641 DMC RxPDO-Map Outputs 32 Bit

Index	Name	Meaning	Data type	Flags	Default
1641:0	DMC RxPDO-Map Outputs 32 Bit	PDO Mapping RxPDO 66	UINT8	RO	0x14 (20_{dec})
1641:01	SubIndex 001	1. PDO Mapping entry (1 bit align)	UINT32	RO	0x0000:00, 1
1641:02	SubIndex 002	2. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x02 (DMC__FeedbackControl__Enable latch extern on positive edge))	UINT32	RO	0x7040:02, 1
1641:03	SubIndex 003	3. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x03 (DMC__FeedbackControl__Set counter))	UINT32	RO	0x7040:03, 1
1641:04	SubIndex 004	4. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x04 (DMC__FeedbackControl__Enable latch extern on negative edge))	UINT32	RO	0x7040:04, 1
1641:05	SubIndex 005	5. PDO Mapping entry (12 bits align)	UINT32	RO	0x0000:00, 12
1641:06	SubIndex 006	6. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x11 (DMC__DriveControl__Enable))	UINT32	RO	0x7040:11, 1
1641:07	SubIndex 007	7. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x12 (DMC__DriveControl__Reset))	UINT32	RO	0x7040:12, 1
1641:08	SubIndex 008	8. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14
1641:09	SubIndex 009	9. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x21 (DMC__PositioningControl__Execute))	UINT32	RO	0x7040:21, 1
1641:0A	SubIndex 010	10. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x22 (DMC__PositioningControl__Emergency stop))	UINT32	RO	0x7040:22, 1
1641:0B	SubIndex 011	11. PDO Mapping entry (14 bits align)	UINT32	RO	0x0000:00, 14
1641:0C	SubIndex 012	12. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x31 (DMC__Set counter value))	UINT32	RO	0x7040:31, 32
1641:0D	SubIndex 013	13. PDO Mapping entry (32 bits align)	UINT32	RO	0x0000:00, 32
1641:0E	SubIndex 014	14. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x32 (DMC__Target position))	UINT32	RO	0x7040:32, 32
1641:0F	SubIndex 015	15. PDO Mapping entry (32 bits align)	UINT32	RO	0x0000:00, 32
1641:10	SubIndex 016	16. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x33 (DMC__Target velocity))	UINT32	RO	0x7040:33, 16
1641:11	SubIndex 017	17. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x34 (DMC__Start type))	UINT32	RO	0x7040:34, 16

Index	Name	Meaning	Data type	Flags	Default
1641:12	SubIndex 018	18. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x35 (DMC__Target acceleration))	UINT32	RO	0x7040:35, 16
1641:13	SubIndex 019	19. PDO Mapping entry (object 0x7040 (DMC Outputs), entry 0x36 (DMC__Target deceleration))	UINT32	RO	0x7040:36, 16
1641:14	SubIndex 020	20. PDO Mapping entry (80 bits align)	UINT32	RO	0x0000:00, 80

Index 1A00 FB TxPDO-Map Position

Index	Name	Meaning	Data type	Flags	Default
1A00:0	FB TxPDO-Map Position	PDO Mapping TxPDO 1	UINT8	RO	0x01 (1_{dec})
1A00:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (FB Inputs), entry 0x11 (Position))	UINT32	RO	0x6000:11, 32

Index 1A01 DRV TxPDO-Map Statusword

Index (hex)	Name	Meaning	Data type	Flags	Default
1A01:0	DRV TxPDO-Map Statusword	PDO Mapping TxPDO 2	UINT8	RO	0x01 (1_{dec})
1A01:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (DRV Inputs), entry 0x01 (Statusword))	UINT32	RO	0x6010:01, 16

Index 1A02 DRV TxPDO-Map Velocity actual value

Index (hex)	Name	Meaning	Data type	Flags	Default
1A02:0	DRV TxPDO-Map Velocity actual value	PDO Mapping TxPDO 3	UINT8	RO	0x01 (1_{dec})
1A02:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (DRV Inputs), entry 0x07 (Velocity actual value))	UINT32	RO	0x6010:07, 32

Index 1A03 DRV TxPDO-Map Torque actual value

Index (hex)	Name	Meaning	Data type	Flags	Default
1A03:0	DRV TxPDO-Map Torque actual value	PDO Mapping TxPDO 4	UINT8	RO	0x01 (1_{dec})
1A03:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (DRV Inputs), entry 0x08 (Torque actual value))	UINT32	RO	0x6010:08, 16

Index 1A04 DRV TxPDO-Map Info data 1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A04:0	DRV TxPDO-Map Info data 1	PDO Mapping TxPDO 5	UINT8	RO	0x01 (1_{dec})
1A04:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (DRV Inputs), entry 0x12 (Info data 1))	UINT32	RO	0x6010:12, 16

Index 1A05 DRV TxPDO-Map Info data 2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A05:0	DRV TxPDO-Map Info data 2	PDO Mapping TxPDO 6	UINT8	RO	0x01 (1_{dec})
1A05:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (DRV Inputs), entry 0x13 (Info data 2))	UINT32	RO	0x6010:13, 16

Index 1A06 DRV TxPDO-Map Following error actual value

Index (hex)	Name	Meaning	Data type	Flags	Default
1A06:0	DRV TxPDO-Map Following error actual value	PDO Mapping TxPDO 7	UINT8	RO	0x01 (1_{dec})
1A06:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (DRV Inputs), entry 0x06 (Following error actual value))	UINT32	RO	0x6010:06, 32

Index 1A07 FB TxPDO-Map Touch probe status

Index (hex)	Name	Meaning	Data type	Flags	Default
1A07:0	FB TxPDO-Map Touch probe status	PDO Mapping TxPDO 8	UINT8	RO	0x0A (10_{dec})
1A07:01	SubIndex 001	1. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x01 (TP1 Enable))	UINT32	RO	0x6001:01, 1
1A07:02	SubIndex 002	2. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x02 (TP1 Pos value stored))	UINT32	RO	0x6001:02, 1
1A07:03	SubIndex 003	3. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x03 (TP1 Neg value stored))	UINT32	RO	0x6001:03, 1
1A07:04	SubIndex 004	4. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1A07:05	SubIndex 005	5. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x08 (TP1 Input))	UINT32	RO	0x6001:08, 1
1A07:06	SubIndex 006	6. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x09 (TP2 Enable))	UINT32	RO	0x6001:09, 1
1A07:07	SubIndex 007	7. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x0A (TP2 Pos value stored))	UINT32	RO	0x6001:0A, 1
1A07:08	SubIndex 008	8. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x0B (TP2 Neg value stored))	UINT32	RO	0x6001:0B, 1
1A07:09	SubIndex 009	9. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1A07:0A	SubIndex 010	10. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x10 (TP2 Input))	UINT32	RO	0x6001:10, 1

Index 1A08 FB TxPDO-Map Touch probe 1 pos position

Index (hex)	Name	Meaning	Data type	Flags	Default
1A08:0	FB TxPDO-Map Touch probe 1 pos position	PDO Mapping TxPDO 9	UINT8	RO	0x01 (1_{dec})
1A08:01	SubIndex 001	1. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x11 (TP1 Pos position))	UINT32	RO	0x6001:11, 32

Index 1A09 FB TxPDO-Map Touch probe 1 neg position

Index (hex)	Name	Meaning	Data type	Flags	Default
1A09:0	FB TxPDO-Map Touch probe 1 neg position	PDO Mapping TxPDO 10	UINT8	RO	0x01 (1_{dec})
1A09:01	SubIndex 001	1. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x12 (TP1 Neg position))	UINT32	RO	0x6001:12, 32

Index 1A0A FB TxPDO-Map Touch probe 2 pos position

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0A:0	FB TxPDO-Map Touch probe 2 pos position	PDO Mapping TxPDO 11	UINT8	RO	0x01 (1_{dec})
1A0A:01	SubIndex 001	1. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x13 (TP2 Pos position))	UINT32	RO	0x6001:13, 32

Index 1A0B FB TxPDO-Map Touch probe 2 neg position

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0B:0	FB TxPDO-Map Touch probe 2 neg position	PDO Mapping TxPDO 12	UINT8	RO	0x01 (1_{dec})
1A0B:01	SubIndex 001	1. PDO Mapping entry (object 0x6001 (FB Touch probe inputs), entry 0x14 (TP2 Neg position))	UINT32	RO	0x6001:14, 32

Index 1A0D DRV TxPDO-Map Info data 3

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0D:0	DRV TxPDO-Map Info data 3	PDO Mapping TxPDO 14	UINT8	RO	0x01 (1_{dec})
1A0D:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (DRV Inputs), entry 0x14 (Info data 3))	UINT32	RO	0x6010:14, 16

Index 1A0E FB TxPDO-Map Enc Position

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0E:0	FB TxPDO-Map Enc Position	PDO Mapping TxPDO 15	UINT8	RO	0x01 (1_{dec})
1A0E:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (FB Inputs), entry 0x12 (Enc Position))	UINT32	RO	0x6000:12, 32

Index 1A0F FAN TxPDO-Map Inputs Device

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0F:0	FAN TxPDO-Map Inputs Device	PDO Mapping TxPDO 16	UINT8	RO	0x02 (2_{dec})
1A0F:01	SubIndex 001	1. PDO Mapping entry (object 0xF600 (FAN Inputs), entry 0x01 (Fan diag/Standard input))	UINT32	RO	0xF600:01, 1
1A0F:02	SubIndex 002	2. PDO Mapping entry (15 bits align)	UINT32	RO	0x0000:00, 15

Index 1A10 DI TxPDO-Map Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1A10:0	DI TxPDO-Map Inputs	PDO Mapping TxPDO 17	UINT8	RO	0x0E (14_{dec})
1A10:01	SubIndex 001	1. PDO Mapping entry (object 0x6020 (DI Inputs), entry 0x01 (Input 1))	UINT32	RO	0x6020:01, 1
1A10:02	SubIndex 002	2. PDO Mapping entry (object 0x6020 (DI Inputs), entry 0x02 (Input 2))	UINT32	RO	0x6020:02, 1
1A10:03	SubIndex 003	3. PDO Mapping entry (object 0x6020 (DI Inputs), entry 0x03 (Fan Diag))	UINT32	RO	0x6020:03, 1
1A10:04	SubIndex 004	4. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A10:05	SubIndex 005	5. PDO Mapping entry (object 0x6020 (DI Inputs), entry 0x05 (Encoder A))	UINT32	RO	0x6020:05, 1
1A10:06	SubIndex 006	6. PDO Mapping entry (object 0x6020 (DI Inputs), entry 0x06 (Encoder B))	UINT32	RO	0x6020:06, 1
1A10:07	SubIndex 007	7. PDO Mapping entry (object 0x6020 (DI Inputs), entry 0x07 (Encoder C))	UINT32	RO	0x6020:07, 1
1A10:08	SubIndex 008	8. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A10:09	SubIndex 009	9. PDO Mapping entry (object 0x6020 (DI Inputs), entry 0x09 (Hall A))	UINT32	RO	0x6020:09, 1
1A10:0A	SubIndex 010	10. PDO Mapping entry (object 0x6020 (DI Inputs), entry 0x0A (Hall B))	UINT32	RO	0x6020:0A, 1
1A10:0B	SubIndex 011	11. PDO Mapping entry (object 0x6020 (DI Inputs), entry 0x0B (Hall C))	UINT32	RO	0x6020:0B, 1
1A10:0C	SubIndex 012	12. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1A10:0D	SubIndex 013	13. PDO Mapping entry (object 0x6020 (DI Inputs), entry 0x0D (Level of STO input))	UINT32	RO	0x6020:0D, 1
1A10:0E	SubIndex 014	14. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3

Index 1A40 DMC TxPDO-Map Inputs

Index	Name	Meaning	Data type	Flags	Default
1A40:0	DMC TxPDO-Map Inputs	PDO Mapping TxPDO 65	UINT8	RO	0x26 (38_{dec})
1A40:01	SubIndex 001	1. PDO Mapping entry (1 bit align)	UINT32	RO	0x0000:00, 1
1A40:02	SubIndex 002	2. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x02 (DMC__FeedbackStatus__Latch extern valid))	UINT32	RO	0x6040:02, 1
1A40:03	SubIndex 003	3. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x03 (DMC__FeedbackStatus__Set counter done))	UINT32	RO	0x6040:03, 1
1A40:04	SubIndex 004	4. PDO Mapping entry (9 bits align)	UINT32	RO	0x0000:00, 9
1A40:05	SubIndex 005	5. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x0D (DMC__FeedbackStatus__Status of extern latch))	UINT32	RO	0x6040:0D, 1
1A40:06	SubIndex 006	6. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1A40:07	SubIndex 007	7. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x11 (DMC__DriveStatus__Ready to enable))	UINT32	RO	0x6040:11, 1
1A40:08	SubIndex 008	8. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x12 (DMC__DriveStatus__Ready))	UINT32	RO	0x6040:12, 1
1A40:09	SubIndex 009	9. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x13 (DMC__DriveStatus__Warning))	UINT32	RO	0x6040:13, 1
1A40:0A	SubIndex 010	10. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x14 (DMC__DriveStatus__Error))	UINT32	RO	0x6040:14, 1
1A40:0B	SubIndex 011	11. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x15 (DMC__DriveStatus__Moving positive))	UINT32	RO	0x6040:15, 1
1A40:0C	SubIndex 012	12. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x16 (DMC__DriveStatus__Moving negative))	UINT32	RO	0x6040:16, 1
1A40:0D	SubIndex 013	13. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1A40:0E	SubIndex 014	14. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x1C (DMC__DriveStatus__Digital input 1))	UINT32	RO	0x6040:1C, 1
1A40:0F	SubIndex 015	15. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x1D (DMC__DriveStatus__Digital input 2))	UINT32	RO	0x6040:1D, 1
1A40:10	SubIndex 016	16. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1A40:11	SubIndex 017	17. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x21 (DMC__PositioningStatus__Busy))	UINT32	RO	0x6040:21, 1

Index	Name	Meaning	Data type	Flags	Default
1A40:12	SubIndex 018	18. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x22 (DMC__PositioningStatus__In-Target))	UINT32	RO	0x6040:22, 1
1A40:13	SubIndex 019	19. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x23 (DMC__PositioningStatus__Warning))	UINT32	RO	0x6040:23, 1
1A40:14	SubIndex 020	20. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x24 (DMC__PositioningStatus__Error))	UINT32	RO	0x6040:24, 1
1A40:15	SubIndex 021	21. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x25 (DMC__PositioningStatus__Calibrated))	UINT32	RO	0x6040:25, 1
1A40:16	SubIndex 022	22. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x26 (DMC__PositioningStatus__Accelerate))	UINT32	RO	0x6040:26, 1
1A40:17	SubIndex 023	23. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x27 (DMC__PositioningStatus__Decelerate))	UINT32	RO	0x6040:27, 1
1A40:18	SubIndex 024	24. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x28 (DMC__PositioningStatus__Ready to execute))	UINT32	RO	0x6040:28, 1
1A40:19	SubIndex 025	25. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1A40:1A	SubIndex 026	26. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x31 (DMC__Set position))	UINT32	RO	0x6040:31, 64
1A40:1B	SubIndex 027	27. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x32 (DMC__Set velocity))	UINT32	RO	0x6040:32, 16
1A40:1C	SubIndex 028	28. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x33 (DMC__Actual drive time))	UINT32	RO	0x6040:33, 32
1A40:1D	SubIndex 029	29. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x34 (DMC__Actual position lag))	UINT32	RO	0x6040:34, 64
1A40:1E	SubIndex 030	30. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x35 (DMC__Actual velocity))	UINT32	RO	0x6040:35, 16
1A40:1F	SubIndex 031	31. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x36 (DMC__Actual position))	UINT32	RO	0x6040:36, 64
1A40:20	SubIndex 032	32. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x37 (DMC__Error id))	UINT32	RO	0x6040:37, 32
1A40:21	SubIndex 033	33. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x38 (DMC__Input cycle counter))	UINT32	RO	0x6040:38, 8
1A40:22	SubIndex 034	34. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x39 (DMC__Channel id))	UINT32	RO	0x6040:39, 8
1A40:23	SubIndex 035	35. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x3A (DMC__Latch value))	UINT32	RO	0x6040:3A, 64
1A40:24	SubIndex 036	36. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x3B (DMC__Cyclic info data 1))	UINT32	RO	0x6040:3B, 16

Index	Name	Meaning	Data type	Flags	Default
1A40:25	SubIndex 037	37. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x3C (DMC__Cyclic info data 2))	UINT32	RO	0x6040:3C, 16
1A40:26	SubIndex 038	38. PDO Mapping entry (64 bits align)	UINT32	RO	0x0000:00, 64

Index 1A41 DMC TxPDO-Map Inputs 32 Bit

Index	Name	Meaning	Data type	Flags	Default
1A41:0	DMC TxPDO-Map Inputs 32 Bit	PDO Mapping TxPDO 66	UINT8	RO	0x2A (42_{dec})
1A41:01	SubIndex 001	1. PDO Mapping entry (1 bit align)	UINT32	RO	0x0000:00, 1
1A41:02	SubIndex 002	2. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x02 (DMC__FeedbackStatus__Latch extern valid))	UINT32	RO	0x6040:02, 1
1A41:03	SubIndex 003	3. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x03 (DMC__FeedbackStatus__Set counter done))	UINT32	RO	0x6040:03, 1
1A41:04	SubIndex 004	4. PDO Mapping entry (9 bits align)	UINT32	RO	0x0000:00, 9
1A41:05	SubIndex 005	5. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x0D (DMC__FeedbackStatus__Status of extern latch))	UINT32	RO	0x6040:0D, 1
1A41:06	SubIndex 006	6. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1A41:07	SubIndex 007	7. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x11 (DMC__DriveStatus__Ready to enable))	UINT32	RO	0x6040:11, 1
1A41:08	SubIndex 008	8. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x12 (DMC__DriveStatus__Ready))	UINT32	RO	0x6040:12, 1
1A41:09	SubIndex 009	9. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x13 (DMC__DriveStatus__Warning))	UINT32	RO	0x6040:13, 1
1A41:0A	SubIndex 010	10. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x14 (DMC__DriveStatus__Error))	UINT32	RO	0x6040:14, 1
1A41:0B	SubIndex 011	11. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x15 (DMC__DriveStatus__Moving positive))	UINT32	RO	0x6040:15, 1
1A41:0C	SubIndex 012	12. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x16 (DMC__DriveStatus__Moving negative))	UINT32	RO	0x6040:16, 1
1A41:0D	SubIndex 013	13. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1A41:0E	SubIndex 014	14. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x1C (DMC__DriveStatus__Digital input 1))	UINT32	RO	0x6040:1C, 1
1A41:0F	SubIndex 015	15. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x1D (DMC__DriveStatus__Digital input 2))	UINT32	RO	0x6040:1D, 1
1A41:10	SubIndex 016	16. PDO Mapping entry (3 bits align)	UINT32	RO	0x0000:00, 3
1A41:11	SubIndex 017	17. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x21 (DMC__PositioningStatus__Busy))	UINT32	RO	0x6040:21, 1

Index	Name	Meaning	Data type	Flags	Default
1A41:12	SubIndex 018	18. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x22 (DMC__PositioningStatus__In-Target))	UINT32	RO	0x6040:22, 1
1A41:13	SubIndex 019	19. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x23 (DMC__PositioningStatus__Warning))	UINT32	RO	0x6040:23, 1
1A41:14	SubIndex 020	20. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x24 (DMC__PositioningStatus__Error))	UINT32	RO	0x6040:24, 1
1A41:15	SubIndex 021	21. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x25 (DMC__PositioningStatus__Calibrated))	UINT32	RO	0x6040:25, 1
1A41:16	SubIndex 022	22. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x26 (DMC__PositioningStatus__Accelerate))	UINT32	RO	0x6040:26, 1
1A41:17	SubIndex 023	23. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x27 (DMC__PositioningStatus__Decelerate))	UINT32	RO	0x6040:27, 1
1A41:18	SubIndex 024	24. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x28 (DMC__PositioningStatus__Ready to execute))	UINT32	RO	0x6040:28, 1
1A41:19	SubIndex 025	25. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1A41:1A	SubIndex 026	26. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x31 (DMC__Set position))	UINT32	RO	0x6040:31, 32
1A41:1B	SubIndex 027	27. PDO Mapping entry (32 bits align)	UINT32	RO	0x0000:00, 32
1A41:1C	SubIndex 028	28. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x32 (DMC__Set velocity))	UINT32	RO	0x6040:32, 16
1A41:1D	SubIndex 029	29. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x33 (DMC__Actual drive time))	UINT32	RO	0x6040:33, 32
1A41:1E	SubIndex 030	30. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x34 (DMC__Actual position lag))	UINT32	RO	0x6040:34, 32
1A41:1F	SubIndex 031	31. PDO Mapping entry (32 bits align)	UINT32	RO	0x0000:00, 32
1A41:20	SubIndex 032	32. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x35 (DMC__Actual velocity))	UINT32	RO	0x6040:35, 16
1A41:21	SubIndex 033	33. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x36 (DMC__Actual position))	UINT32	RO	0x6040:36, 32
1A41:22	SubIndex 034	34. PDO Mapping entry (32 bits align)	UINT32	RO	0x0000:00, 32
1A41:23	SubIndex 035	35. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x37 (DMC__Error id))	UINT32	RO	0x6040:37, 32
1A41:24	SubIndex 036	36. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x38 (DMC__Input cycle counter))	UINT32	RO	0x6040:38, 8
1A41:25	SubIndex 037	37. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x39 (DMC__Channel id))	UINT32	RO	0x6040:39, 8

Index	Name	Meaning	Data type	Flags	Default
1A41:26	SubIndex 038	38. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x3A (DMC__Latch value))	UINT32	RO	0x6040:3A, 32
1A41:27	SubIndex 039	39. PDO Mapping entry (32 bits align)	UINT32	RO	0x0000:00, 32
1A41:28	SubIndex 040	40. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x3B (DMC__Cyclic info data 1))	UINT32	RO	0x6040:3B, 16
1A41:29	SubIndex 041	41. PDO Mapping entry (object 0x6040 (DMC Inputs), entry 0x3C (DMC__Cyclic info data 2))	UINT32	RO	0x6040:3C, 16
1A41:2A	SubIndex 042	42. PDO Mapping entry (64 bits align)	UINT32	RO	0x0000:00, 64

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x02 (2_{dec})
1C12:01	Subindex 001	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1600 (5632_{dec})
1C12:02	Subindex 002	2. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1606 (5638_{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})
1C12:08	Subindex 008	8. 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	0x03 (3_{dec})
1C13:01	Subindex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A00 (6656_{dec})
1C13:02	Subindex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A01 (6657_{dec})
1C13:03	Subindex 003	3. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A06 (6662_{dec})
1C13:04	Subindex 004	4. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:05	Subindex 005	5. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:06	Subindex 006	6. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:07	Subindex 007	7. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:08	Subindex 008	8. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:09	Subindex 009	9. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:0A	Subindex 010	10. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:0B	Subindex 011	11. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:0C	Subindex 012	12. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:0D	Subindex 013	13. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:0E	Subindex 014	14. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:0F	Subindex 015	15. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
1C13:10	Subindex 016	16. 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	0x0000 (0_{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	0x00000000 (0_{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 Standard objects [▶ 189])	UINT16	RO	0x0000 (0_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x00000000 (0_{dec})
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:07	Minimum delay time		UINT32	RO	0x00000000 (0_{dec})
1C32:08	Command	0: Measurement of the local cycle time is stopped 1: Measurement of the local cycle time is started The entries Standard objects [▶ 189], Standard objects [▶ 189], Standard objects [▶ 189], Standard objects [▶ 189], Standard objects [▶ 189], Standard objects [▶ 189], Standard objects [▶ 189] 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})

Index (hex)	Name	Meaning	Data type	Flags	Default
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	0x0000 (0_{dec})
1C33:02	Cycle time	as Standard objects [▶ 189]	UINT32	RW	0x00000000 (0_{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 Standard objects [▶ 189] or Standard objects [▶ 189])	UINT16	RO	0x0000 (0_{dec})
1C33:05	Minimum cycle time	as Standard objects [▶ 189]	UINT32	RO	0x00000000 (0dec)
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 Standard objects [▶ 189]	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 Standard objects [▶ 189]	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	as Standard objects [▶ 189]	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	as Standard objects [▶ 189]	UINT16	RO	0x0000 (0_{dec})
1C33:20	Sync error	as Standard objects [▶ 189]	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	0x0003 (3_{dec})

Index F008 Code word

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

Index F010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default
F010:0	Module list	Module list	UINT8	RW	0x03 (3_{dec})
F010:01	SubIndex 001	Encoder profile number DS402 / MDP 513	UINT32	RW	0x00000201 (513_{dec})
F010:02	SubIndex 002	Profile number drive DS402 / MDP 742	UINT32	RW	0x000002E6 (742_{dec})
F010:03	SubIndex 003	Reserved	UINT32	RW	0x00000064 (100_{dec})

Index F081 Download revision

Index (hex)	Name	Meaning	Data type	Flags	Default
F081:0	Download revision	Download revision	UINT8	RO	0x01 (1_{dec})
F081:01	Revision number	Revision number of the terminal Relevant as a startup list entry for compatibility	UINT32	RW	0x00000000 (0_{dec})

Index F083 BTN

Index (hex)	Name	Meaning	Data type	Flags	Default
F083:0	BTN	Beckhoff Traceability Number	STRING	RO	

9 Appendix

9.1 EtherCAT AL Status Codes

For detailed information please refer to the EtherCAT system description.

9.2 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.
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!

Hardware (HW)	Firmware (FW)	Revision no.	Release date
00-02*	01	EL7411-0000-0016	
	02	EL7411-0000-0017	
	03		
	04	EL7411-0000-0018	

*) This is the current compatible firmware/hardware version at the time of the preparing this documentation. Check on the Beckhoff web page whether more up-to-date documentation is available.

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

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

NOTE

Only use TwinCAT 3 software!

A firmware update of Beckhoff IO devices must only be performed with a TwinCAT 3 installation. It is recommended to build as up-to-date as possible, available for free download on the Beckhoff website https://www.beckhoff.com/en-us/.

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

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

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

Storage locations

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

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

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

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

Simplified update by bundle firmware

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

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

Following the update, its success should be verified

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

NOTE

Risk of damage to the device!

✓ Note the following when downloading new device files

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

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

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

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

9.3.1 Device description ESI file/XML

NOTE

Attention regarding update of the ESI description/EEPROM

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

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

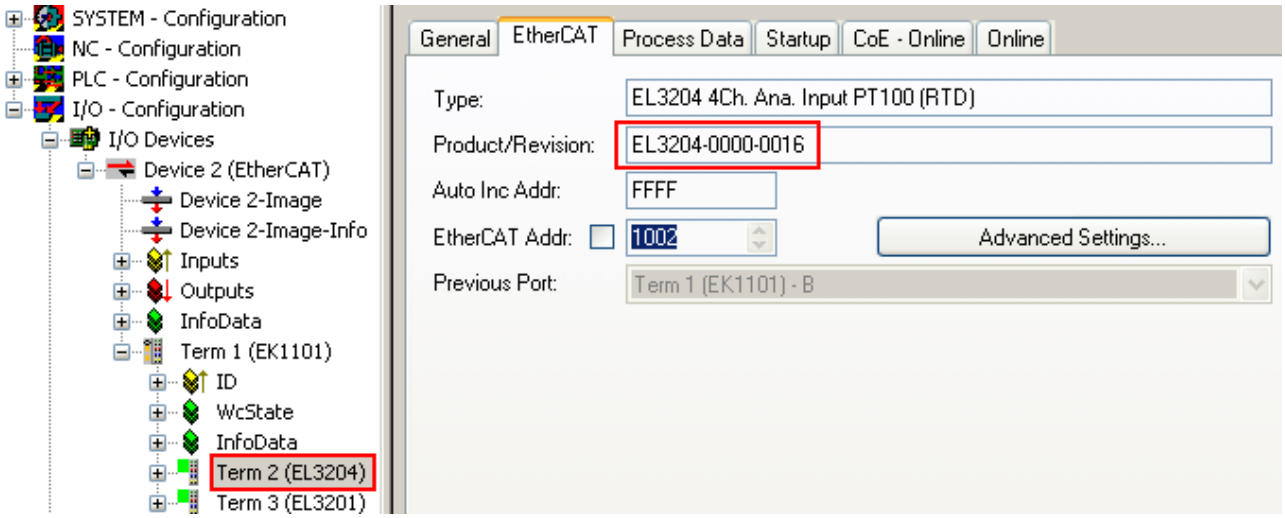


Fig. 101: Device identifier consisting of name EL3204-0000 and revision -0016

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:

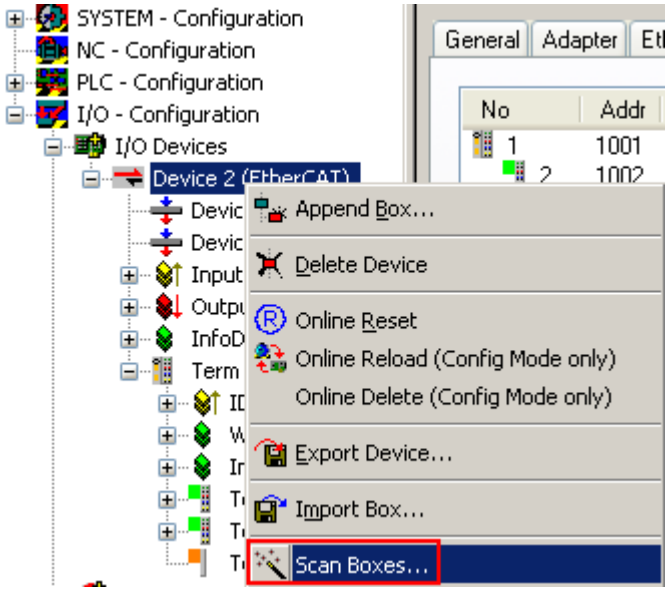


Fig. 102: Scan the subordinate field by right-clicking on the EtherCAT device

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

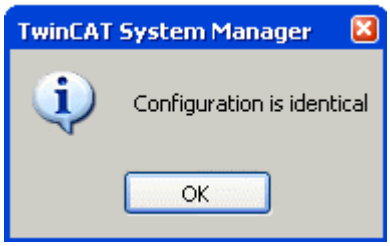


Fig. 103: Configuration is identical

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

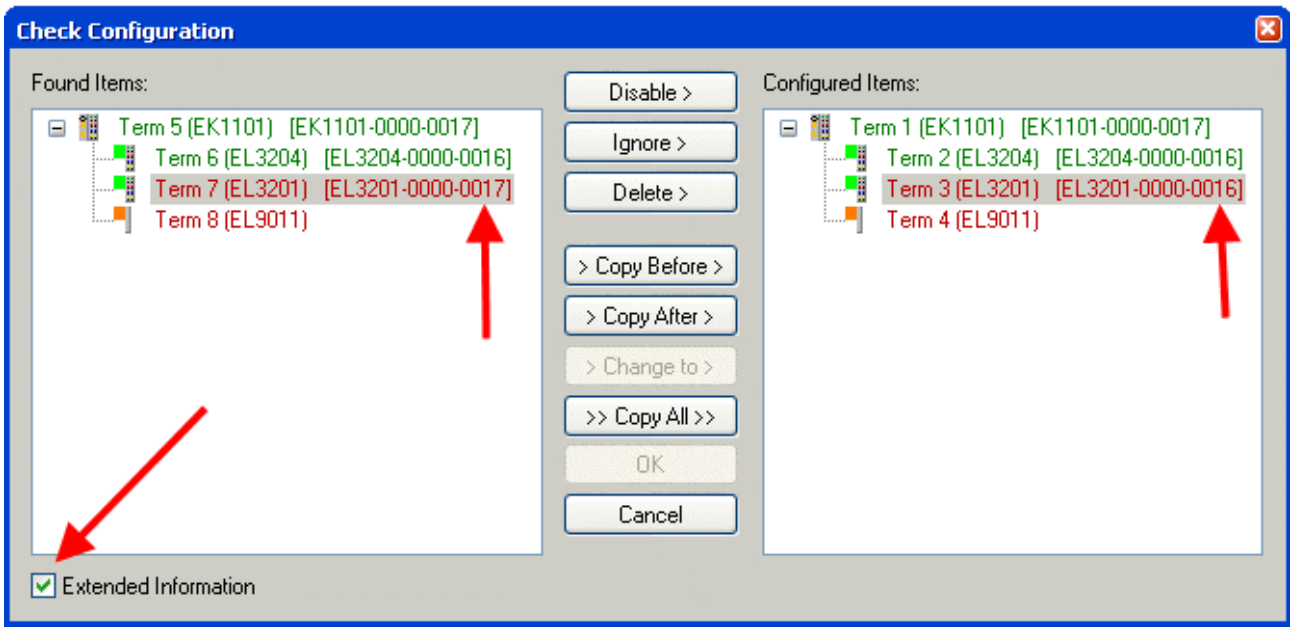


Fig. 104: Change dialog

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

Changing the ESI slave identifier

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

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

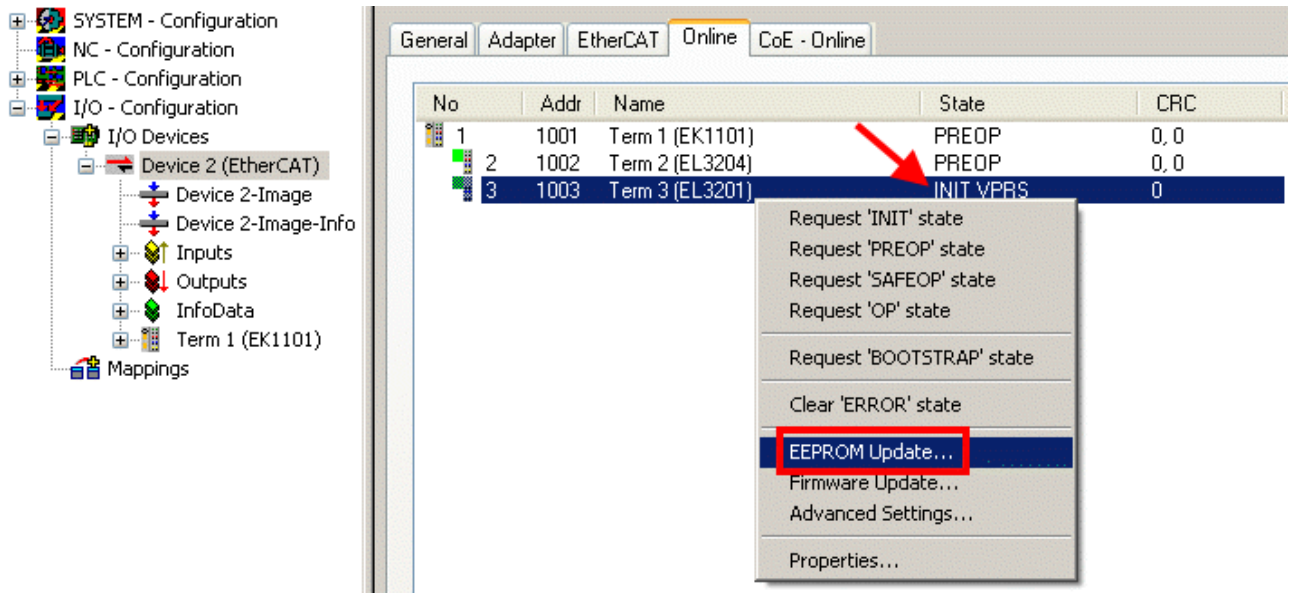


Fig. 105: EEPROM Update

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

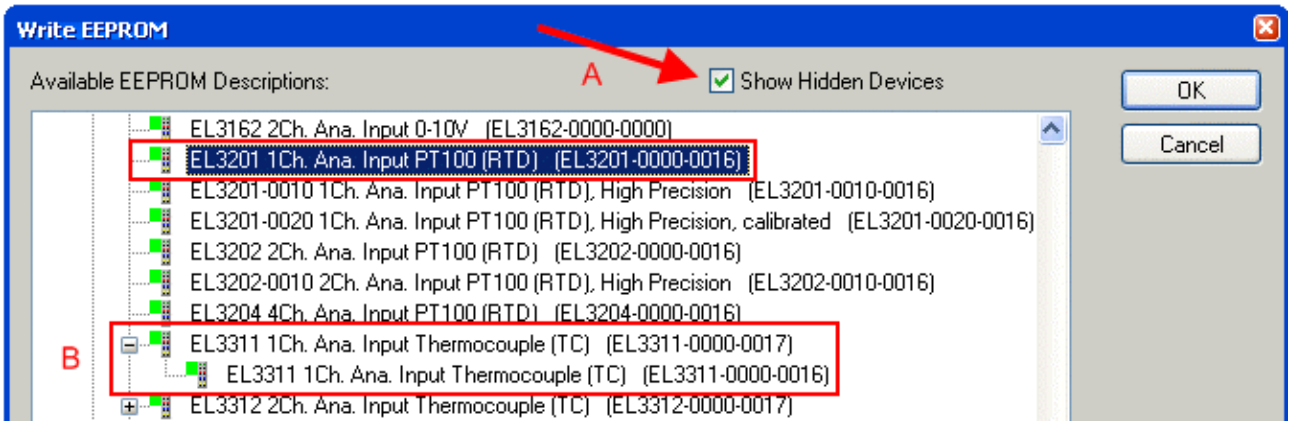


Fig. 106: Selecting the new ESI

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

The change only takes effect after a restart.

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

9.3.2 Firmware explanation

Determining the firmware version

Determining the version via the System Manager

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

CoE Online and Offline CoE

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

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

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

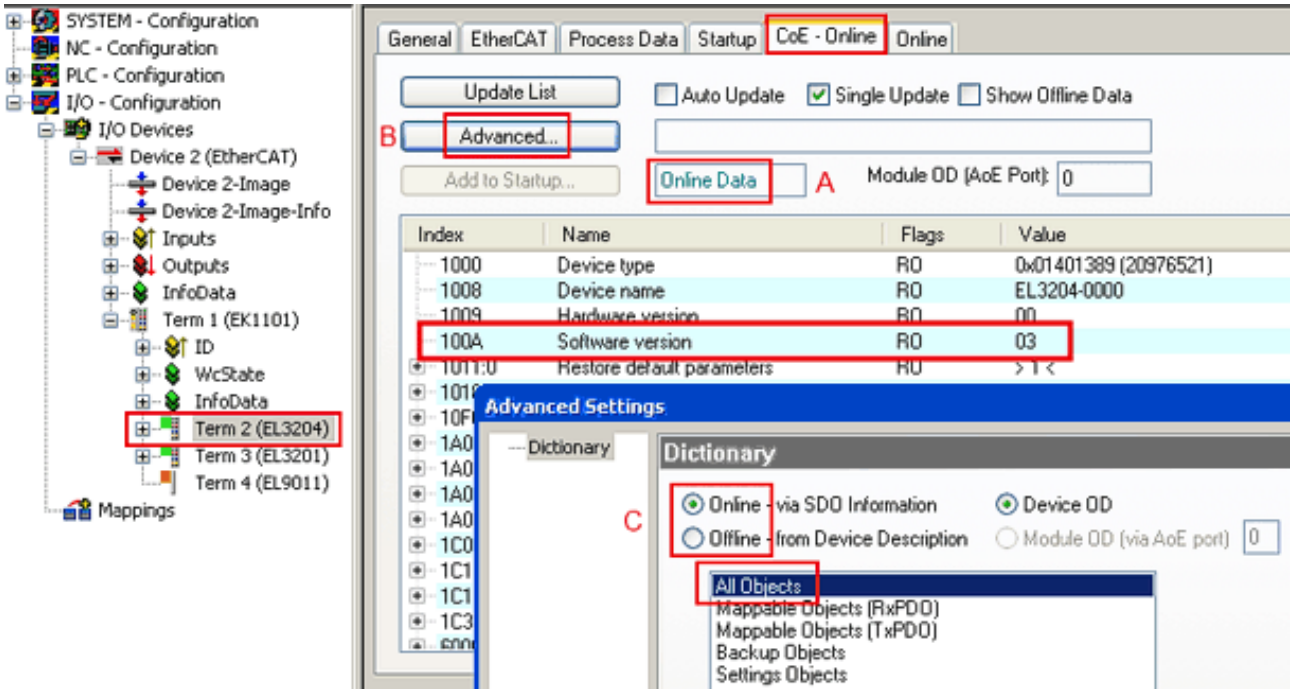


Fig. 107: Display of EL3204 firmware version

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

9.3.3 Updating controller firmware *.efw

CoE directory

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

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

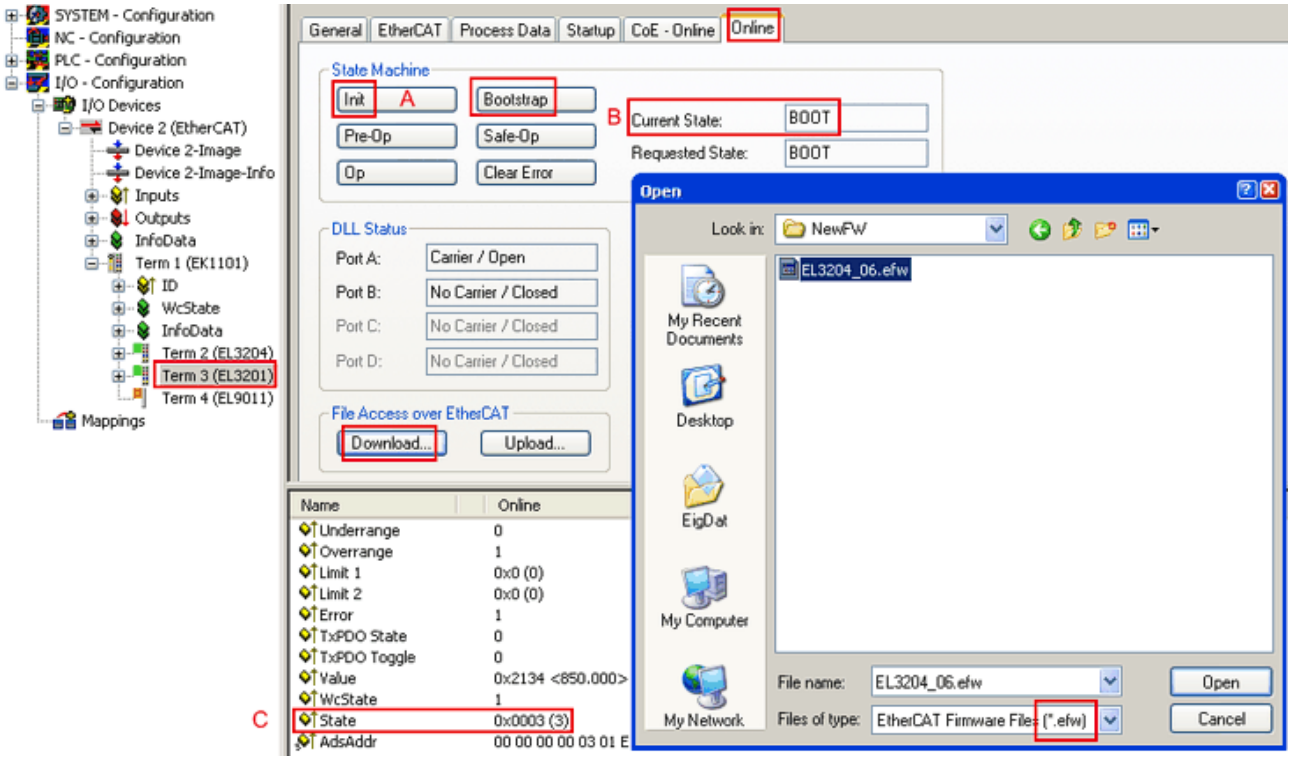
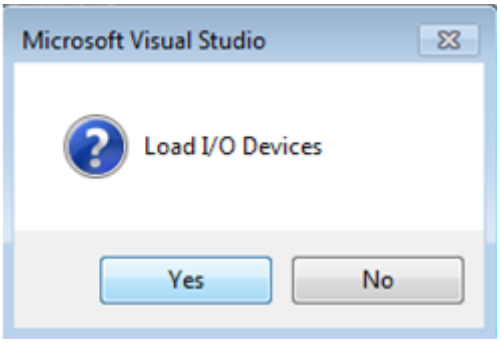
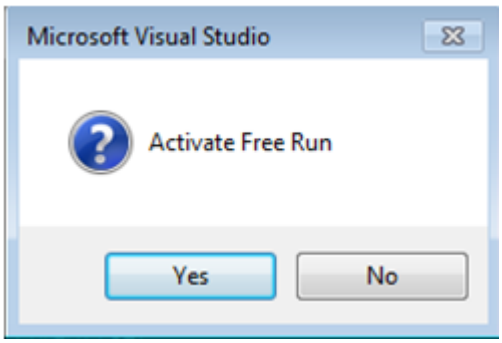


Fig. 108: Firmware Update

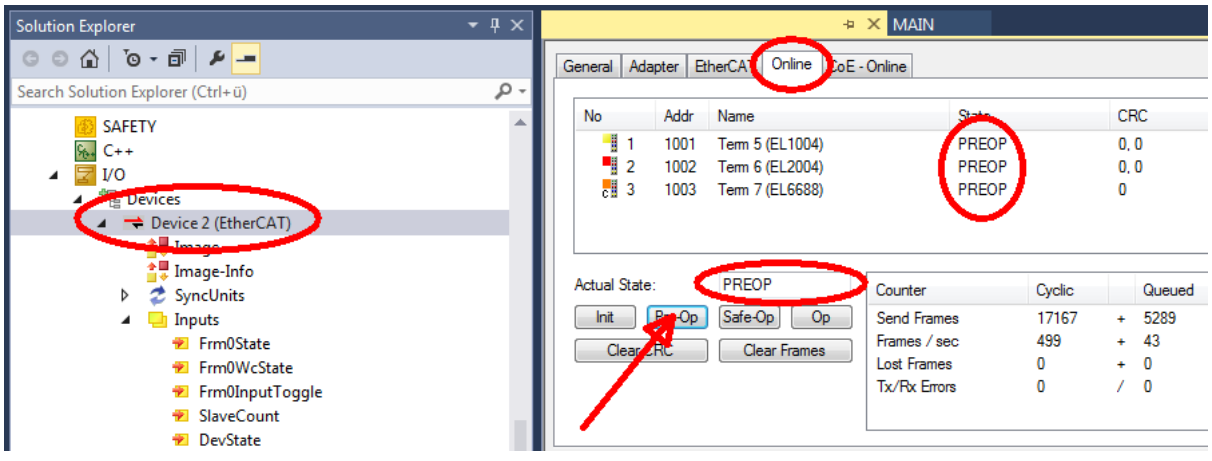
Proceed as follows, unless instructed otherwise by Beckhoff support. Valid for TwinCAT 2 and 3 as EtherCAT master.

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



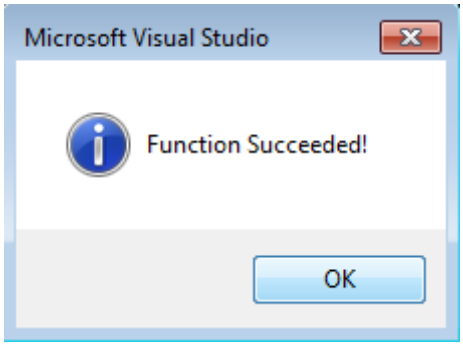


- Switch EtherCAT Master to PreOP



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

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



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

9.3.4 FPGA firmware *.rbf

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

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

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

Determining the version via the System Manager

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

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

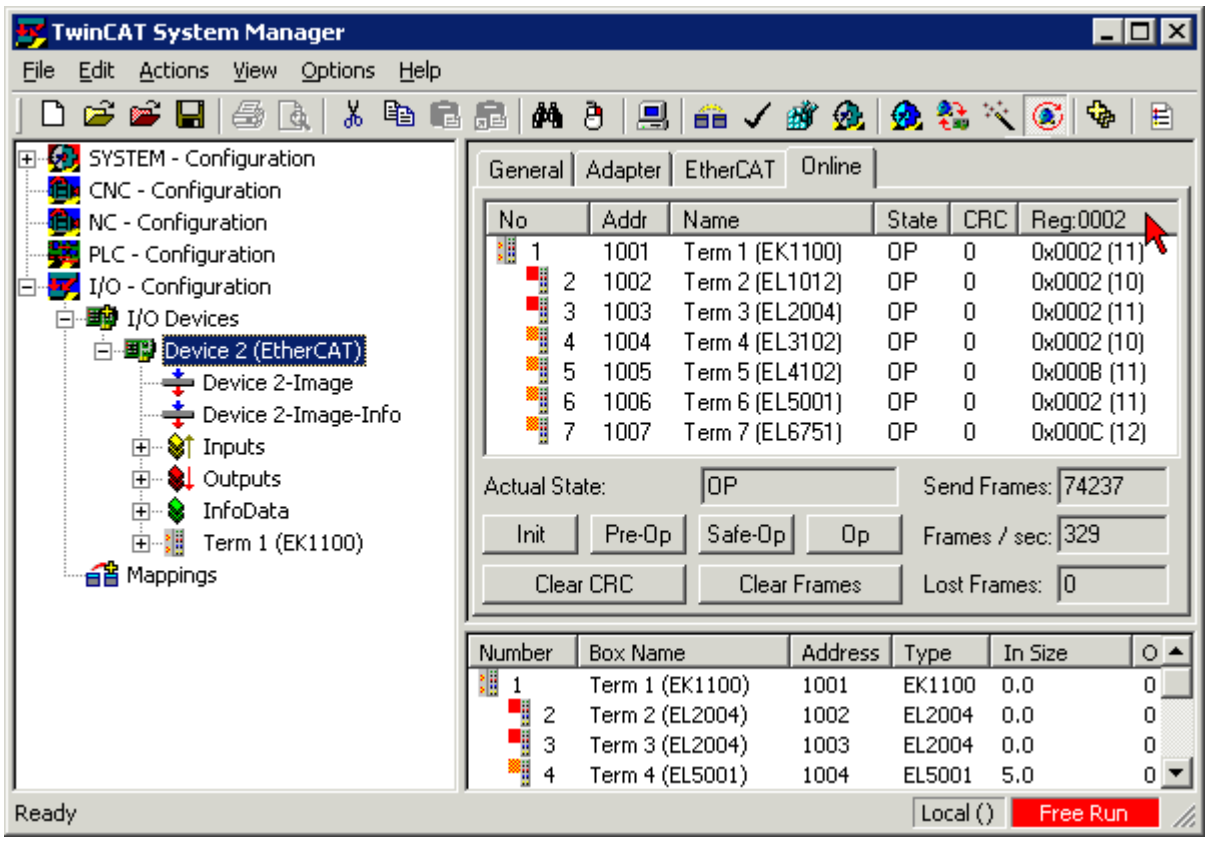


Fig. 109: FPGA firmware version definition

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

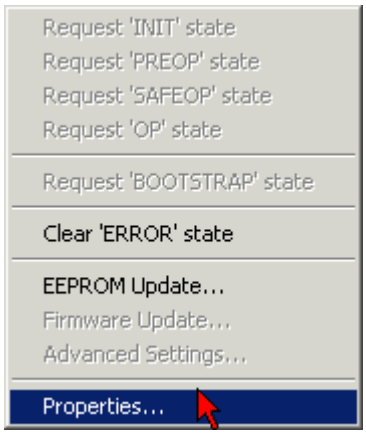


Fig. 110: Context menu *Properties*

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

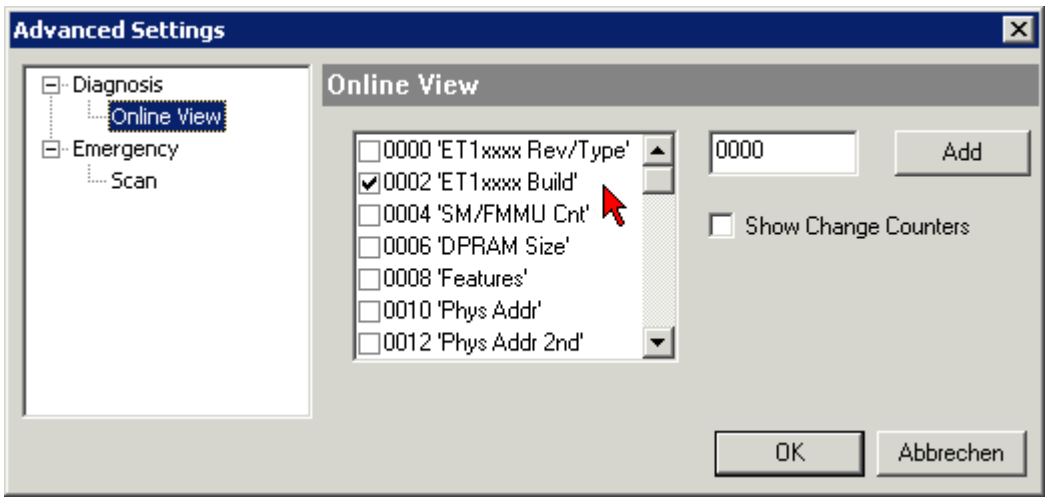


Fig. 111: Dialog *Advanced Settings*

Update

For updating the FPGA firmware

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

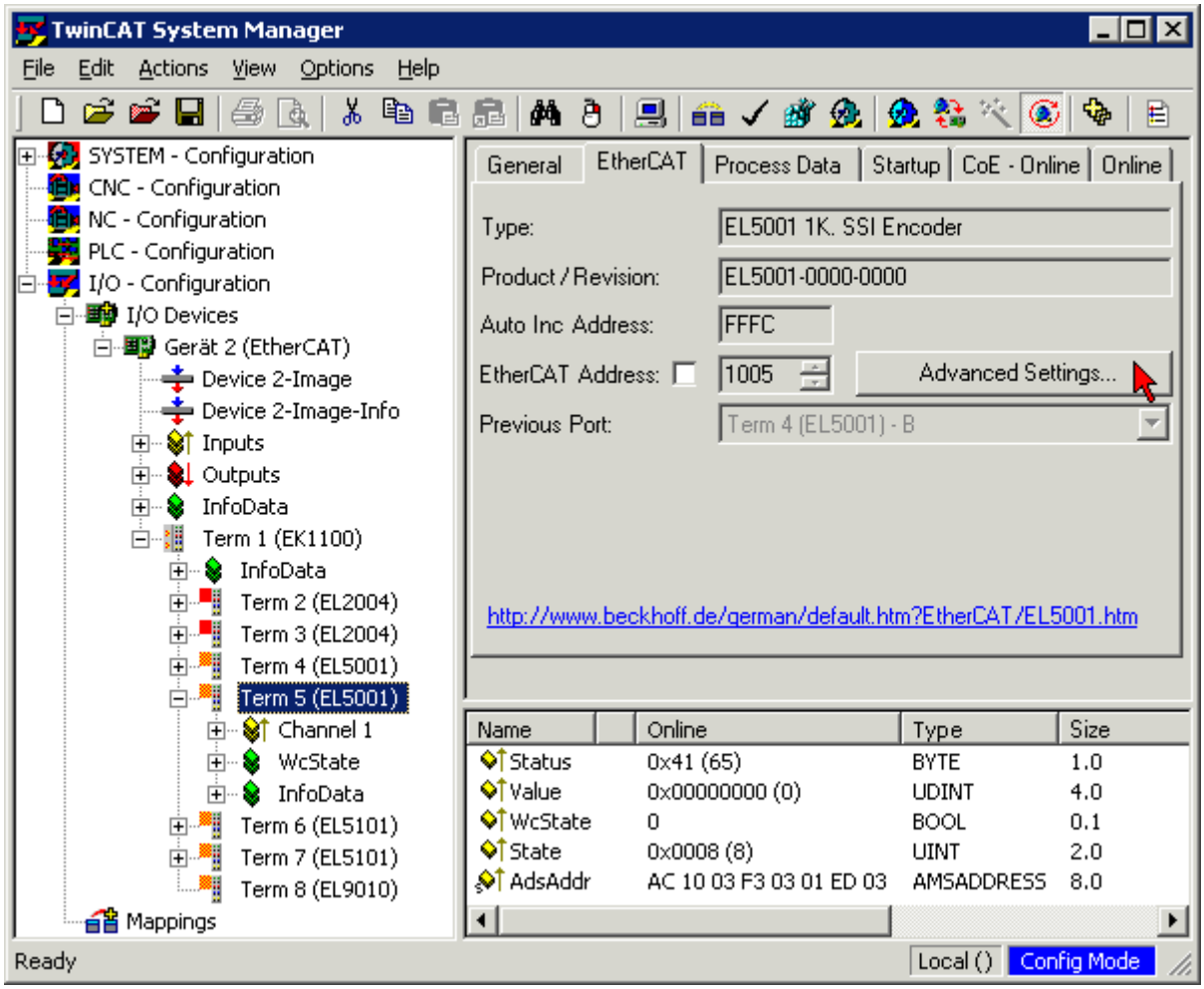
Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

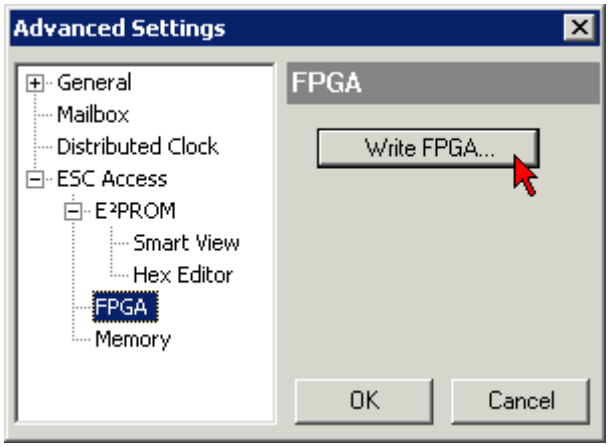
The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

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

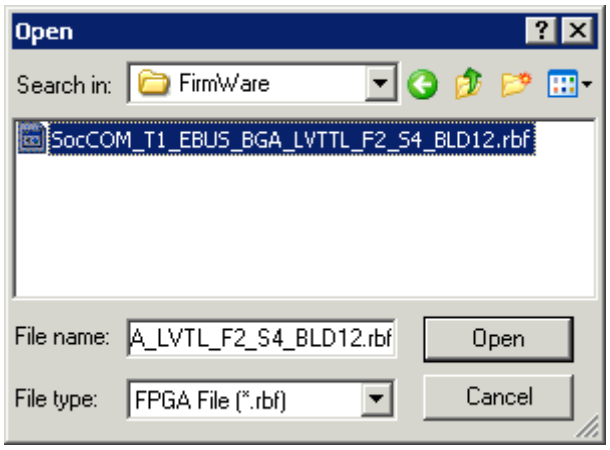
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and
  click the *Advanced Settings* button in the *EtherCAT* tab:



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



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



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

NOTE

Risk of damage to the device!

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

9.3.5 Simultaneous updating of several EtherCAT devices

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

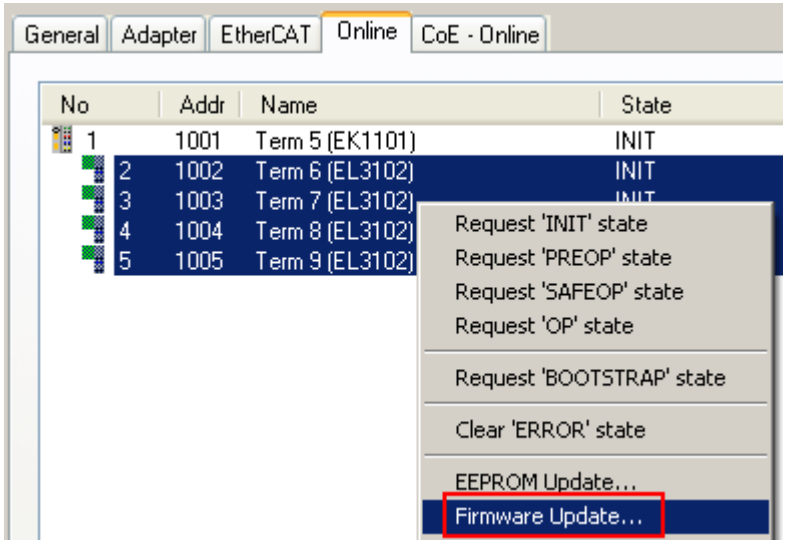


Fig. 112: Multiple selection and firmware update

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

9.4 Restoring the delivery state

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

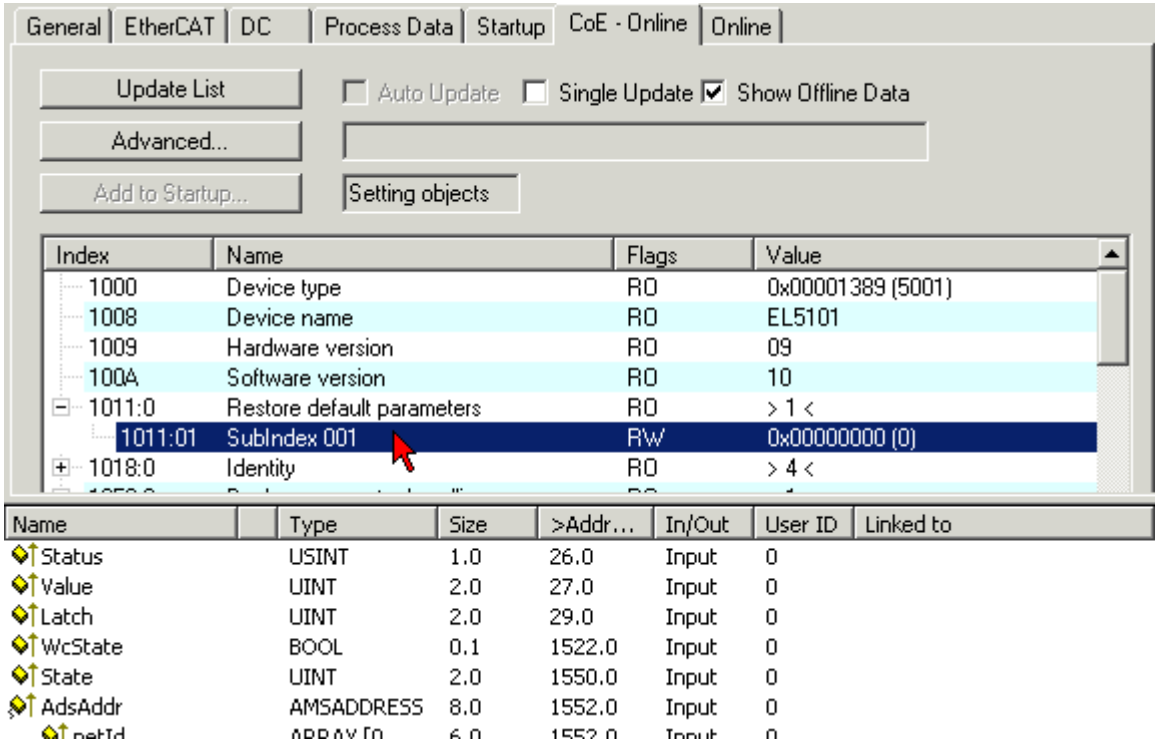


Fig. 113: Selecting the *Restore default parameters* PDO

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

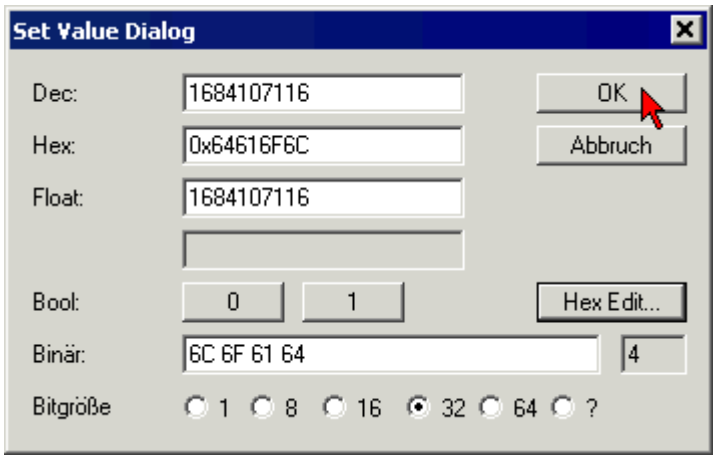


Fig. 114: Entering a restore value in the Set Value dialog

Alternative restore value

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

9.5 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/el7411/

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