BECKHOFF New Automation Technology

Documentation | EN

EL6821

EtherCAT Terminal, 1-channel Communckation interface, DALI-2, master/ power supply

2025-09-10 | Version: 1.0.0

Table of Contents

1 Foreword

1.1 Notes on the documentation

Intended audience

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.
The qualified personnel is obliged to always use the currently valid documentation.

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®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®,
Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® and XTS® 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.

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

Copyright

Third-party brands

Trademarks of third parties may be used in this documentation. You can find the trademark notices here: https://www.beckhoff.com/trademarks

1.2 Guide through documentation

NOTICE

Further components of documentation

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

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

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

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

If you have any suggestions or proposals for our documentation, please send us an e-mail stating the documentation title and version number to: documentation@beckhoff.com

1.3 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER
Hazard with high risk of death or serious injury.

⚠ WARNING
Hazard with medium risk of death or serious injury.

⚠ CAUTION
There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE
The environment, equipment, or data may be damaged.

Information on handling the product

This information includes, for example:
recommendations for action, assistance or further information on the product.

1.4 Documentation issue status

Version	Comment
1.0.0	• First release
0.3.0	• Chapter *Technical data* updated
0.2.0	• Chapter *Technical data* updated • Chapter *DALI* updated • Chapter *Device diagnostic functions* updated • Chapter *EL6821 - Object description and parameterization* updated • Document structure updated
0.1.0	• First draft

1.5 Version identification of EtherCAT devices

1.5.1 General notes on marking

Designation

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

- family key
- type
- version
- revision

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

Notes

- The elements mentioned above result in the **technical designation**. EL3314-0000-0016 is used in the example below.
- EL3314-0000 is the order identifier, in the case of “-0000” usually abbreviated to EL3314. “-0016” is the EtherCAT revision.
- The **order identifier** is made up of
  - family key (EL, EP, CU, ES, KL, CX, etc.)
  - type (3314)
  - version (-0000)
- The **revision** -0016 shows the technical progress, such as the extension of features with regard to the EtherCAT communication, and is managed by Beckhoff.
  In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation.
  Associated and synonymous with each revision there is usually a description (ESI, EtherCAT Slave Information) in the form of an XML file, which is available for download from the Beckhoff web site.
  From 2014/01 the revision is shown on the outside of the IP20 terminals, see Fig. *“EL2872 with revision 0022 and serial number 01200815”*.
- The type, version and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

1.5.2 Version identification of EL terminals

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

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

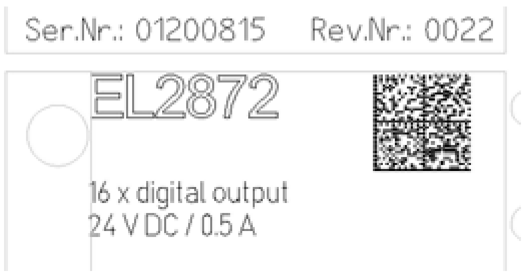


Fig. 1: EL2872 with revision 0022 and serial number 01200815

1.5.3 Beckhoff Identification Code (BIC)

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

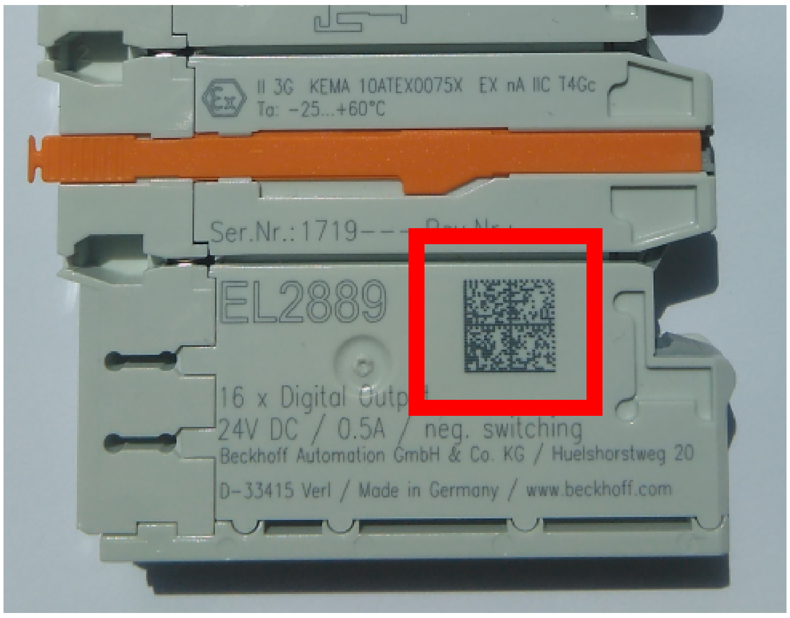

II 3G KEMA 10ATEX0075X EX nA IIC T4Gc
Ta: -25...+60°C
Ser.Nr.:1719---
EL2889
16 x Digital Outp
24V DC / 0.5A / neg. switching
Beckhoff Automation GmbH & Co. KG / Huelshorstweg 20
D-33415 Verl / Made in Germany / www.beckhoff.com

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

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

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

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

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

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

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

Posi-tion	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	**Beckhoff order number**	1P	8	**1P**072222
2	Beckhoff Traceability Number (BTN**)**	**Unique serial number, see note below**	SBTN	12	**SBTN**k4p562d7
3	Article description	**Beckhoff article description, e.g. EL1008**	1K	32	**1K**EL1809
4	Quantity	**Quantity in packaging unit, e.g. 1, 10, etc.**	Q	6	**Q**1
5	Batch number	Optional: Year and week of production	2P	14	**2P**401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	**51S**678294
7	Variant number	Optional: Product variant number on the basis of standard products	30P	12	**30P**F971, 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**SBTN**k4p562d7**1K**EL1809 **Q**1 **51S**678294

Accordingly as DMC:

Fig. 3: Example DMC **1P**072222**SBTN**k4p562d7**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.

NOTICE

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 documentation.

1.5.4 Electronic access to the BIC (eBIC)

Electronic BIC (eBIC)

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

The interface that the product can be electronically addressed by is crucial for the electronic readout.

K-bus devices (IP20, IP67)

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

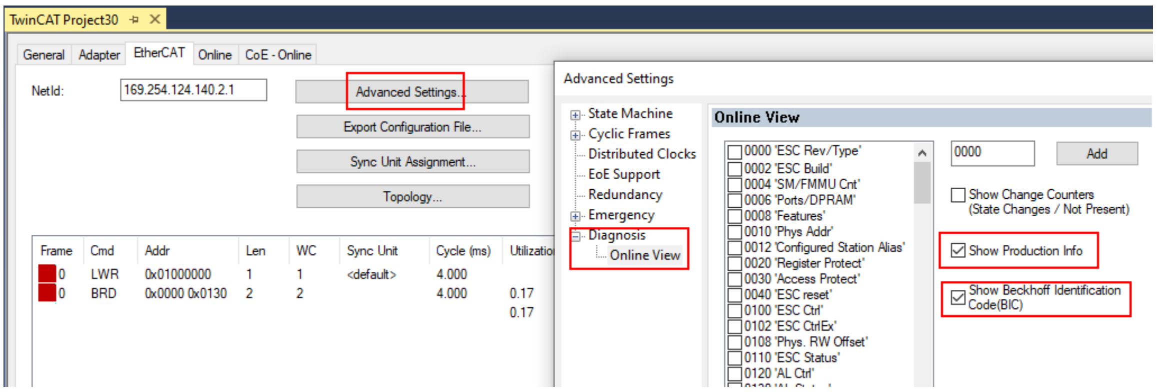
EtherCAT devices (IP20, IP67)

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

Beckhoff also stores the eBIC in the ESI-EEPROM. The eBIC was introduced into Beckhoff IO production (terminals, box modules) in 2020; as of 2023, implementation is largely complete.

The user can electronically access the eBIC (if present) 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 "Show Beckhoff Identification Code (BIC)" checkbox under EtherCAT → Advanced Settings → Diagnostics:



NetId: 169.254.124.140.2.1

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

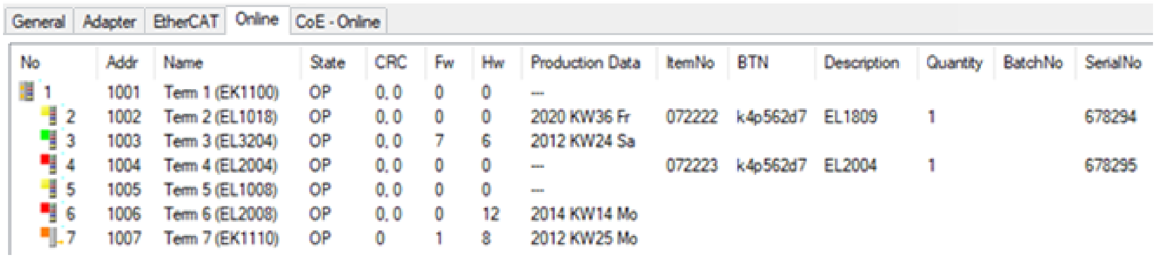
Online View

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

0000

- [ ] Show Change Counters (State Changes / Not Present)
- [✓] Show Production Info
- [✓] Show Beckhoff Identification Code(BIC)

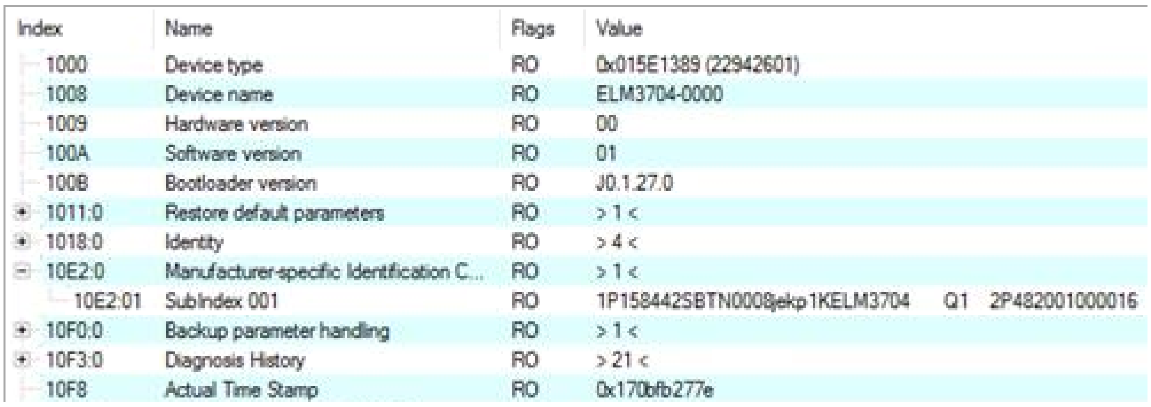
  - The BTN and its contents are then displayed:



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

  - Note: As shown in the figure, the production data HW version, FW version, and production date, which have been programmed since 2012, can also be displayed with "Show production info".
  - Access from the PLC: From TwinCAT 3.1. build 4024.24, the functions *FB_EcReadBIC* and *FB_EcReadBTN* for reading into the PLC are available in the Tc2_EtherCAT library from v3.3.19.0.
- EtherCAT devices with a CoE directory may also have the object 0x10E2:01 to display their own eBIC, which can also be easily accessed by the PLC:

- 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 preferentially introduced into stock products in the course of necessary firmware revision.
- From TwinCAT 3.1. build 4024.24, the functions *FB_EcCoEReadBIC* and *FB_EcCoEReadBTN* for reading into the PLC are available in the Tc2_EtherCAT library from v3.3.19.0

• The following auxiliary functions are available for processing the BIC/BTN data in the PLC in *Tc2_Utilities* as of TwinCAT 3.1 build 4024.24
  - F_SplitBIC: The function splits the Beckhoff Identification Code (BIC) sBICValue into its components using known identifiers and returns the recognized substrings in the ST_SplittedBIC structure as a return value
  - BIC_TO_BTN: The function extracts the BTN from the BIC and returns it as a return value

• Note: If there is further electronic 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 written as an additional category in the ESI-EEPROM during device production. The structure of the ESI content is largely dictated by the ETG specifications, therefore the additional vendor-specific content is stored using a category in accordance with the ETG.2010. ID 03 tells all EtherCAT masters that they may not overwrite these data in the event of an update or restore the data after an ESI update.
  The structure follows the content of the BIC, see here. The EEPROM therefore requires approx. 50..200 bytes of memory.

• 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 which each have 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, and DeviceNet devices

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

2 Product description

2.1 Introduction

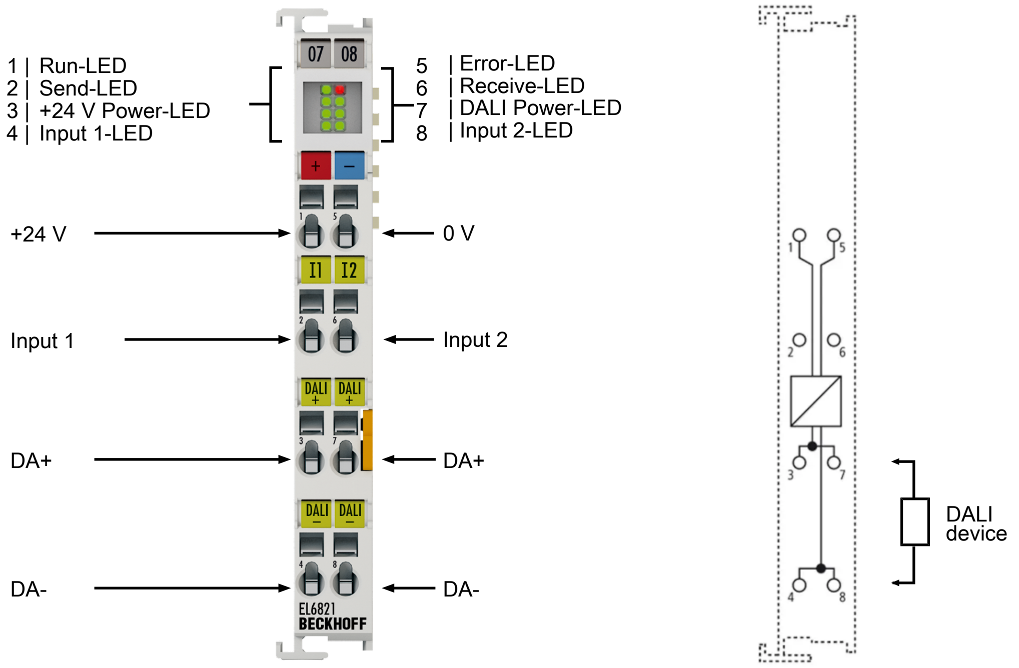


Fig. 4: EL6821

1-channel communication interface, DALI-2, master/power supply

The EL6821 allows the connection of up to 64 DALI/DALI-2 slaves and 64 DALI-2 input devices.

The TwinCAT 3 System Manager makes it easy to configure and parameterize DALI devices. Programming is performed exclusively via TwinCAT 3 function blocks.

The EL6821 contains an integrated DALI bus power supply that can be switched off. The electrically isolated input voltage of 24 V_{DC} must be supplied via the EL9562 power supply terminal or via the PS1111-2402-0002, PS1111-2403-0000 or PS1111-2403-0002 power supplies in order to comply with the certification requirements.

The EL6821 is certified in accordance with the DALI-2 standard.

DALI-2 Part 101, Version 2.0 and Part 103, Version 2.0

Quick links

EtherCAT basics	Technical data [▸ 16]	Integration in TwinCAT [▸ 107]
DALI basics [▸ 17]	Connection [▸ 54]	Settings [▸ 111]
	LEDs [▸ 57]	Programming [▸ 128]
		Diagnostics [▸ 115]

2.2 Technical data

Communication and function	**EL6821**
Technology	DALI-2
Data transfer channels	1
DALI slaves / groups	max. 64 DALI control gears and max. 64 DALI control devices / max. 16 groups
Transmission standard	DALI + DALI-2
Data transfer rates	1200 baud
Configuration	Via TwinCAT 3 System Manager
Special features	• 2 digital inputs for simplified commissioning; • TwinCAT 3 PLC library: Tc3_DALI

DALI power supply	**EL6821**
DALI power supply	max. 250 mA (can be switched off)
Guaranteed DALI supply current	220 mA
DALI current with power supply switched off	2 mA
Surge voltage resistance	Continuous load 275 V AC effective

Supply and potentials	**EL6821**
Power supply	via the E-bus
Current consumption via E-bus	80 mA typ.
Current consumption of the 24 V connections	typ. 25 mA + DALI current
Input voltage	24 V_{DC} (-15 %/+ 20 %)
Insulation voltage	DALI bus/E-bus: 1,500 V_{DC}

General data	**EL6821**
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)
Weight	approx. 80 g
Installation [▶ 43]	on 35 mm mounting rail, conforms to EN 60715
Installation position	Any, see note [▶ 46]!

Environmental conditions	**EL6821**
Permissible ambient temperature range during operation	-25 °C ... +60 °C
Permissible ambient temperature range during storage	-40 °C ... +85 °C
Relative humidity	95%, no condensation

Standards and approvals	**EL6821**
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 rating	IP20
Identification / approval*	CE, UKCA

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

2.3 DALI

DALI (Digital Addressable Lighting Interface) is a definition for the standardization of digital interfaces between control gears (lamps) and control devices (sensors and application controllers). The standard (IEC 62386) allows the manufacturers of lighting components to implement complex lighting tasks easily and conveniently.

The KL6811 (DALI version-1/DSI) and KL6821/EL6821 (DALI-2) Bus Terminals are integrated into the bus terminal system and are therefore fieldbus-independent. The DALI data is forwarded to the DALI devices via the respective bus coupler. Bus controllers also offer the option of running PLC programs locally in IEC 61131-3.

To ensure the interoperability of DALI-2 devices with each other, the DALI Alliance (DiiA) provides a certification program. Products that have successfully completed the DALI-2 certification process may use the DALI-2 logo.

All certified DALI-2 devices are entered in the DiiA product database. The product database can be accessed via the DiiA homepage:

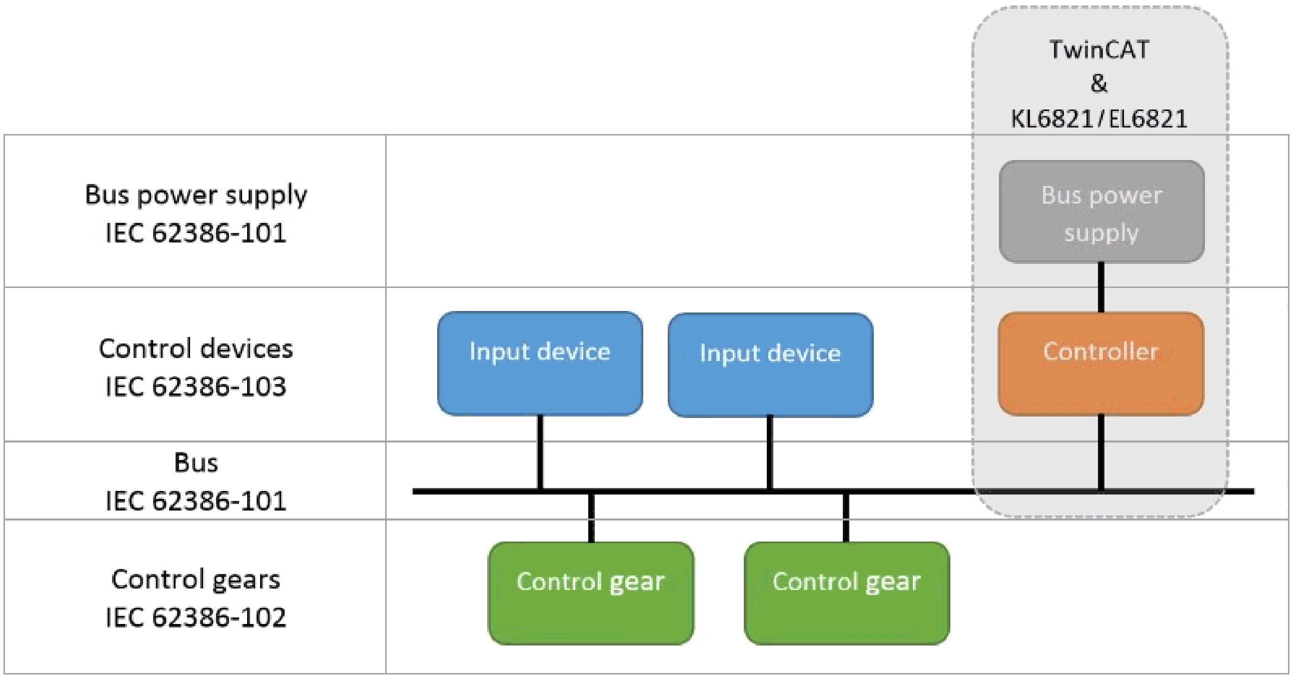
https://www.dali-alliance.org/products/4844/kl6821-dali-dali-2-multi-master-and-power-supply

2.3.1 IEC 62386

DALI is specified in the IEC 62386 standard and offers advantages such as flexibility, simplicity, user friendliness and robustness. IEC 62386 has been revised several times and was extended considerably in November 2014 with the publication of the second revision. While in the first revision only DALI control gears (lamps) were considered, from the second revision onwards DALI control devices (sensors and application controllers) are also included. These are described in the respective section of IEC 62386:

IEC 62386-101	General system properties such as cabling, mains supply and frame structure
IEC 62386-102	General properties of the DALI control gears
	IEC 62386-201: Fluorescent lamps (device type 0) IEC 62386-202: Emergency lighting (device type 1) IEC 62386-203: Discharge lamps (device type 2) IEC 62386-207: LED modules (device type 6) …
IEC 62386-103	General properties of the DALI control devices
	IEC 62386-301: Push buttons IEC 62386-302: Absolute input devices IEC 62386-303: Occupancy sensor IEC 62386-304: Light sensor …

The IEC 62386-101, IEC 62386-102 and IEC 62386-103 standards describe general properties, while the IEC 62386-2xx and IEC 62386-3xx standards specify the individual device types. IEC 62386-103 and IEC 62386-3xx were included in Revision 2 of the DALI standard.

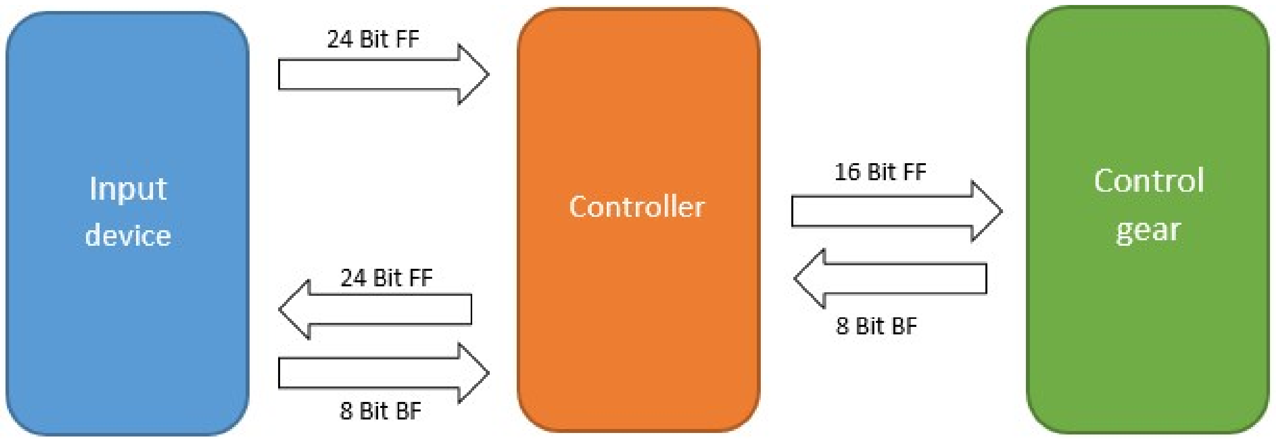


Up to 64 DALI control gears and up to 64 DALI control devices can be connected to the KL6821/EL6821 per DALI line. The KL6821/EL6821 represents the DALI controller. One such device exists for each DALI line. Up to 64 control gears, which have to be DALI/DSI devices, can be connected to the KL6811. Any number of DALI lines (KL6811, KL6821 or EL6821) can be operated with a single TwinCAT controller.

2.3.2 Communication

With regard to the communication, a distinction is made between three frame types:

- 16-bit query, configuration and control frame.
- 24-bit query, configuration and control frame.
- 24-bit event frame.



BF: backward frame
FF: forward frame

16-bit frames

16-bit frames are always sent from a DALI controller to a DALI control gear. They are used for configuring the devices, querying parameters or sending control commands. For certain DALI commands the DALI control gear sends an 8-bit backward frame. DALI control gears only send an 8-bit frame when requested.

In the Tc3_DALI library these commands are identified by the PLC function blocks with the prefix FB_DALI102 or FB_DALI2xx, e.g. FB_DALI102QueryActualLevel or FB_DALI207QueryFeatures.

24-bit frames

24-bit frames are always sent from a DALI controller to a DALI control device. They are used for configuring the devices, querying parameters or sending control commands. For certain DALI commands the DALI control device sends an 8-bit backward frame.

In the Tc3_DALI library these commands are identified by the PLC function blocks with the prefix FB_DALI103 or FB_DALI3xx, e.g. FB_DALI103QueryOperatingMode or FB_DALI303SetHoldTimer.

24-bit events

DALI control devices are able to send events. They are always evaluated by the DALI controller and have a length of 24 bits.

Individual events can be filtered out and further processed with the function blocks FB_DALIGetInputNotification and FB_DALIGetPowerCycleNotification.

Further information on DALI can be found on the homepage of the DALI Alliance (https://www.dali-alliance.org) and in the IEC 62386 standard.

The KL6811 only supports the first revision of the DALI standard. It is not possible to operate control devices with the KL6811.

2.3.3 Priorities

If several DALI control devices are connected to a DALI line, priorities control concurrent access to the DALI bus. According to IEC 62386-103, all DALI-2 devices that can initiate sending of a DALI command (controllers) or sending of an event (input devices) on the DALI bus are referred to as DALI control device.

All DALI-2 devices of a DALI line must share the same data line. To avoid collisions during sending, the sending device checks whether the DALI bus has already been assigned. Sending takes place after a certain settling time, once the DALI bus is free. For high-priority DALI commands the bus access takes place after a short settling time, for low-priority commands the settling time is longer. In other words, high-priority DALI commands are given preference over low-priority DALI commands.

DALI control gears are defined in IEC 62386-102. They are not capable of sending DALI commands or events independently. DALI control gears may only return the 8-bit backward frame to forward frames sent by a DALI controller (see also Communication [▶ 18]). Since a DALI controller waits for the backward frame, the 8-bit backward frame has the shortest settling time. This settling time is shorter than for DALI commands with the highest priority. This means that DALI forward frames can be processed without interference from other DALI commands.

The priorities used by a DALI controller for sending the DALI commands are referred to as command priorities and are mapped by the data type E_DALICommandPriority. Command priorities can have 5 different values:

- **Low**: DALI priority 5
- **Middle low**: DALI priority 4
- **Middle**: DALI priority 3
- **Middle high**: DALI priority 2
- **High**: DALI priority 1

Most function blocks referred to in chapter Part 102 (control gears) have the input *eCommandPriority*. This input is used to specify the priority with which the DALI commands are to be sent via the KL6821/EL6821.

Events also have a priority (event priority), which is represented by the data type E_DALIEventPriority. Event priorities can have 4 values in the range *Low* (DALI priority 5) to *Middle high* (DALI priority 2). The event priority is written as a parameter (see instance variable eventPriority) into the respective instances of the DALI devices.

Priority *High* (DALI priority 1) is only allowed for DALI-2 commands and cannot be used for events.

Tc3_DALI uses the following values for the priorities:

E_DALICommandPriority/ E_DALIEventPriority	Application
Low	-
MiddleLow	Light sensor events (Part 304). All other DALI commands.
Middle	Events of push buttons (Part 301), absolute input devices (Part 302) and occupancy sensors (Part 303).
MiddleHigh	DALI commands for writing parameters and for addressing DALI devices.
High	DALI commands for transactions (from the second DALI command).

When selecting priorities, care should always be taken to ensure that time-critical events that are important for switching the lighting have a higher priority than the DALI commands themselves. Non-system-critical DALI commands, such as the cyclic querying of states for the display in a visualization, should be sent with a lower priority.

Priorities for DALI commands (E_DALICommandPriority) are supported from Tc3_DALI V3.11.0.0. If the KL6821 is used, it must contain the firmware BD or newer. Older firmware versions always send DALI commands with the priority *High*.

Neither the KL6811 nor the Tc2_DALI library support priorities for DALI commands.

Priorities are always important in situations where DALI sensors (input devices) send events and DALI commands are sent in parallel via the KL6821/EL6821. If only DALI control gears and a DALI controller (KL6821/EL6821) are connected to a DALI line, the priorities of the DALI commands are of secondary importance. The priorities of the DALI commands can also be neglected if the DALI sensors on the DALI line do not send notifications.

Further details about the DALI priorities can also be found in the following chapter Bus Timing [▸ 21].

2.3.4 Bus Timing

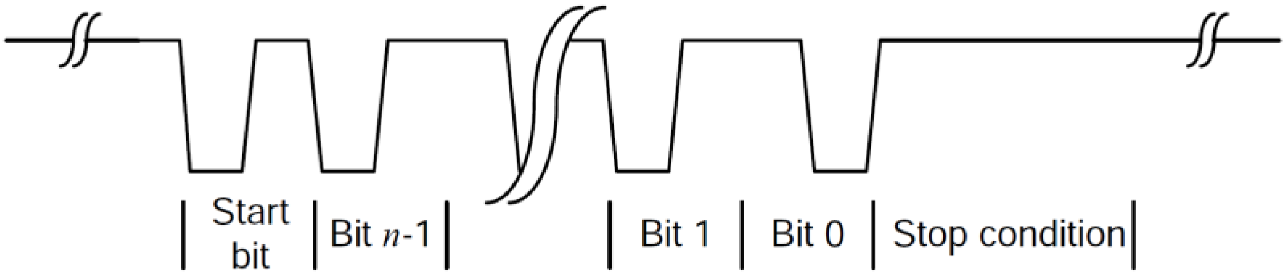
The following describes the structure and operation of the DALI protocol. This description focuses on the most important basic principles. For a full explanation, the IEC 62386 standard, in particular Part 101, should be consulted.

2.3.4.1 Structure data frame

Each Forward Frame (FF) and Backward Frame (BF) basically consists of:

- 1 start bit
- n data bits
- 1 stop condition

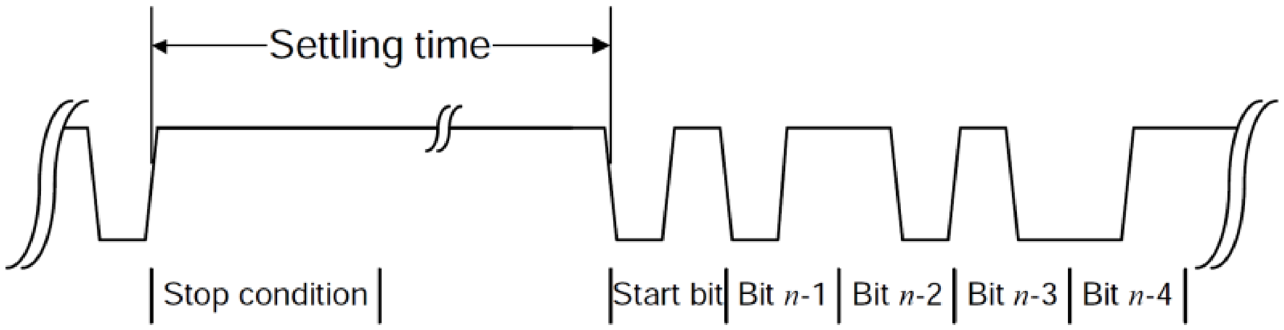
The *most significant bit* (MSB) is transmitted at the beginning.



Transmission is via Manchester coding with a data transfer rate of 1200 bits per second. Thus, each bit has a length of 0.833 ms (1 / 1200 = 0.000833).

The *Stop condition* has a length of at least 2.45 ms.

A fixed time (*settling time*) must be waited between the sending of two data frames before the sending of the next data frame can be started.



The length of the *settling time* depends on the DALI priority with which a data frame is sent. The higher the DALI priority, the smaller the *settling time*.

Settling time	Minimum	Mean value	Maximum
between FF and BF	5.5 ms	8.0 ms	10.5 ms
before each FF (DALI priority *High*)	13.5 ms	14.1 ms	14.7 ms
before each FF (DALI priority *Middle high*)	14.9 ms	15.5 ms	16.1 ms
before each FF (DALI priority *Middle*)	16.3 ms	17.0 ms	17.7 ms
before each FF (DALI priority *Middle low*)	17.9 ms	18.6 ms	19.3 ms
before each FF (DALI priority *Low*)	19.5 ms	20.3 ms	21.2 ms

More information about DALI priorities is also available in the chapter Priorities [▸ 20].

Thus, data frames with a higher DALI priority (low *settling time*) occupy the DALI bus earlier, compared to data frames with a lower DALI priority (high *settling time*).

Certain DALI commands must be sent twice within 100 ms (send-twice) without the receiver being allowed to receive another DALI command in the meantime. Only then will the DALI command be recognized as valid by the receiver. This is primarily used with DALI commands that are used to configure DALI devices. So that the two DALI commands are not interrupted by another DALI command, the 2nd DALI command is always sent with DALI priority *High*. The DALI priority *High* is reserved for these DALI commands and must not be used in any other context.

2.3.4.2 Transmission length

The approximate transmission length can be determined from the bit length and the structure of the data frame. For further (simplified) consideration, a mean *settling time* of 17.0 ms is used for the forward frames (FF), and a mean settling time of 8 ms for the backward frames (BF). Between the two DALI commands sent within 100 ms (send-twice), a *settling time* of 14.1 ms is used.

The number of bits results from the number of data bits (8, 16 or 24) plus the start bit. Thus 9, 17 or 25 bits are transmitted with the respective frames.

16-bit frame without backward frame:
17.0 ms + (17 x 0.833 ms) = **31.2 ms**.

16-bit frame with backward frame:
17.0 ms + (17 x 0.833 ms) + 8.0 ms + (9 x 0.833 ms) = **46.7 ms**.

16-bit frame send-twice:
17.0 ms + (17 x 0.833 ms) + 14.1 ms + (17 x 0.833 ms) = **59.4 ms**.

24-bit frame without backward frame:
17.0 ms + (25 x 0.833 ms) = **37.8 ms**.

24-bit frame with backward frame:
17.0 ms + (25 x 0.833 ms) + 8.0 ms + (9 x 0.833 ms) = **53.3 ms**.

24-bit frame send-twice:
17.0 ms + (25 x 0.833 ms) + 14.1 ms + (25 x 0.833 ms) = **72.8 ms**.

The following table lists the average transmission lengths of the individual frames. This results in a maximum possible number of frames per second. The PLC program should be designed in such a way that the number of frames always falls below the maximum.

Frame	Transmission length	Frames per second
16-bit frame without backward frame	31.2 ms	approx. 32
16-bit frame with backward frame	46.7 ms	approx. 21
16-bit frame send-twice	59.4 ms	approx. 16
24-bit frame without backward frame	37.8 ms	approx. 26
24-bit frame with backward frame	53.3 ms	approx. 18
24-bit frame send-twice	72.8 ms	approx. 13

2.3.4.3 Collision detection

The generation of the DALI frames is achieved in a DALI device by changing between high and low level within defined times. At a low level, the DALI bus is pulled towards 0 V against a current limitation. With a high level, the DALI connection from the DALI device is high-resistance.

With DALI-2, it may happen that several DALI devices try to send DALI frames independently of each other. For this reason, DALI-2 includes collision avoidance, collision detection and collision resolution.

Collision avoidance is achieved by using the DALI priorities. Before a DALI device sends a DALI frame, it is checked whether the DALI bus is free. Only if the DALI bus is free (high level), a bus access may take place. Correct use of DALI priorities reduces the probability of simultaneous bus access and thus minimizes the number of collisions.

Nevertheless, the DALI priorities cannot completely avoid collisions on the DALI bus, since DALI frames from different DALI devices may have the same DALI priority. For this reason, DALI-2 has collision detection and collision resolution.

The following diagram explains the collision resolution.

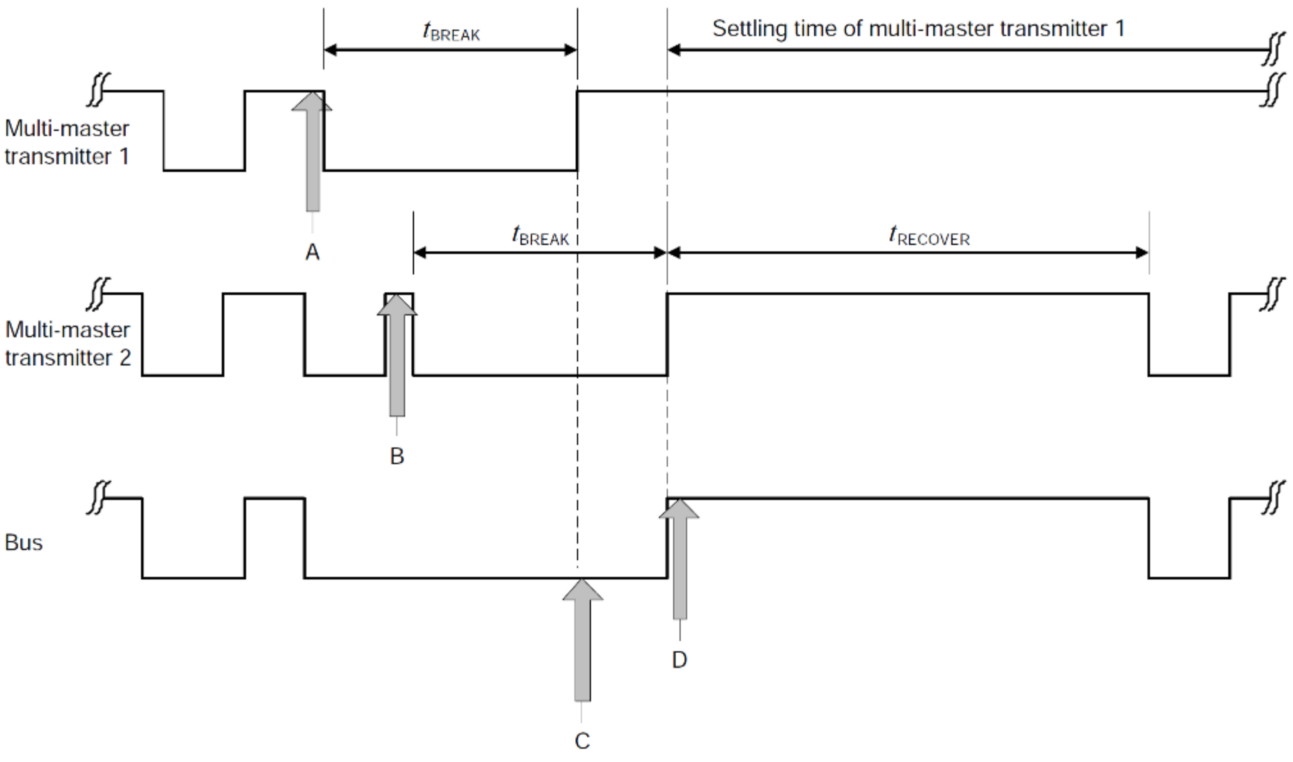
If several DALI devices send a high level, this cannot be detected by the DALI devices. The resulting voltage on the DALI bus is also a high level in this case.

At point A DALI device 1 detects a collision, because an attempt is made to generate a high signal at this point, but the DALI bus is pulled to low level by DALI device 2. DALI device 1 starts the break sequence for this reason. During this time the DALI device pulls the DALI bus to low level.

At point B DALI device 2 tries to generate a high level. However, since the DALI bus is pulled to low level by DALI device 1, DALI device 2 also detects a collision and also starts the break sequence.

At point C the break sequence of DALI device 1 has expired. Subsequently, it is checked whether the DALI bus is still at low level. Since this is the case, the system waits until the DALI bus is free again. DALI device 1 then starts sending the DALI frame again, including the *settling time*.

If the break sequence is finished at DALI device 2, the DALI bus is not occupied by any other DALI device (point D). Therefore the recover sequence is started at DALI device 2 and then the DALI frame is sent again directly (without *settling time*).


t_{BREAK}
Settling time of multi-master transmitter 1
Multi-master transmitter 1
A
t_{BREAK}
$t_{RECOVER}$
Multi-master transmitter 2
B
Bus
D
C

The break sequence has a length of 1.2 ms to 1.4 ms, while the recover sequence can be between 4.0 ms and 4.6 ms.

Collisions on the DALI bus interrupt the transmission on the DALI bus for several milliseconds. This further reduces the data throughput. For this reason, a DALI system should be put together and configured in such a way that as few collisions as possible occur.

2.3.4.4 Cycle times PLC tasks

For practical application, the cycle times of the PLC tasks should, if possible, always be set so that the maximum data transfer rate from the DALI bus is achieved.

Tests were performed to determine the number of frames at different cycle times of the PLC tasks. For this purpose, a PLC program was used, which cyclically sends 6 frames independently of each other. Three 16-bit frames (2 x without backward frame, 1 x with backward frame) and three 24-bit frames (2 x without backward frame, 1 x with backward frame) were sent. Since the send-twice frames are only of importance for the configuration of DALI devices, they were not considered further. The total transmission length of the 6 frames was thus 238 ms (2 x 21.2 ms + 46.7 ms + 2 x 37.8 ms + 53.3 ms). This means that the sample program could send a maximum of 25 frames per second (1000 ms / 238 ms x 6). Events that are additionally sent by possible DALI sensors (input devices) are not taken into account.

	80 ms	**60 ms**	**40 ms**	**30 ms**	**20 ms**	**10 ms**	**8 ms**	**6 ms**	**4 ms**
30 ms	6	7	7	8	8	8	8	8	8
15 ms	11	11	12	12	13	13	13	13	13
10 ms	15	15	16	17	17	17	17	17	17
8 ms	18	18	19	19	20	20	20	20	20
6 ms	20	21	21	22	22	22	22	22	22
4 ms	23	23	24	25	25	25	25	25	25
2 ms	23	24	25	25	25	25	25	25	25

The times in the top line (4 ms ... 80 ms) specify the cycle time of the PLC task from which the DALI commands are started. The times (2 ms ... 30 ms) in the first column specify the cycle time of the PLC task for background communication.

2.3.4.5 Summary

Even though the test program is only representative, it can be clearly seen that the cycle time of the background communication has a decisive influence on the data throughput. If a maximum data transfer rate is required on the DALI bus, the following points must be observed:

- K-bus and fieldbus should be arranged so that the cycle time for the PLC task accessing the DALI terminal (background communication) does not exceed 6 ms.
- The number of events of the DALI sensors (input devices) should be as low as possible. The more events are sent, the higher the probability of collisions on the DALI bus. The DALI sensors should be configured so that the number of events is minimal.
- To reduce the number of collisions on the DALI bus, the DALI priorities should be used. Recommendations for this can be found in the chapter DALI priorities [▸ 20].
- A large number of DALI control devices also increases the probability of collisions on the DALI bus. If necessary, the DALI control devices must be divided among different DALI lines. DALI control devices are DALI controllers and DALI sensors (see chapter Communication [▸ 18]).

2.3.5 Memory banks

Memory banks are freely accessible memory areas in which device-specific information and properties can be stored. The contents of the memory banks can be read with FB_DALI10xReadMemoryLocation (see FB_DALI102ReadMemoryLocation and FB_DALI103ReadMemoryLocation) and, if enabled, written with FB_DALI10xWriteMemoryLocationNoReply (see FB_DALI102WriteMemoryLocationNoReply and FB_DALI103WriteMemoryLocationNoReply).

Part of the memory banks can be provided with write protection.

A DALI device can support a maximum of 256 memory banks, each with up to 255 bytes, with memory banks 200 to 255 being reserved. Memory bank 0 and memory bank 1 are predefined by IEC 62386.

Structure of memory bank 0:

Memory bank 0 is read only and contains general, vendor-specific information about the DALI control gear or DALI control device. Every certified DALI device must implement memory bank 0. Up to offset 16#1A the fields are defined by IEC 62386 as follows.

Offset	Description	Default values
16#00	Offset of the last memory area inside the memory bank that can be accessed.	Vendor-specific
16#01	Reserved, not implemented	
16#02	Number of the last memory bank that can be accessed.	Vendor-specific
16#03	GTIN byte 0 (MSB)	Vendor-specific
16#04	GTIN byte 1	Vendor-specific
16#05	GTIN byte 2	Vendor-specific
16#06	GTIN byte 3	Vendor-specific
16#07	GTIN byte 4	Vendor-specific
16#08	GTIN byte 5 (LSB)	Vendor-specific
16#09	Firmware Version (major)	Vendor-specific
16#0A	Firmware Version (minor)	Vendor-specific
16#0B	Identification number byte 0 (MSB)	Vendor-specific
16#0C	Identification number byte 1	Vendor-specific
16#0D	Identification number byte 2	Vendor-specific
16#0E	Identification number byte 3	Vendor-specific
16#0F	Identification number byte 4	Vendor-specific
16#10	Identification number byte 5	Vendor-specific
16#11	Identification number byte 6	Vendor-specific
16#12	Identification number byte 7 (LSB)	Vendor-specific
16#13	Hardware version (major)	Vendor-specific
16#14	Hardware version (minor)	Vendor-specific
16#15	101 Version number of the current DALI standard	Vendor-specific
16#16	102 Version numbers of all integrated DALI control gears	Vendor-specific
16#17	103 Version numbers of all integrated DALI control devices	Vendor-specific
16#18	Number of logical control units in the device	Vendor-specific
16#19	Number of logical control gears in the device	Vendor-specific
16#1A	Index number of this logical DALI control gear or DALI control device	Vendor-specific
16#1B… 16#7F	Reserved, not implemented	
16#80… 16#FE	Additional device information	Vendor-specific
16#FF	Reserved, not implemented	

Structure of memory bank 1:

Memory bank 1 can be used by the device vendor to store further information in the DALI device. Up to offset 16#10 the fields are defined by IEC 62386 as follows.

Offset	Description	Default values	Memory
16#00	Offset of the last memory area inside the memory bank that can be accessed.	Vendor-specific (16#10… 16#FE)	
16#01	Indicator byte	Vendor-specific	
16#02	Lock byte for memory bank 1. Writeable bytes become changeable through the value 16#55. No other values make writing possible.	16#FF	
16#03	OEM GTIN byte 0 (MSB)	16#FF	Lockable by byte 16#02
16#04	OEM GTIN byte 1	16#FF	Lockable by byte 16#02
16#05	OEM GTIN byte 2	16#FF	Lockable by byte 16#02
16#06	OEM GTIN byte 3	16#FF	Lockable by byte 16#02
16#07	OEM GTIN byte 4	16#FF	Lockable by byte 16#02
16#08	OEM GTIN byte 5 (LSB)	16#FF	Lockable by byte 16#02
16#09	OEM Identification number byte 0 (MSB)	16#FF	Lockable by byte 16#02
16#0A	OEM Identification number byte 1	16#FF	Lockable by byte 16#02
16#0B	OEM Identification number byte 2	16#FF	Lockable by byte 16#02
16#0C	OEM Identification number byte 3	16#FF	Lockable by byte 16#02
16#0D	OEM Identification number byte 4	16#FF	Lockable by byte 16#02
16#0E	OEM Identification number byte 5	16#FF	Lockable by byte 16#02
16#0F	OEM Identification number byte 6	16#FF	Lockable by byte 16#02
16#10	OEM Identification number byte 7 (LSB)	16#FF	Lockable by byte 16#02
16#11… 16#FE	Additional device information	Vendor-specific	
16#FF	Reserved, not implemented		

Structure of memory banks 2 to 199:

The device vendor can use memory banks 2 to 199 to supply further parameters. The structure of the memory banks is always as shown below. The vendor of the DALI device must be consulted regarding the contents and the possibility to write individual bytes.

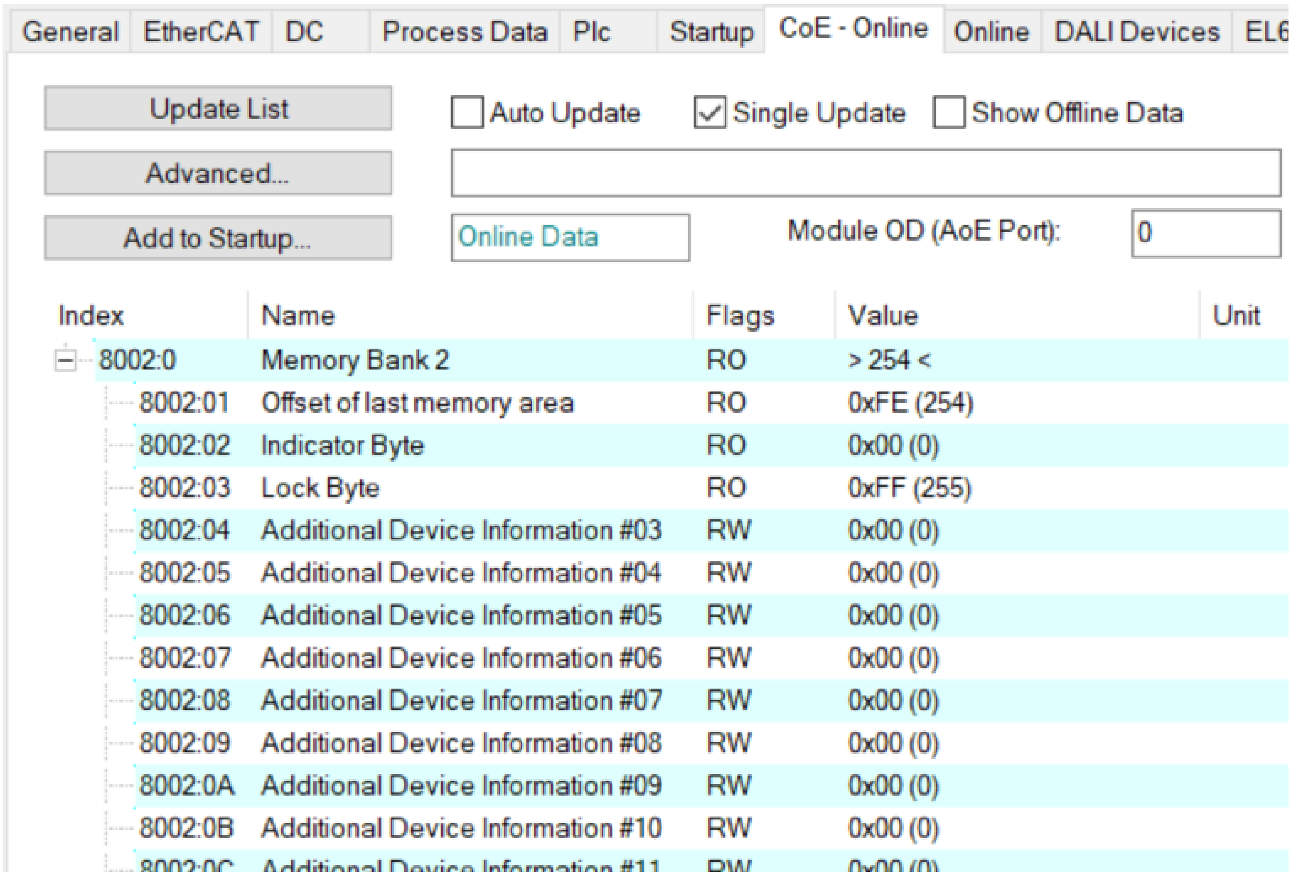
Offset	Description	Default values	Memory
16#00	Offset of the last memory area inside the memory bank that can be accessed.	Vendor-specific (16#03… 16#FE)	
16#01	Indicator byte	Vendor-specific	
16#02	Lock byte for the memory bank. Writeable bytes become changeable through the value 16#55. No other values make writing possible.	16#FF	
16#03… 16#FE	Additional device information	Vendor-specific	Vendor-specific; lockable by byte 16#02 if enabled by the vendor
16#FF	Reserved, not implemented		

Access to memory bank 2 on the EL6821

Since the KL6821/EL6821 represents a DALI control device according to IEC 62386, the Bus Terminal must offer memory bank 0 and memory bank 1. Other DALI control devices can access these memory banks via the corresponding DALI commands.

In addition, the EL6821 offers memory bank 2. In the EL6821 memory bank 2 is mapped by CoE object 16#8002. The first three fields (16#8002:01 ... 16#8002:03) are defined by IEC 62386 (see above). The fields 16#8002:04 to 16#8002:FF can be written and/or read individually.

Note that the offset in memory bank 2 and the subindex are shifted from the CoE. Thus, offset 10 in memory bank 2 corresponds to field 16#8002:0B.



Auto Update [✓ Single Update] Show Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
8002:0	Memory Bank 2	RO	> 254 <	
8002:01	Offset of last memory area	RO	0xFE (254)	
8002:02	Indicator Byte	RO	0x00 (0)	
8002:03	Lock Byte	RO	0xFF (255)	
8002:04	Additional Device Information #03	RW	0x00 (0)	
8002:05	Additional Device Information #04	RW	0x00 (0)	
8002:06	Additional Device Information #05	RW	0x00 (0)	
8002:07	Additional Device Information #06	RW	0x00 (0)	
8002:08	Additional Device Information #07	RW	0x00 (0)	
8002:09	Additional Device Information #08	RW	0x00 (0)	
8002:0A	Additional Device Information #09	RW	0x00 (0)	
8002:0B	Additional Device Information #10	RW	0x00 (0)	
8002:0C	Additional Device Information #11	RW	0x00 (0)	

The function blocks FB_DALI102ReadMemoryLocation/FB_DALI103ReadMemoryLocation or FB_DALI102WriteMemoryLocationNoReply/FB_DALI103WriteMemoryLocationNoReply are available for accessing the memory banks of a DALI device via the DALI bus.

If memory bank 2 of the own EL6821 is to be accessed from a PLC program, access is via the EtherCAT CoE interface. The TwinCAT library Tc2_EtherCAT offers the necessary function blocks for this (see FB_EcCoeSdoRead and FB_EcCoeSdoWrite).

The following example reads offset 3 (subindex 4) and writes offset 4 (subindex 5) of memory bank 2. The EL6821 must be located at the same controller where the PLC program is executed.

```
VAR
  fbCoERead      : FB_EcCoESdoRead;
  fbCoEWrite     : FB_EcCoESdoWrite;
  nValue         : USINT;
  bExecuteRead   : BOOL;
  bExecuteWrite  : BOOL;
END_VAR
```

```
fbCoERead(sNetId := F_CreateAmsNetId(GVL.stEL6821InData01.stAdsAddr.netId),
          nSlaveAddr := GVL.stEL6821InData01.stAdsAddr.port,
          nIndex := 16#8002,
          nSubIndex := 16#04,
          pDstBuf := ADR(nValue),
          cbBufLen := SIZEOF(nValue),
          bExecute := bExecuteRead);
IF (NOT fbCoERead.bBusy) THEN
  bExecuteRead := FALSE;
END_IF

fbCoEWrite(sNetId := F_CreateAmsNetId(GVL.stEL6821InData01.stAdsAddr.netId),
           nSlaveAddr := GVL.stEL6821InData01.stAdsAddr.port,
           nIndex := 16#8002,
           nSubIndex := 16#05,
           pSrcBuf := ADR(nValue),
           cbBufLen := SIZEOF(nValue),
           bExecute := bExecuteWrite);
```

```
IF (NOT fbCoEWrite.bBusy) THEN
  bExecuteWrite := FALSE;
END_IF
```

2.3.6 DALI-2 current

The DALI power supply of the EL6821/KL6821 has a maximum output current of 250 mA and a guaranteed DALI current of 220 mA.

To ensure safe operation of the DALI bus, the connected DALI devices must never consume more than the guaranteed DALI-2 current in total.

The current consumption of the DALI devices (control gears, sensors, etc.) can be found in the manufacturers' data sheets or on the DiiA product list (www.dali-alliance.org/products).

Furthermore, a reserve specified in IEC 62386 (guaranteed DALI current / 1.2) should be maintained. This reserve is intended for the subsequent installation of additional devices that may be added in the course of a project.

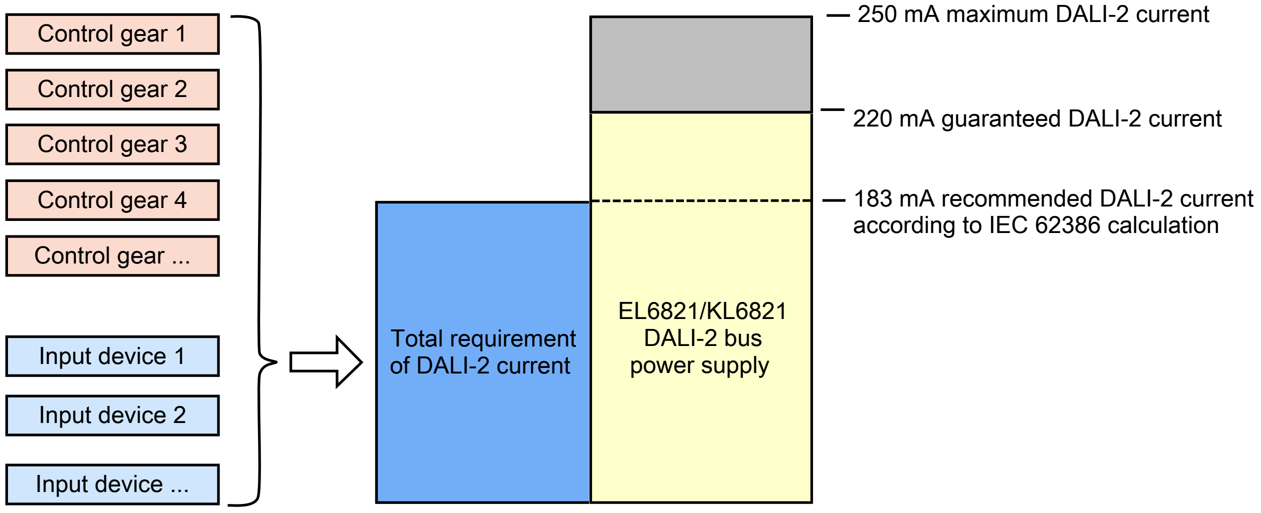

Control gear 1
Control gear 2
Control gear 3
Control gear 4
Control gear ...
Input device 1
Input device 2
Input device ...
Total requirement of DALI-2 current
EL6821/KL6821 DALI-2 bus power supply
250 mA maximum DALI-2 current
220 mA guaranteed DALI-2 current
183 mA recommended DALI-2 current according to IEC 62386 calculation

Fig. 5: DALI-2 current

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

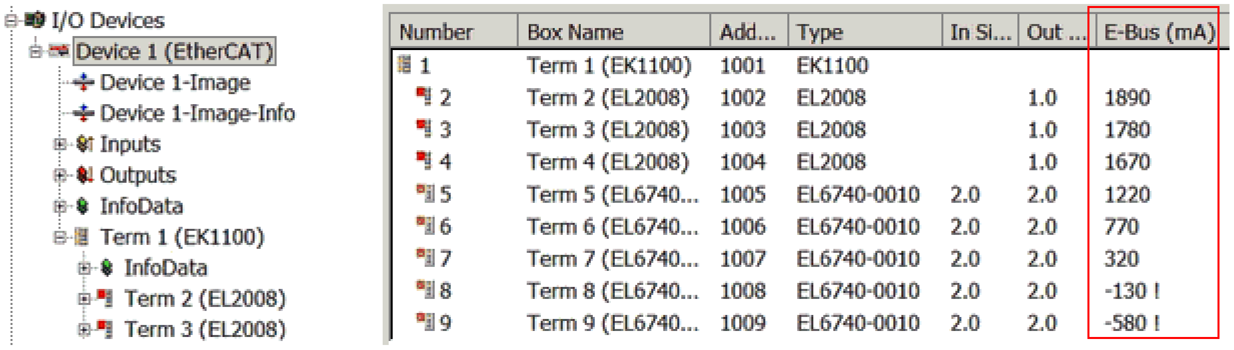
It is recommended to use the appropriate Beckhoff components e.g.
- cable sets ZK1090-9191-xxxx respectively
- RJ45 connector, field assembly ZS1090-0005
- EtherCAT cable, field assembly ZB9010, ZB9020

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.



- I/O Devices
  - Device 1 (EtherCAT)
    - Device 1-Image
    - Device 1-Image-Info
    - Inputs
    - Outputs
    - InfoData
    - Term 1 (EK1100)
      - InfoData
      - Term 2 (EL2008)
      - Term 3 (EL2008)

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

Fig. 6: System manager current calculation

NOTICE

Malfunction possible!

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

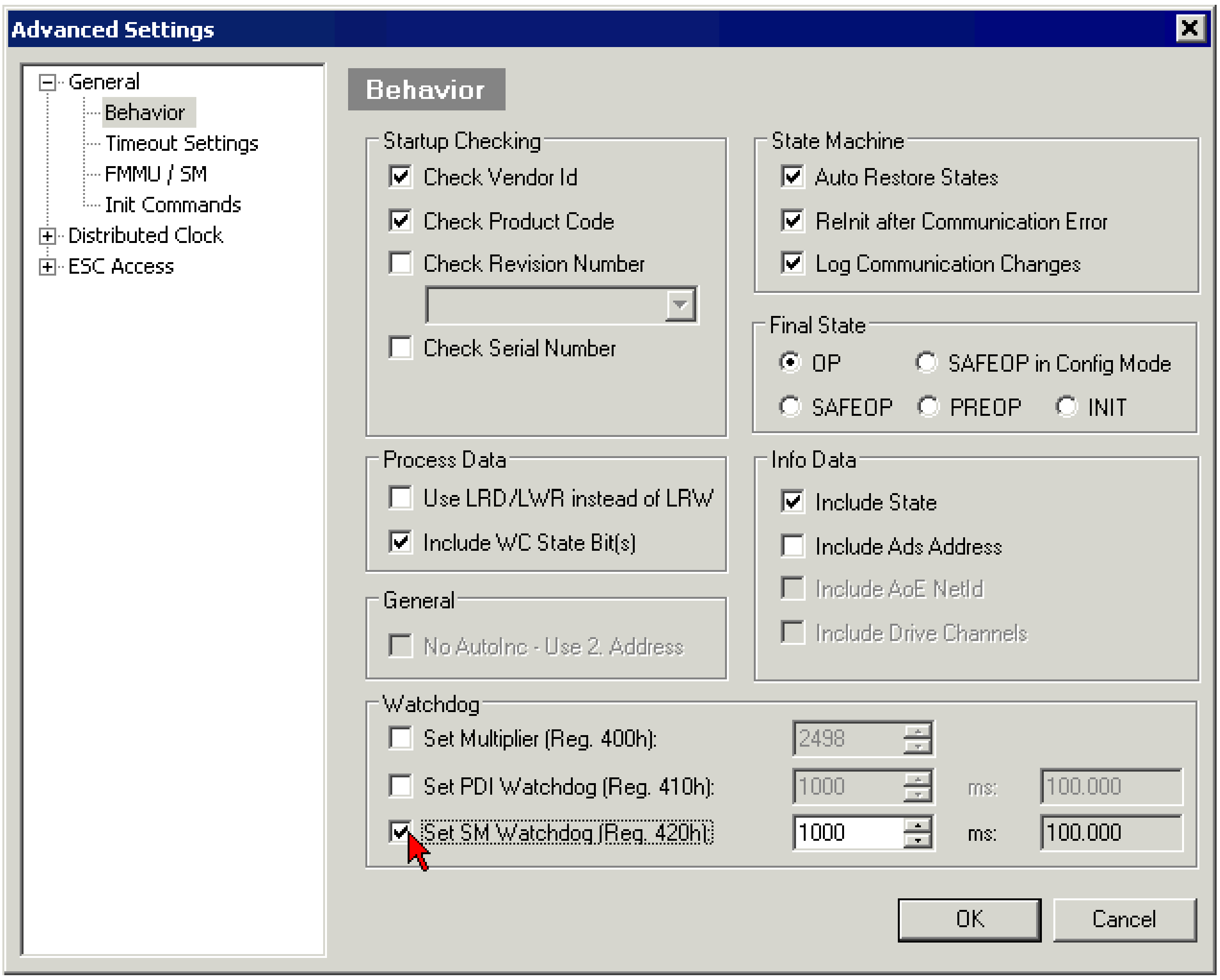
3.3 General notes for setting the watchdog

The EtherCAT 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 features two watchdogs:

- Sync Manager (SM) watchdog (default: 100 ms)
- Process Data (PDI) watchdog (default: 100 ms)

Their times are individually parameterized in TwinCAT as follows:


Advanced Settings

- General
  - Behavior
  - Timeout Settings
  - FMMU / SM
  - Init Commands
- Distributed Clock
- ESC Access

Behavior

Startup Checking
- [✓] Check Vendor Id
- [✓] Check Product Code
- [ ] Check Revision Number
- [ ] Check Serial Number

State Machine
- [✓] Auto Restore States
- [✓] ReInit after Communication Error
- [✓] Log Communication Changes

Final State
[✓ OP] SAFEOP in Config Mode
SAFEOP PREOP INIT

Process Data
- [ ] Use LRD/LWR instead of LRW
- [✓] Include WC State Bit(s)

Info Data
- [✓] Include State
- [ ] Include Ads Address
- [ ] Include AoE NetId
- [ ] Include Drive Channels

General
- [ ] No AutoInc - Use 2. Address

Watchdog
- [ ] Set Multiplier (Reg. 400h): 2498
- [ ] Set PDI Watchdog (Reg. 410h): 1000 ms: 100.000
- [✓] Set SM Watchdog (Reg. 420h): 1000 ms: 100.000

Fig. 7: eEtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

- the Multiplier Register 400h (hexadecimal, i. e. 0x0400) 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 400h, 410h and 420h: 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 register 400h/420h but executed by the microcontroller (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 ESC for longer than the set and activated Process Data Interface (PDI) watchdog time, this watchdog is triggered.

The PDI 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)] * SM/PDI watchdog

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

The value in “Watchdog multiplier + 2” in the formula above corresponds to the number of 40ns base ticks representing one watchdog tick.

⚠ CAUTION

Undefined state possible!

The function for switching off the SM watchdog via SM watchdog = 0 is only implemented in terminals from revision -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
- Operational
- Bootstrap

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

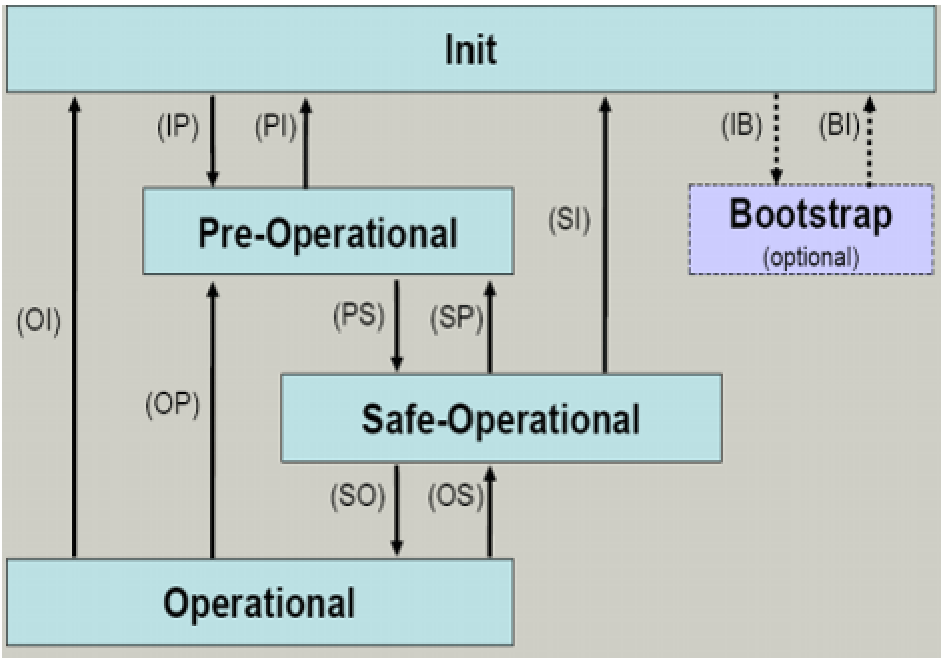


Fig. 8: States of the EtherCAT State Machine

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 Fieldbus Memory Management Unit (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 Dual Port (DP)-RAM areas of the ESC.

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 monitoring sets the outputs of the ESC 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 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 interface) 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 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 data 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.

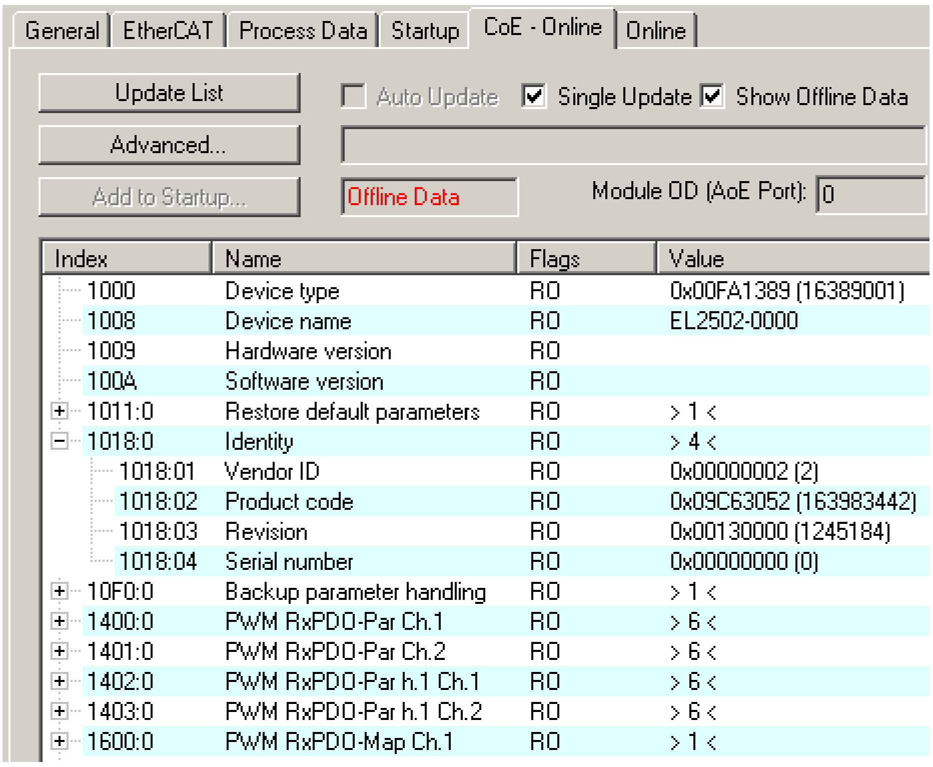
Other important ranges are:

- 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 (“inputs” from the perspective of the EtherCAT master)
- 0x7000: Output PDOs (“outputs” 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. 9: “CoE Online” tab

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

NOTICE

Changes in the CoE directory (CAN over EtherCAT directory), 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 TwinCAT 3 | 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,

- via the System Manager (Fig. “CoE Online” tab) by clicking.
  This is useful for commissioning of the system or slaves. Click on the row of the index to be parameterized and enter a value in the “SetValue” dialog.
- from the control system or 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 index 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.
- If the function is not supported: continuous changing of CoE values is not permissible in view of the lifetime limit.

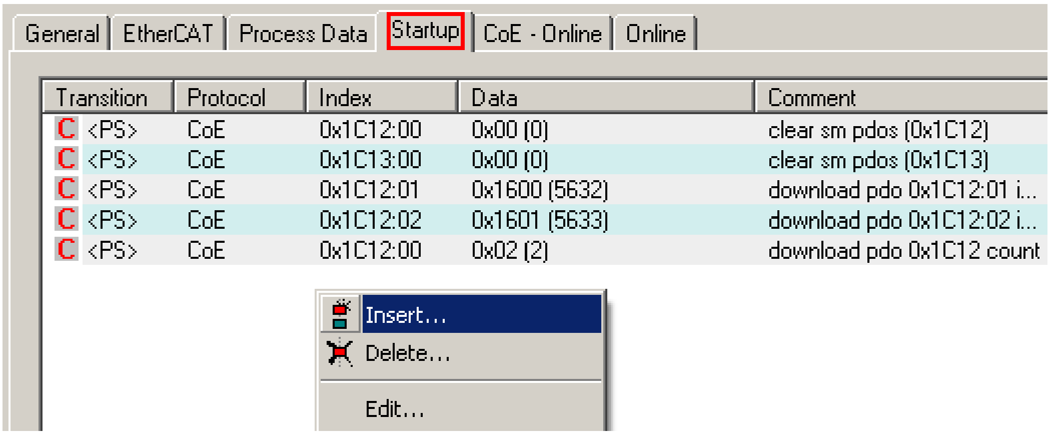
Startup list

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

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

Recommended approach for manual modification of CoE parameters

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



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

Fig. 10: Startup list in the TwinCAT System Manager

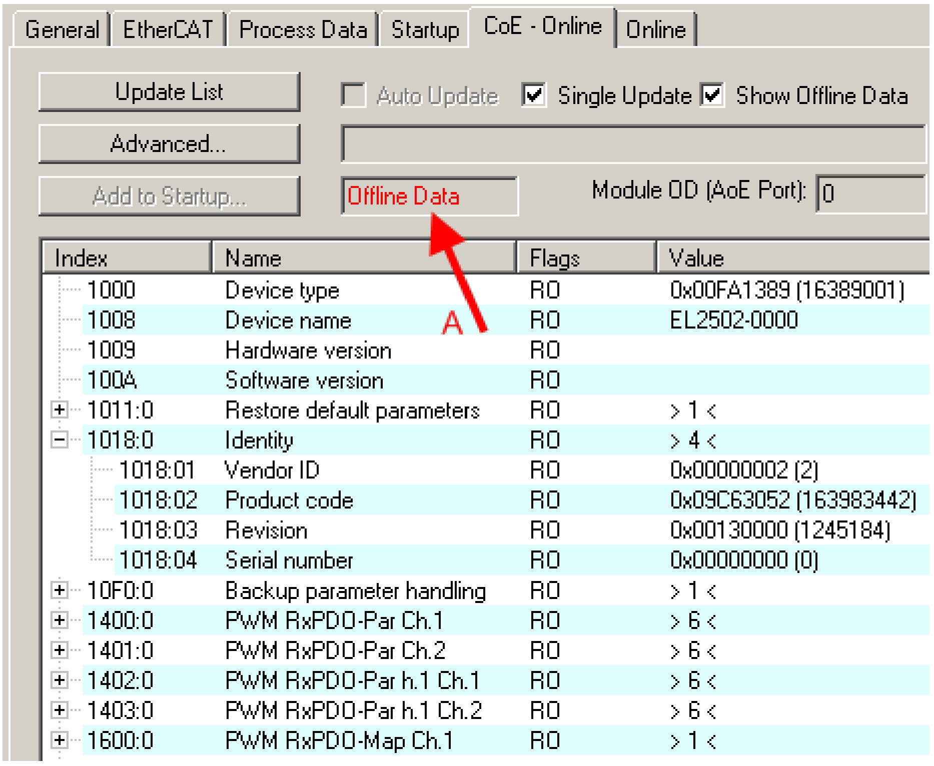
The Startup list may already contain values that were configured by the System Manager based on the ESI specifications. Additional application-specific entries can also be created.

Online / offline list

When working with the TwinCAT System Manager, a distinction must be made as to whether the EtherCAT device is currently "available", i.e. switched on and connected via EtherCAT - i.e. **online** - or whether a configuration is created **offline** without slaves being connected.

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

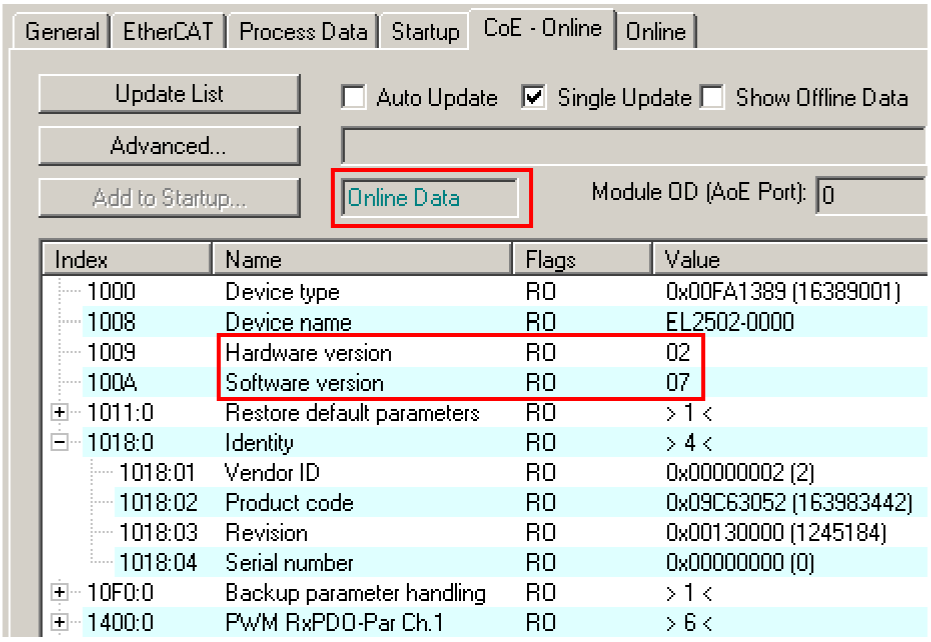
- If the slave is offline:
  - The offline list from the ESI file is displayed. In this case modifications are not meaningful or possible.
  - The configured status is shown under Identity.
  - No firmware or hardware version is displayed since these are features of the physical device.
  - **Offline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



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

Fig. 11: 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 status of the device is displayed in the CoE.
  - **Online Data** is shown in green.



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

Fig. 12: 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 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} or 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.

4 Installation

4.1 Instructions for ESD protection

NOTICE

Destruction of the devices by electrostatic discharge possible!

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

- When handling the components, ensure that there is no electrostatic discharge; also avoid touching the spring contacts directly (see illustration).
- Contact with highly insulating materials (synthetic fibers, plastic films, etc.) should be avoided when handling components at the same time.
- When handling the components, ensure that the environment (workplace, packaging and persons) is properly earthed.
- Each bus station must be terminated on the right-hand side with the EL9011 or EL9012 end cap to ensure the degree of protection and ESD protection.

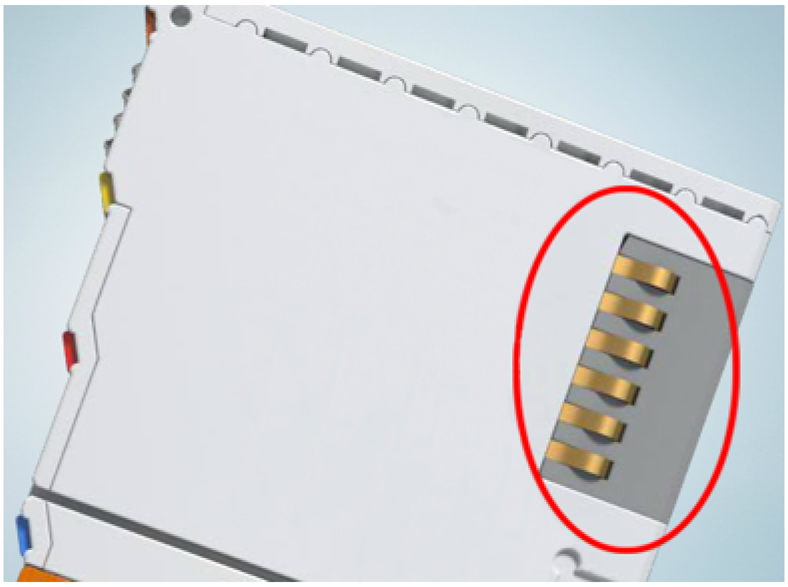


Fig. 13: Spring contacts of the Beckhoff I/O components

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

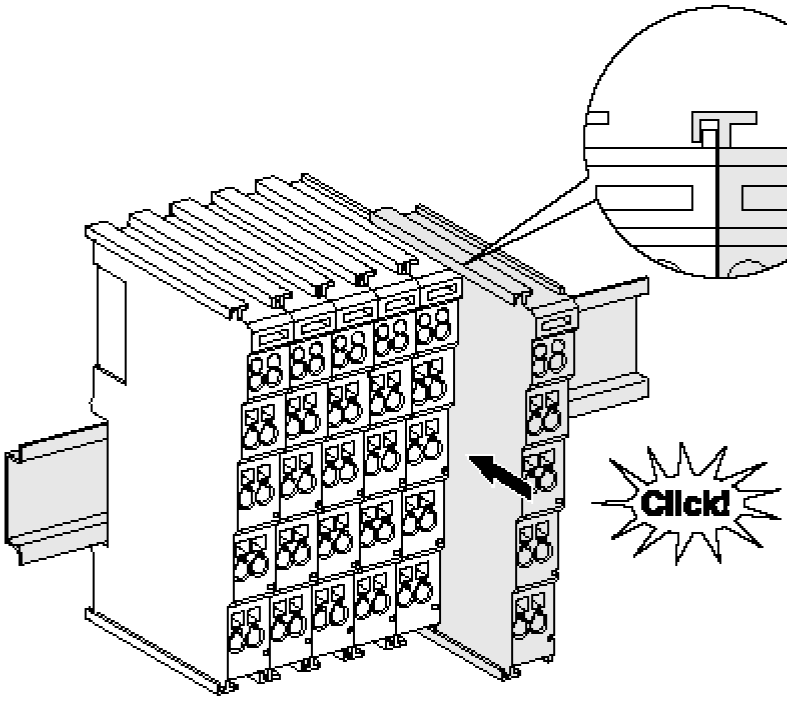


Fig. 14: Attaching on mounting rail

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

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

Fixing of mounting rails

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

NOTICE

Ground the mounting rail!

Ensure that the mounting rail is sufficiently grounded.

Connections within a bus terminal block

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

- The six spring contacts of the E-Bus/K-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 points 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 (EL91xx, EL92xx or KL91xx, KL92xx) interrupt the power contacts and thus represent the start of a new supply rail.

Power contact ⏚

The power contact labeled ⏚ (earthing connection according to IEC 60417-5017, British English: earth, American English: ground) can be used as grounding. For safety reasons this contact mates first when plugging together, and can ground short-circuit currents of up to 125 A.

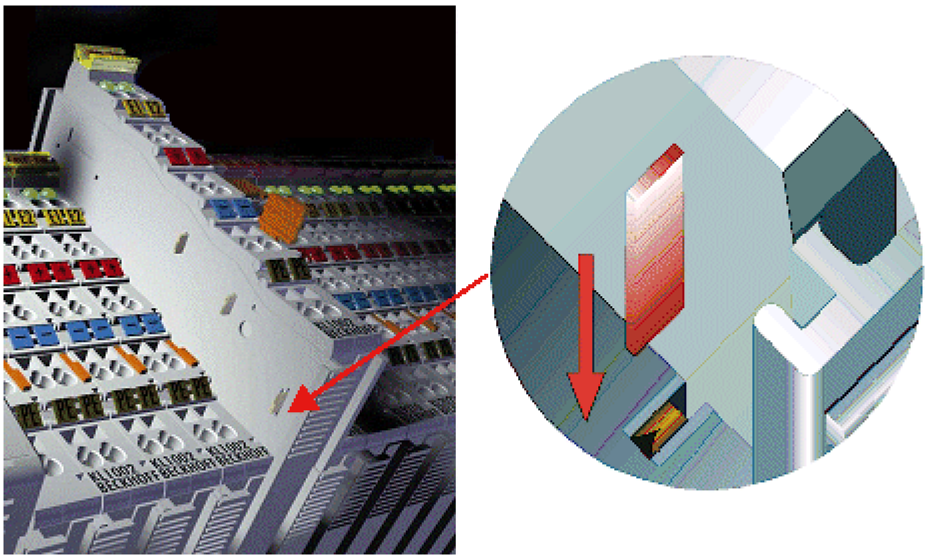


Fig. 15: Power contact on left side

⚠ WARNING

Risk of electric shock!

The power contact labeled ⏚ must not be used for other potentials!

NOTICE

Possible damage of the device

Note that, for reasons of electromagnetic compatibility, the earthing 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 earthing line during insulation testing of a consumer with a nominal voltage of 230 V). For insulation testing, disconnect the earthing 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.

Disassembly

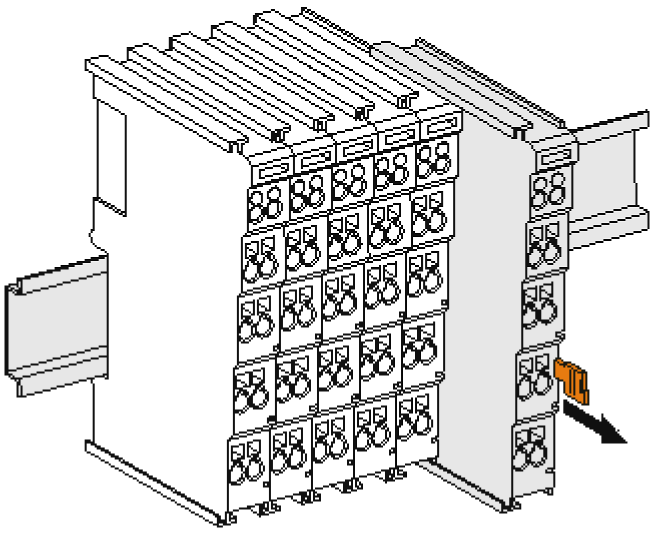


Fig. 16: Disassembling of terminal

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

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

4.3 Disposal

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

4.4 Installation positions

NOTICE

Constraints regarding installation position and operating temperature range

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

Optimum installation position (standard)

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

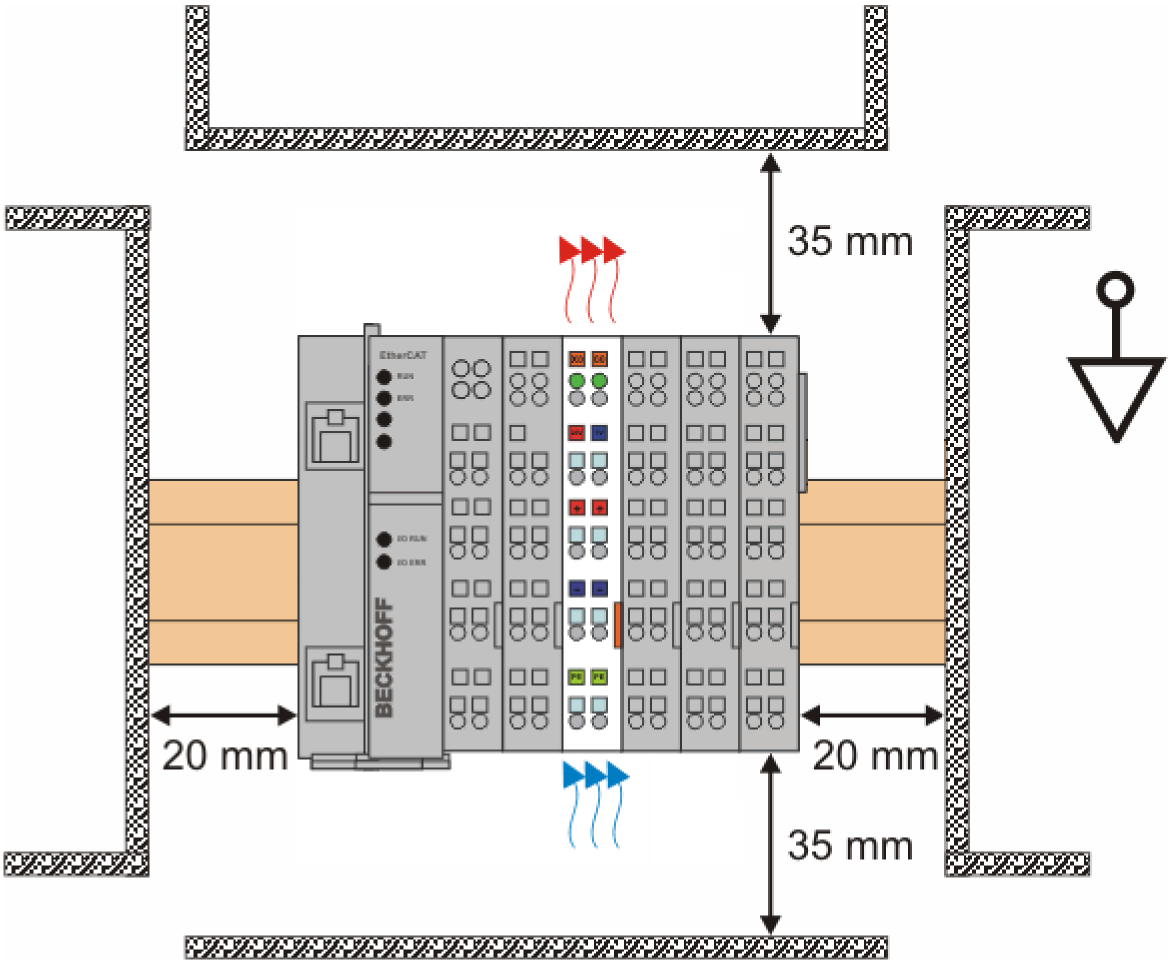


Fig. 17: Recommended distances for standard installation position

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

Other installation positions

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

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

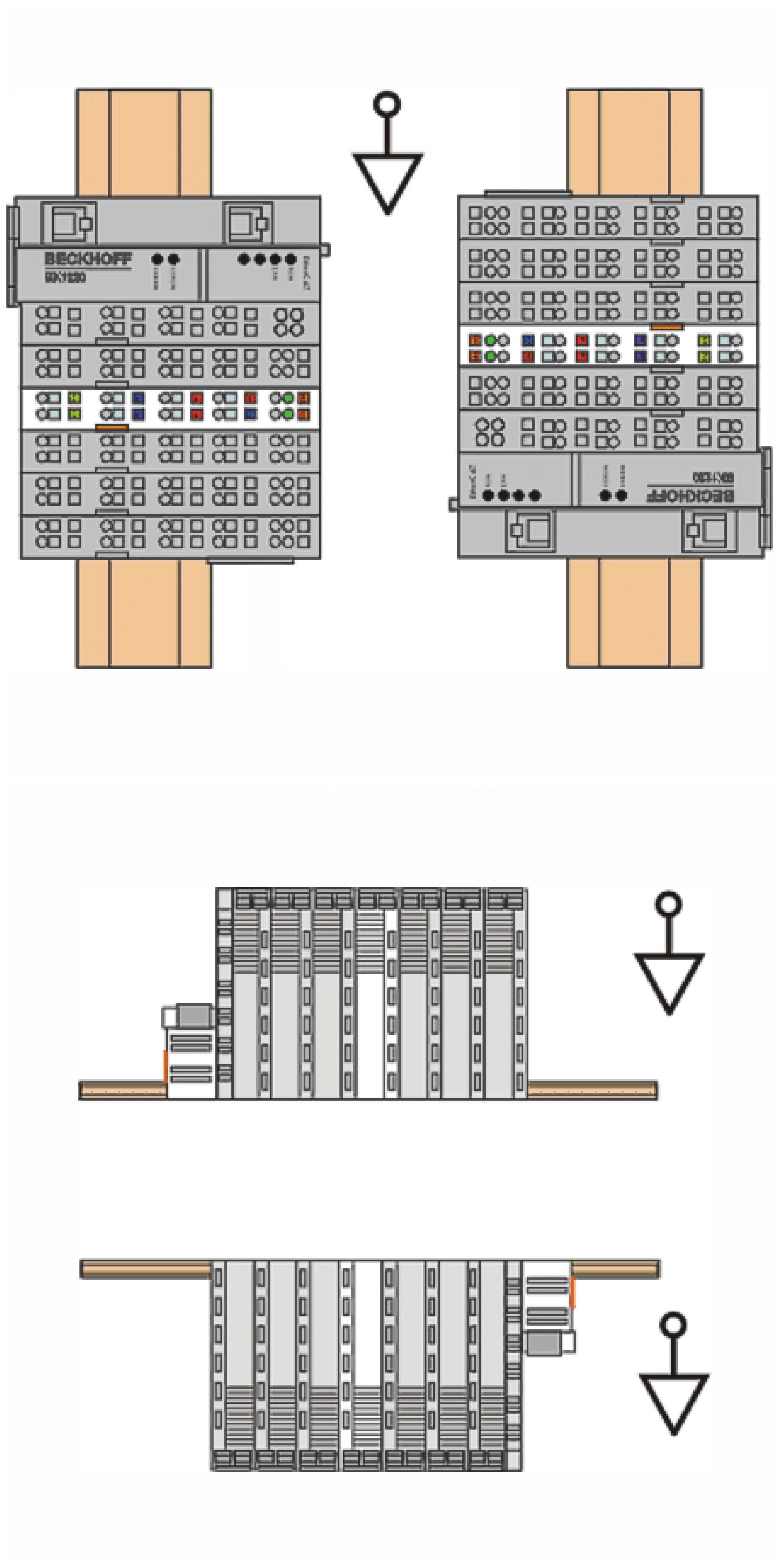


Fig. 18: Other installation positions

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)

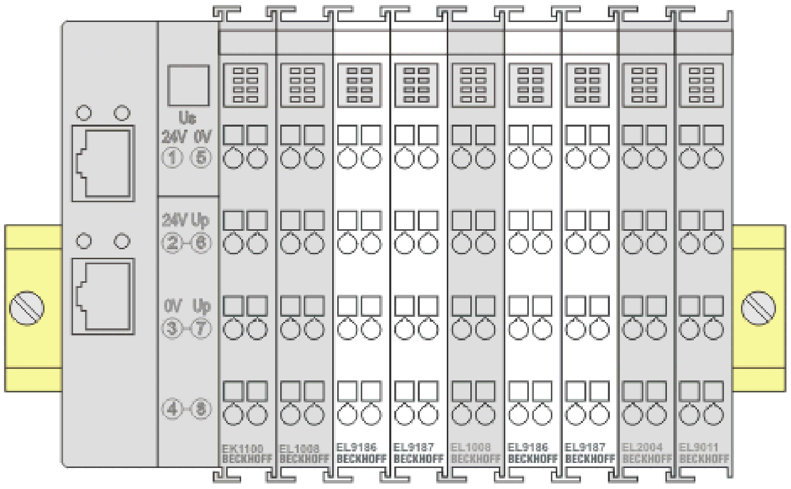


Fig. 19: Correct positioning

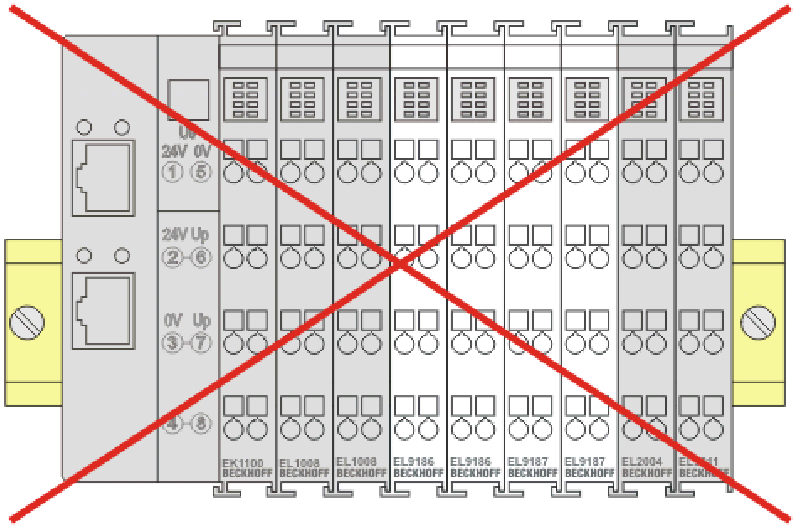


Fig. 20: Incorrect positioning

4.6 Connection

4.6.1 Connection system

⚠ **WARNING**

Risk of electric shock and damage of device!

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

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)

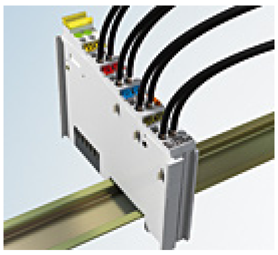


Fig. 21: Standard wiring

The terminals of the ELxxxx and KLxxxx series integrate screwless spring-cage technology for quick and easy wiring.

Pluggable wiring (ESxxxx / KSxxxx)

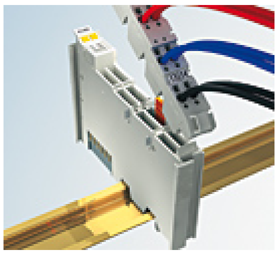


Fig. 22: Pluggable wiring

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

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

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

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

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

High Density Terminals (HD Terminals)

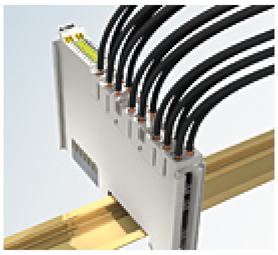


Fig. 23: High Density Terminals

The terminals from these series with 16/32 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 compacted (ultrasonically welded) strands

Ultrasonically compacted (ultrasonically welded) strands

Ultrasonically compacted (ultrasonically welded) strands can also be connected to the standard and high-density terminals. In this case, please note the tables concerning the wire-size width [▶ 52]!

4.6.2 Wiring

⚠ WARNING

Risk of electric shock and damage of device!

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

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

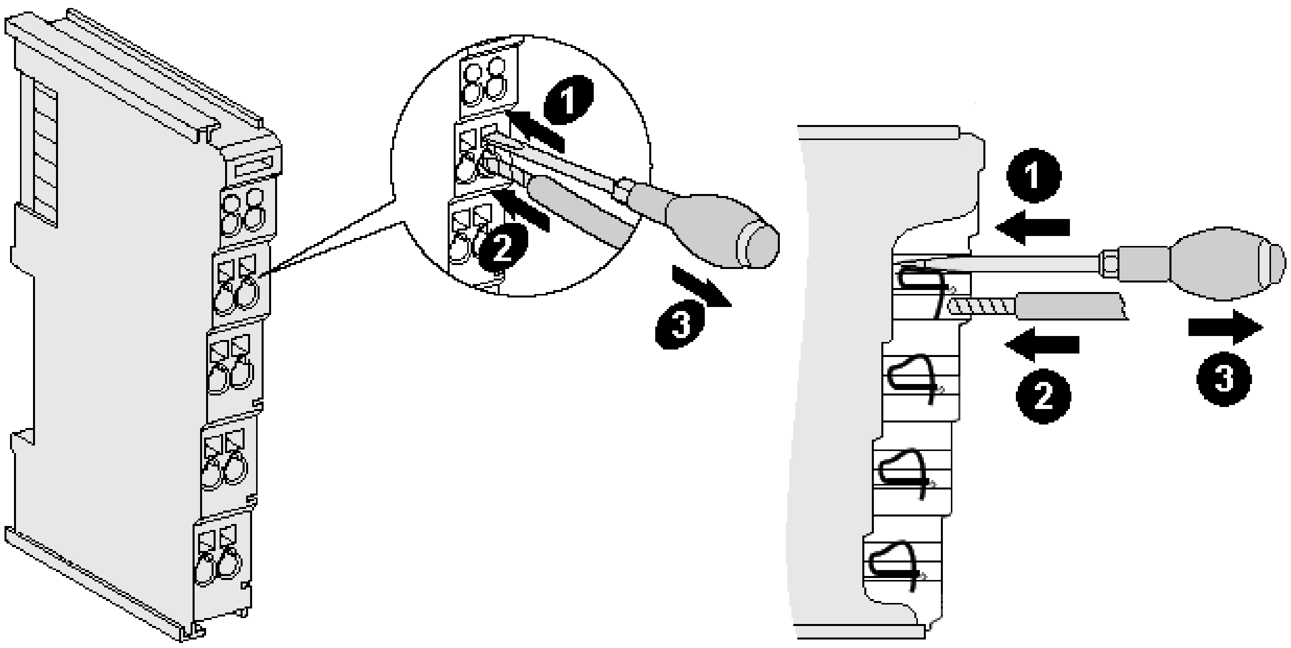


Fig. 24: Connecting a cable on a terminal point

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 (see fig. “Connecting a cable on a terminal point”:

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. When the screwdriver is removed, the terminal point closes automatically and holds the wire securely and permanently in place

See the following table for the suitable wire size width:

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

High Density Terminals (HD Terminals [▶ 50]) with 16/32 terminal points

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

Terminal housing	High Density Housing
Wire size width (single core wires)	0.08 ... 1.5 mm^2
Wire size width (fine-wire conductors)	0.25 ... 1.5 mm^2
Wire size width (conductors with a wire end sleeve)	0.14 ... 0.75 mm^2
Wire size width (ultrasonically compacted [ultrasonically welded] strands)	only 1.5 mm^2 (see notice [▶ 50])
Wire stripping length	8 ... 9 mm

4.6.3 Shielding

Shielding

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

4.7 Note - power supply

⚠ **WARNING**

Power supply from SELV / PELV power supply unit!

SELV / PELV circuits (safety extra-low voltage / protective extra-low voltage) according to IEC 61010-2-201 must be used to supply this device.

Notes:

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

4.8 EL6821 – connection

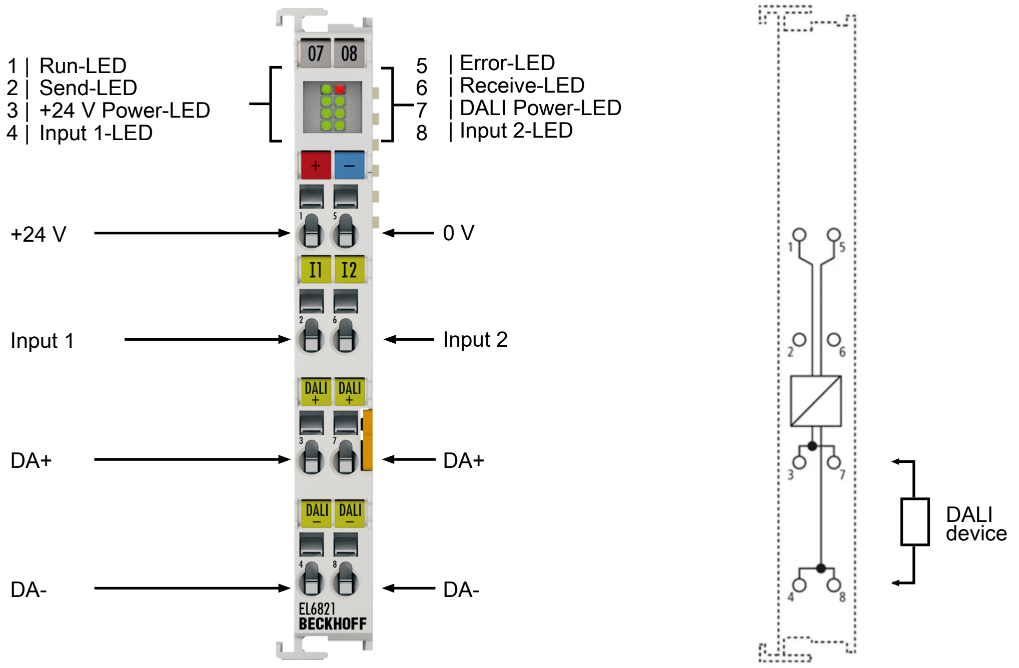


Fig. 25: EL6821

Terminal point	No.	Connection for
+24 V	1	Power supply connection +24 V
Input 1	2	Digital input 1
DA +	3	DALI-2 bus + (internally connected to terminal point 7)
DA -	4	DALI-2 bus - (internally connected to terminal point 8)
0 V	5	Power supply connection 0 V
Input 2	6	Digital input 2
DA +	7	DALI-2 bus + (internally connected to terminal point 3)
DA -	8	DALI-2 bus - (internally connected to terminal point 4)

Connection instructions

Notice Digital inputs "Input 1" (terminal point 2) and "Input 2" (terminal point 6)

- The digital inputs only work if the supply voltage (24 V) is applied and the E-bus is supplied with voltage. The potential for the 24 V inputs is the potential of the electrically isolated input voltage.
- Depending on the settings via the "FB_EL6821Communication" function block, other PLC-controlled DALI commands are blocked by activating the digital inputs (see Priority of the digital inputs [▸ 55]).

Notice Topology, cable length and cable cross-section in DALI operation

- The DALI bus can be configured in a line or star topology, or in a mix of the two.
- The maximum cable length must not exceed 300 m!
- The cable cross-section depends on the cable length:
  - for cable lengths up to 100 m: ≥ 0.5 mm^2
  - for cable lengths up to 150 m: ≥ 0.75 mm^2
  - for cable lengths up to 300 m: ≥ 1.5 mm^2

Notice Further important boundary conditions derived from IEC 62386:

- The electrically isolated input voltage of 24 V_{DC} must be supplied to comply with the certification conditions
  - via the EL9562 power supply terminal (see connection examples) or
  - via the power supplies PS1111-2402-0002, PS1111-2403-0000 or PS1111-2403-0002 (see connection example).
- The DALI cables must not be terminated with resistors.
- The maximum voltage drop between the sender and the receiver must not exceed 2 V.
- If the maximum cable length is utilized, it is not advisable to lay DALI in combination with the power cable.

Priority of the digital inputs

The digital Input 1 and Input 2 have priority over automatic control through the PLC program. If the digital inputs are operated, other PLC-controlled DALI commands are blocked. For release, a positive edge must be applied to the “bResetInactiveProcessImage” input of the “FB_EL6821Communication” function block (see documentation TwinCAT 3 | PLC libraries: Tc3_DALI). Alternatively, the priority rule can also be changed via the setting in TwinCAT 3 (do not lock process image).

Behavior of the digital inputs

The behavior of the digital inputs "Input 1" and "Input 2" can be changed via the library function blocks or via the settings in TwinCAT 3. To support commissioning, the following default behavior is assigned in the delivery state.

Signal	**Broadcast DALI command**	**Action**	**Comment**
Rising edge at DI1	00_{hex}	Turns OFF all control gears (without fading).	Factory setting
Rising edge at DI2	05_{hex}	Switches all control gears to maximum brightness. If a control gear is switched off, it is switched on.	Factory setting

4.8.1 Connection examples

The electrically isolated input voltage of 24 V_{DC} must be supplied to comply with the certification conditions via the EL9562 power supply terminal or the PS1111-2402-0002, PS1111-2403-0000 or PS1111-2403-0002 power supplies.

1. Up to 2 x EL6821 can be supplied via the EL9562 power supply terminal.

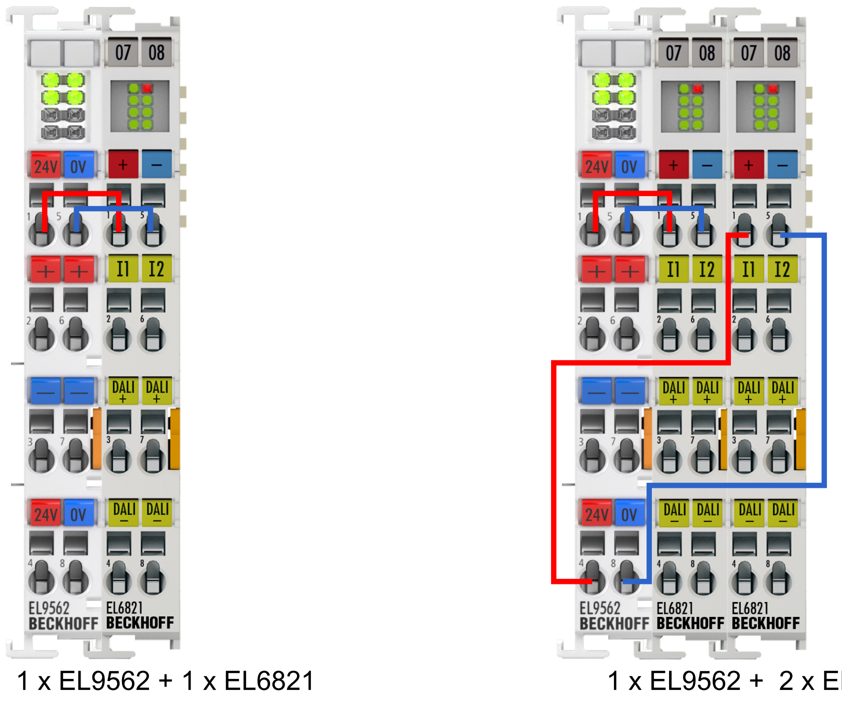


Fig. 26: Connection examples with EL9562 power supply terminal and 1 x EL6821 (left) / 2 x EL6821 (right)

2. Up to 12 x EL6821 or up to 19 x EL6821 can be supplied via the power supplies:
   ⇨ PS1111-2402-0002 (2.5 A) up to 12 x EL6821 or
   ⇨ PS1111-2403-0000 (3.8 A) up to 19 x EL6821 or
   ⇨ PS1111-2403-0002 (3.8 A) up to 19 x EL6821.

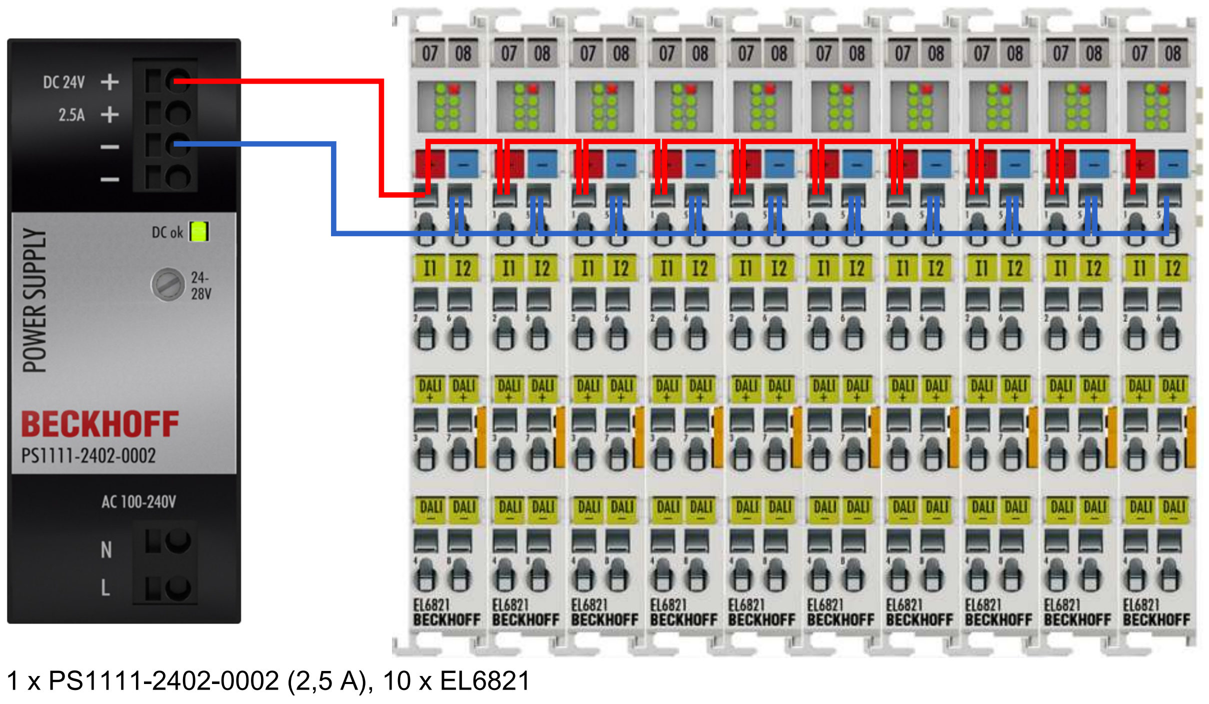


Fig. 27: Connection example with power supply PS1111-2402-0002 and 10 x EL6821

4.9 EL6821 - LEDs

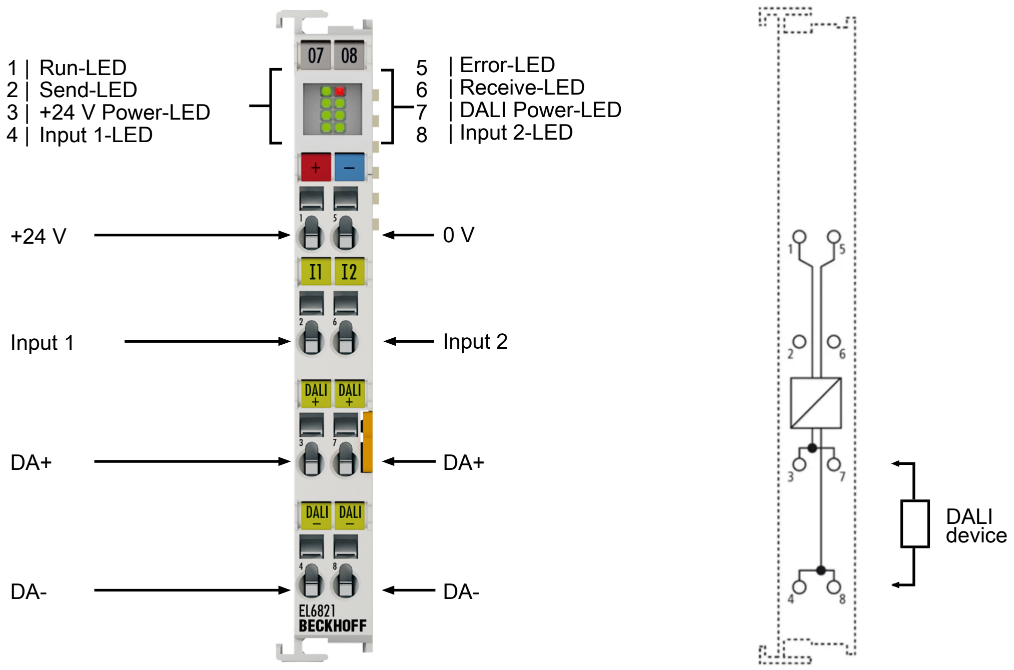


Fig. 28: EL6821

LED	Color	Meaning	
RUN	Green	This LED indicates the terminal's operating state:	
		Off	State of the EtherCAT State Machine [▶ 35]: INIT = initialization of the terminal or BOOTSTRAP = function for terminal firmware updates [▶ 130]
		Flashing	State of the EtherCAT State Machine: PREOP = function for mailbox communication and different default settings set
		Single flash	State of the EtherCAT State Machine: SAFEOP = verification of the Sync Manager [▶ 83] channels and the distributed clocks. Outputs remain in safe state.
		On	State of the EtherCAT State Machine: OP = normal operating state; mailbox and process data communication is possible
Send	green	flashes	A DALI telegram is sent every time the light came on.
+24 V Power	green	off	No 24 V_{DC} power supply at contact 1 and 5
		on	24 V_{DC} power supply at contact 1 and 5 okay
Input 1	green	off	The digital input "Input 1" is switched off.
		on	Digital input “Input 1” is switched on or was active and has not yet been acknowledged.
Error	red	on	No 24 V_{DC} power supply at contact 1 and 5 or hardware error
Receive	green	flashes	A DALI telegram was received each time the light came on.
DALI Power	green	off	The integrated DALI power supply is switched off.
		on	The terminal supplies the DALI bus with power.
Input 2	green	off	The digital input "Input 2" is switched off.
		on	The digital input “Input 2” is switched on or was active and has not yet been acknowledged.
The status of the LEDs can also be read via CoE object 0xF915 (see chapter EL6821 - Device diagnostic functions [▶ 115]).			

5 Configuration via TwinCAT

5.1 TwinCAT Development Environment

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

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

Details:

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

Additional features:

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

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

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

5.1.1 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.

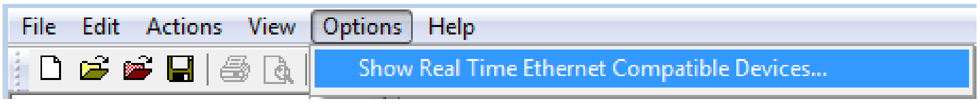


Fig. 29: System Manager “Options” (TwinCAT 2)

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

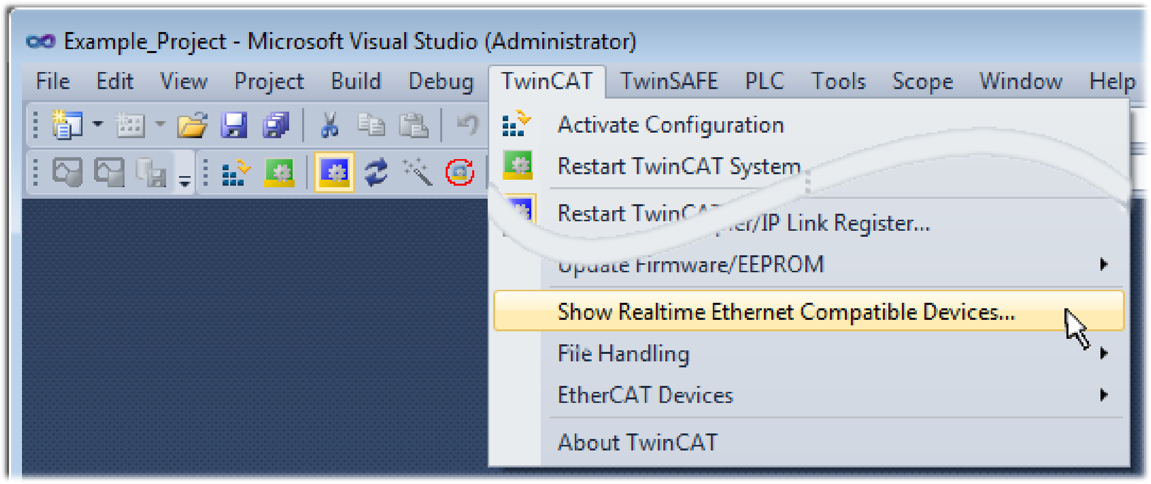


Fig. 30: Call up under VS Shell (TwinCAT 3)

B: Via TcRteInstall.exe in the TwinCAT directory

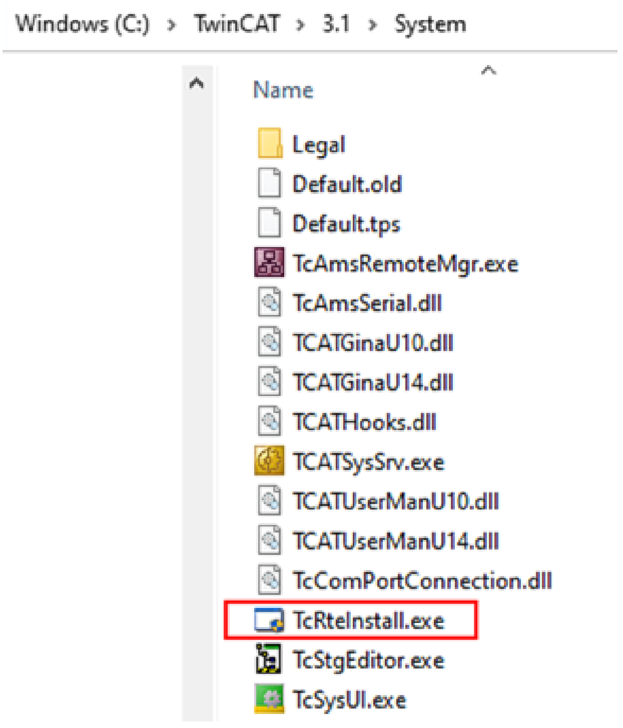


Fig. 31: TcRteInstall in the TwinCAT directory

In both cases, the following dialog appears:

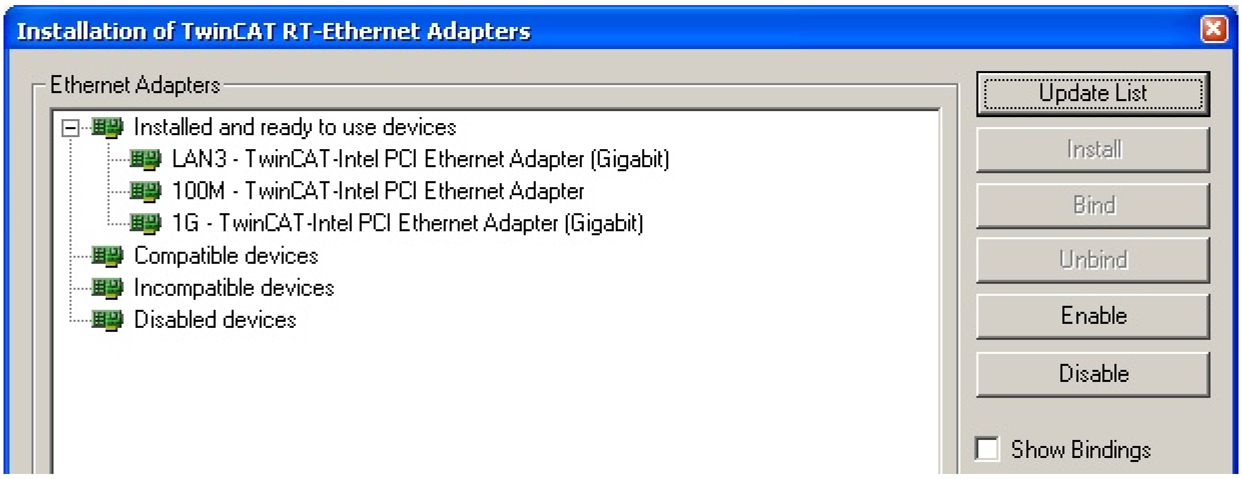
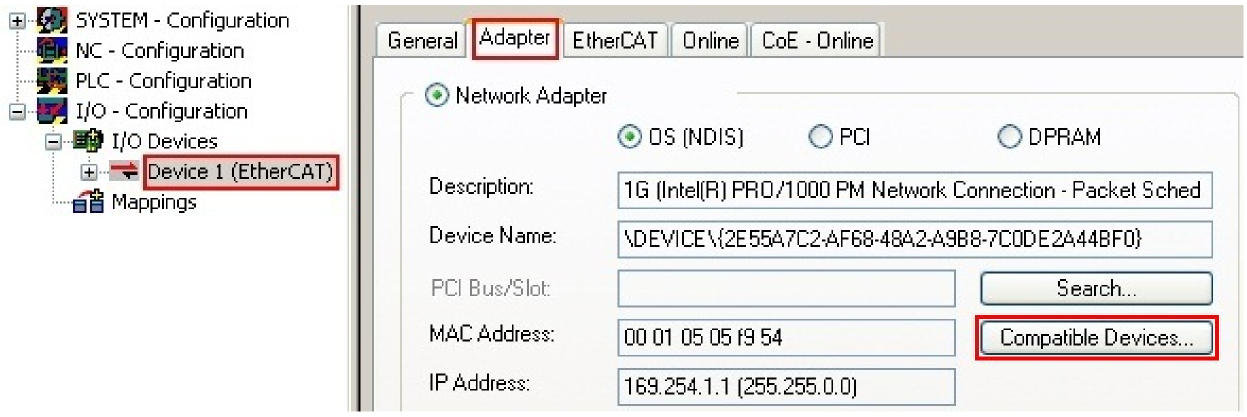

Installation of TwinCAT RT-Ethernet Adapters
Ethernet Adapters
Installed and ready to use devices
LAN3 - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
100M - TwinCAT-Intel PCI Ethernet Adapter
1G - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
Compatible devices
Incompatible devices
Disabled devices
Update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

Fig. 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.

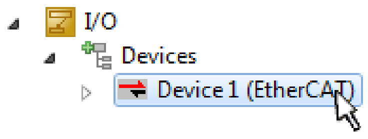
Alternatively an EtherCAT-device can be inserted first of all as described in chapter Offline configuration creation, section “Creating the EtherCAT device” [▸ 69] in order to view the compatible ethernet ports via its EtherCAT properties (tab “Adapter”, button “Compatible Devices…”):



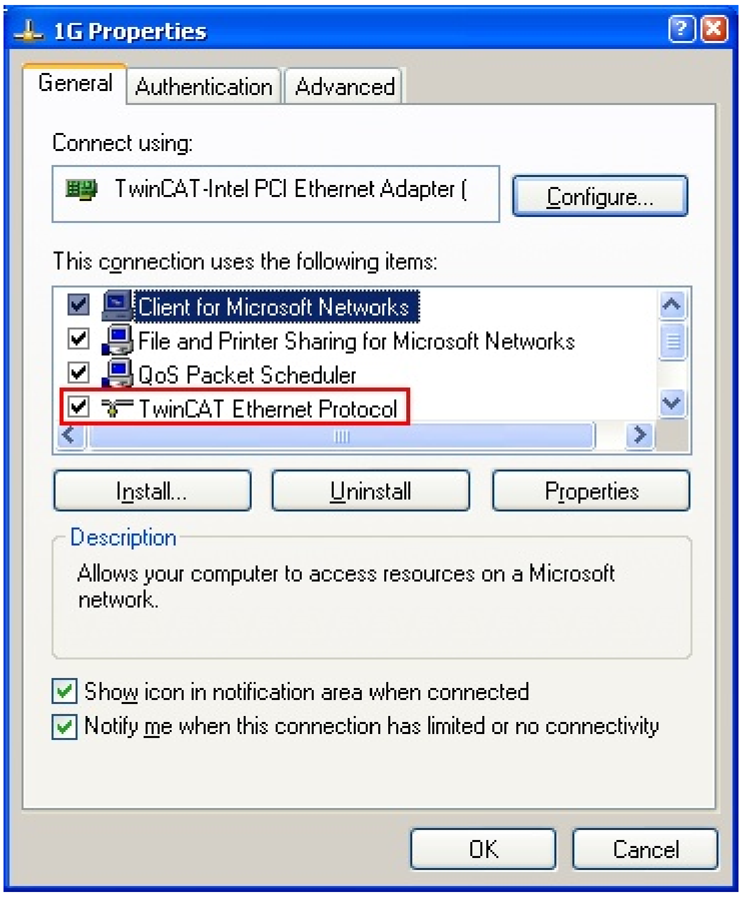
Network Adapter
[✓ OS (NDIS)] PCI DPRAM
Description: 1G (Intel(R) PRO/1000 PM Network Connection - Packet Sched
Device Name: \DEVICE\{2E55A7C2-AF68-48A2-A9B8-7C0DE2A44BF0}
PCI Bus/Slot:
MAC Address: 00 01 05 05 f9 54
IP Address: 169.254.1.1 (255.255.0.0)

Fig. 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)



1G Properties

General | Authentication | Advanced

Connect using:

TwinCAT-Intel PCI Ethernet Adapter (

Configure...

This connection uses the following items:

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

Install... | Uninstall | Properties

Description

Allows your computer to access resources on a Microsoft network.

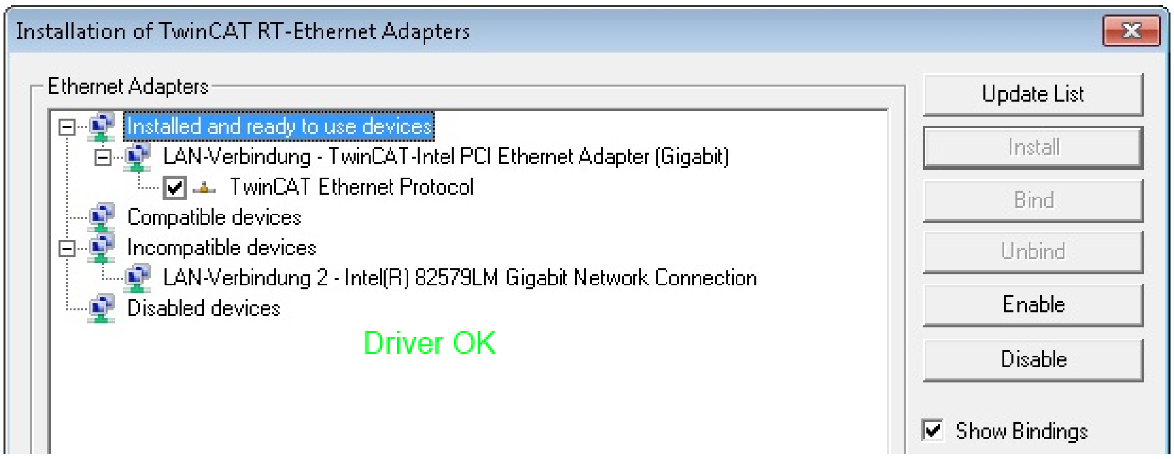
☑ Show icon in notification area when connected

☑ Notify me when this connection has limited or no connectivity

OK | Cancel

Fig. 34: Windows properties of the network interface

A correct setting of the driver could be:



Installation of TwinCAT RT-Ethernet Adapters

Ethernet Adapters

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

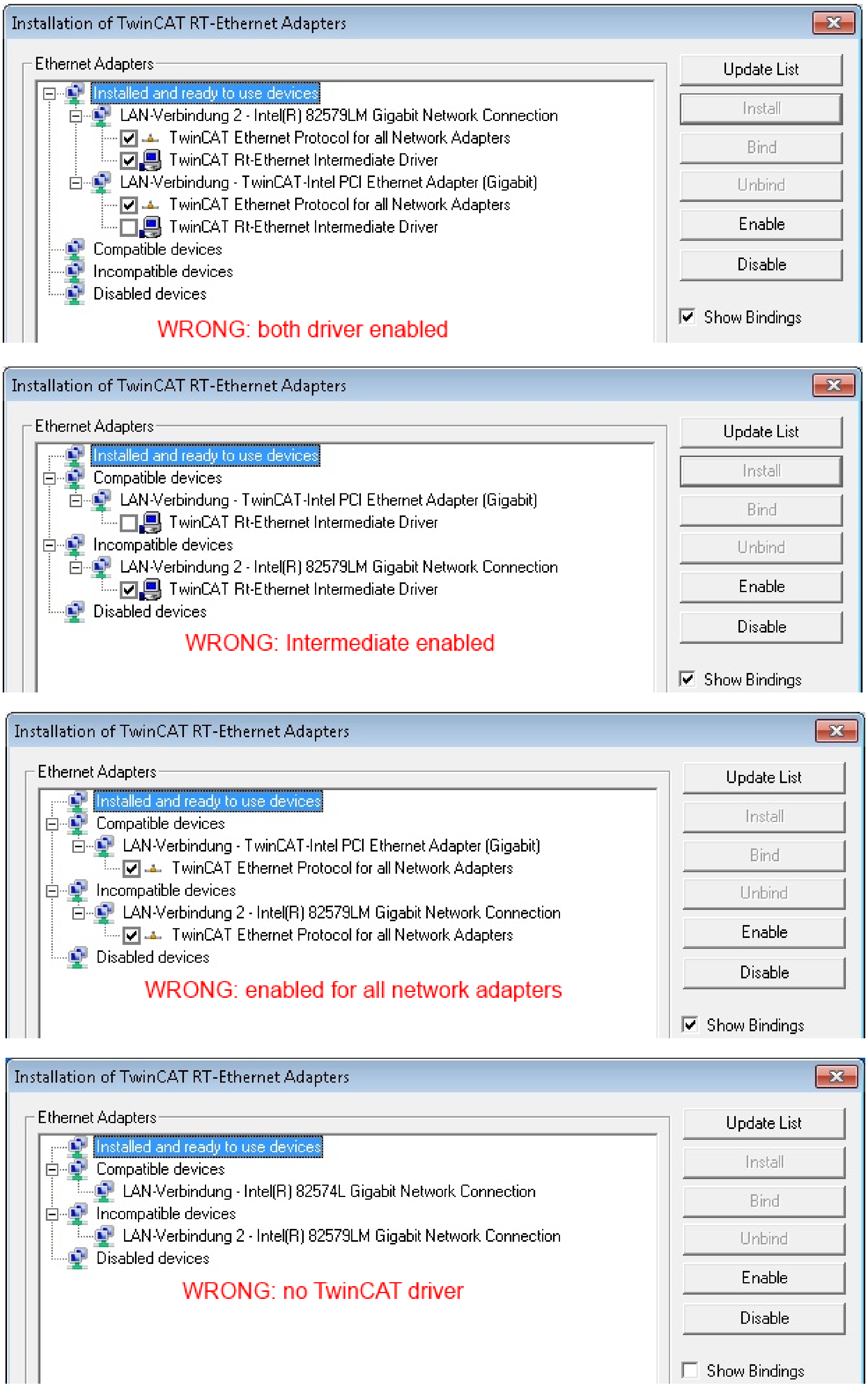
Driver OK

Update List | Install | Bind | Unbind | Enable | Disable

☑ Show Bindings

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

Other possible settings have to be avoided:


Installation of TwinCAT RT-Ethernet Adapters

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

WRONG: both driver enabled

[✓] Show Bindings

Installation of TwinCAT RT-Ethernet Adapters

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

WRONG: Intermediate enabled

[✓] Show Bindings

Installation of TwinCAT RT-Ethernet Adapters

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

WRONG: enabled for all network adapters

[✓] Show Bindings

Installation of TwinCAT RT-Ethernet Adapters

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

WRONG: no TwinCAT driver

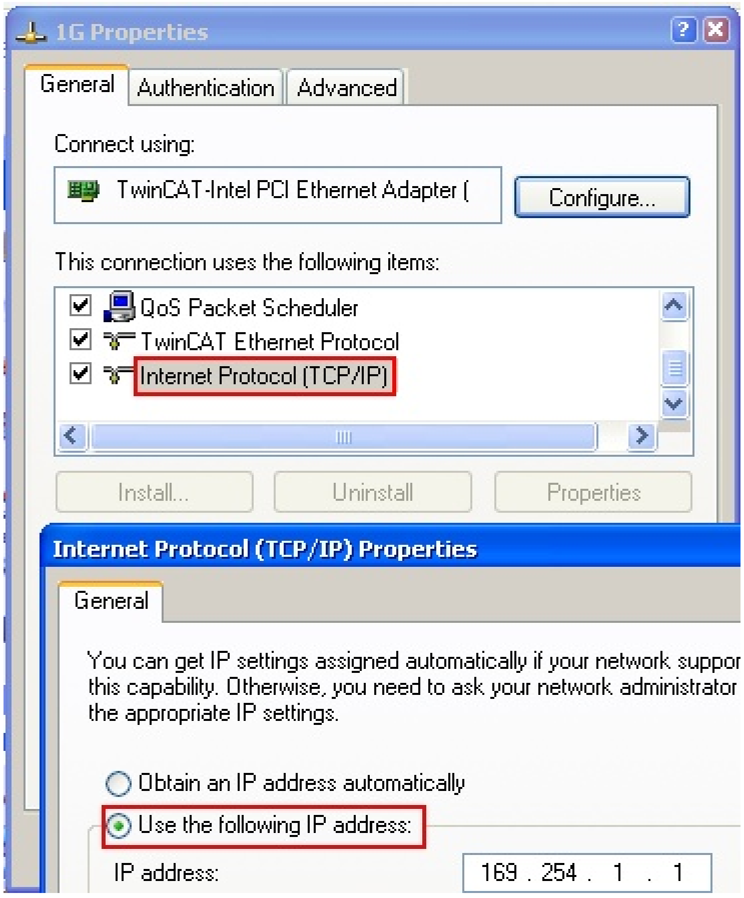
[ ] Show Bindings

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.



Connect using:

TwinCAT-Intel PCI Ethernet Adapter (

This connection uses the following items:

- [✓] QoS Packet Scheduler
- [✓] TwinCAT Ethernet Protocol
- [✓] Internet Protocol (TCP/IP)

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

Obtain an IP address automatically [✓ Use the following IP address:]

IP address: 169 . 254 . 1 . 1

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

5.1.2 Notes regarding ESI device description

Installation of the latest ESI device description

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

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

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

Default settings:

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

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

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

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

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

The TwinCAT ESI Updater [▶ 68] 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”

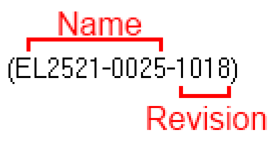
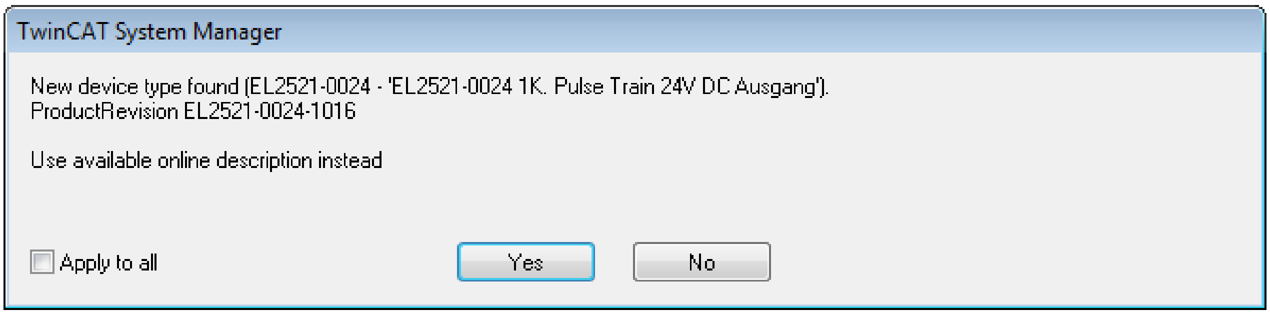


Fig. 38: Identifier structure

The order identifier consisting of name + type (here: EL2521-0025) 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 [▶ 9].

Online description

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

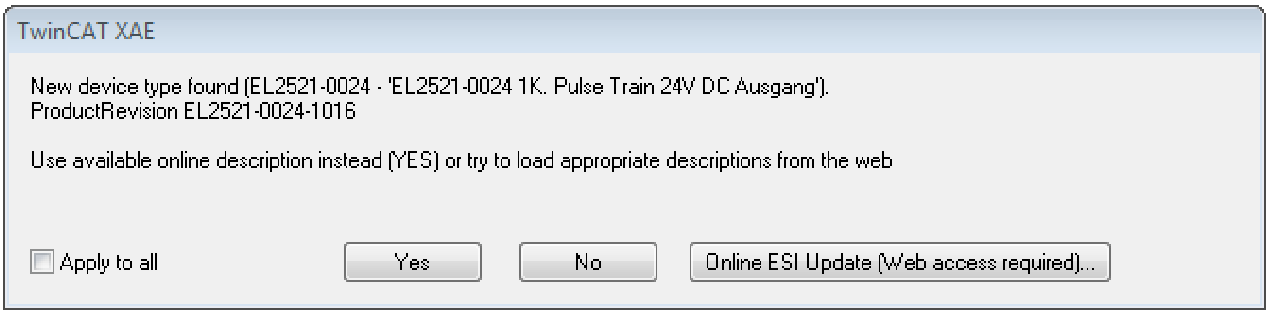

TwinCAT System Manager

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

Use available online description instead

Fig. 39: OnlineDescription information window (TwinCAT 2)

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


TwinCAT XAE

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

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

Fig. 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.

NOTICE

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 [▶ 69]”.

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.

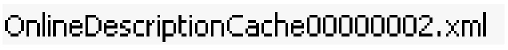

OnlineDescriptionCache00000002.xml

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)*.

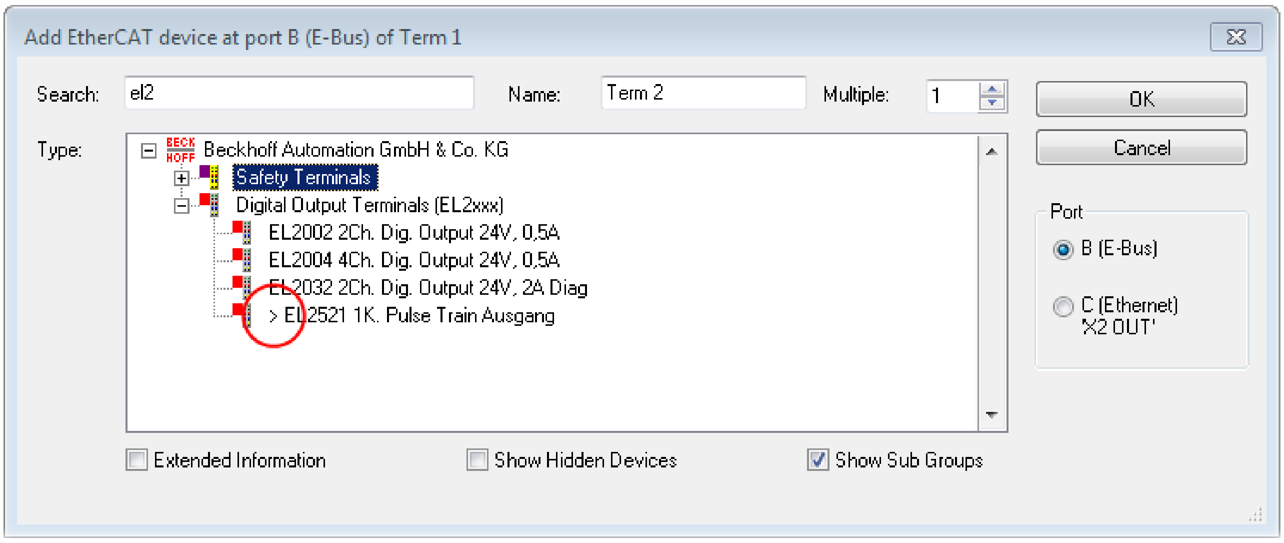

Add EtherCAT device at port B (E-Bus) of Term 1
Search: el2
Name: Term 2
Multiple: 1
OK
Cancel
Type:
Beckhoff Automation GmbH & Co. KG
Safety Terminals
Digital Output Terminals (EL2xxx)
EL2002 2Ch. Dig. Output 24V, 0,5A
EL2004 4Ch. Dig. Output 24V, 0,5A
EL2032 2Ch. Dig. Output 24V, 2A Diag
> EL2521 1K. Pulse Train Ausgang
Port
B (E-Bus)
C (Ethernet) 'X2 OUT'
Extended Information
Show Hidden Devices
Show Sub Groups

Fig. 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

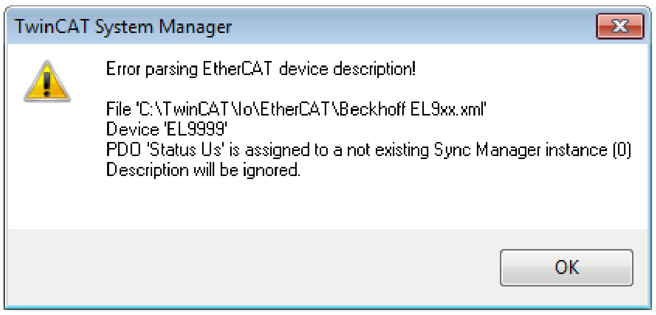
In addition to the file described above “OnlineDescription0000...xml”, a so called EtherCAT cache with new discovered devices is created by TwinCAT 3.x, e.g. under Windows 7:

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

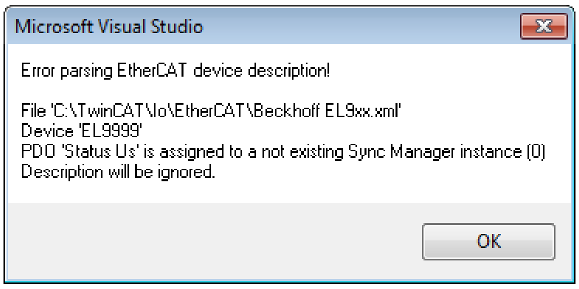
If an ESI file is faulty and the System Manager is unable to read it, the System Manager brings up an information window.


TwinCAT System Manager

Error parsing EtherCAT device description!

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

OK


Microsoft Visual Studio

Error parsing EtherCAT device description!

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

OK

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

5.1.3 TwinCAT ESI Updater

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

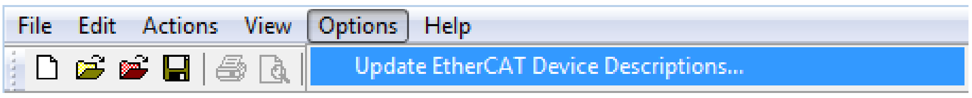
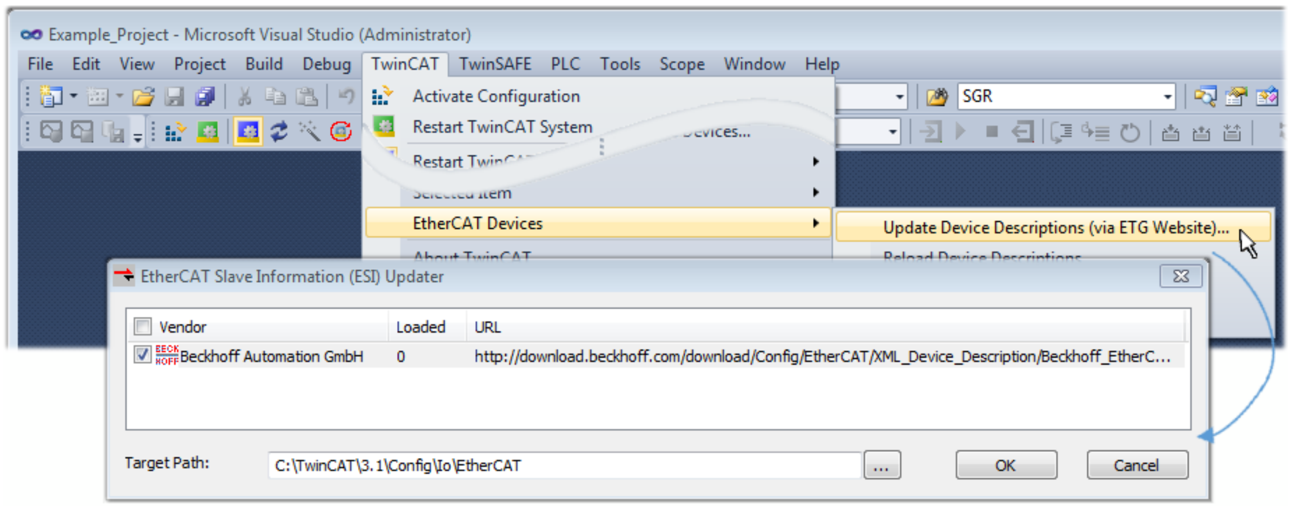


Fig. 44: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:


Example_Project - Microsoft Visual Studio (Administrator)
File
Edit
View
Project
Build
Debug
TwinCAT
TwinSAFE
PLC
Tools
Scope
Window
Help
Activate Configuration
Restart TwinCAT System
EtherCAT Devices
Update Device Descriptions (via ETG Website)...
SGR
EtherCAT Slave Information (ESI) Updater
OK
Cancel

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

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

Fig. 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)…”.

5.1.4 Distinction between Online and Offline

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

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

In any case, during each startup the EtherCAT master checks whether the slaves it finds match the configuration. This test can be parameterised in the extended slave settings. Refer to note “Installation of the latest ESI-XML device description” [▶ 64].

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 [▶ 74] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 75]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 78]

The scan with existing configuration [▶ 79] can also be carried out for comparison.

5.1.5 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

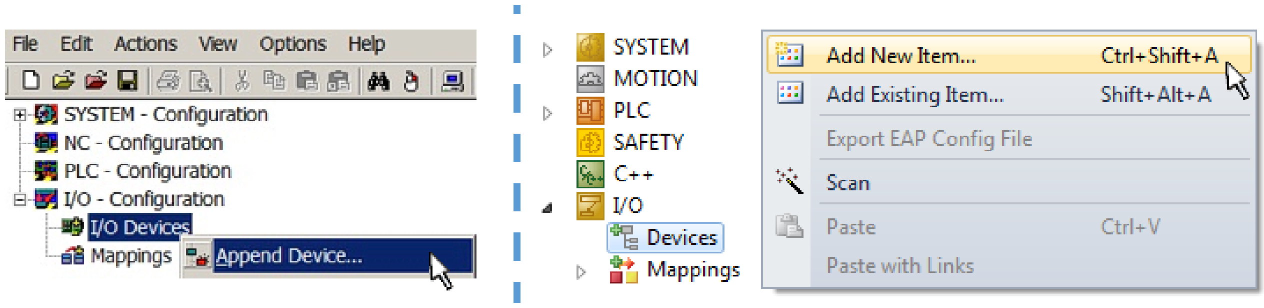


Fig. 46: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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”.

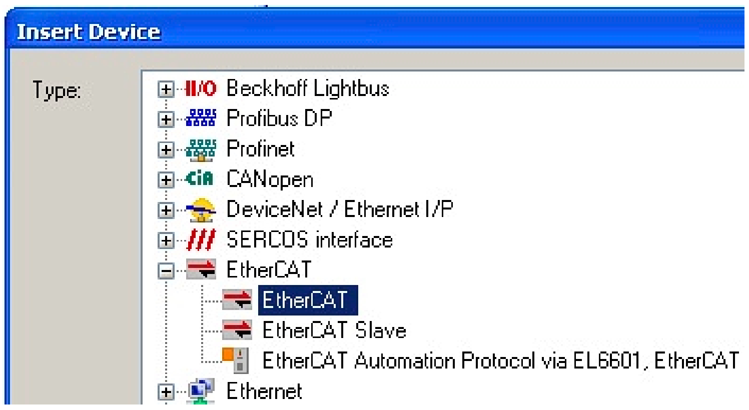


Fig. 47: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

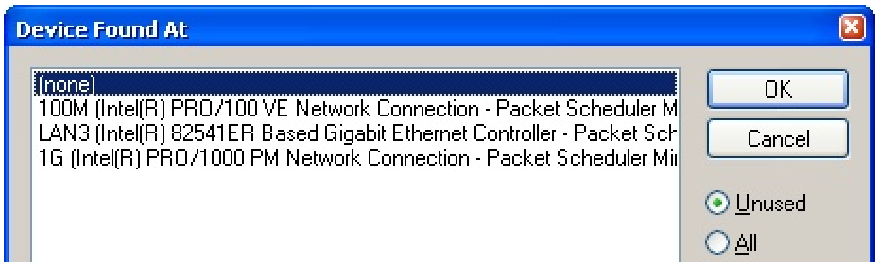
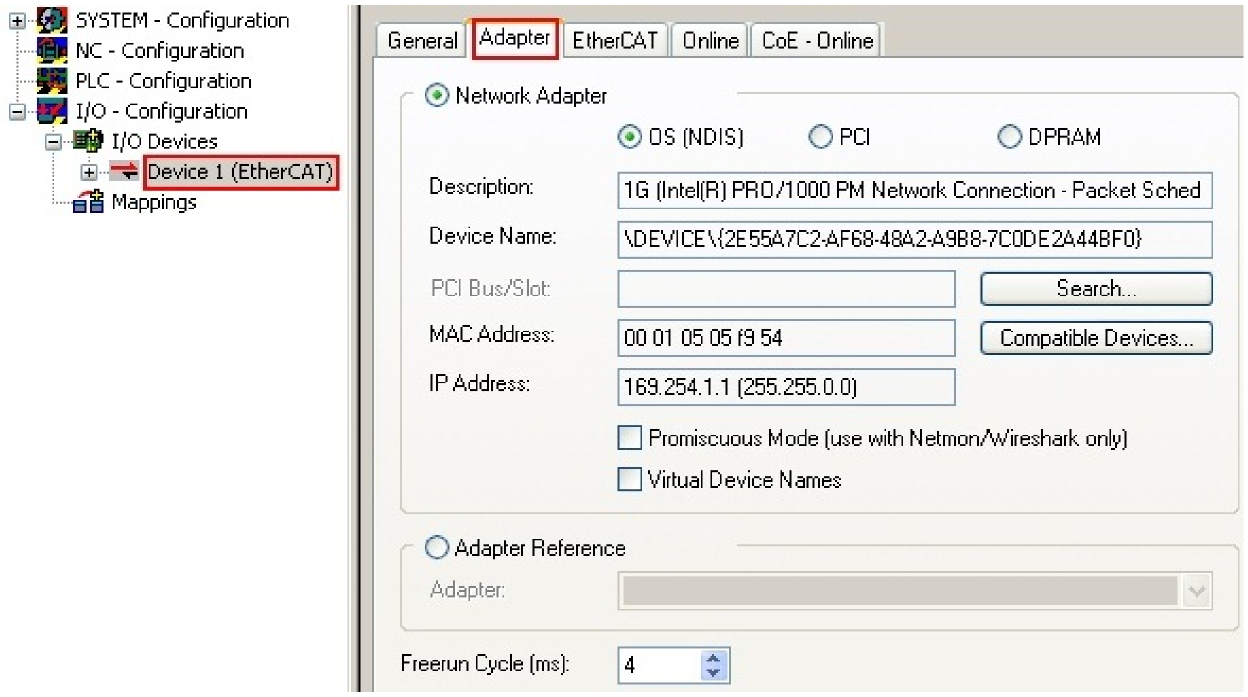

Device Found At
(none)
100M (Intel(R) PRO/100 VE Network Connection - Packet Scheduler M
LAN3 (Intel(R) 82541ER Based Gigabit Ethernet Controller - Packet Sch
1G (Intel(R) PRO/1000 PM Network Connection - Packet Scheduler Mi
OK
Cancel
Unused
All

Fig. 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)”.


SYSTEM - Configuration
NC - Configuration
PLC - Configuration
I/O - Configuration
I/O Devices
Device 1 (EtherCAT)
Mappings

General | Adapter | EtherCAT | Online | CoE - Online

(●) Network Adapter
(●) OS (NDIS) ( ) PCI ( ) DPRAM

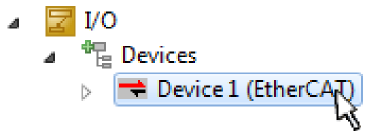
Description: 1G (Intel(R) PRO/1000 PM Network Connection - Packet Sched
Device Name: \DEVICE\{2E55A7C2-AF68-48A2-A9B8-7C0DE2A44BF0}
PCI Bus/Slot: [Search...]
MAC Address: 00 01 05 05 f9 54 [Compatible Devices...]
IP Address: 169.254.1.1 (255.255.0.0)
☐ Promiscuous Mode (use with Netmon/Wireshark only)
☐ Virtual Device Names

( ) Adapter Reference
Adapter:

Freerun Cycle (ms): 4

Fig. 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 [▸ 58].

Defining EtherCAT slaves

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

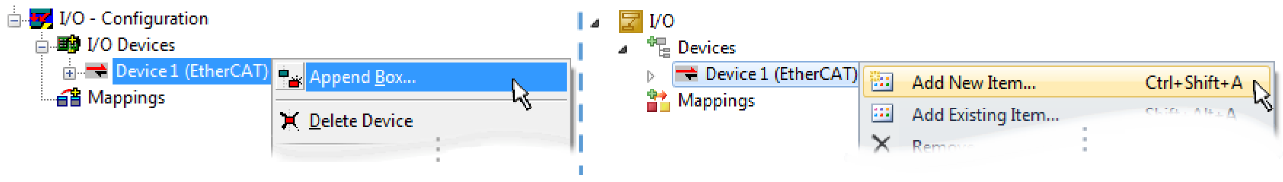


Fig. 50: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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

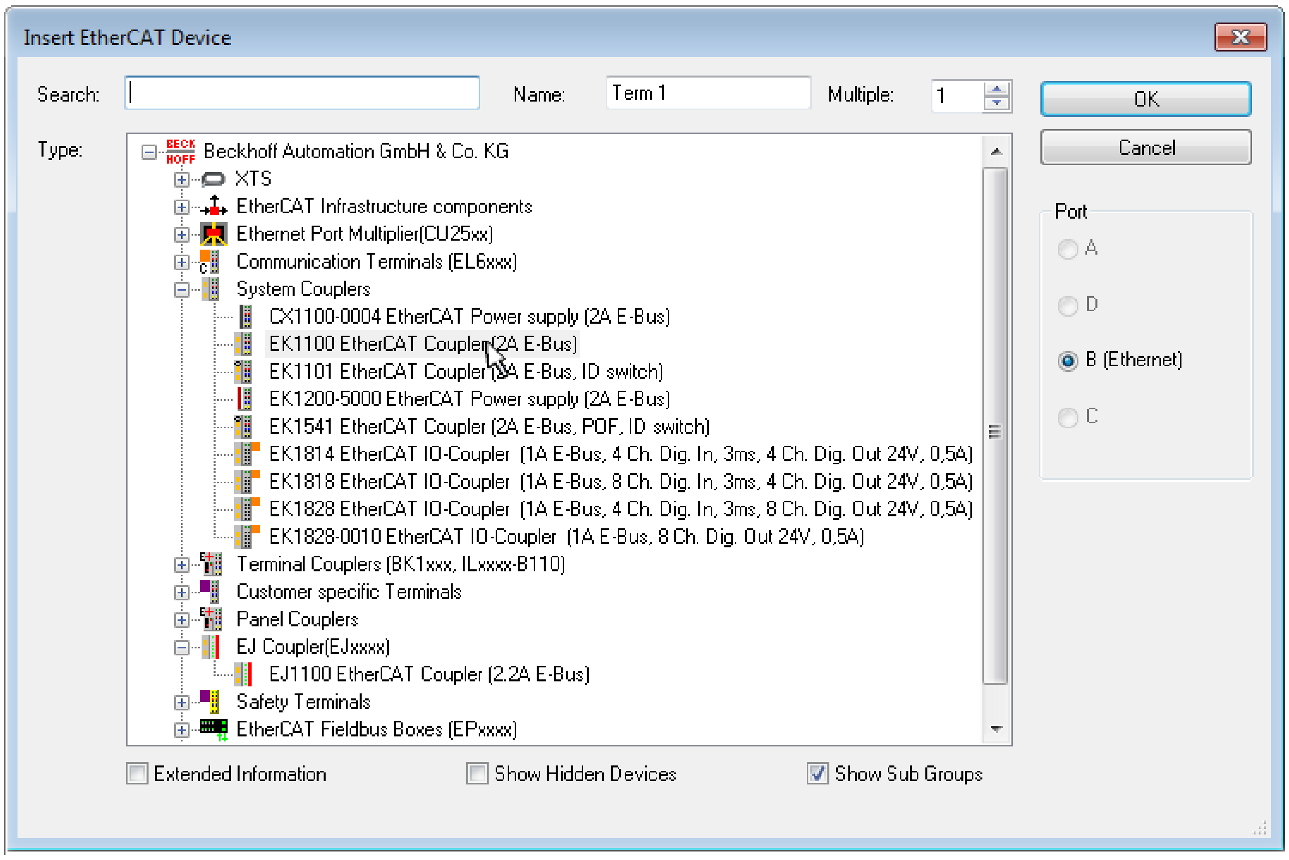
Only devices are offered for selection that can be appended to the previously selected device. Therefore, the physical layer available for this port is also displayed (Fig. “Selection dialog for new EtherCAT device”, A). In the case of cable-based Fast-Ethernet physical layer with PHY transfer, then 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).



Insert EtherCAT Device

Search: | Name: Term 1 | Multiple: 1 | OK | Cancel

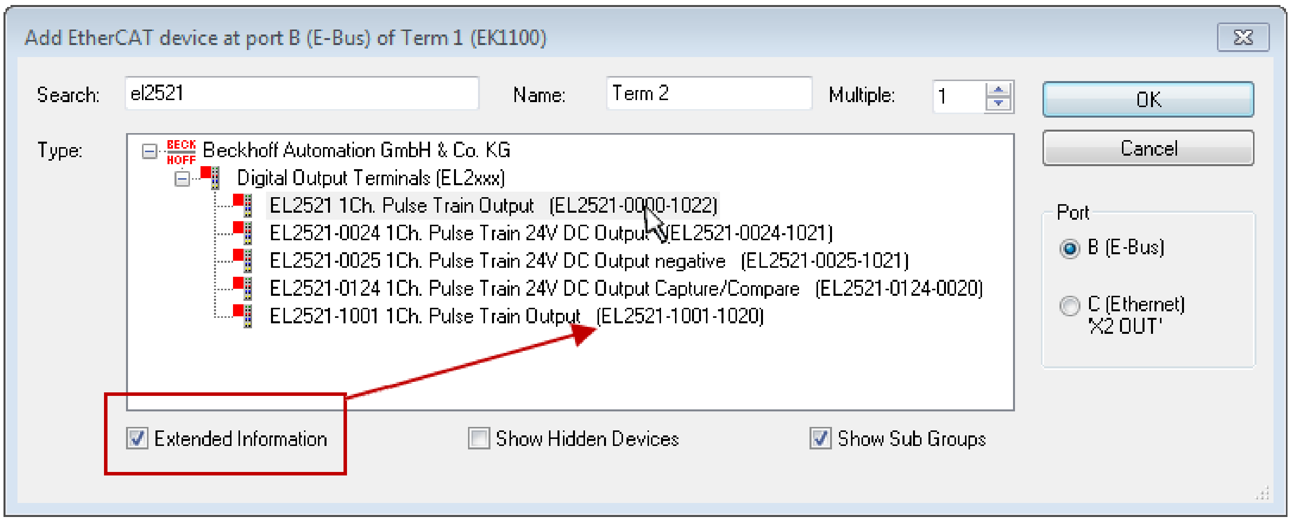
Type:
- Beckhoff Automation GmbH & Co. KG
  - XTS
  - EtherCAT Infrastructure components
  - Ethernet Port Multiplier(CU25xx)
  - Communication Terminals (EL6xxx)
  - System Couplers
    - CX1100-0004 EtherCAT Power supply (2A E-Bus)
    - EK1100 EtherCAT Coupler (2A E-Bus)
    - EK1101 EtherCAT Coupler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

Port: A, D, [✓ B (Ethernet)], C

☐ Extended Information ☐ Show Hidden Devices ☑ Show Sub Groups

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".



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

Search: el2521 | Name: Term 2 | Multiple: 1 | OK | Cancel

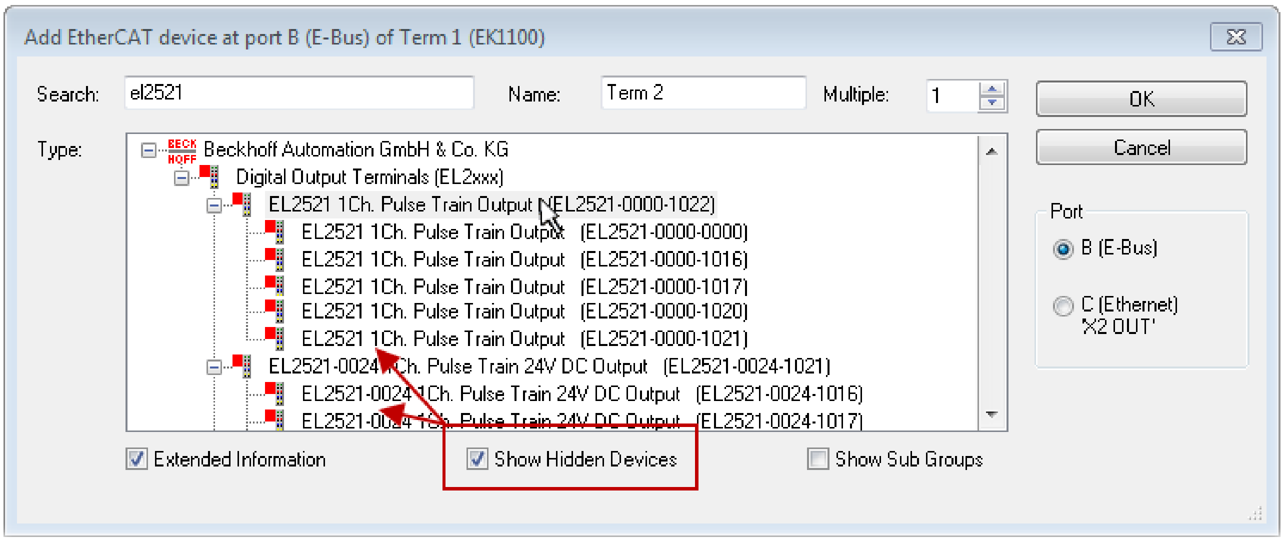
Type:
- Beckhoff Automation GmbH & Co. KG
  - Digital Output Terminals (EL2xxx)
    - EL2521 1Ch. Pulse Train Output (EL2521-0000-1022)
    - EL2521-0024 1Ch. Pulse Train 24V DC Output (EL2521-0024-1021)
    - EL2521-0025 1Ch. Pulse Train 24V DC Output negative (EL2521-0025-1021)
    - EL2521-0124 1Ch.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

Port: [✓ B (E-Bus)], C (Ethernet) 'X2 OUT'

☑ Extended Information ☐ Show Hidden Devices ☑ Show Sub Groups

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".


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

Search: el2521
Name: Term 2
Multiple: 1

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

Port: [✓ B (E-Bus)] C (Ethernet) 'X2 OUT'

[✓] Extended Information [✓] Show Hidden Devices [ ] Show Sub Groups

Fig. 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.

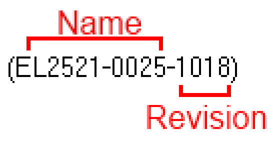


Fig. 54: Name/revision of the terminal

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, ...

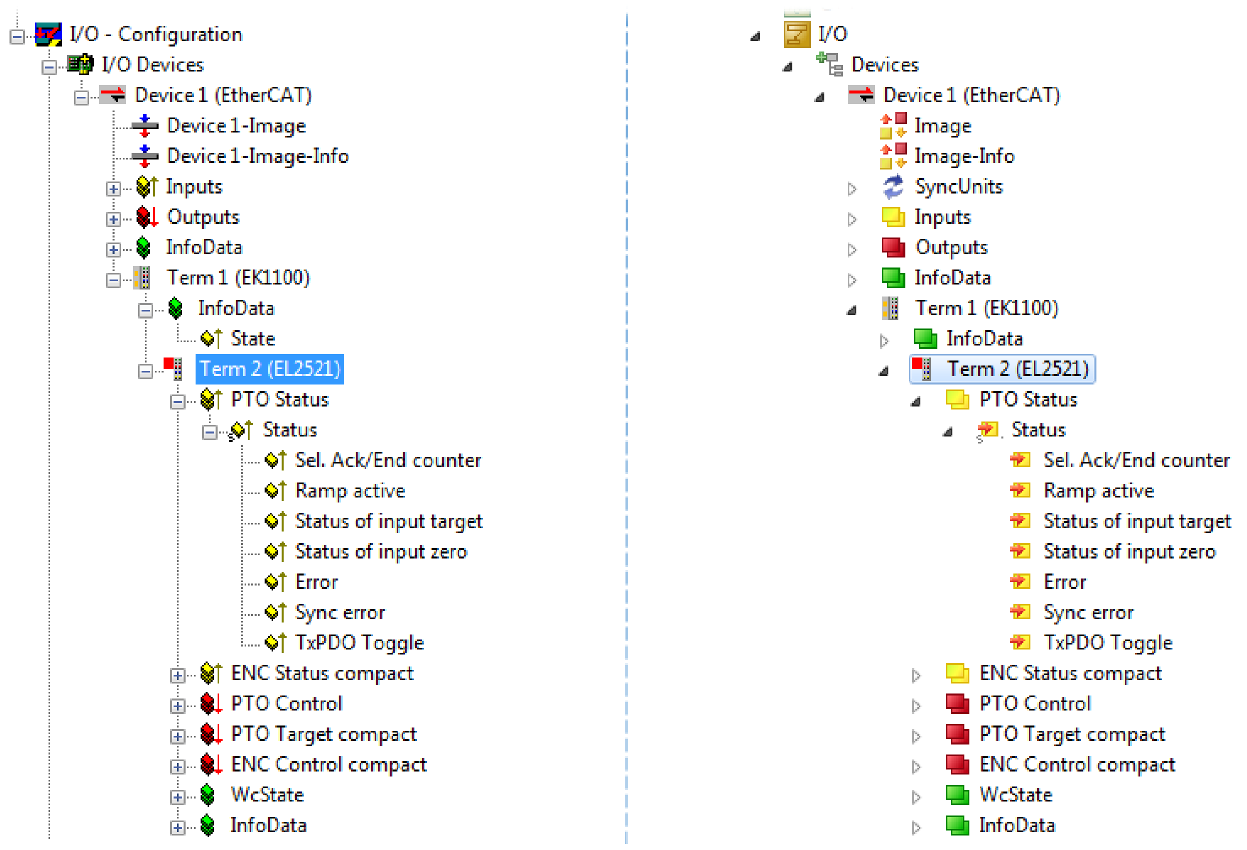


Fig. 55: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)

5.1.6 ONLINE configuration creation

Detecting/scanning of the EtherCAT device

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

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

TwinCAT can be set into this mode:

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

Online scanning in Config mode

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

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

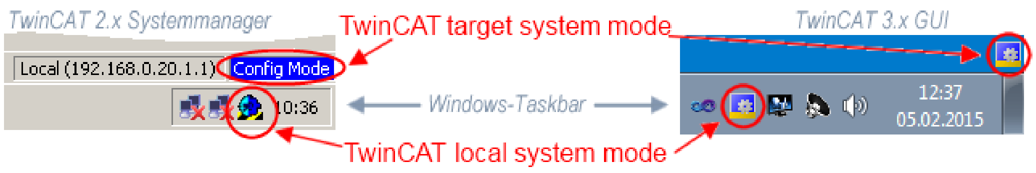


Fig. 56: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

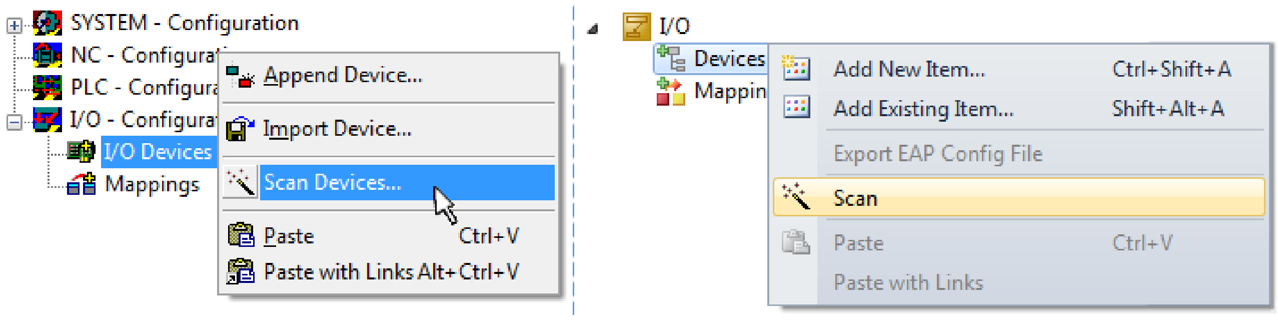
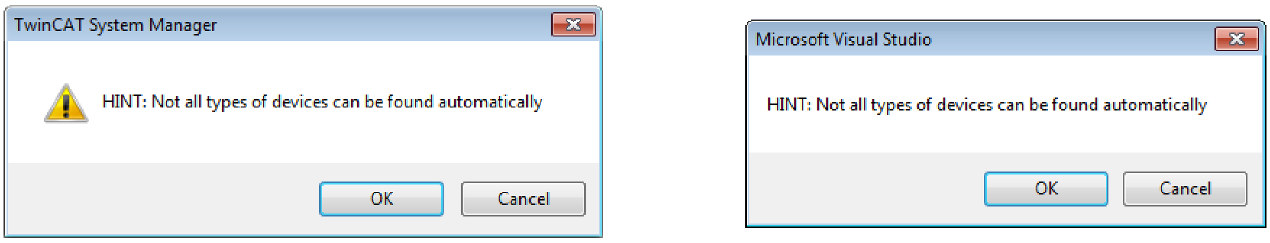


Fig. 57: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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.

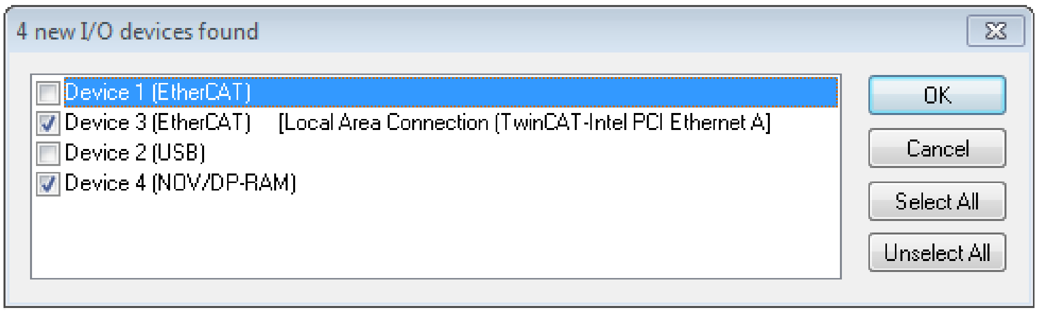


HINT: Not all types of devices can be found automatically

HINT: Not all types of 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” .


4 new I/O devices found

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

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 [▸ 58].

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.

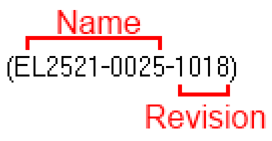


Fig. 60: Example default state

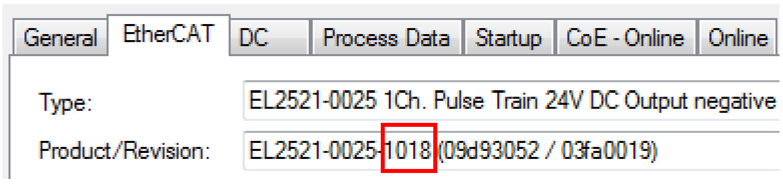
NOTICE

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

Example:

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



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

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

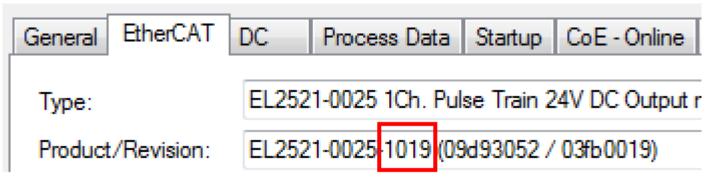
Fig. 61: Installing EthetCAT terminal with revision -1018

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

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



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

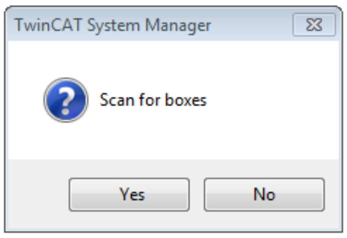
Product/Revision: EL2521-0025-1019 (09d93052 / 03fb0019)

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

This is usually not noticed by the commissioning engineers. TwinCAT cannot signal anything either, since a new configuration is essentially 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.



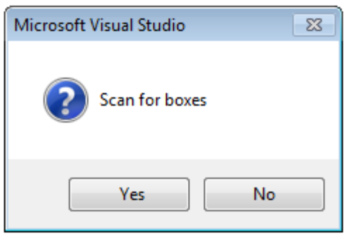


Fig. 63: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

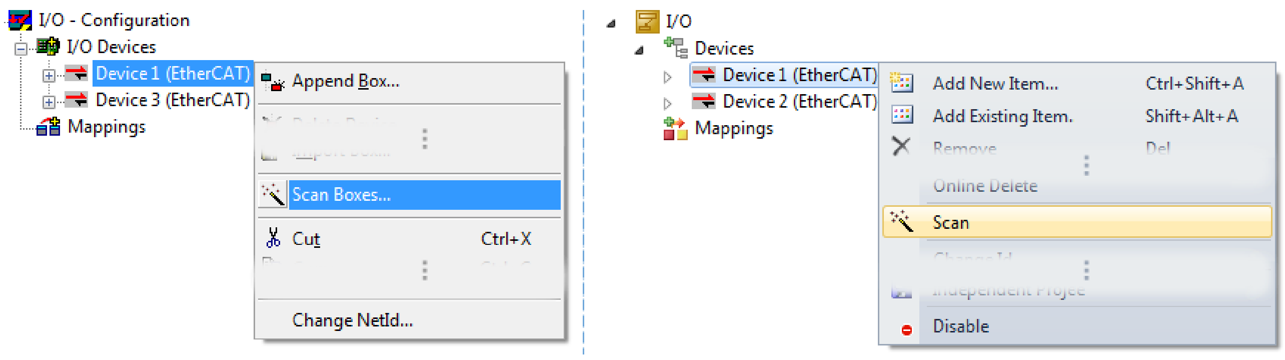


Fig. 64: Manual scanning for devices on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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.

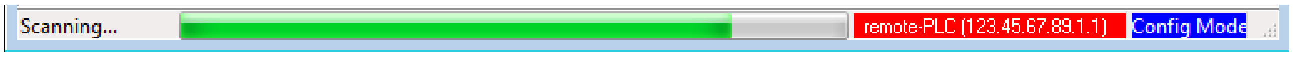


Fig. 65: *Scan progressexemplary by TwinCAT 2*

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

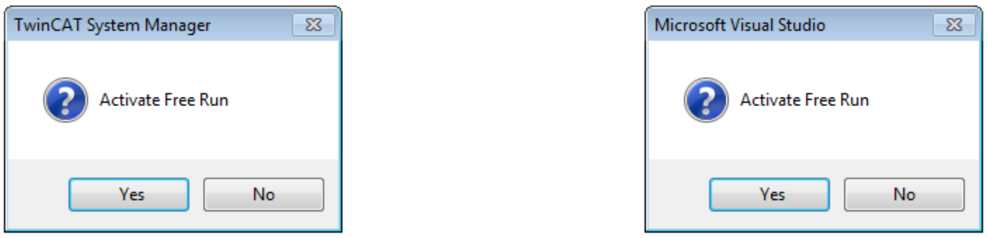


Fig. 66: Config/FreeRun query (left: TwinCAT 2; right: TwinCAT 3)

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).

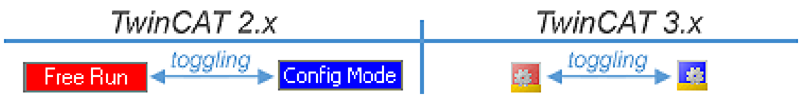


Fig. 67: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar

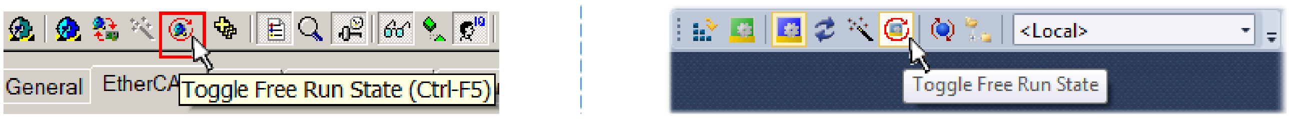
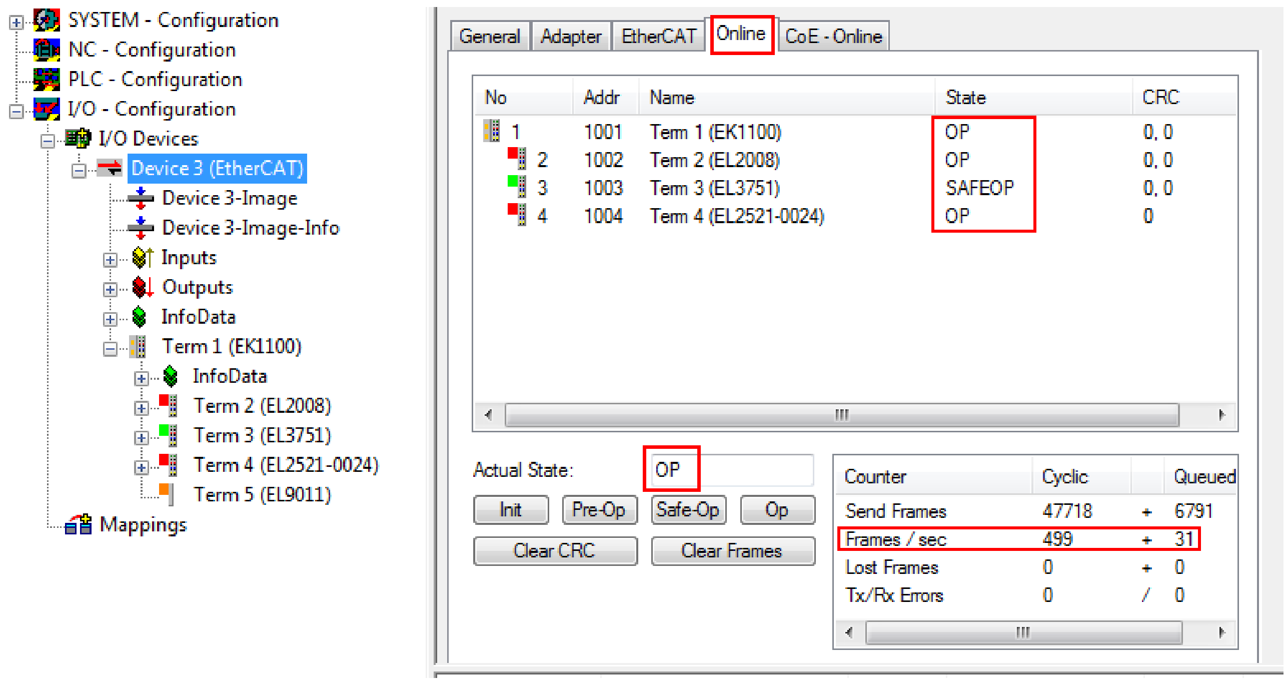


Fig. 68: TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3)

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



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

Actual State: OP

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

Fig. 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 [▶ 69].

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.

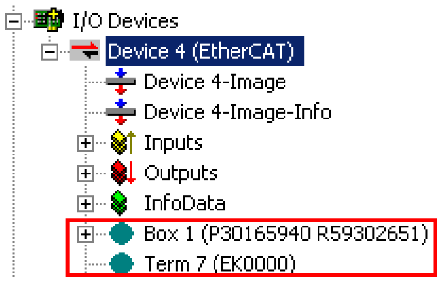


Fig. 70: Faulty identification

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

NOTICE

Change of the configuration after comparison

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

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

TwinCAT System Manager
Configuration is identical
OK
Microsoft Visual Studio
Configuration is identical
OK

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.

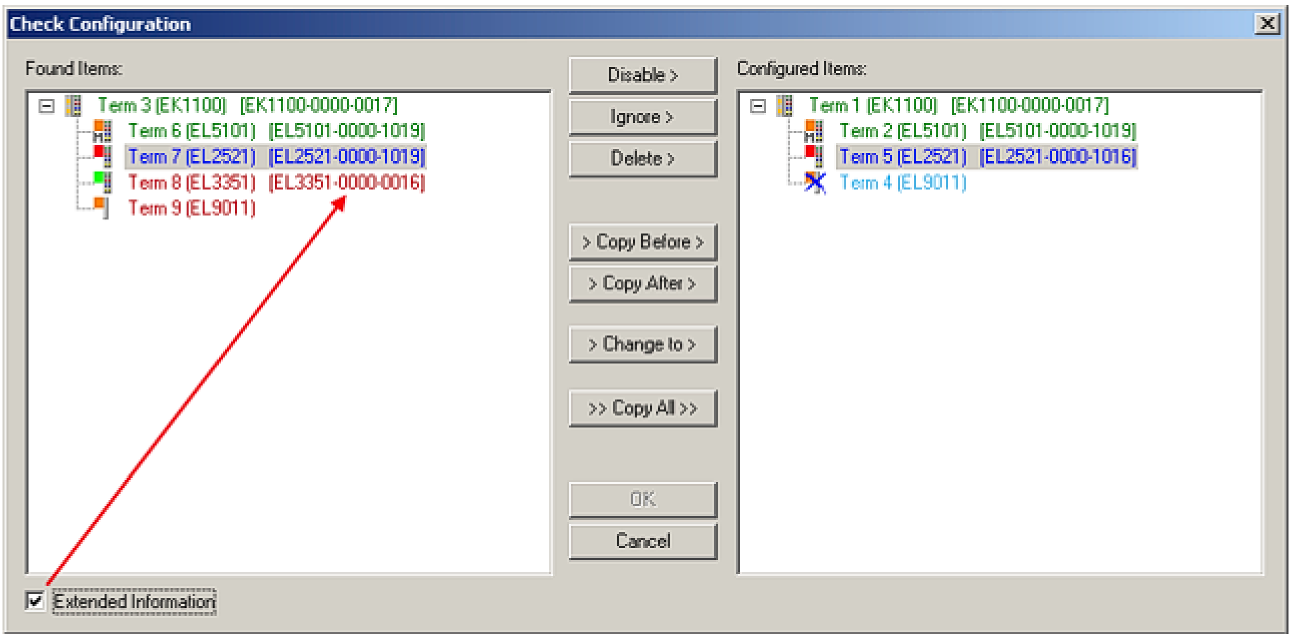

Check Configuration
Found Items:
Term 3 (EK1100) [EK1100-0000-0017]
Term 6 (EL5101) [EL5101-0000-1019]
Term 7 (EL2521) [EL2521-0000-1019]
Term 8 (EL3351) [EL3351-0000-0016]
Term 9 (EL9011)
Disable >
Ignore >
Delete >
> Copy Before >
> Copy After >
> Change to >
>> Copy All >>
OK
Cancel
Configured Items: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

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.

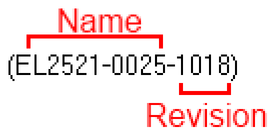


Fig. 73: Name/revision of the terminal

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, ...

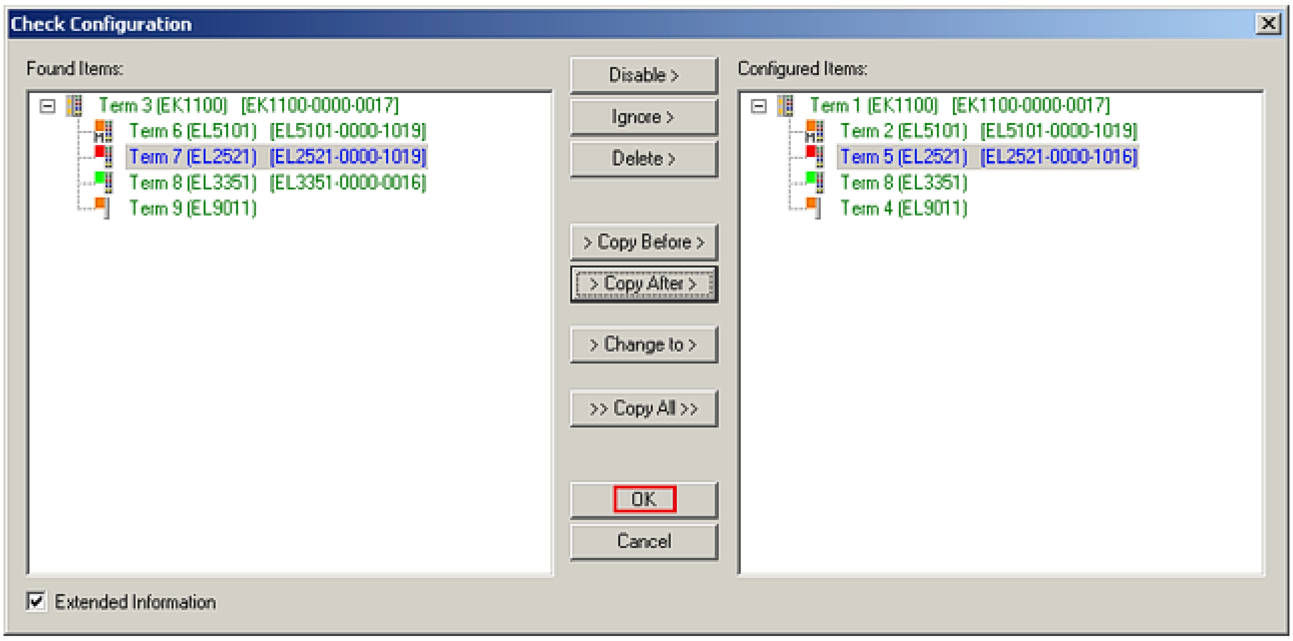

Check Configuration
Found Items:
Term 3 (EK1100) [EK1100-0000-0017]
Term 6 (EL5101) [EL5101-0000-1019]
Term 7 (EL2521) [EL2521-0000-1019]
Term 8 (EL3351) [EL3351-0000-0016]
Term 9 (EL9011)
Disable >
Ignore >
Delete >
> Copy Before >
> Copy After >
> Change to >
>> Copy All >>
OK
Cancel
Configured Items: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

Fig. 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.

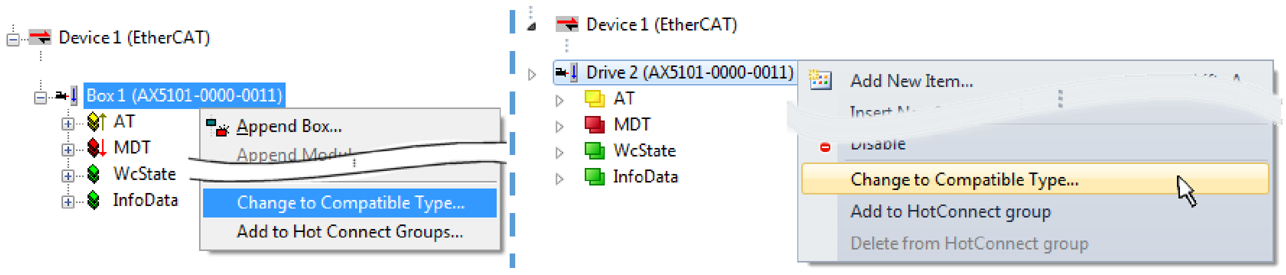


Fig. 75: Dialog “Change to Compatible Type…” (left: TwinCAT 2; right: TwinCAT 3)

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

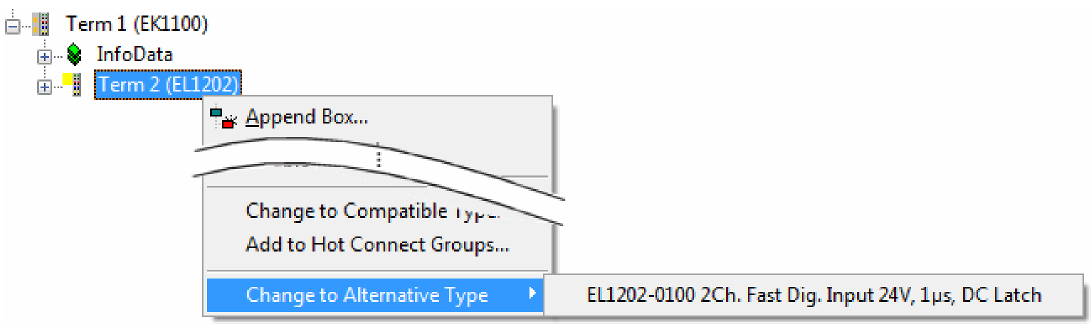

Term 1 (EK1100)
InfoData
Term 2 (EL1202)
Append Box...
Change to Compatible Type...
Add to Hot Connect Groups...
Change to Alternative Type
EL1202-0100 2Ch. Fast Dig. Input 24V, 1µs, DC Latch

Fig. 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).

5.1.7 EtherCAT subscriber configuration

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

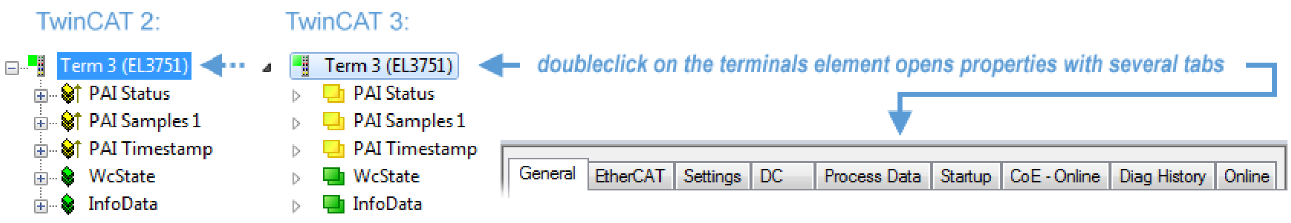
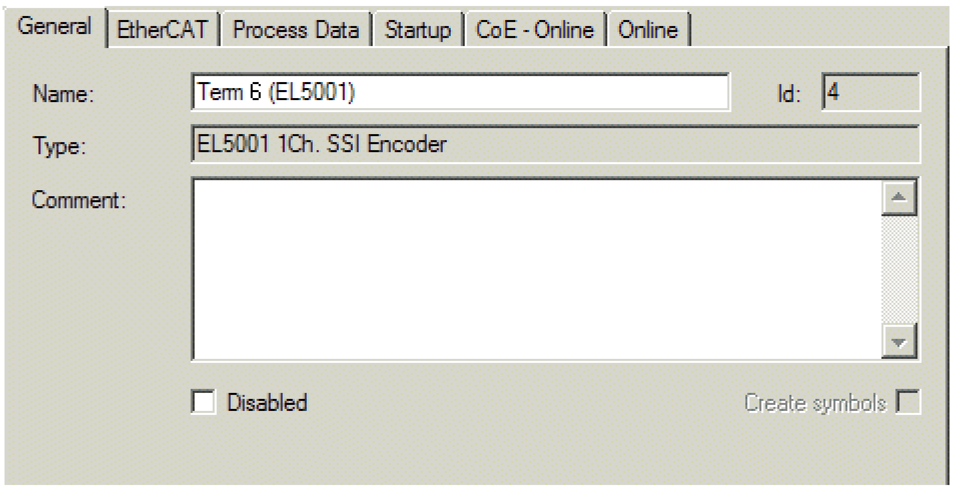


Fig. 77: Branch element as terminal EL3751

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



Name: Term 6 (EL5001)

Id: 4

Type: EL5001 1Ch. SSI Encoder

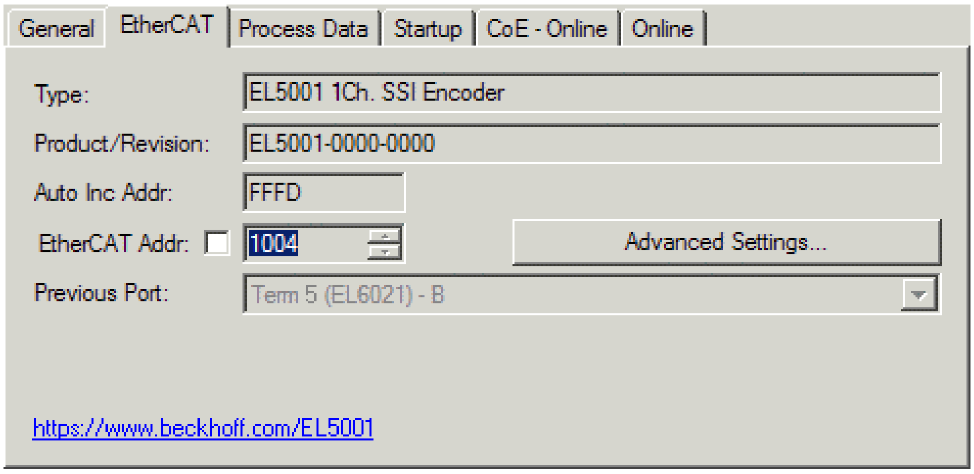
Comment:

☐ Disabled

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



Type: EL5001 1Ch. SSI Encoder
Product/Revision: EL5001-0000-0000
Auto Inc Addr: FFFD
EtherCAT Addr: ☐ 1004
Previous Port: Term 5 (EL6021) - B
https://www.beckhoff.com/EL5001

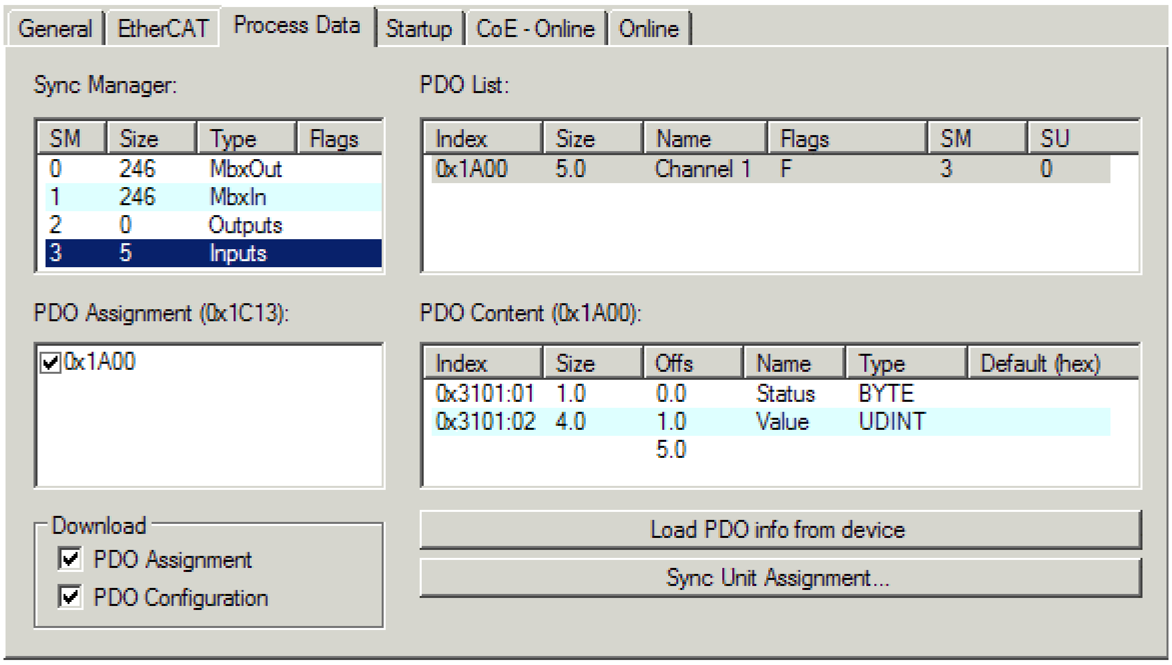
Fig. 79: “EtherCAT” tab

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.



Sync Manager:

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

PDO List:

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

PDO Assignment (0x1C13):

☑ 0x1A00

PDO Content (0x1A00):

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

Download
- ☑ PDO Assignment
- ☑ PDO Configuration

Fig. 80: “Process Data” tab

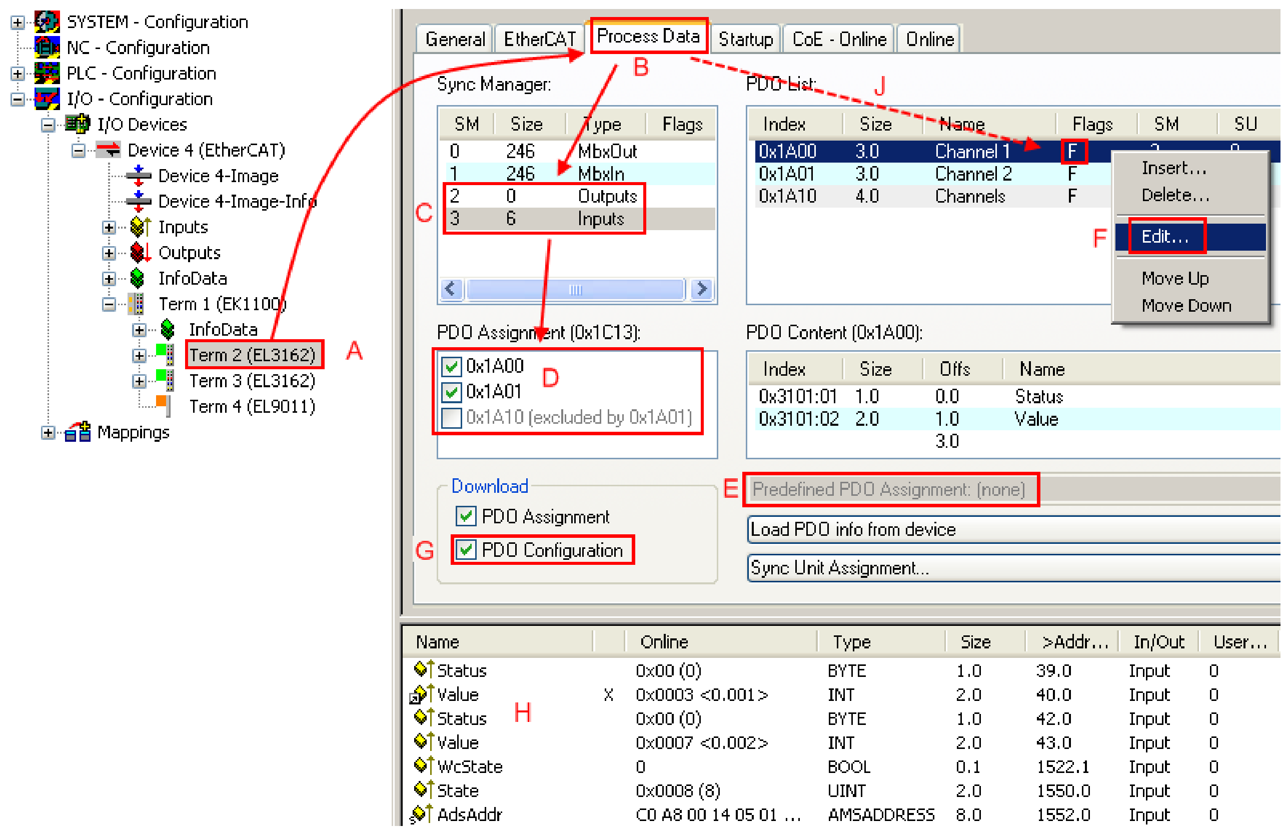
The process data (PDOs) transferred by an EtherCAT slave during each cycle are user data which the application expects to be updated cyclically or which are sent to the slave. To this end the EtherCAT master (Beckhoff TwinCAT) parameterizes each EtherCAT slave during the start-up phase to define which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



Sync Manager:

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

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	3.0	Channel 1	F	3	0
0x1A01	3.0	Channel 2	F		
0x1A10	4.0	Channels	F		

PDO Assignment (0x1C13):

- ✓ 0x1A00
- ✓ 0x1A01
- ☐ 0x1A10 (excluded by 0x1A01)

PDO Content (0x1A00):

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

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

Fig. 81: Configuring the process data

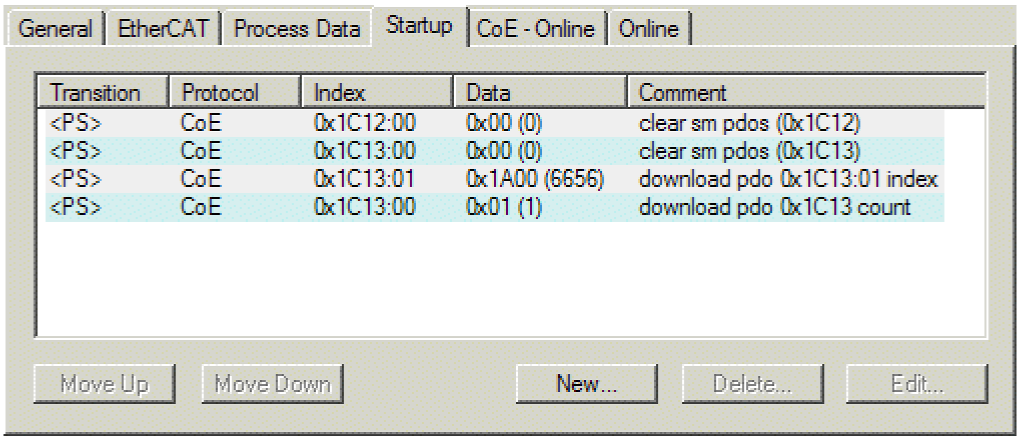
Manual modification of the process data

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

“Startup” tab

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



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

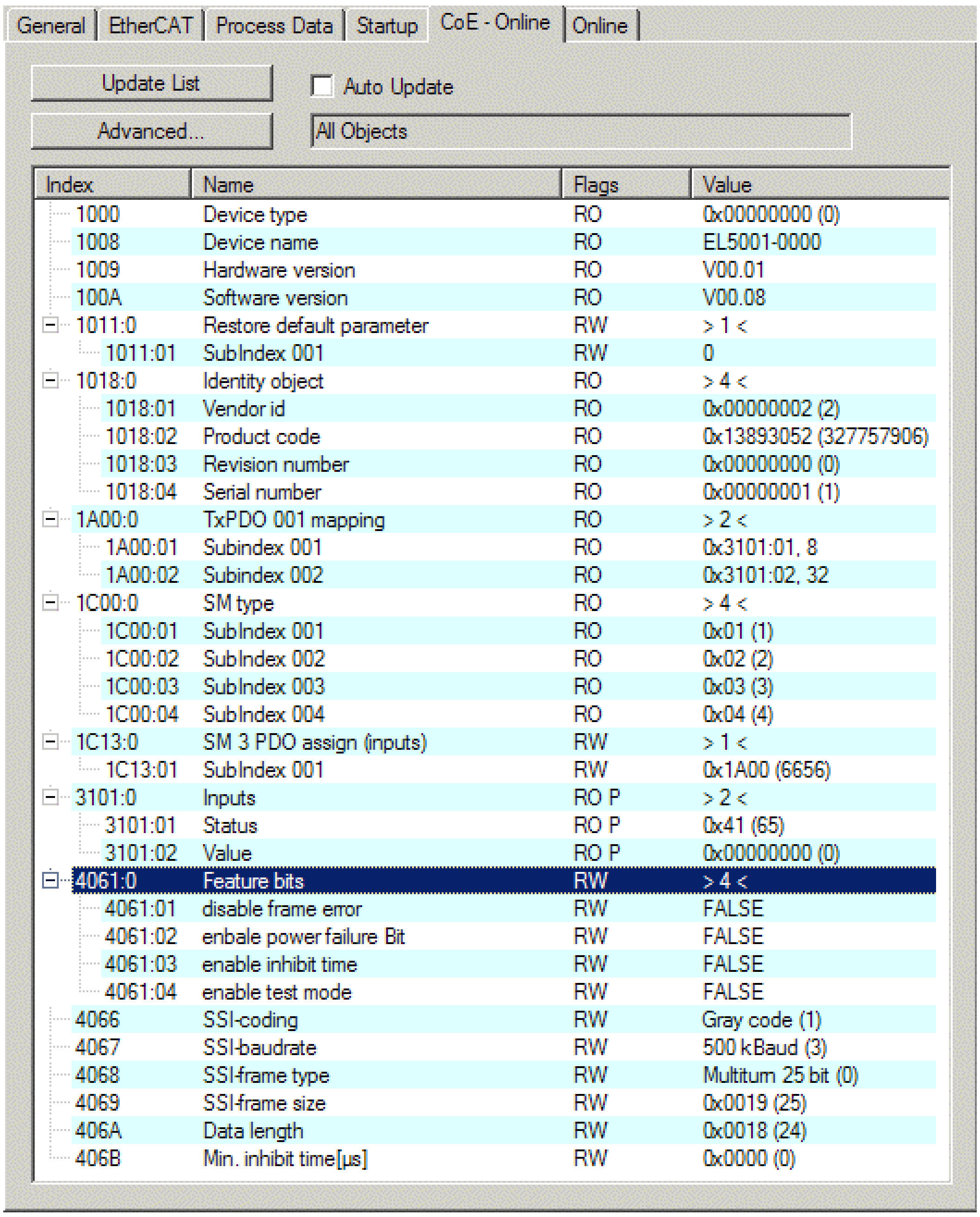
Fig. 82: “Startup” tab

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.



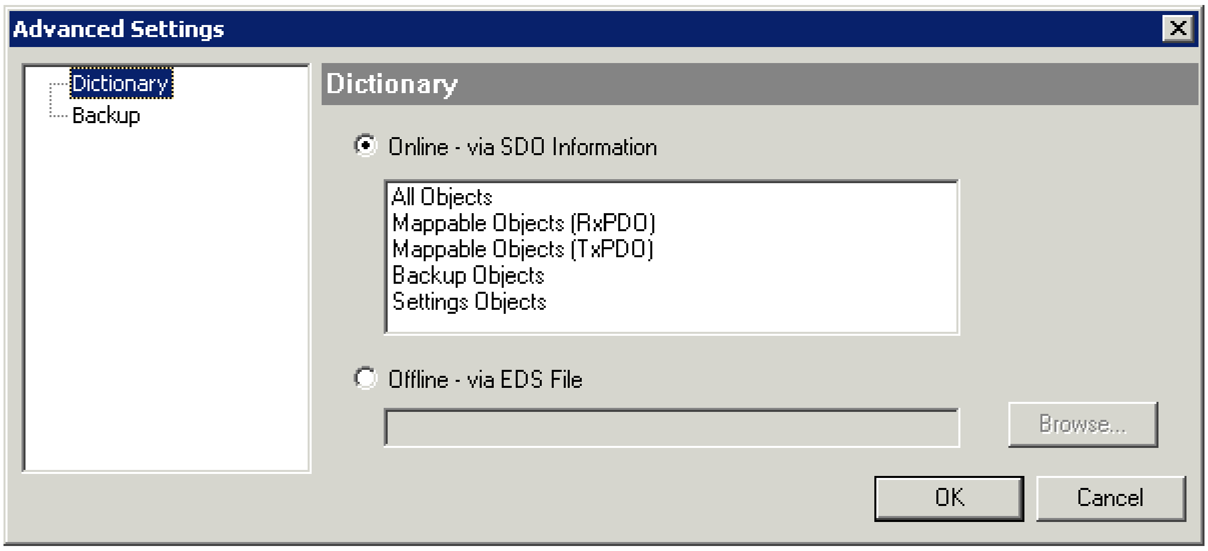
Index	Name	Flags	Value
1000	Device type	RO	0x00000000 (0)
1008	Device name	RO	EL5001-0000
1009	Hardware version	RO	V00.01
100A	Software version	RO	V00.08
1011:0	Restore default parameter	RW	> 1 <
1011:01	SubIndex 001	RW	0
1018:0	Identity object	RO	> 4 <
1018:01	Vendor id	RO	0x00000002 (2)
1018:02	Product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

Fig. 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.


Advanced Settings

- Dictionary
- Backup

Dictionary

[✓ Online - via SDO Information]

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

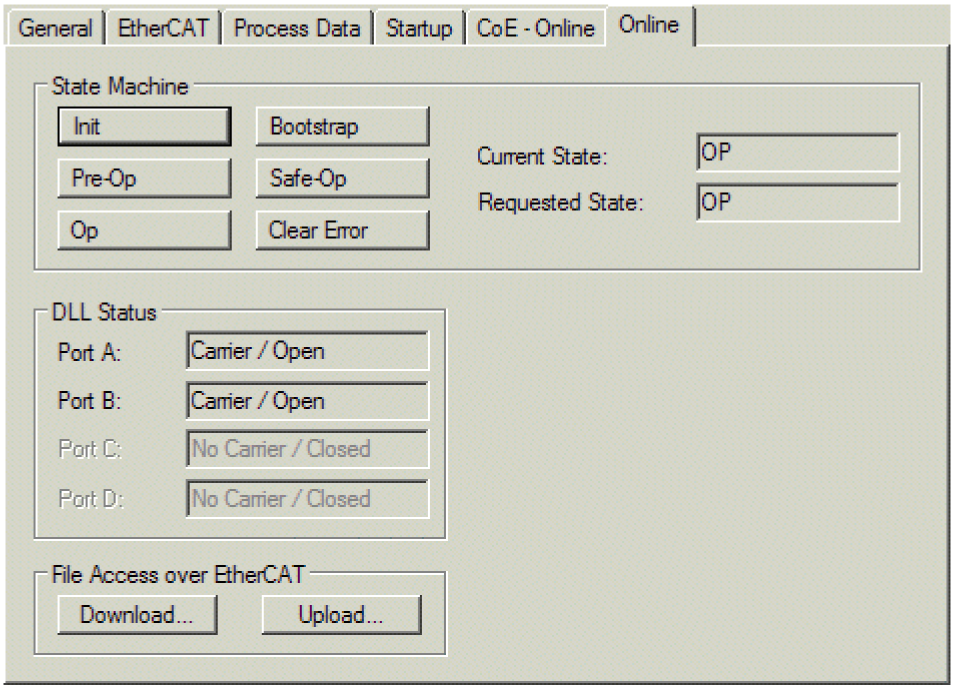
Offline - via EDS File

Browse... OK Cancel

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


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

State Machine

Init | Bootstrap | Pre-Op | Safe-Op | Op | Clear Error

Current State: OP
Requested State: OP

DLL Status

Port A: Carrier / Open
Port B: Carrier / Open
Port C: No Carrier / Closed
Port D: No Carrier / Closed

File Access over EtherCAT

Download... Upload...

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

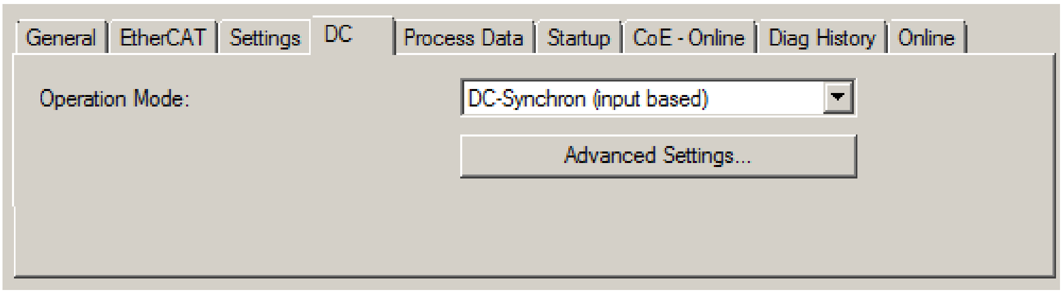
Indicates the DLL status (data link layer status) of the individual ports of the EtherCAT slave. The DLL status can have four different states:

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)



Operation Mode: DC-Synchron (input based)

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

5.1.7.1 Detailed description of Process Data tab

Sync Manager

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

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

PDO Assignment

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

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

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

Activation of PDO assignment

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

a) the EtherCAT slave has to run through the PS status transition cycle (from pre-operational to safe-operational) once (see Online tab [▶ 88]),

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 [▸ 85] tab.

PDO Configuration

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

5.1.8 Import/Export of EtherCAT devices with SCI and XTI

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

5.1.8.1 Basic principles

An EtherCAT slave is basically parameterized through the following elements:

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

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

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

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

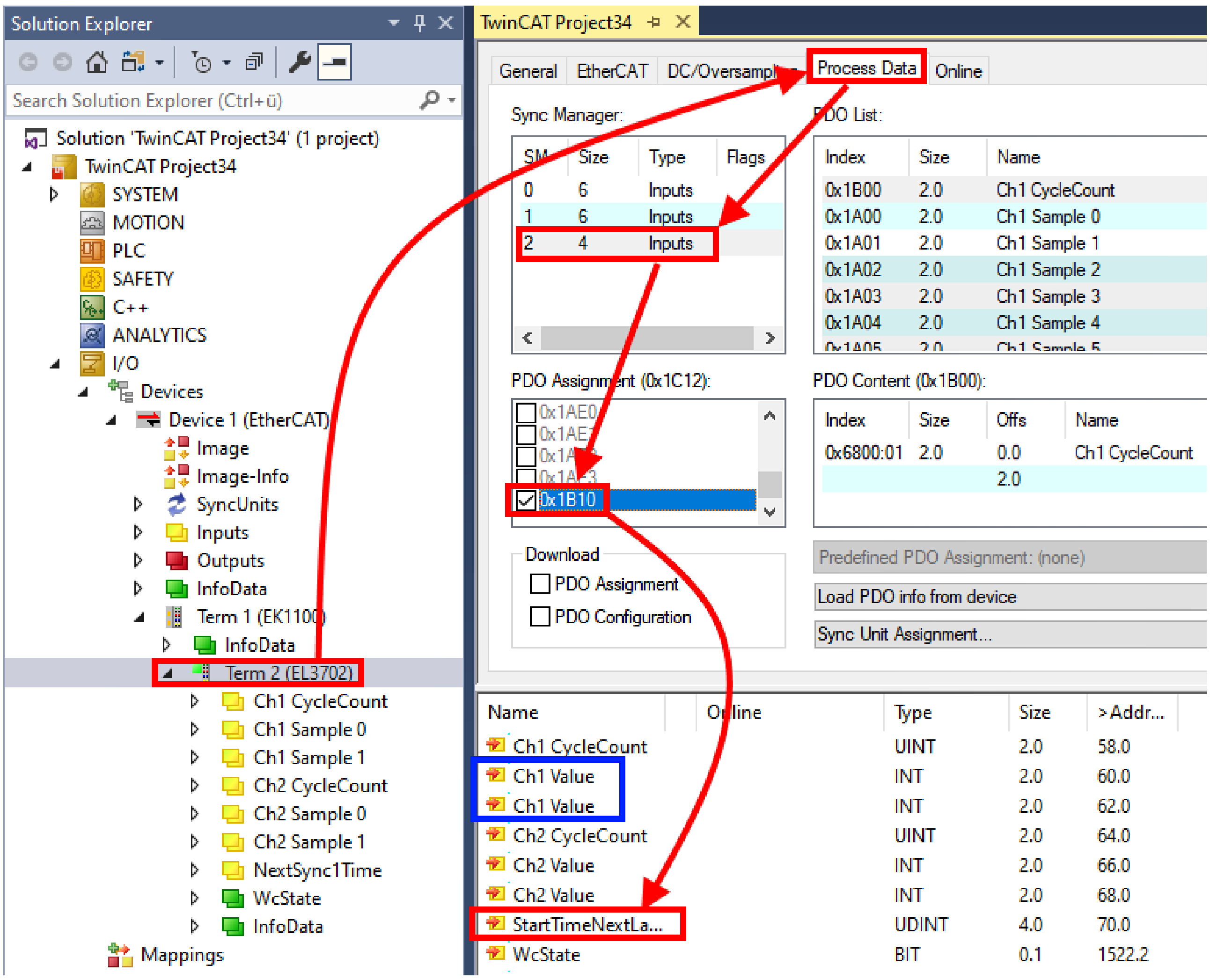
It may become necessary to reuse the modified slave in other projects in this way, without having to make equivalent configuration changes to the slave again. To accomplish this, proceed as follows:

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

TwinCAT offers two methods for this purpose:

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

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


Solution Explorer

Search Solution Explorer (Ctrl+ü)

- Solution 'TwinCAT Project34' (1 project)
  - TwinCAT Project34
    - SYSTEM
    - MOTION
    - PLC
    - SAFETY
    - C++
    - ANALYTICS
    - I/O
      - Devices
        - Device 1 (EtherCAT)
          - Image
          - Image-Info
          - SyncUnits
          - Inputs
          - Outputs
          - InfoData
          - Term 1 (EK1100)
            - InfoData
            - Term 2 (EL3702)
              - Ch1 CycleCount
              - Ch1 Sample 0
              - Ch1 Sample 1
              - Ch2 CycleCount
              - Ch2 Sample 0
              - Ch2 Sample 1
              - NextSync1Time
              - WcState
              - InfoData
      - Mappings

TwinCAT Project34

General | EtherCAT | DC/Oversampl... | Process Data | Online

Sync Manager:

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

PDO List:

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

PDO Assignment (0x1C12):

- [ ] 0x1AE0
- [ ] 0x1AE
- [ ] 0x1A
- [ ] 0x1AE3
- [✓] 0x1B10

PDO Content (0x1B00):

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

Download
- [ ] PDO Assignment
- [ ] PDO Configuration

Predefined PDO Assignment: (none)

Load PDO info from device

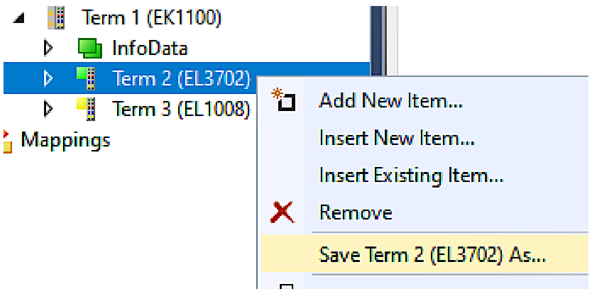
Sync Unit Assignment...

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

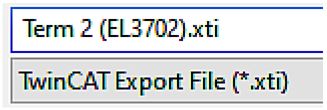
The two methods for exporting and importing the modified terminal referred to above are demonstrated below.

5.1.8.2 Procedure within TwinCAT with xti files

Each IO device can be exported/saved individually:

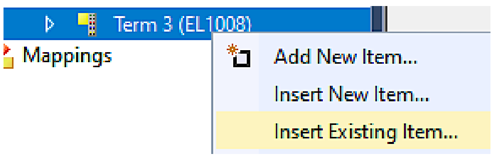

- Term 1 (EK1100)
  - InfoData
  - Term 2 (EL3702)
  - Term 3 (EL1008)
- Mappings

The xti file can be stored:


Term 2 (EL3702).xti

TwinCAT Export File (*.xti)

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



5.1.8.3 Procedure within and outside TwinCAT with sci file

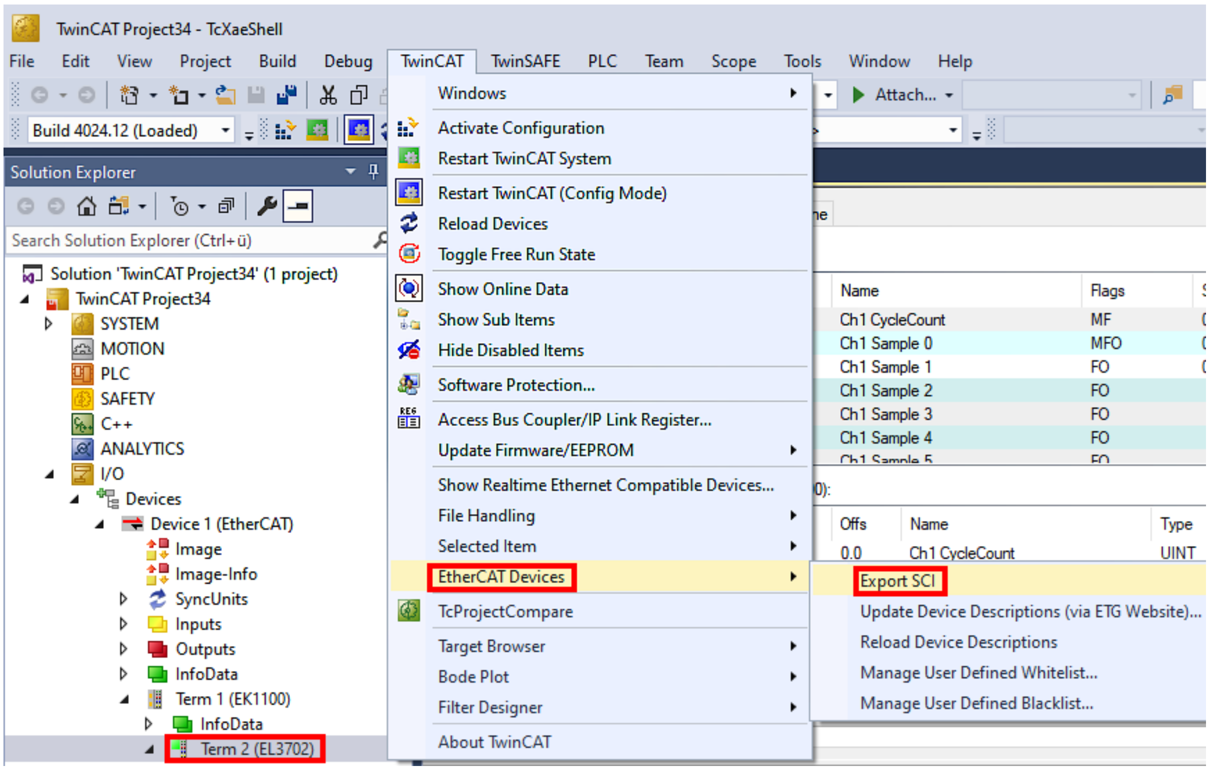
Note regarding availability (2021/01)

The SCI method is available from TwinCAT 3.1 build 4024.14.

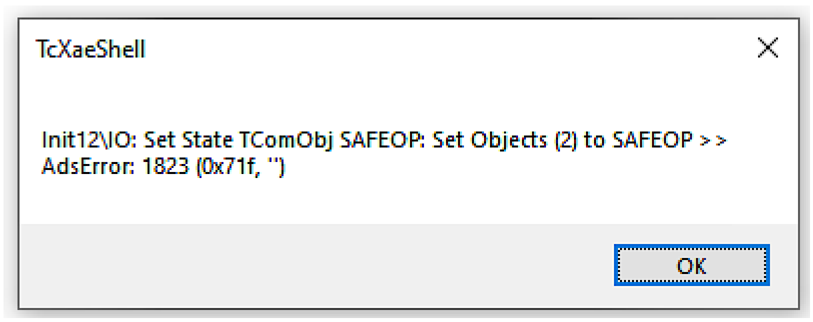
The Slave Configuration Information (SCI) describes a specific complete configuration for an EtherCAT slave (terminal, box, drive...) based on the setting options of the device description file (ESI, EtherCAT Slave Information). That is, it includes PDO, CoE, synchronization.

Export:

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



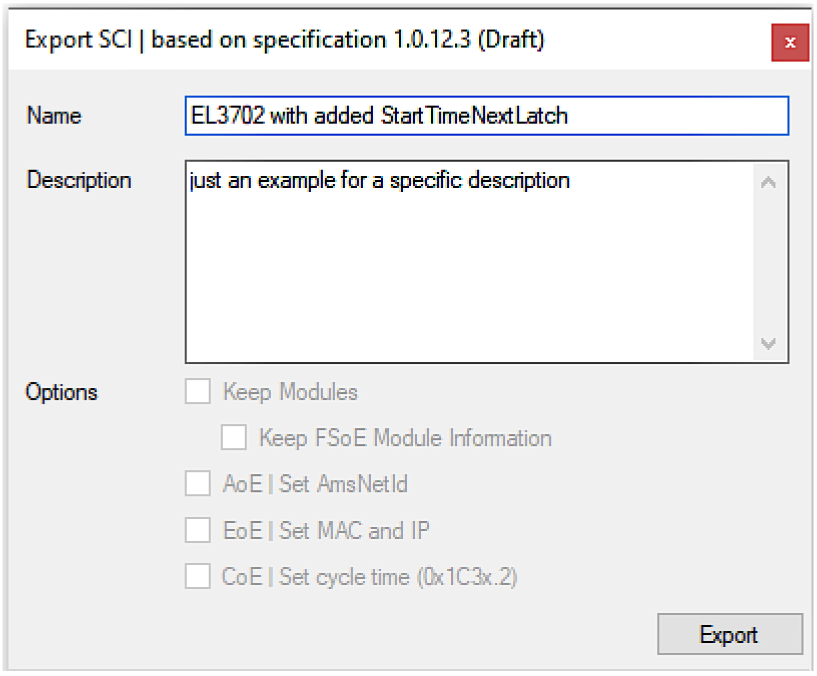
- If TwinCAT is offline (i.e. if there is no connection to an actual running controller) a warning message may appear, because after executing the function the system attempts to reload the EtherCAT segment. However, in this case this is not relevant for the result and can be acknowledged by clicking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


TcXaeShell

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

OK

- A description may also be provided:


Export SCI | based on specification 1.0.12.3 (Draft)
Options
Keep Modules
Keep FSoE Module Information
AoE | Set AmsNetId
EoE | Set MAC and IP
CoE | Set cycle time (0x1C3x.2)
Export

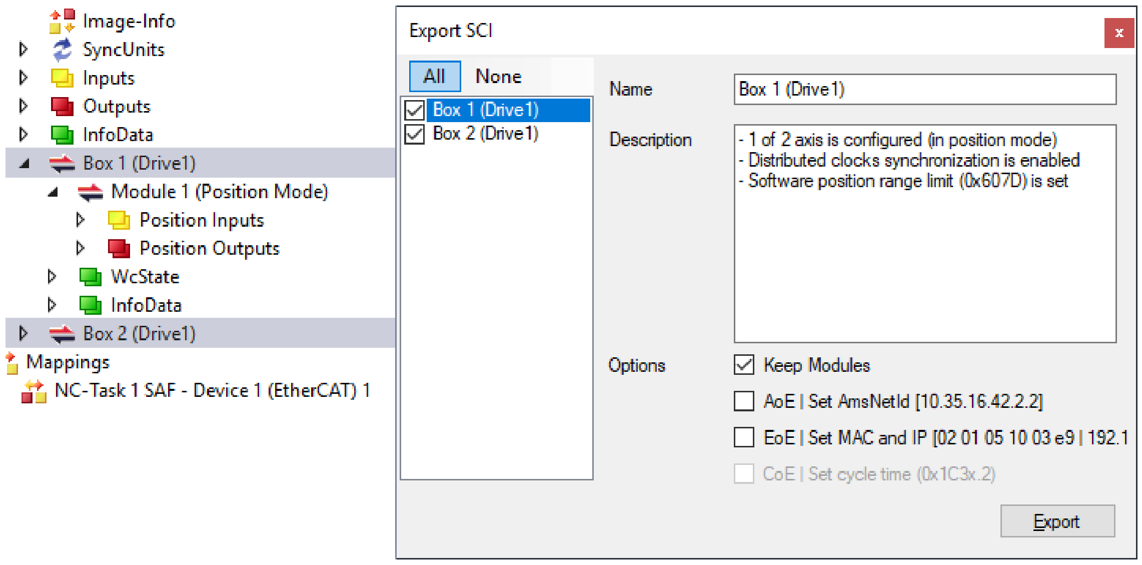
Name: EL3702 with added StartTimeNextLatch

Description: just an example for a specific description

- 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*):



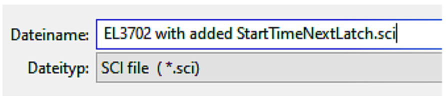
Name: Box 1 (Drive1)

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

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

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

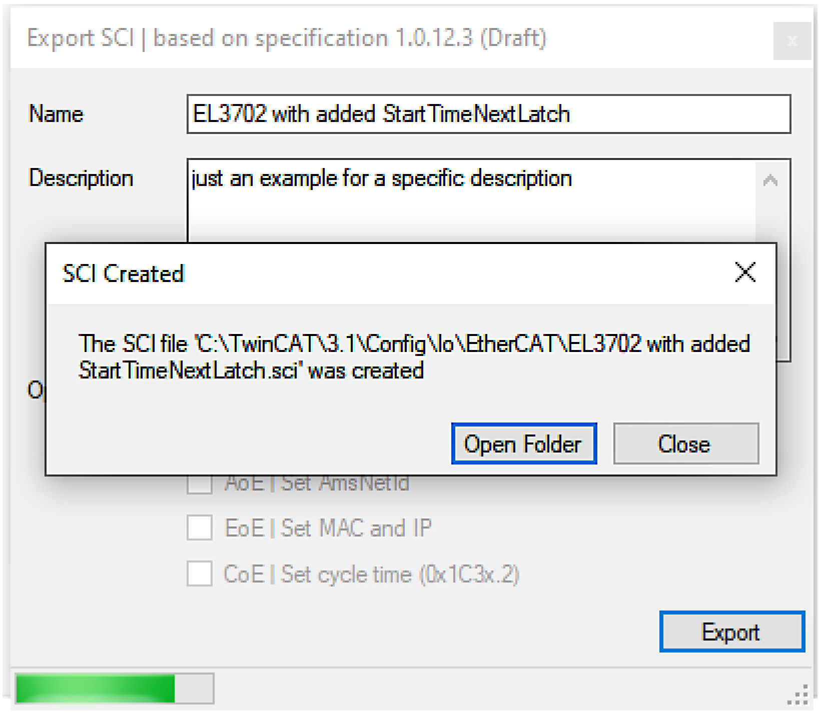
   - None:
     All slaves are deselected.
- The sci file can be saved locally:



Dateiname: EL3702 with added StartTimeNextLatch.sci

Dateityp: SCI file (*.sci)

- The export takes place:



Export SCI | based on specification 1.0.12.3 (Draft)

Name: EL3702 with added StartTimeNextLatch

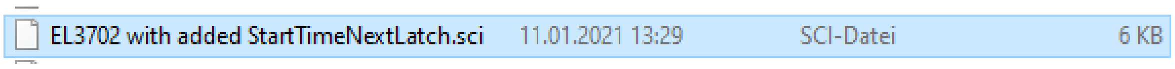
Description: just an example for a specific description

SCI Created

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

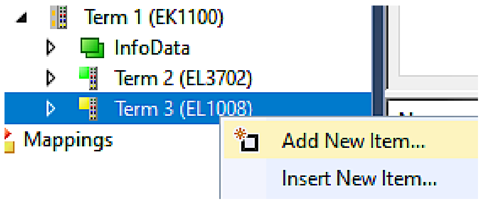
Import

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

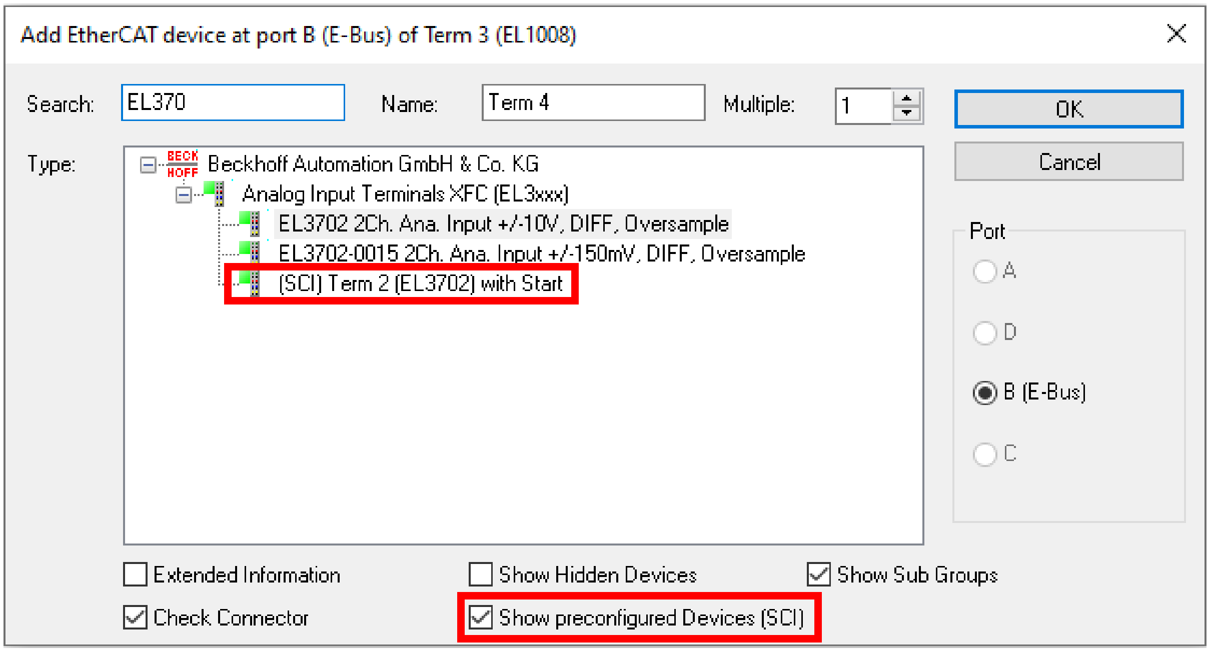


EL3702 with added StartTimeNextLatch.sci 11.01.2021 13:29 SCI-Datei 6 KB

- Open the selection dialog:



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


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

Search: EL370 Name: Term 4 Multiple: 1

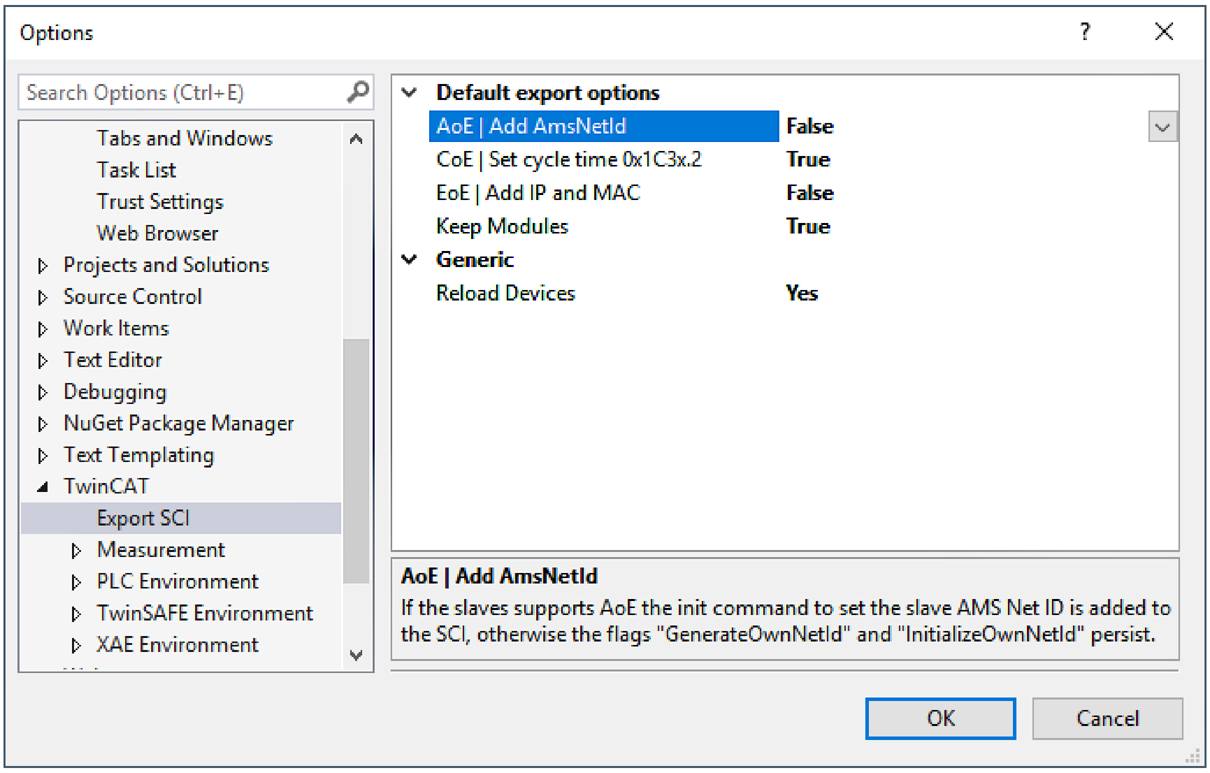
Type:
- Beckhoff Automation GmbH & Co. KG
  - Analog Input Terminals XFC (EL3xxx)
    - EL3702 2Ch. Ana. Input +/-10V, DIFF, Oversample
    - EL3702-0015 2Ch. Ana. Input +/-150mV, DIFF, Oversample
    - (SCI) Term 2 (EL3702) with Start

Port: A / D / [✓ B (E-Bus)] / C

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

Additional Notes

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


Options

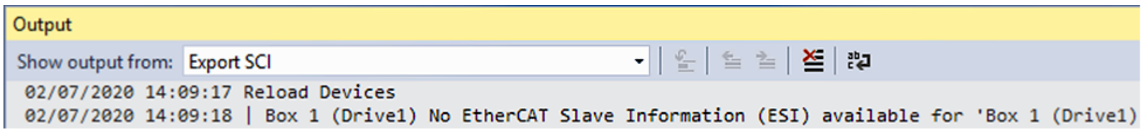
Default export options	
AoE | Add AmsNetId	False
CoE | Set cycle time 0x1C3x.2	True
EoE | Add IP and MAC	False
Keep Modules	True
Generic	
Reload Devices	Yes

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

Explanation of the settings:

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

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



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

5.2 General Commissioning Instructions for an EtherCAT Slave

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

Diagnosis in real time: WorkingCounter, EtherCAT State and Status

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

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

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

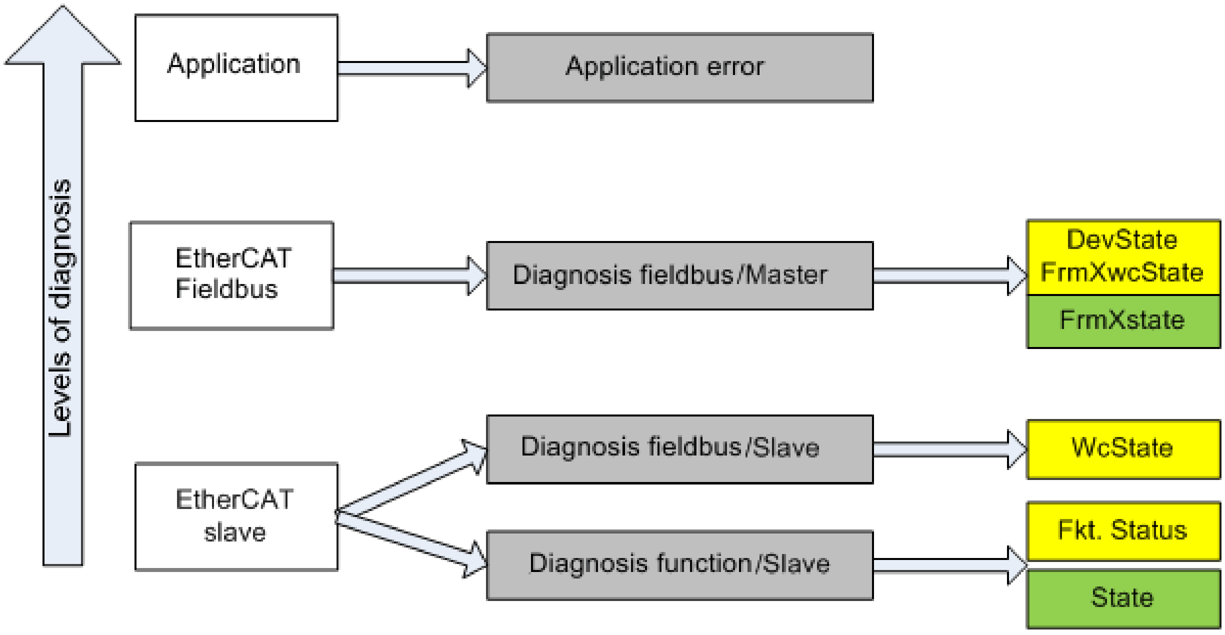


Fig. 87: Selection of the diagnostic information of an EtherCAT Slave

In general, an EtherCAT Slave offers

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

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

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

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

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

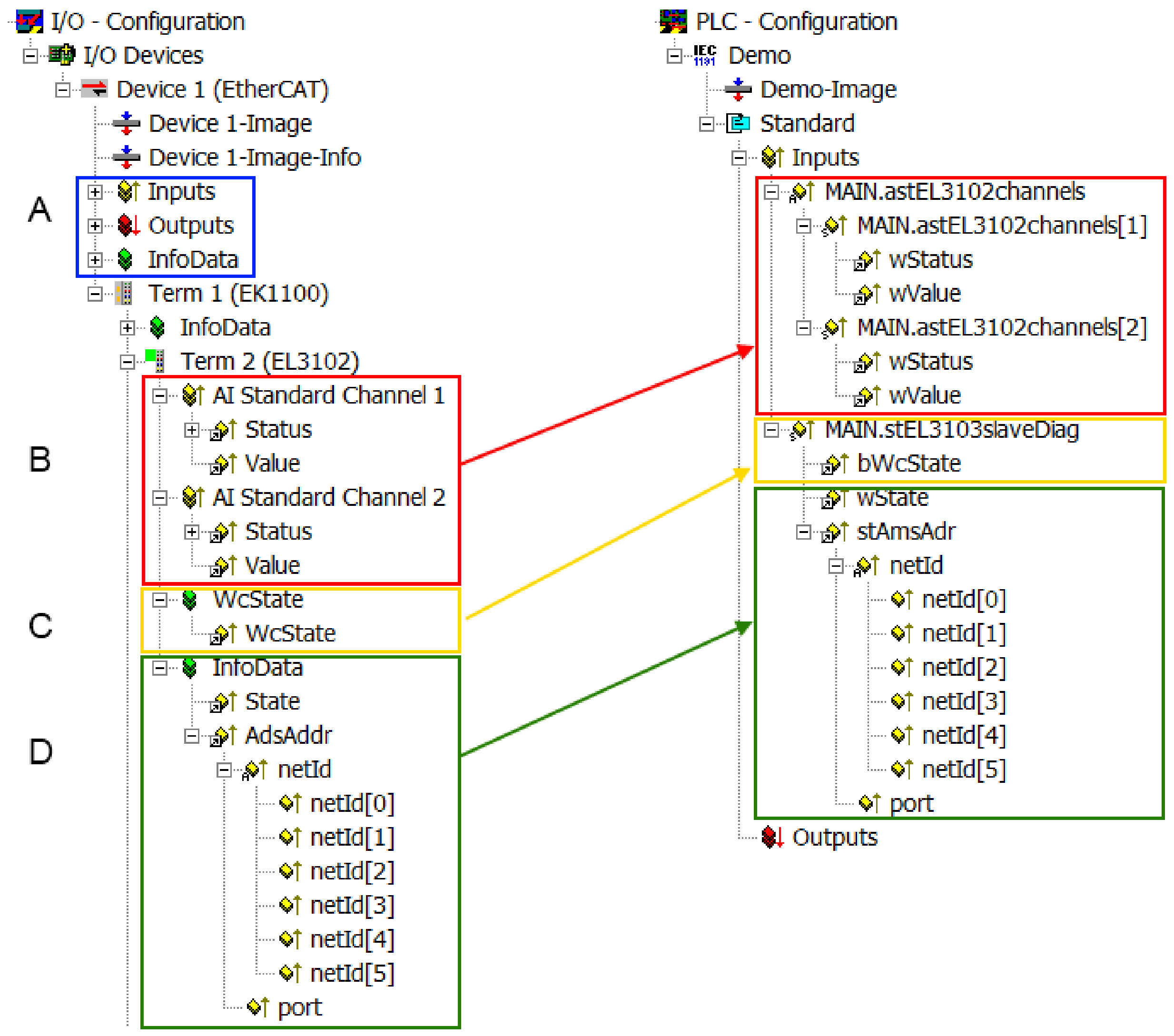


Fig. 88: Basic EtherCAT Slave Diagnosis in the PLC

The following aspects are covered here:

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

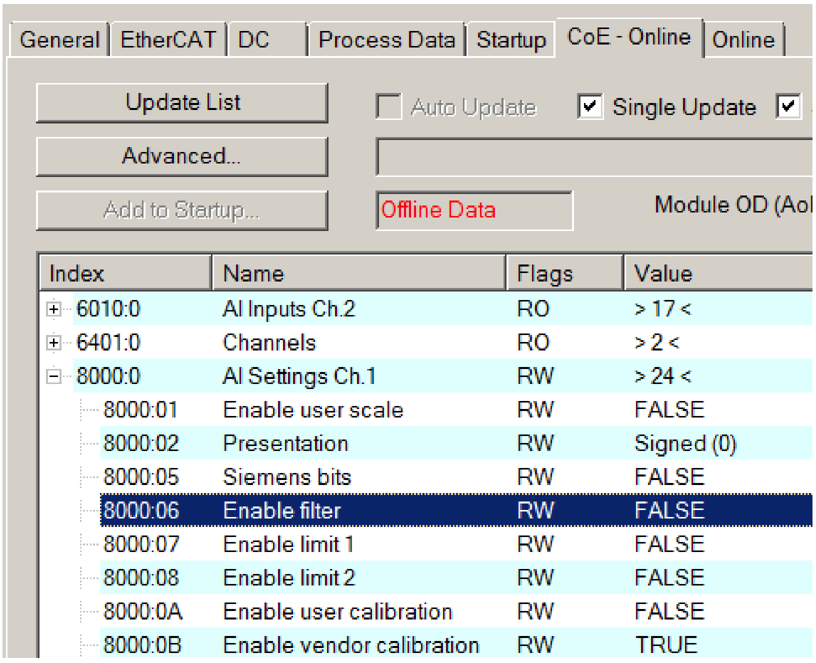
NOTICE

Diagnostic information

It is strongly recommended that the diagnostic information made available is evaluated so that the application can react accordingly.

CoE Parameter Directory

The CoE parameter directory (CanOpen-over-EtherCAT) is used to manage the set values for the slave concerned. Changes may, in some circumstances, have to be made here when commissioning a relatively complex EtherCAT Slave. It can be accessed through the TwinCAT System Manager, see Fig. *EL3102, CoE directory*:



Index	Name	Flags	Value
6010:0	AI Inputs Ch.2	RO	> 17 <
6401:0	Channels	RO	> 2 <
8000:0	AI Settings Ch.1	RW	> 24 <
8000:01	Enable user scale	RW	FALSE
8000:02	Presentation	RW	Signed (0)
8000:05	Siemens bits	RW	FALSE
8000:06	Enable filter	RW	FALSE
8000:07	Enable limit 1	RW	FALSE
8000:08	Enable limit 2	RW	FALSE
8000:0A	Enable user calibration	RW	FALSE
8000:0B	Enable vendor calibration	RW	TRUE

Fig. 89: EL3102, CoE directory

EtherCAT System Documentation

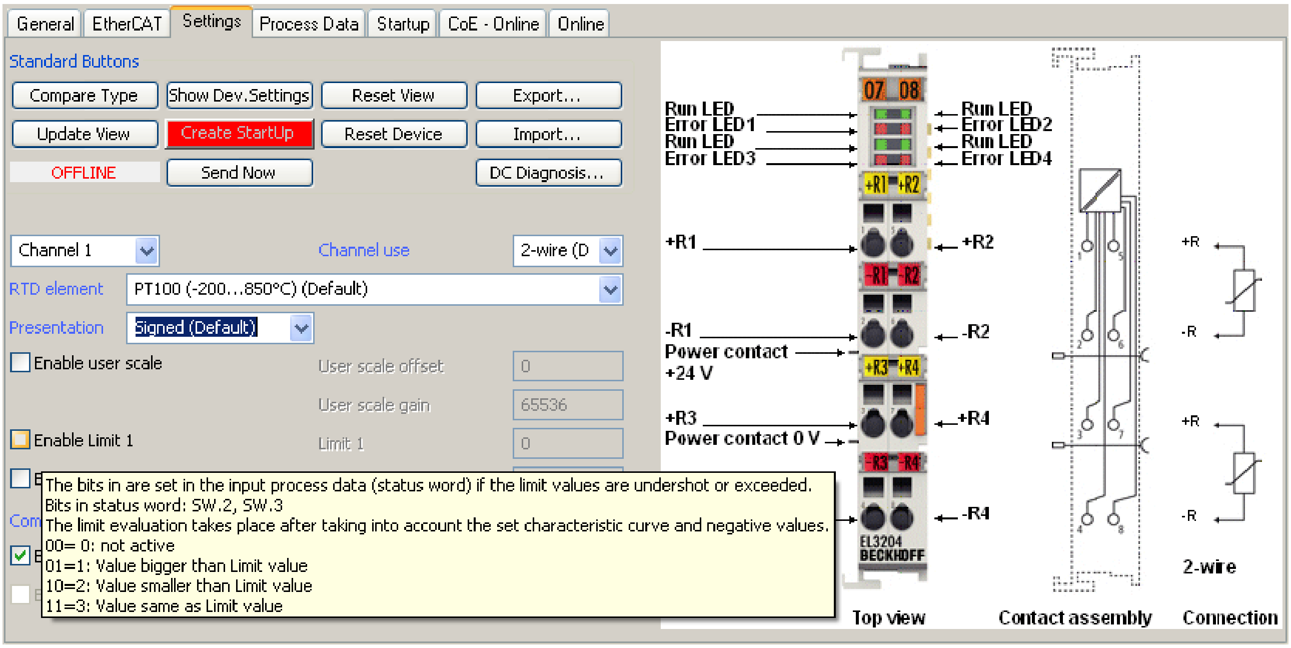
The comprehensive description in the EtherCAT System Documentation (EtherCAT Basics --> CoE Interface) must be observed!

A few brief extracts:

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

Commissioning aid in the TwinCAT System Manager

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



Channel 1 | Channel use: 2-wire (D

RTD element: PT100 (-200...850°C) (Default)

Presentation: Signed (Default)

Enable user scale | User scale offset: 0 | User scale gain: 65536

Enable Limit 1 | Limit 1: 0

The bits in are set in the input process data (status word) if the limit values are undershot or exceeded.
Bits in status word: SW.2, SW.3
The limit evaluation takes place after taking into account the set characteristic curve and negative values.
00= 0: not active
01=1: Value bigger than Limit value
10=2: Value smaller than Limit value
11=3: Value same as Limit value

Fig. 90: Example of commissioning aid for a EL3204

This commissioning process simultaneously manages

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

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

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

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

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

- INIT
- PREOP
- SAFEOP
- OP

to ensure sound operation. The EtherCAT Master directs these statuses in accordance with the initialization routines that are defined for commissioning the device by the ES/XML and user settings (Distributed Clocks (DC), PDO, CoE). See also the section on "Principles of Communication, EtherCAT State Machine [▶ 35]" in this connection. Depending how much configuration has to be done, and on the overall communication, booting can take up to a few seconds.

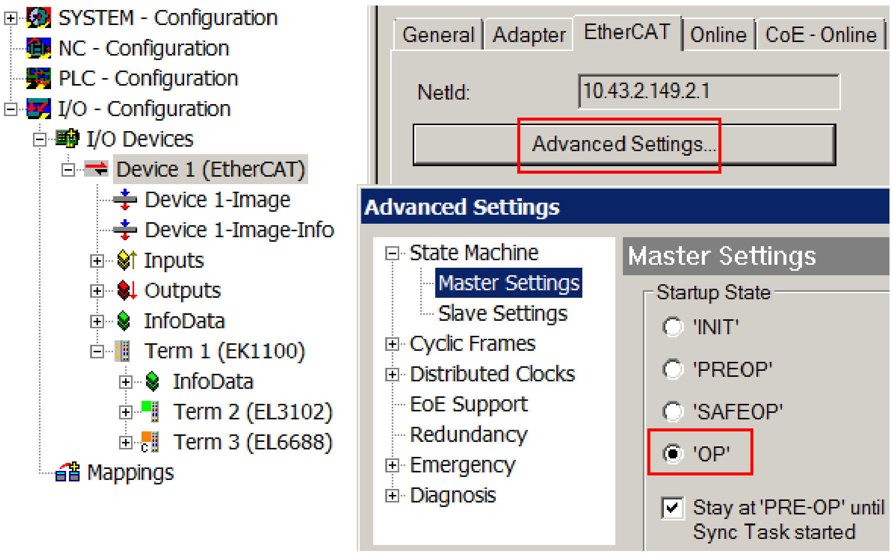
The EtherCAT Master itself must go through these routines when starting, until it has reached at least the OP target state.

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

Standard setting

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

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



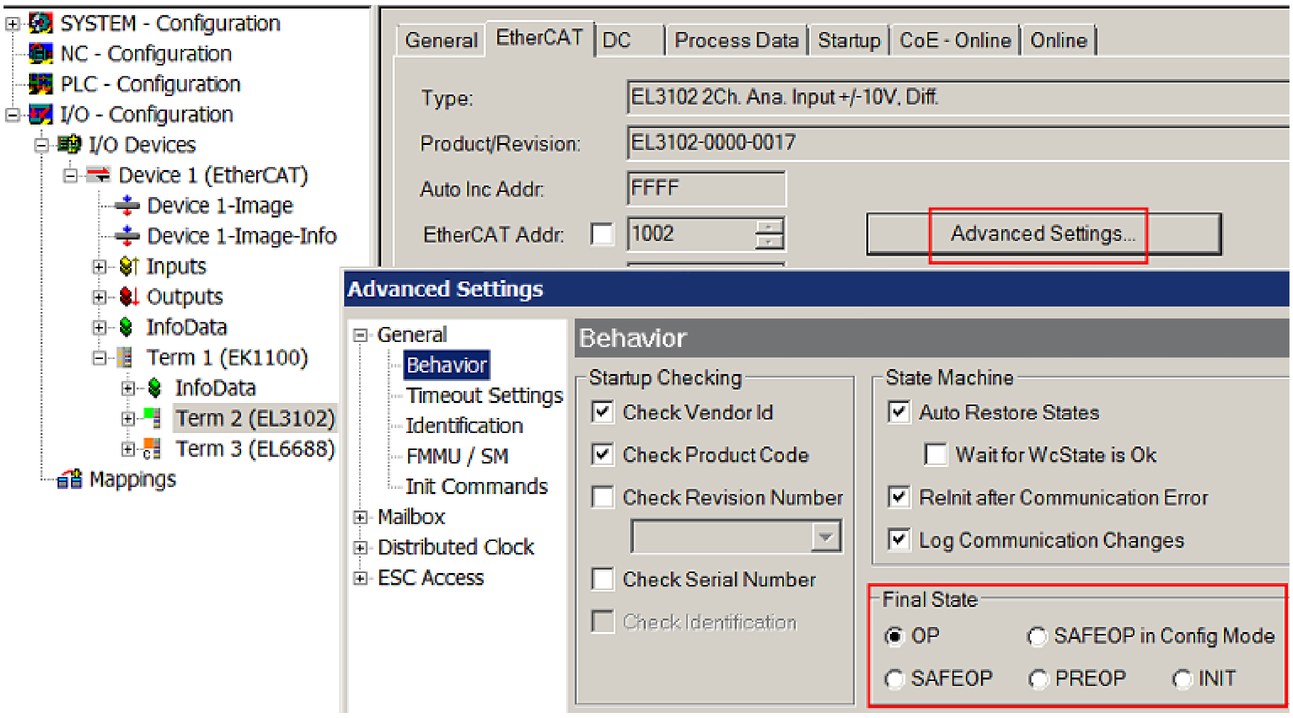
NetId: 10.43.2.149.2.1

Startup State
'INIT' 'PREOP' 'SAFEOP' [✓ 'OP']

[✓] Stay at 'PRE-OP' until Sync Task started

Fig. 91: Default behaviour of the System Manager

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



Type: EL3102 2Ch. Ana. Input +/-10V, Diff.
Product/Revision: EL3102-0000-0017
Auto Inc Addr: FFFF
EtherCAT Addr: ☐ 1002

Behavior

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

State Machine
- ☑ Auto Restore States
- ☐ Wait for WcState is Ok
- ☑ ReInit after Communication Error
- ☑ Log Communication Changes

Final State
[✓ OP] SAFEOP in Config Mode SAFEOP PREOP INIT

Fig. 92: Default target state in the Slave

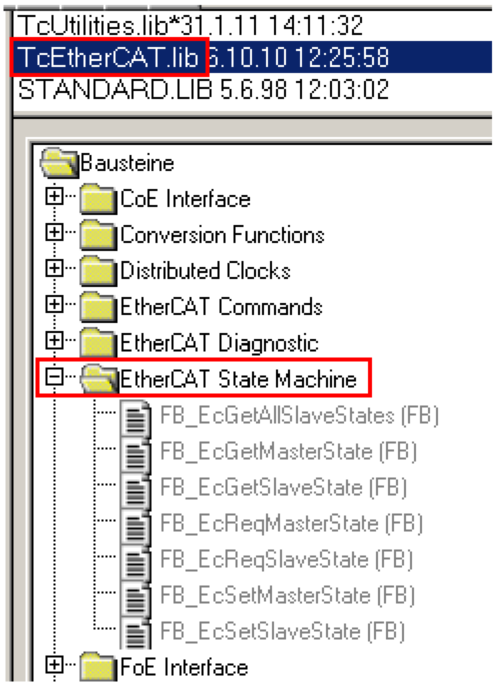
Manual Control

There are particular reasons why it may be appropriate to control the states from the application/task/PLC. For instance:

- for diagnostic reasons
- to induce a controlled restart of axes
- because a change in the times involved in starting is desirable

In that case it is appropriate in the PLC application to use the PLC function blocks from the *TcEtherCAT.lib*, which is available as standard, and to work through the states in a controlled manner using, for instance, *FB_EcSetMasterState*.

It is then useful to put the settings in the EtherCAT Master to INIT for master and slave.



TcUtilities.lib*31.1.11 14:11:32
TcEtherCAT.lib 6.10.10 12:25:58
STANDARD.LIB 5.6.98 12:03:02

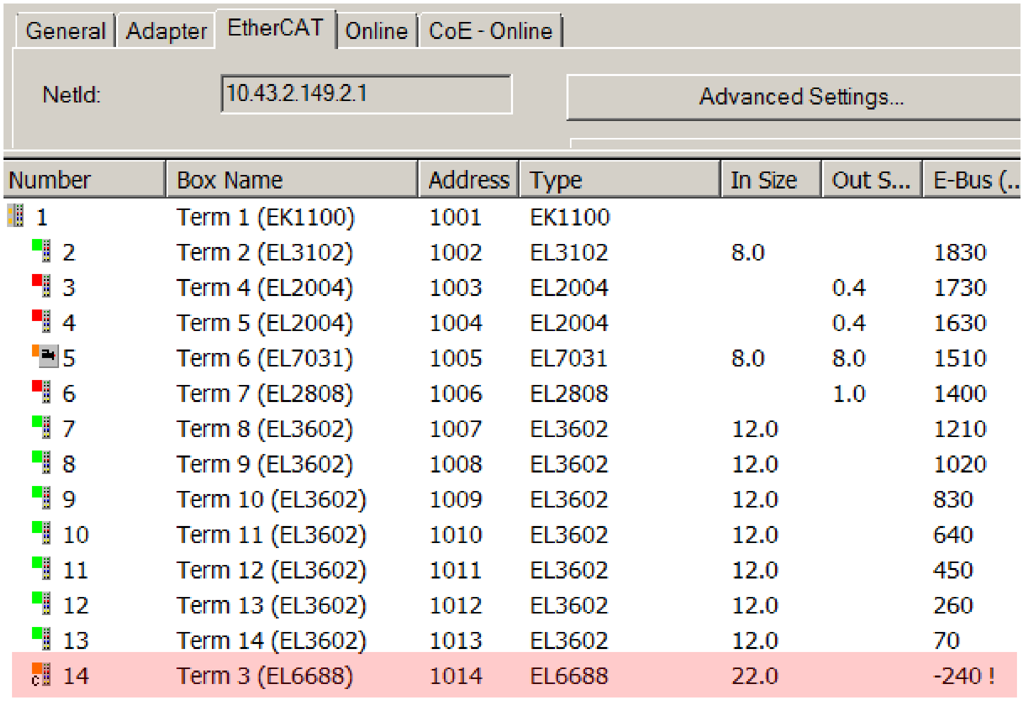
- Bausteine
  - CoE Interface
  - Conversion Functions
  - Distributed Clocks
  - EtherCAT Commands
  - EtherCAT Diagnostic
  - EtherCAT State Machine
    - FB_EcGetAllSlaveStates (FB)
    - FB_EcGetMasterState (FB)
    - FB_EcGetSlaveState (FB)
    - FB_EcReqMasterState (FB)
    - FB_EcReqSlaveState (FB)
    - FB_EcSetMasterState (FB)
    - FB_EcSetSlaveState (FB)
  - FoE Interface

Fig. 93: PLC function blocks

Note regarding E-Bus current

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

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

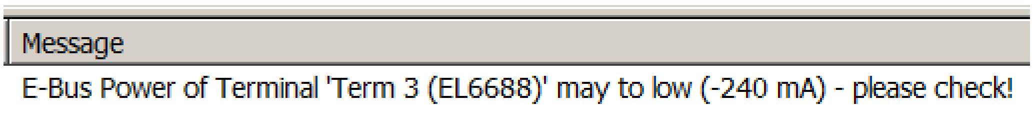


NetId: 10.43.2.149.2.1

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

Fig. 94: Illegally exceeding the E-Bus current

From TwinCAT 2.11 and above, a warning message “E-Bus Power of Terminal...” is output in the logger window when such a configuration is activated:



```
Message
E-Bus Power of Terminal 'Term 3 (EL6688)' may to low (-240 mA) - please check!
```

Fig. 95: Warning message for exceeding E-Bus current

NOTICE

Caution! Malfunction possible!

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

6 EL6821 - Commissioning

6.1 EL6821 - Integration in TwinCAT

6.1.1 EL6821 with CX5120

This sample explains how to write a simple PLC program for DALI in TwinCAT and how to link it with the hardware.

A single lamp is to be controlled and switched to the maximum output value or switched off with a push button.

Sample: https://infosys.beckhoff.com/content/1033/el6821/Resources/13899865611.zip

The TwinCAT project is available for download as *.zip file. This must first be unpacked locally so that the archive (*.tnzip file) is available for import into the TwinCAT project.

Hardware

Setting up the components

1x CX5120 Embedded PC

1x EL1008 digital 8-channel input terminal (for the switch-on/switch-off function)

1 x power supply terminal EL9562

1x DALI terminal EL6821

1 x end terminal EL9011

Set up the hardware and the DALI components as described in the documentation.

This sample assumes that an On button was connected to the first EL1008 input and an Off button to the second. There is a lamp at DALI address 0.

Software

Creation of the PLC program

Create a new **TwinCAT XAE Project** and a **Standard PLC Project**. Add the Tc3_DALI library in the PLC project under **References**. Generate a global variable list with the name *GVL_DALI* and create the following variables:

```
VAR_GLOBAL
  bSwitchOn             AT %I* : BOOL;
  bSwitchOff            AT %I* : BOOL;
  stEL6821InData        AT %I* : ST_EL6821InData;
  stEL6821OutData       AT %Q* : ST_EL6821OutData;
END_VAR
```

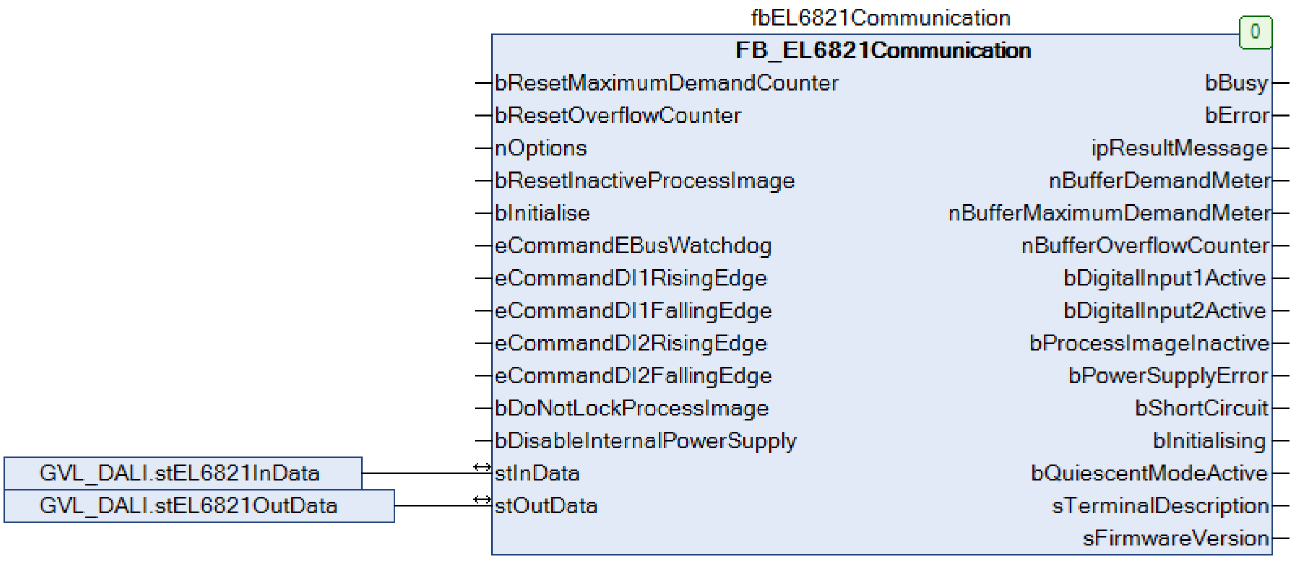
bSwitchOn: Input variable for the On button.

bSwitchOff: Input variable for the Off button.

stEL6821InData Input variable for the DALI terminal (see ST_EL6821InData).

stEL6821OutData: Output variable for the DALI terminal (see ST_EL6821OutData).

Create a program (CFC) for the background communication with the DALI terminal. The function block FB_EL6821Communication is called in the program. In the communication block ensure that the structures *stInData* and *stOutData* are linked.



Create a MAIN program (CFC) in which the function blocks FB_DALI102RecallMaxLevel and FB_DALI102Off are declared as follows.

```
PROGRAM MAIN
VAR
  fb102RecallMaxLevel   : FB_DALI102RecallMaxLevel(Communication.fbEL6821Communication);
  fb102Off              : FB_DALI102Off(Communication.fbEL6821Communication);
END_VAR
```

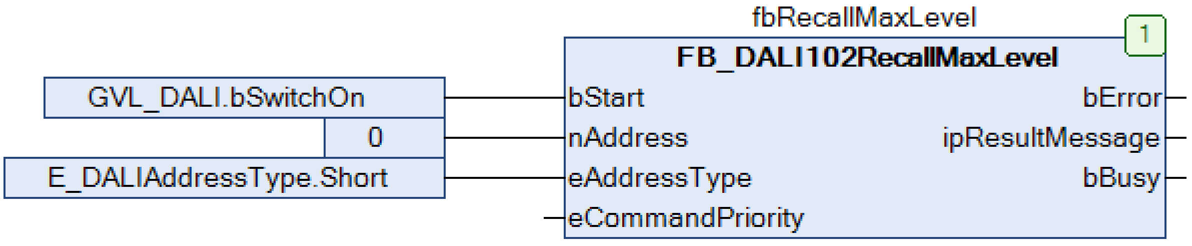
The communication block is specified in the round brackets after the declaration. The reference to the desired DALI terminal is defined via this specification.

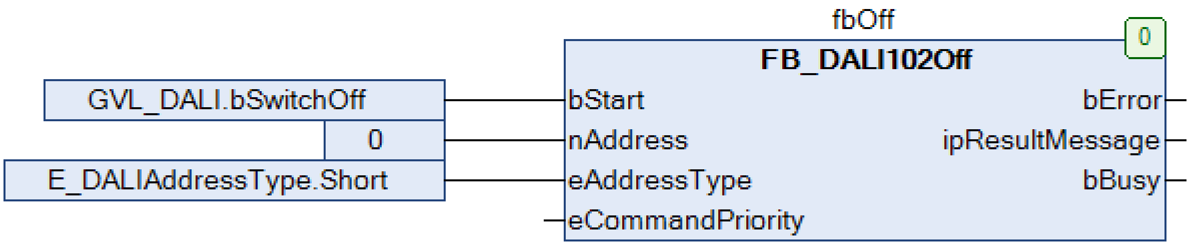
For more information, see Transfer of the reference to the communication block.

Call the two instances of the function blocks FB_DALI102RecallMaxLevel and FB_DALI102Off with the following variables.

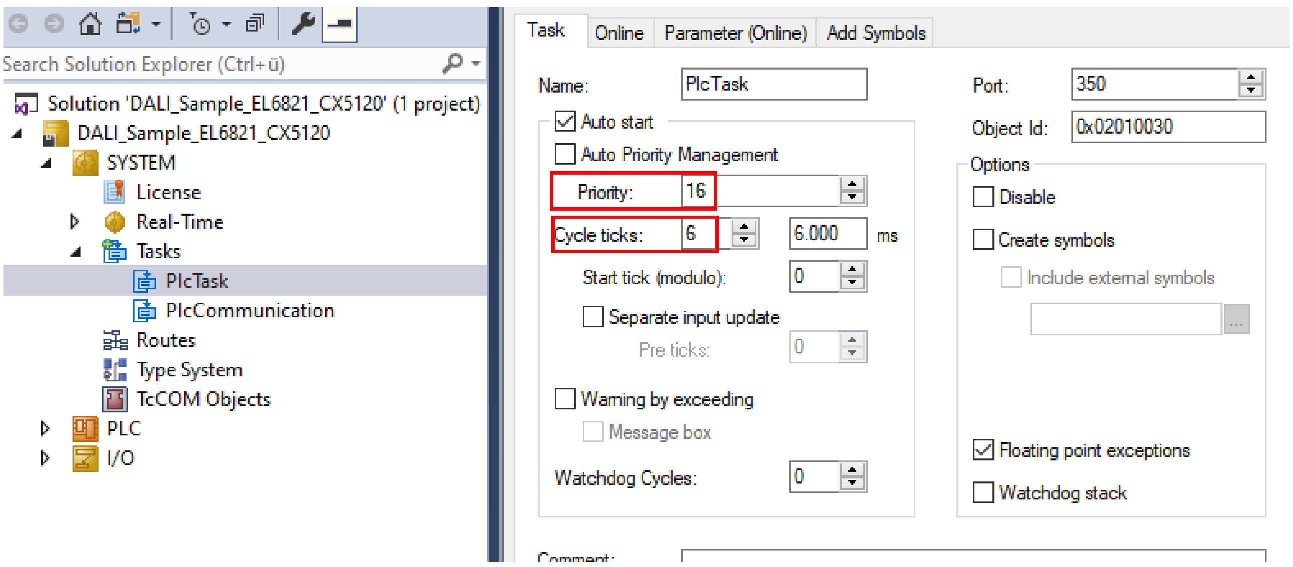
The input *bStart* of the function block for switching on a lamp with the maximum output value is linked to the global variable *bSwitchOn*.

The input *bStart* of the function block for switching off a lamp is linked to the global variable *bSwitchOff*.





Navigate to the task configuration section and configure the **PlcTask**. By way of example, the task is assigned priority 16 and a cycle time of 6 ms.



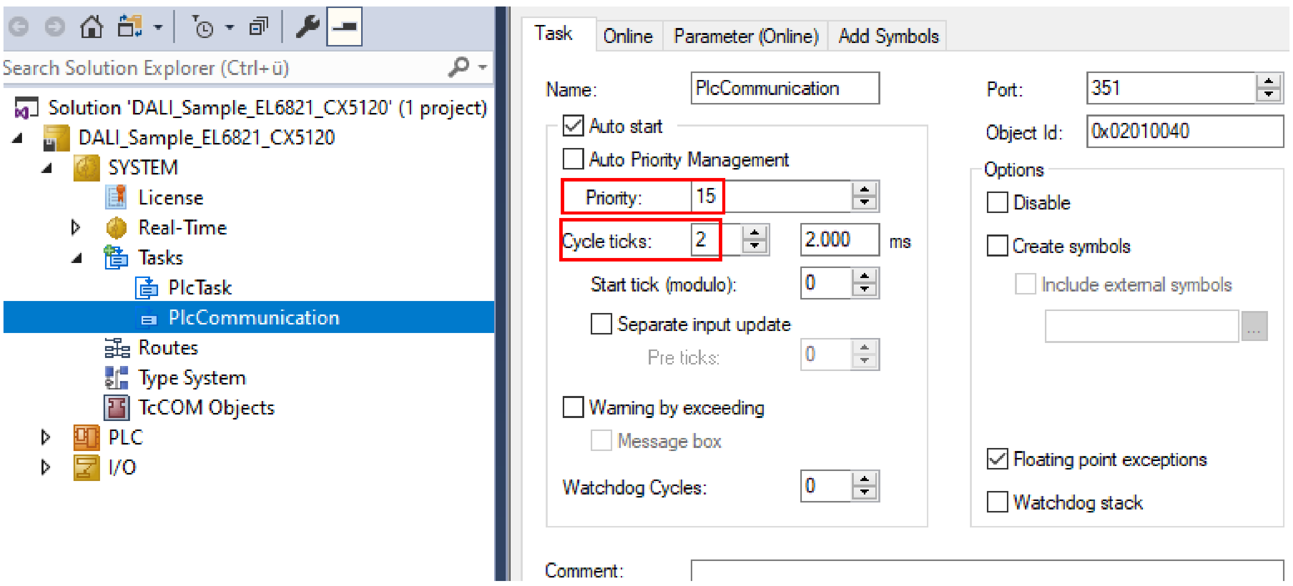
Search Solution Explorer (Ctrl+ü)

- Solution 'DALI_Sample_EL6821_CX5120' (1 project)
  - DALI_Sample_EL6821_CX5120
    - SYSTEM
      - License
      - Real-Time
      - Tasks
        - PlcTask
        - PlcCommunication
      - Routes
      - Type System
      - TcCOM Objects
    - PLC
    - I/O

Task | Online | Parameter (Online) | Add Symbols

Name: PlcTask
Port: 350
Object Id: 0x02010030
[✓] Auto start
[ ] Auto Priority Management
Priority: 16
Cycle ticks: 6 — 6.000 ms
Start tick (modulo): 0
[ ] Separate input update
Pre ticks: 0
[ ] Warning by exceeding
[ ] Message box
Watchdog Cycles: 0
Options
[ ] Disable
[ ] Create symbols
[ ] Include external symbols
[✓] Floating point exceptions
[ ] Watchdog stack
Comment:

Create another task for background communication. Give this task a higher priority (lower number) and a lower interval time than the Plc task.



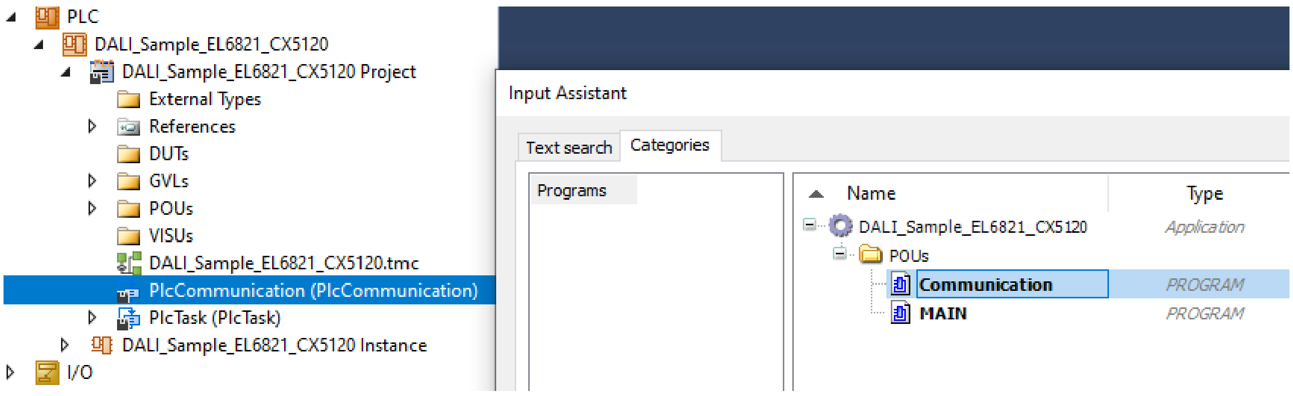
Search Solution Explorer (Ctrl+ü)

- Solution 'DALI_Sample_EL6821_CX5120' (1 project)
  - DALI_Sample_EL6821_CX5120
    - SYSTEM
      - License
      - Real-Time
      - Tasks
        - PlcTask
        - PlcCommunication
      - Routes
      - Type System
      - TcCOM Objects
    - PLC
    - I/O

Task | Online | Parameter (Online) | Add Symbols

Name: PlcCommunication
Port: 351
Object Id: 0x02010040
[✓] Auto start
[ ] Auto Priority Management
Priority: 15
Cycle ticks: 2 — 2.000 ms
Start tick (modulo): 0
[ ] Separate input update
Pre ticks: 0
[ ] Warning by exceeding
[ ] Message box
Watchdog Cycles: 0
Options
[ ] Disable
[ ] Create symbols
[ ] Include external symbols
[✓] Floating point exceptions
[ ] Watchdog stack
Comment:

Add the program for the communication to this task. Further information on task configuration can be found in the description of the function block FB_EL6821Communication.



- PLC
  - DALI_Sample_EL6821_CX5120
    - DALI_Sample_EL6821_CX5120 Project
      - External Types
      - References
      - DUTs
      - GVLs
      - POUs
      - VISUs
      - DALI_Sample_EL6821_CX5120.tmc
      - PlcCommunication (PlcCommunication)
      - PlcTask (PlcTask)
    - DALI_Sample_EL6821_CX5120 Instance
- I/O

Input Assistant

Text search | Categories

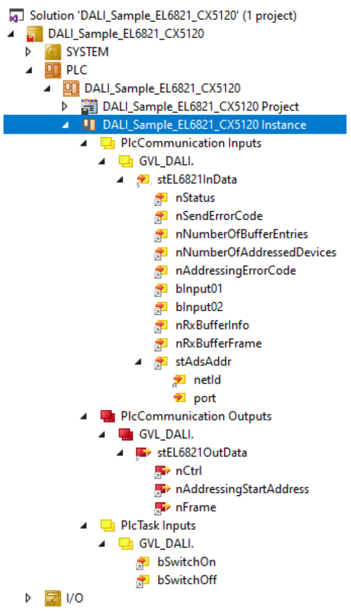
Programs

Name	Type
DALI_Sample_EL6821_CX5120	Application
POUs	
Communication	PROGRAM
MAIN	PROGRAM

I/O configuration

Select the CX as target system and initiate a search for its hardware.

In the instance (**DALI_Sample_EL6821_CX5120 Instance**) of the PLC project, you can see that the input and output variables of the PLC program are assigned to the corresponding tasks (**PlcCommunication** and **PlcTask**).



```
Solution 'DALI_Sample_EL6821_CX5120' (1 project)
  DALI_Sample_EL6821_CX5120
    SYSTEM
    PLC
      DALI_Sample_EL6821_CX5120
        DALI_Sample_EL6821_CX5120 Project
        DALI_Sample_EL6821_CX5120 Instance
          PlcCommunication Inputs
            GVL_DALI.
              stEL6821InData
                nStatus
                nSendErrorCode
                nNumberOfBufferEntries
                nNumberOfAddressedDevices
                nAddressingErrorCode
                bInput01
                bInput02
                nRxBufferInfo
                nRxBufferFrame
                stAdsAddr
                  netId
                  port
          PlcCommunication Outputs
            GVL_DALI.
              stEL6821OutData
                nCtrl
                nAddressingStartAddress
                nFrame
          PlcTask Inputs
            GVL_DALI.
              bSwitchOn
              bSwitchOff
    I/O
```

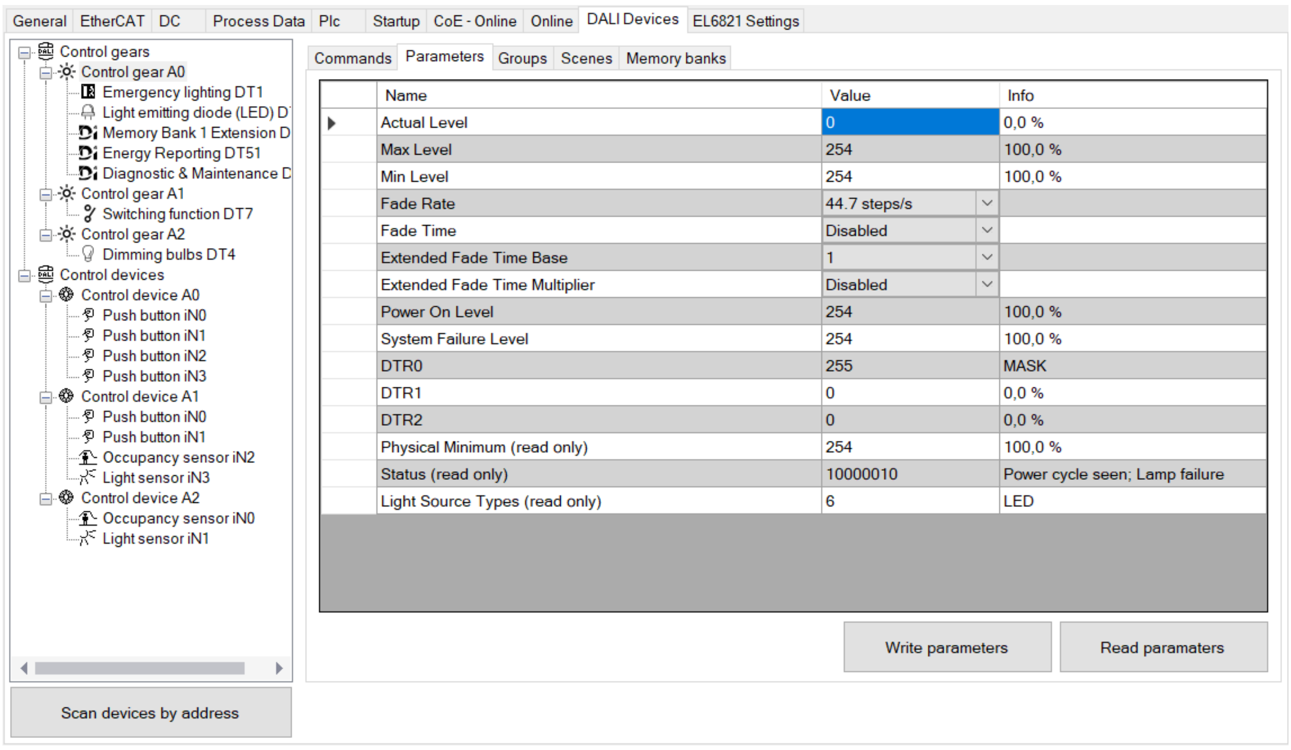
Now link the global variables of PLC program with the inputs and outputs of the bus terminals. Create the Solution and enable the configuration.

The lamp with the maximum brightness value is switched on by pressing the first push button. The second push button can be used to switch it off again.

6.2 Commissioning EL6821

From TwinCAT 3.1.4024.57, dialogs are available for the commissioning and diagnosis of DALI devices on the EL6821.

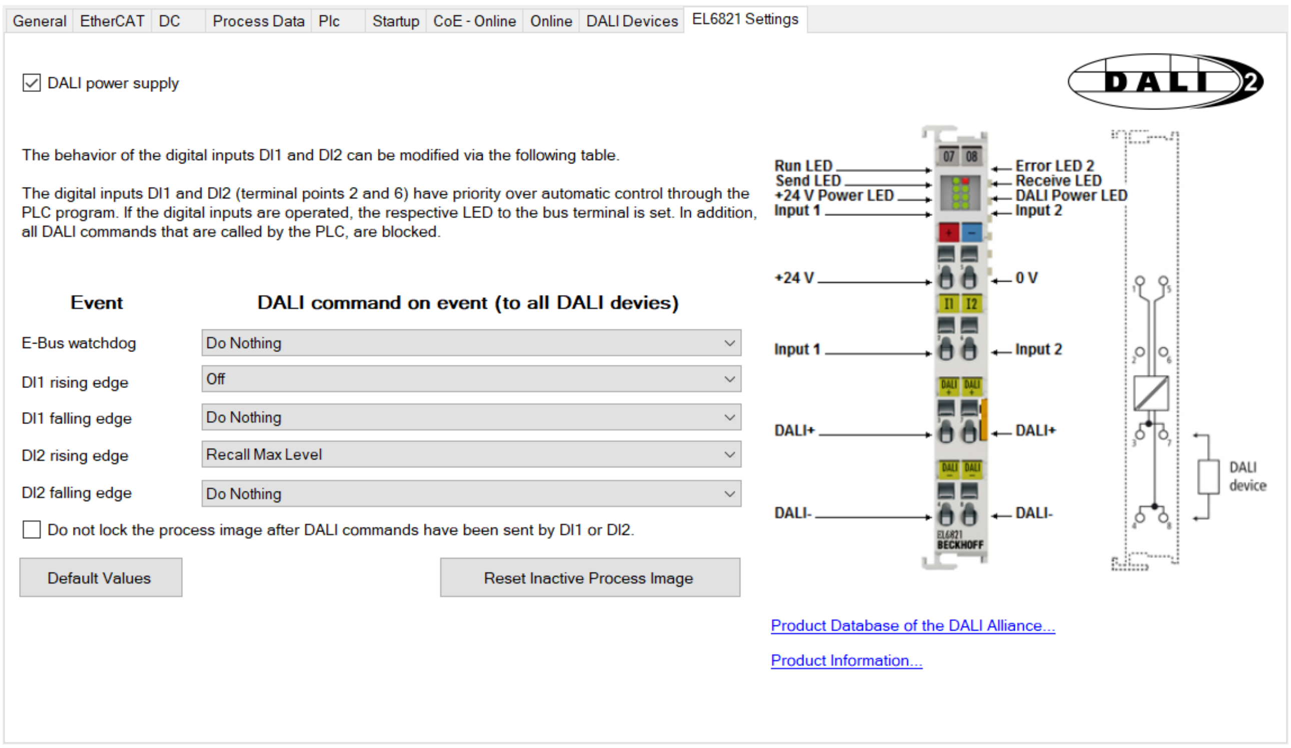
The dialogs offer not only functions for addressing DALI control gears and DALI control devices, but also for writing and reading parameters. The search function automatically detects the DALI device types and displays them in a tree structure.



- Control gears
  - Control gear A0
    - Emergency lighting DT1
    - Light emitting diode (LED) D
    - Memory Bank 1 Extension D
    - Energy Reporting DT51
    - Diagnostic & Maintenance D
  - Control gear A1
    - Switching function DT7
  - Control gear A2
    - Dimming bulbs DT4
- Control devices
  - Control device A0
    - Push button iN0
    - Push button iN1
    - Push button iN2
    - Push button iN3
  - Control device A1
    - Push button iN0
    - Push button iN1
    - Occupancy sensor iN2
    - Light sensor iN3
  - Control device A2
    - Occupancy sensor iN0
    - Light sensor iN1

Name	Value	Info
Actual Level	0	0,0 %
Max Level	254	100,0 %
Min Level	254	100,0 %
Fade Rate	44.7 steps/s	
Fade Time	Disabled	
Extended Fade Time Base	1	
Extended Fade Time Multiplier	Disabled	
Power On Level	254	100,0 %
System Failure Level	254	100,0 %
DTR0	255	MASK
DTR1	0	0,0 %
DTR2	0	0,0 %
Physical Minimum (read only)	254	100,0 %
Status (read only)	10000010	Power cycle seen; Lamp failure
Light Source Types (read only)	6	LED

Furthermore, all parameters of the EL6821 can be set via TwinCAT XAE. This makes it possible, for example, to define the DALI commands that are sent when the digital inputs on the EL6821 are actuated.



☑ DALI power supply

The behavior of the digital inputs DI1 and DI2 can be modified via the following table.

The digital inputs DI1 and DI2 (terminal points 2 and 6) have priority over automatic control through the PLC program. If the digital inputs are operated, the respective LED to the bus terminal is set. In addition, all DALI commands that are called by the PLC, are blocked.

Event	DALI command on event (to all DALI devices)
E-Bus watchdog	Do Nothing
DI1 rising edge	Off
DI1 falling edge	Do Nothing
DI2 rising edge	Recall Max Level
DI2 falling edge	Do Nothing

☐ Do not lock the process image after DALI commands have been sent by DI1 or DI2.

Overview of the most important functions in the TwinCAT XAE for the EL6821:

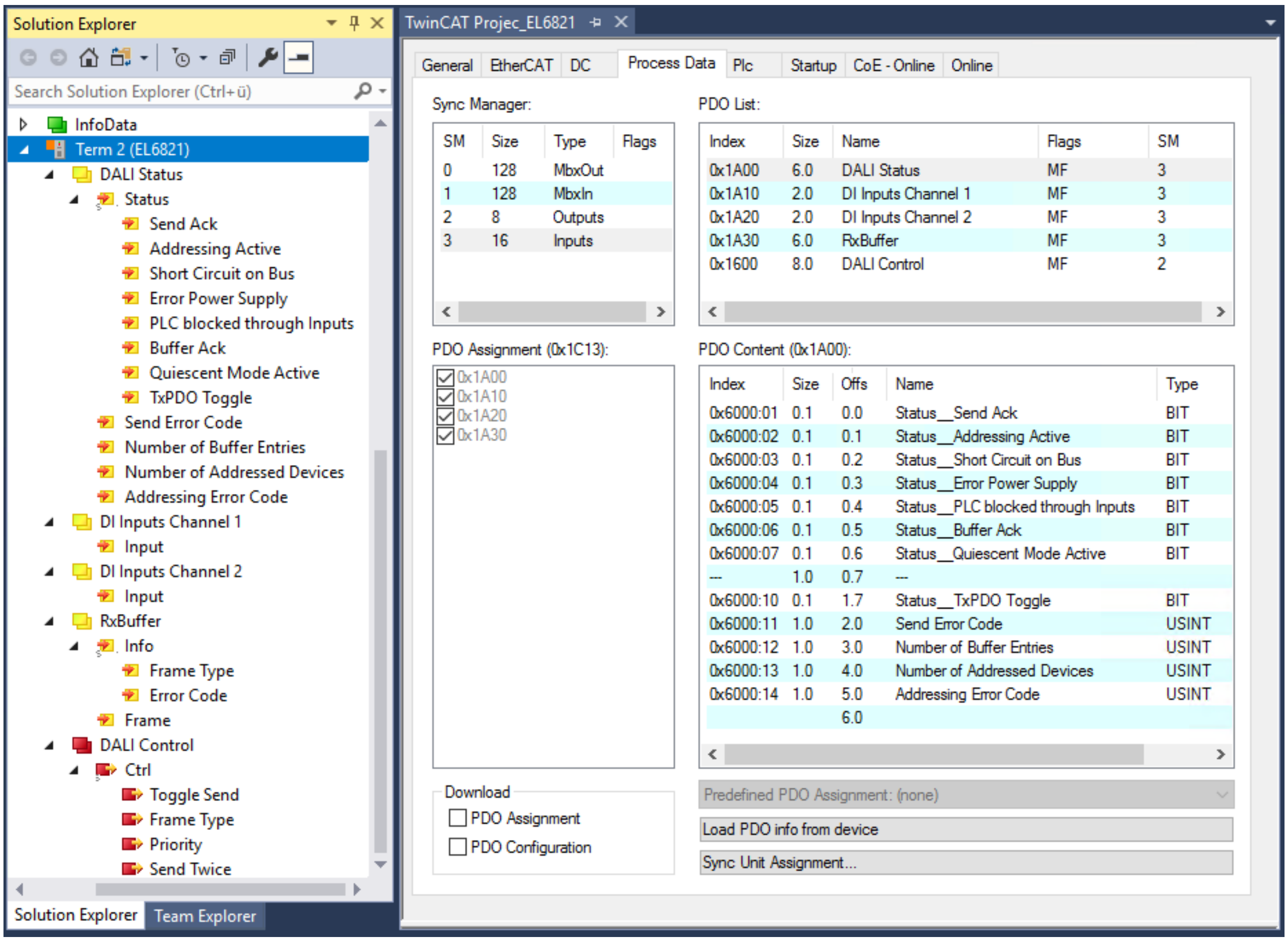
- Find DALI devices

- Addressing the DALI devices, including adapting the short addresses
- Scene and group assignment of the DALI control gears
- Configuration of the DALI control gears, including the parameters for the different device types
- Configuration of the DALI control devices, including the parameters for the different sensor types
- Writing/reading of the memory banks
- Definition of DALI commands for DI1 and DI2 (separate for rising and falling edge)
- Definition of the DALI command for the E-bus watchdog (E-bus failure)
- Switching the DALI power supply unit on/off
- Activation/deactivation of the blocking of the process image for the PLC as soon as a DALI command is sent by actuating a digital input on the EL6821.

6.3 EL6821 - Process data

The available process data objects (PDOs) are displayed in the “PDO List” in the “Process Data” tab. "PDO Content" and "PDO Assignment" of the process data objects cannot be changed for the EL6821. The input process data is assigned to Sync Manager 3 (SM 3), the output process data to Sync Manager 2 (SM 2).

Sync Manager SM 3 "Inputs"



Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	8	Outputs	
3	16	Inputs	

PDO List:

Index	Size	Name	Flags	SM
0x1A00	6.0	DALI Status	MF	3
0x1A10	2.0	DI Inputs Channel 1	MF	3
0x1A20	2.0	DI Inputs Channel 2	MF	3
0x1A30	6.0	RxBuffer	MF	3
0x1600	8.0	DALI Control	MF	2

PDO Assignment (0x1C13):

- [x] 0x1A00
- [x] 0x1A10
- [x] 0x1A20
- [x] 0x1A30

PDO Content (0x1A00):

Index	Size	Offs	Name	Type
0x6000:01	0.1	0.0	Status__Send Ack	BIT
0x6000:02	0.1	0.1	Status__Addressing Active	BIT
0x6000:03	0.1	0.2	Status__Short Circuit on Bus	BIT
0x6000:04	0.1	0.3	Status__Error Power Supply	BIT
0x6000:05	0.1	0.4	Status__PLC blocked through Inputs	BIT
0x6000:06	0.1	0.5	Status__Buffer Ack	BIT
0x6000:07	0.1	0.6	Status__Quiescent Mode Active	BIT
---	1.0	0.7	---	
0x6000:10	0.1	1.7	Status__TxPDO Toggle	BIT
0x6000:11	1.0	2.0	Send Error Code	USINT
0x6000:12	1.0	3.0	Number of Buffer Entries	USINT
0x6000:13	1.0	4.0	Number of Addressed Devices	USINT
0x6000:14	1.0	5.0	Addressing Error Code	USINT
		6.0		

Download
- [ ] PDO Assignment
- [ ] PDO Configuration

Predefined PDO Assignment: (none)

Load PDO info from device

Sync Unit Assignment...

Fig. 96: EL6821 – Input process data SM 3

See also:

0x6xxx - Input data [▸ 120]

0x1A00 "DALI status [▸ 124]",

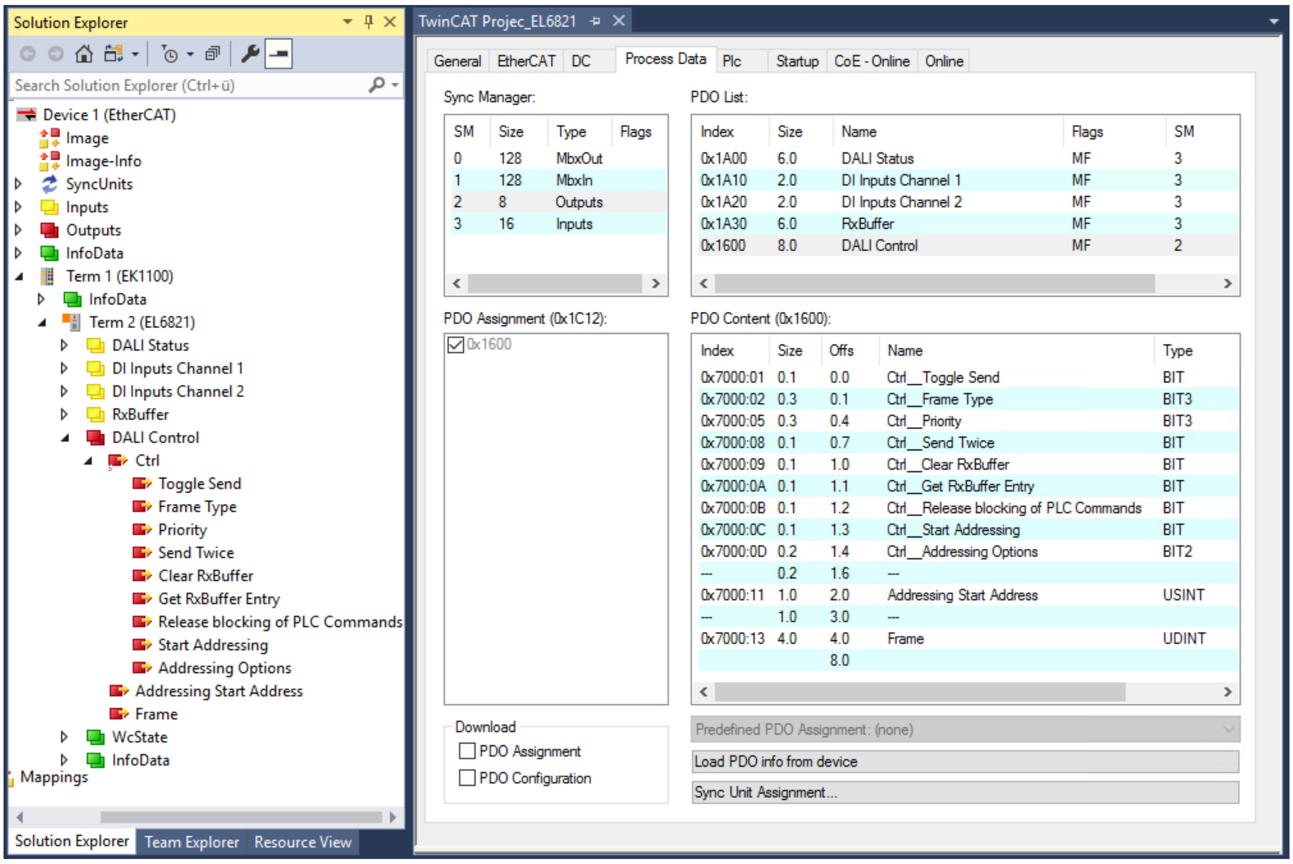
0x1A10 "DI Inputs Channel 1", [▸ 125]

0x1A20 "DI Inputs Channel 1" [▸ 125],

0x1A30 "RxBuffer" [▸ 125],

0x1600 "DALI Control" [▸ 124]

Sync Manager SM 2 "Outputs"



- Device 1 (EtherCAT)
  - Image
  - Image-Info
  - SyncUnits
  - Inputs
  - Outputs
  - InfoData
  - Term 1 (EK1100)
    - InfoData
    - Term 2 (EL6821)
      - DALI Status
      - DI Inputs Channel 1
      - DI Inputs Channel 2
      - RxBuffer
      - DALI Control
        - Ctrl
          - Toggle Send
          - Frame Type
          - Priority
          - Send Twice
          - Clear RxBuffer
          - Get RxBuffer Entry
          - Release blocking of PLC Commands
          - Start Addressing
          - Addressing Options
        - Addressing Start Address
        - Frame
      - WcState
      - InfoData
- Mappings

Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	8	Outputs	
3	16	Inputs	

PDO List:

Index	Size	Name	Flags	SM
0x1A00	6.0	DALI Status	MF	3
0x1A10	2.0	DI Inputs Channel 1	MF	3
0x1A20	2.0	DI Inputs Channel 2	MF	3
0x1A30	6.0	RxBuffer	MF	3
0x1600	8.0	DALI Control	MF	2

PDO Assignment (0x1C12):

- [✓] 0x1600

PDO Content (0x1600):

Index	Size	Offs	Name	Type
0x7000:01	0.1	0.0	Ctrl__Toggle Send	BIT
0x7000:02	0.3	0.1	Ctrl__Frame Type	BIT3
0x7000:05	0.3	0.4	Ctrl__Priority	BIT3
0x7000:08	0.1	0.7	Ctrl__Send Twice	BIT
0x7000:09	0.1	1.0	Ctrl__Clear RxBuffer	BIT
0x7000:0A	0.1	1.1	Ctrl__Get RxBuffer Entry	BIT
0x7000:0B	0.1	1.2	Ctrl__Release blocking of PLC Commands	BIT
0x7000:0C	0.1	1.3	Ctrl__Start Addressing	BIT
0x7000:0D	0.2	1.4	Ctrl__Addressing Options	BIT2
---	0.2	1.6	---	
0x7000:11	1.0	2.0	Addressing Start Address	USINT
---	1.0	3.0	---	
0x7000:13	4.0	4.0	Frame	UDINT
		8.0		

Download
- [ ] PDO Assignment
- [ ] PDO Configuration

Predefined PDO Assignment: (none)

Load PDO info from device

Sync Unit Assignment...

Fig. 97: EL6821 – Output process data SM 2

See also:

0x7000 – Output data [▶ 121]

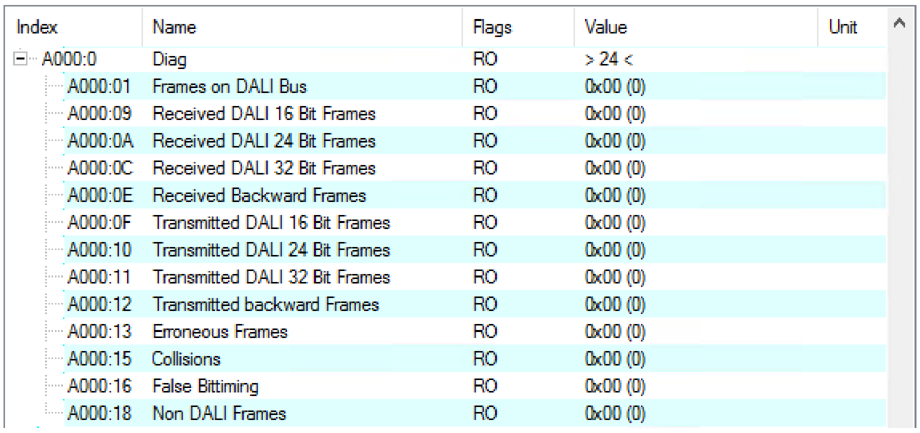
0x1600 - "DALI Control" [▶ 124]

6.4 EL6821 - Devices diagnostic functions

The CoE directory can be used to read diagnostic information on DALI communication (0xA000:0 "Diag") and the LED status (object 0xF915:0 "LED status").

Diagnostics via object 0xA000:0

The terminal has integrated diagnostics that can be read in object 0xA000:0.

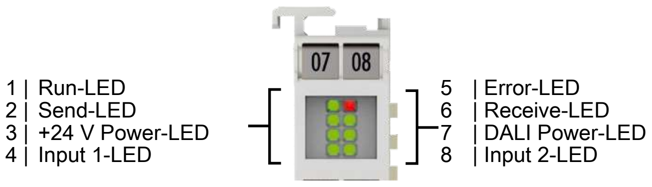


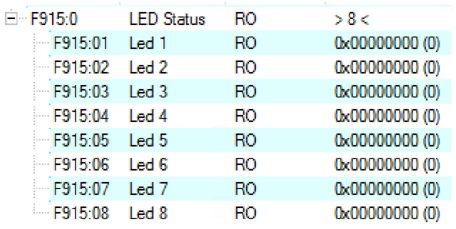
Index	Name	Flags	Value	Unit
A000:0	Diag	RO	> 24 <	
A000:01	Frames on DALI Bus	RO	0x00 (0)	
A000:09	Received DALI 16 Bit Frames	RO	0x00 (0)	
A000:0A	Received DALI 24 Bit Frames	RO	0x00 (0)	
A000:0C	Received DALI 32 Bit Frames	RO	0x00 (0)	
A000:0E	Received Backward Frames	RO	0x00 (0)	
A000:0F	Transmitted DALI 16 Bit Frames	RO	0x00 (0)	
A000:10	Transmitted DALI 24 Bit Frames	RO	0x00 (0)	
A000:11	Transmitted DALI 32 Bit Frames	RO	0x00 (0)	
A000:12	Transmitted backward Frames	RO	0x00 (0)	
A000:13	Erroneous Frames	RO	0x00 (0)	
A000:15	Collisions	RO	0x00 (0)	
A000:16	False Bittiming	RO	0x00 (0)	
A000:18	Non DALI Frames	RO	0x00 (0)	

Fig. 98: EL6821 - CoE object 0xA000 "Diag"

Index	Name	Description
0xA000:01	Frames on DALI Bus	Frames on DALI bus
0xA000:09	Received DALI 16 Bit Frames	Received DALI 16-bit frames
0xA000:0A	Received DALI 24 Bit Frames	Received DALI 24-bit frames
0xA000:0C	Received DALI 32 Bit Frames	Received DALI 32-bit frames
0xA000:0E	Received Backward Frames	Received DALI 8-bit response frames
0xA000:0F	Transmitted DALI 16 Bit Frames	Transmitted DALI 16-bit frames
0xA000:0F	Transmitted DALI 24 Bit Frames	Transmitted DALI 24-bit frames
0xA000:11	Transmitted DALI 32 Bit Frames	Transmitted DALI 32-bit frames
0xA000:12	Transmitted backward Frames	Transmitted DALI 8-bit backward frames
0xA000:13	Erroneous Frames	Erroneous frames
0xA000:15	Collisions	Collisions Observe the information and notes in the chapters "Priorities [▸ 20]" and "Bus timing [▸ 21]"
0xA000:16	False Bittiming	A frame was received that deviates from the prescribed bit timing (including tolerance) and is therefore recognized as incorrect.
0xA000:18	Non DALI Frames	Non DALI frames

LED status in object 0xF915:0 using the EL6821 terminal as an example





```
F915:0      LED Status   RO   > 8 <
  F915:01   Led 1        RO   0x00000000 (0)
  F915:02   Led 2        RO   0x00000000 (0)
  F915:03   Led 3        RO   0x00000000 (0)
  F915:04   Led 4        RO   0x00000000 (0)
  F915:05   Led 5        RO   0x00000000 (0)
  F915:06   Led 6        RO   0x00000000 (0)
  F915:07   Led 7        RO   0x00000000 (0)
  F915:08   Led 8        RO   0x00000000 (0)
```

LED	Index	Value	LED Status	Meaning
1 | Run-LED (green)	0xF915:01	0x00 00 FF 00	Off	EtherCAT State Machine: INIT
		0x80 00 FF 00	Flashing uniformly	EtherCAT State Machine: PREOP
		0x81 00 FF 00	Flashing slowly	EtherCAT State Machine: SAFEOP
		0x82 00 FF 00	Flashing rapidly	EtherCAT State Machine: BOOT
		0xFF 00 FF 00	On	EtherCAT State Machine: OP
Status LEDs (green) 2 | Send 3 | +24 V Power 4 | Input 1 6 | Receive 7 | DALI Power 8 | Input 2-LED	 0xF915:02 0xF915:03 0xF915:04 0xF915:06 0xF915:07 0xF915:08	0xFF 00 FF 00	On	2/6 | Signal is sent/received 3 | 24 V power supply OK 4/8 | Signal at input 1/2 at 7 | Integrated DALI power supply is on
		0x00 00 FF 00	Off	2/6 | no signal is sent/received 3 | no 24 V power supply 4/8 | no signal at input 1/2 7 | integrated DALI power supply is off
5 | Error LED (red)	0xF915:05	0xFF 00 FF 00	On	No 24 V power supply or hardware error
		0x00 00 FF 00	Off	No error
s. LED description [▸ 57]				

6.5 EL6821 - object description and parameterization

EtherCAT XML Device Description

The display matches that of the CoE objects from the EtherCAT ESI Device Description (XML).We recommend downloading the latest XML file from the download area of the Beckhoff website and installing it according to installation instructions.

Parameterization via the CoE list (CAN over EtherCAT)

The EtherCAT device is parameterized via the CoE-Online tab [▸ 86] (double-click on the respective object) or via the Process Data tab [▸ 83](allocation of PDOs). Please note the following general CoE notes [▸ 37] when using/manipulating the CoE parameters:

- Keep a startup list if components have to be replaced
- Differentiation between online/offline dictionary, existence of current XML description
- use "CoE reload [▸ 142]" for resetting changes

6.5.1 Restore object (0x1011)

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})

6.5.2 Configuration data (0x8xxx)

Index 8000 Emergency Message

Index (hex)	Name	Meaning	Data type	Flags	Default
8000:0	Emergency Message	Maximum subindex	UINT8	RO	>14<
8000:01	Frame Type	Frame type: 0: "No Frame" (factory setting) 1: "16 Bit" 2: "24 Bit" 3: "32 Bit"	BIT2	RW	0x00 (0_{dec})
8000:03	Frame Priority	Frame priority 0: "High" (factory setting) 1: "Middle High" 2: "Middle" 3: "Middle Low" 4: "Low"	BIT3	RW	0x00 (0_{dec})
8000:06	Block PLC Options	0: "Do Nothing" (factory setting) 1: "Block PLC Options" PLC-controlled DALI commands are ignored	BIT2	RW	0x00 (0_{dec})
8000:0E	Frame	The received DALI telegram (raw data)	UINT32	RW	0x00000000 (0_{dec})

Index 8001 Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8001:0	Settings	Maximum subindex	UINT8	RO	>1<
8001:01	Ignore non affecting Collisions	TRUE (factory setting): Non affecting collisions are ignored FALSE: Non affecting collisions are not ignored	BOOLEAN	RW	0x00 (0_{dec})
8001:02	Ignore 16 Bit Frames	TRUE (factory setting): 16-bit frames are ignored FALSE: 16-bit frames are not ignored	BOOLEAN	RW	0x00 (0_{dec})
8001:03	Ignore 24 Bit Frames	0: "Off" (factory setting) 24-bit frames are not ignored 1: "Ignore all but Events" Everything is ignored except events 2: "Ignore all" Everything is ignored	BIT3	RW	0x00 (0_{dec})
8001:05	Ignore 32 Bit Frames	TRUE: 32-bit frames are ignored FALSE (factory setting): 32 bit frames are not ignored	BOOLEAN	RW	0x00 (0_{dec})
8001:08	Enable Power Supply	Enable power supply TRUE (factory setting): Power supply is switched on FALSE: Power supply is switched off	BOOLEAN	RW	0x01 (1_{dec})
8001:09	DALI Short Address	DALI short address	UINT8	RW	0xFF (255_{dec})

Index 8002 Memory Bank 2

Index (hex)	Name	Meaning	Data type	Flags	Default
8002:0	Memory Bank 2	Maximum subindex	UINT8	RO	>1<
8002:01	Offset of last memory area	Offset of the last memory area inside the memory bank that can be accessed.	UINT8	RO	0xFE (254_{dec})
8002:02	Indicator Byte	Indicator Byte	UINT8	RO	0x00 (0_{dec})
8002:03	Lock Byte	Lock byte for the memory bank	UINT8	RO	0xFF (255_{dec})
8002:04	Additional Device Information #03	Additional device information	UINT8	RW	0x00 (0_{dec})
…	…	…	UINT8	RW	0x00 (0_{dec})
8002:FF	Additional Device Information #254	Additional device information	UINT8	RW	0x00 (0_{dec})

Index 8010 Input Settings Ch. 1

Index (hex)	Name	Meaning	Data type	Flags	Default
8010:0	Input Settings Ch. 1	Maximum subindex	UINT8	RO	>1<
8010:01	Frame Type falling Edge	Frame type on falling edge at "Input 1" 0: "No Frame" (factory setting) 1: "16 Bit" 2: "24 Bit" 3: "32 Bit"	BIT3	RW	0x00 (0_{dec})
8010:03	Frame Priority falling Edge	Frame priority on falling edge at "Input 1" 0: "High" (factory setting) 1: "Middle High" 2: "Middle" 3: "Middle Low" 4: "Low"	BIT3	RW	0x00 (0_{dec})
8010:06	Block PLC Options falling Edge	0: "Do Nothing" (factory setting) 1: "Block PLC Options" Priority of the digital input "Input 1", PLC-controlled DALI commands are blocked on a falling edge at "Input 1". 2: "Release" The process image is released again on a falling edge at "Input 1".	BIT3	RW	0x00 (0_{dec})
8010:08	Frame Type rising Edge	Frame type on rising edge at "Input 1" 0: "No Frame" (factory setting) 1: "16 Bit" 2: "24 Bit" 3: "32 Bit"	BIT3	RW	0x00 (0_{dec})
8010:0A	Frame Priority rising Edge	Frame priority on rising edge at "Input 1" 0: "High" (factory setting) 1: "Middle High" 2: "Middle" 3: "Middle Low" 4: "Low"	BIT3	RW	0x00 (0_{dec})
8010:0D	Block PLC Options rising Edge	0: "Do Nothing" 1: "Block PLC Options" (factory setting) Priority setting of the digital input "Input 1", PLC-controlled DALI commands are blocked on a rising edge at "Input 1". 2: "Release" The process image is released again on a rising edge at "Input 1".	BIT3	RW	0x01 (1_{dec})
8010:11	Frame falling Edge	Frame on falling edge at "Input 1"	UINT32	RW	0x00000000 (0_{dec})
8010:12	Frame rising Edge	Frame on rising edge at "Input 1"	UINT32	RW	0x0000FF05 (65285_{dec})

Index 8020 Input Settings Ch. 2

Index (hex)	Name	Meaning	Data type	Flags	Default
8020:0	Input Settings Ch. 2	Maximum subindex	UINT8	RO	>18<
8020:01	Frame Type falling Edge	Frame type on falling edge at "Input 2" 0: "No Frame" (factory setting) 1: "16 Bit" 2: "24 Bit" 3: "32 Bit"	BIT3	RW	0x00 (0_{dec})
8020:03	Frame Priority falling Edge	Frame priority on falling edge at "Input 2" 0: "High" (factory setting) 1: "Middle High" 2: "Middle" 3: "Middle Low" 4: "Low"	BIT3	RW	0x00 (0_{dec})
8020:06	Block PLC Options falling Edge	0: "Do Nothing" (factory setting) 1: "Block PLC Options" Priority of the digital input "Input 2", PLC-controlled DALI commands are blocked on a falling edge at "Input 2". 2: "Release" The process image is released again on a falling edge at "Input 2".	BIT3	RW	0x00 (0_{dec})
8020:08	Frame Type rising Edge	Frame type on rising edge at "Input 2" 0: "No Frame" (factory setting) 1: "16 Bit" 2: "24 Bit" 3: "32 Bit"	BIT3	RW	0x00 (0_{dec})
8020:0A	Frame Priority rising Edge	Frame priority on rising edge at "Input 2" 0: "High" (factory setting) 1: "Middle High" 2: "Middle" 3: "Middle Low" 4: "Low"	BIT3	RW	0x00 (0_{dec})
8020:0D	Block PLC Options rising Edge	0: "Do Nothing" 1: "Block PLC Options" (factory setting) Priority setting of the digital input "Input 2", PLC-controlled DALI commands are blocked on a rising edge at "Input 2". 2: "Release" The process image is released again on a rising edge at "Input 2".	BIT3	RW	0x01 (1_{dec})
8020:11	Frame falling Edge	Frame on falling edge at "Input 2"	UINT32	RW	0x00000000 (0_{dec})
8020:12	Frame rising Edge	Frame on rising edge at "Input 2"	UINT32	RW	0x0000FF05 (0_{dec})

6.5.3 Input data (0x6xxx)

Index 6000 Status

Index (hex)	Name	Meaning	Data type	Flags	Default
6000:0	Status	Maximum subindex	UINT8	RO	>20<
6000:01	Status_Send Ack	Send acknowledgement	BOOLEAN	RO	FALSE
6000:02	Status_Addressing Active	Addressing active	BOOLEAN	RO	FALSE
6000:03	Status_Short Circuit on Bus	Short circuit on bus	BOOLEAN	RO	FALSE
6000:04	Status_Error Power Supply	When using the internal DALI power supply: power supply error detected.	BOOLEAN	RO	FALSE
6000:05	Status_PLC blocked through Inputs	PLC-controlled commands are blocked	BOOLEAN	RO	FALSE
6000:06	Status_Buffer Ack	Command buffer “Buffer”*) Toggled analogously to 0x7000:0A when a new entry is provided in the Rx buffer: Request for the next Rx buffer entry: set 0x7000:0A from 0 to 1. -> If a new entry was provided in the next EtherCAT cycle, this is acknowledged in the status by setting 0x6000:06 to "1" as well.	BOOLEAN	RO	FALSE
6000:07	Status_Quiescent Mode Active	Quiescent Mode is active. In quiescent mode, no commands and events are sent by the DALI control device for approx. 15 minutes.	BOOLEAN	RO	FALSE
6000:10	Status_TxPDO Toggle	Toggles when new data is available.	BOOLEAN	RO	FALSE
6000:11	Send Error Code	An error code has been sent.	UINT8	RO	0x00 (0_{dec})
6000:12	Number of Buffer Entries	Number of buffer entries	UINT8	RO	0x00 (0_{dec})
6000:13	Number of Addressed Devices	Number of addressed devices	UINT8	RO	0x00 (0_{dec})
6000:14	Addressing Error Code	An error occurred during addressing	UINT8	RO	0x00 (0_{dec})
*) The function blocks for the DALI commands do not directly access the process image of the EL6821, but place the individual DALI commands in a command buffer. The "FB_EL6821Communication" function block sequentially reads the DALI commands from the command buffer and forwards them to the EL6821. This prevents multiple function blocks accessing the EL6821 process image at the same time.					

Index 6010 Input Ch. 1

Index (hex)	Name	Meaning	Data type	Flags	Default
6010:0	Input Ch. 1	Maximum subindex	UINT8	RO	>1<
6010:01	Input	Status of "Input 1"	BOOLEAN	RO	FALSE

Index 6020 Input Ch. 2

Index (hex)	Name	Meaning	Data type	Flags	Default
6020:0	Input Ch. 2	Maximum subindex	UINT8	RO	>1<
6020:01	Input	Status of "Input 2"	BOOLEAN	RO	FALSE

Index 6030 RxBuffer Entry

Index (hex)	Name	Meaning	Data type	Flags	Default
6030:0	RxBuffer Entry	Maximum subindex	UINT8	RO	>17<
6030:01	Info_Frame Type	Frame type 0: "No Frame" 1: "8 Bit Answer" 2: "16 Bit" 3: "24 Bit" 4: "32 Bit" 5: "Non DALI"	BIT3	RO	0x00 (0_{dec})
6030:09	Info_Error Code	Error code	UINT8	RO	0x00 (0_{dec})
6030:11	Frame	Frame	UINT32	RO	0x00000000 (0_{dec})

6.5.4 Output data (0x7xxx)

Index 7000 Control

Index (hex)	Name	Meaning	Data type	Flags	Default
7000:0	Control	Maximum subindex	UINT8	RO	>19<
7000:01	Ctrl_Toggle Send	Toggles when sending	BOOLEAN	RW	FALSE
7000:02	Ctrl_Frame Type	Frame type 0: "16 Bit" 1: "24 Bit" 2: "32 Bit"	BIT3	RO	0x00 (0_{dec})
7000:05	Ctrl_Priority	Priority 0: "High" 1: "Middle High" 2: "Middle" 3: "Middle Low" 4: "Low"	BIT3	RO	0x00 (0_{dec})
7000:08	Ctrl_Send Twice	Send DALI command twice (send twice*)	BOOLEAN	RW	FALSE
7000:09	Ctrl_Clear RxBuffer	Clear command buffer	BOOLEAN	RW	FALSE
7000:0A	Ctrl_Get RxBuffer Entry	Toggle to read the next buffer entry (see 0x6000:06).	BOOLEAN	RW	FALSE
7000:0B	Ctrl_Release blocking of PLC Commands	Release blocking of PLC commands	BOOLEAN	RW	FALSE
7000:0C	Ctrl_Start Addressing	Start addressing	BOOLEAN	RW	FALSE
7000:0D	Ctrl Addressing Options	Addressing options	BIT2	RO	0x00 (0_{dec})
7000:11	Addressing Start Address	Addressing Start address	UINT8	RO	0x00 (0_{dec})
7000:13	Frame	Frame	UINT32	RO	0x00000000 (0_{dec})
*) Certain DALI commands must be sent twice within 100 ms (send twice) without the receiver receiving any other DALI commands in the meantime (primarily commands for configuring DALI devices)					

6.5.5 Diagnostic data (0xA000, 0xF000, 0xF915)

Index A000 Diag

Index (hex)	Name	Meaning	Data type	Flags	Default
A000:0	Diag	Maximum subindex	UINT8	RO	>24<
A000:01	Frames on DALI Bus	Frames on DALI bus	UINT8	RO	0x00 (0_{dec})
A000:09	Received DALI 16 Bit Frames	Received DALI 16-bit frames	UINT8	RO	0x00 (0_{dec})
A000:0A	Received DALI 24 Bit Frames	Received DALI 24-bit frames	UINT8	RO	0x00 (0_{dec})
A000:0C	Received DALI 32 Bit Frames	Received DALI 32-bit frames	UINT8	RO	0x00 (0_{dec})
A000:0E	Received backward Frames	Received DALI 8-bit response frames	UINT8	RO	0x00 (0_{dec})
A000:0F	Transmitted DALI 16 Bit Frames	Transmitted DALI 16-bit frames	UINT8	RO	0x00 (0_{dec})
A000:10	Transmitted DALI 24 Bit Frames	Transmitted DALI 24-bit frames	UINT8	RO	0x00 (0_{dec})
A000:11	Transmitted DALI 32 Bit Frames	Transmitted DALI 32-bit frames	UINT8	RO	0x00 (0_{dec})
A000:12	Transmitted backward Frames	Transmitted DALI 8-bit backward frames	UINT8	RO	0x00 (0_{dec})
A000:13	Erroneous Frames	Erroneous frames	UINT8	RO	0x00 (0_{dec})
A000:15	Collisions	Collisions	UINT8	RO	0x00 (0_{dec})
A000:16	False Bittiming	A frame was received that deviates from the prescribed bit timing (including tolerance) and is therefore recognized as incorrect.	UINT8	RO	0x00 (0_{dec})
A000:18	Non DALI Frames	Non DALI frames	UINT8	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	0x0004 (4_{dec})

Index F915 LED Status

Index (hex)	Name	Meaning	Data type	Flags	Default
F915:0	LED Status	Maximum subindex	UINT8	RO	0x10 (16_{dec})
F915:01	Led 1	Run-LED	UINT32	RO	0x00000000 (0_{dec})
F915:02	Led 2	Send-LED	UINT32	RO	0x00000000 (0_{dec})
F915:03	Led 3	+24 V Power-LED	UINT32	RO	0x00000000 (0_{dec})
F915:04	Led 4	Input 1-LED	UINT32	RO	0x00000000 (0_{dec})
F915:05	Led 5	Error-LED	UINT32	RO	0x00000000 (0_{dec})
F915:06	Led 6	Receive-LED	UINT32	RO	0x00000000 (0_{dec})
F915:07	Led 7	DALI Power-LED	UINT32	RO	0x00000000 (0_{dec})
F915:08	Led 8	Input 2-LED	UINT32	RO	0x00000000 (0_{dec})

6.5.6 Standard objects

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default
1000:0	Device type	Device type of the EtherCAT slave: • The Lo-Word contains the CoE profile used (5001). • The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	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	EL6821

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	-

Index 100B Bootloader version

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

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	0x1AA53052 (447033426$_{dec}$)
1018:03	Revision	Revision number of the EtherCAT slave: • The low word (bit 0-15) identifies the special terminal number. • The high word (bit 16-31) refers to the device description.	UINT32	RO	-
1018:04	Serial number	Serial number of the EtherCAT slave: • Low-Byte ◦ 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	-

Index 10E2 Manufacturer-specific Identification Code

Index (hex)	Name	Meaning	Data type	Flags	Default
10E2:0	Manufacturer-specific Identification Code	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	-

Index 1600 DALI RxPDO-Map Control

Index (hex)	Name	Meaning	Data type	Flags	Default
1600:0	DALI RxPDO-Map Control	PDO Mapping DALI RxPDO-Map Control	UINT8	RO	>13<
1600:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (Control), entry 0x01 (Ctrl_Toggle Send))	UINT32	RO	0x7000:01, 1
1600:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (Control), entry 0x02 (Ctrl_Frame Type))	UINT32	RO	0x7000:02, 3
1600:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (Control), entry 0x05 (Ctrl_Priority))	UINT32	RO	0x7000:05, 3
1600:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (Control), entry 0x08 (Ctrl_Send Twice))	UINT32	RO	0x7000:08, 1
1600:05	SubIndex 005	5. PDO Mapping entry (object 0x7000 (Control), entry 0x09 (Ctrl_Clear RxBuffer))	UINT32	RO	0x7000:09, 1
1600:06	SubIndex 006	6. PDO Mapping entry (object 0x7000 (Control), entry 0x0A (Ctrl_Get RxBuffer Entry))	UINT32	RO	0x7000:0A, 1
1600:07	SubIndex 007	7. PDO Mapping entry (object 0x7000 (Control), entry 0x09 (Ctrl_Release blocking of PLC Commands))	UINT32	RO	0x7000:0B, 1
1600:08	SubIndex 008	8. PDO Mapping entry (object 0x7000 (Control), entry 0x09 (Ctrl_Start Addressing))	UINT32	RO	0x7000:0C, 1
1600:09	SubIndex 009	9. PDO Mapping entry (object 0x7000 (Control), entry 0x09 (Ctrl_Addressing Options))	UINT32	RO	0x7000:0D, 2
1600:0A	SubIndex 010	10. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1600:0B	SubIndex 011	11. PDO Mapping entry (object 0x7000 (Control), entry 0x11 (Addressing Start Address))	UINT32	RO	0x7000:11, 8
1600:0C	SubIndex 012	12. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1600:0D	SubIndex 013	13. PDO Mapping entry (object 0x7000 (Control), entry 0x13 (Frame))	UINT32	RO	0x7000:13, 32

Index 1A00 DALI TxPDO-Map Status

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00:0	DALI TxPDO-Map Status	PDO Mapping DALI TxPDO-Map Status	UINT8	RW	>13<
1A00:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (Status), entry 0x01 (Status_Send Ack))	UINT32	RW	0x6000:01, 1
1A00:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (Status), entry 0x02 (Status_Addressing Active))	UINT32	RW	0x6000:02, 1
1A00:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (Status), entry 0x03 (Status_Short Circuit on Bus))	UINT32	RW	0x6000:03, 1
1A00:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (Status), entry 0x04 (Status_Error Power Supply))	UINT32	RW	0x6000:04, 1
1A00:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (Status), entry 0x05 (Status_PLC blocked through Inputs))	UINT32	RW	0x6000:05, 1
1A00:06	SubIndex 006	6. PDO Mapping entry (object 0x6000 (Status), entry 0x06 (Status_Buffer Ack))	UINT32	RW	0x6000:06, 1
1A00:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (Status), entry 0x07 (Status_Quiescent Mode Active))	UINT32	RW	0x6000:07, 1
1A00:08	SubIndex 008	8. PDO Mapping entry (8 bits align)	UINT32	RW	0x0000:00, 8
1A00:09	SubIndex 009	9. PDO Mapping entry (object 0x6000 (Status), entry 0x10 (Status_TxPDO Toggle))	UINT32	RW	0x6000:10, 1
1A00:0A	SubIndex 010	10. PDO Mapping entry (object 0x6000 (Status), entry 0x11 (Send Error Code))	UINT32	RW	0x6000:11, 8
1A00:0B	SubIndex 11	11. PDO Mapping entry (object 0x6000 (Status), entry 0x12 (Number of Buffer Entries))	UINT32	RW	0x6000:12, 8
1A00:0C	SubIndex 12	12. PDO Mapping entry (object 0x6000 (Status), entry 0x13 (Number of Address Devices))	UINT32	RW	0x6000:13, 8
1A00:0D	SubIndex 13	13. PDO Mapping entry (object 0x6000 (Status), entry 0x14 (Addressing Error Code))	UINT32	RW	0x6000:14, 8

Index 1A10 DI TxPDO-Map Ch. 1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A10:0	DI TxPDO-Map Ch. 1	PDO Mapping DI TxPDO-Map Ch. 1	UINT8	RW	>2<
1A10:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (Input Ch. 1), entry 0x01 (Input))	UINT32	RW	0x6010:01, 1
1A10:02	SubIndex 002	2. PDO Mapping entry (15 bits align)	UINT32	RW	0x0000:00, 15

Index 1A20 DI TxPDO-Map Ch. 2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A20:0	DI TxPDO-Map Ch. 2	PDO Mapping DI TxPDO-Map Ch. 2	UINT8	RW	>2<
1A20:01	SubIndex 001	1. PDO Mapping entry (object 0x6020 (Input Ch. 2), entry 0x01 (Input))	UINT32	RW	0x6020:01, 1
1A20:02	SubIndex 002	2. PDO Mapping entry (15 bits align)	UINT32	RW	0x0000:00, 15

Index 1A30 RxBuffer TxPDO-Map

Index (hex)	Name	Meaning	Data type	Flags	Default
1A30:0	RxBuffer TxPDO-Map	PDO Mapping RxBuffer TxPDO-Map	UINT8	RW	>13<
1A30:01	SubIndex 001	1. PDO Mapping entry (object 0x6030 (RxBuffer Entry), entry 0x01 (Info_Frame Type))	UINT32	RW	0x6030:01, 3
1A30:02	SubIndex 002	2. PDO Mapping entry (5 bits align)	UINT32	RW	0x0000:00, 5
1A30:03	SubIndex 003	3. PDO Mapping entry (object 0x6030 (RxBuffer Entry), entry 0x09 (Info_Error Code))	UINT32	RW	0x6030:09, 8
1A30:04	SubIndex 004	4. PDO Mapping entry (object 0x6030 (RxBuffer Entry), entry 0x11 (Frame))	UINT32	RW	0x6030:11, 32

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x01 (1_{dec})
1C12:01	SubIndex 001	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1600 (5632_{dec})

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x02 (2_{dec})
1C13:01	Subindex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A00 (6656_{dec})
1C13:02	Subindex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A10 (6672_{dec})
1C13:03	Subindex 003	3. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A20 (6688_{dec})
1C13:04	Subindex 004	4. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A30 (6704_{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: FreeRun/SM-Synchron	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: cycle time of the master • DC-Mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x000F4240 (1000000$_{dec}$)
1C32:03	Shift time	Time between SYNC0 Event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0$_{dec}$)
1C32:04	Sync modes supported	Sync modes supported: • 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 (DC mode only) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08)	UINT16	RO	0x000B (11$_{dec}$)
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x000186A0 (100000$_{dec}$)
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 Event (in ns, DC Mode only)	UINT32	RO	0x00000000 (0$_{dec}$)
1C32:07	Minimum delay time	Minimum time between SYNC1 Event and output of the outputs (in ns)	UINT32	RO	0x00000000 (0$_{dec}$)
1C32:08	Get Cycle Time	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 0x1C32:03, 0x1C32:05, 0x1C32:06, 0x1C32:09, 0x1C33:03 [▸ 127], 0x1C33:06 [▸ 126], 0x1C33:09 [▸ 127] are updated with the maximum measured values. For a subsequent measurement the measured values are reset.	UINT16	RW	0x0000 (0$_{dec}$)
1C32:09	Maximum Delay time	Time between SYNC1 Event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0$_{dec}$)
1C32:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC Mode only)	UINT16	RO	0x0000 (0$_{dec}$)
1C32:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0$_{dec}$)
1C32:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 Event was too short (DC mode only)	UINT16	RO	0x0000 (0$_{dec}$)
1C32:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC Mode only)	BOOLEAN	RO	0x00 (0$_{dec}$)

Index 1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32$_{dec}$)
1C33:01	Sync mode	• 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	Cycle time (in ns): • Free Run: cycle time of the local timer • Synchron with SM 2 Event: cycle time of the master • DC-Mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x000F4240 (10000$_{dec}$)
1C33:03	Shift time	Time between SYNC0 Event and reading of the inputs (in ns, DC Mode only)	UINT32	RO	0x00000384 (900$_{dec}$)
1C33:04	Sync modes supported	Sync modes supported: • Bit 0: Free Run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron with SM 3 Event is supported (no outputs available) • Bit 2-3 = 01: DC-Mode is supported • Bit 4-5 = 01: Input shift through local event (outputs available) • Bit 4-5 = 10: Input Shift with SYNC1 Event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08 [▶ 126] or 0x1C33:08)	UINT16	RO	0x000B (11$_{dec}$)
1C33:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x000186A0 (100000$_{dec}$)
1C33:06	Calc and copy time	Time between reading of the inputs and the inputs being available for the master (in ns, DC mode only)	UINT32	RO	0x00000000 (0$_{dec}$)
1C33:07	Minimum delay time	Min. time between SYNC1 Event and the reading of the inputs (in ns, DC mode only)	UINT322	RO	0x00000384 (900$_{dec}$)
1C33:08	Get Cycle Time	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 0x1C32:03, 0x1C32:05, 0x1C32:06, 0x1C32:09 are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RW	0x0000 (0$_{dec}$)
1C33:09	Maximum delay time	Time between SYNC1 Event and reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0$_{dec}$)
1C33:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC Mode only)	UINT16	RO	0x0000 (0$_{dec}$)
1C33:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0$_{dec}$)
1C33:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 Event was too short (DC mode only)	UINT16	RO	0x0000 (0$_{dec}$)
1C33:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC Mode only)	BOOLEAN	ROO	0x00 (0$_{dec}$)

Index F008 Code word

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

7 Programming

The EL6821 is programmed exclusively using function blocks from the PLC libraries. Programming is performed exclusively via TwinCAT 3 function blocks.

See software documentation in the Beckhoff Information System.

TwinCAT 3: TwinCAT 3 | PLC Lib: Tc3_DALI

8 Appendix

8.1 EtherCAT AL Status Codes

For detailed information please refer to the EtherCAT system description.

8.2 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.

NOTICE

Risk of damage to the device!

Pay attention to the instructions for firmware updates on the separate page [▸ 130].
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!

EL6821			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Release date**
00	01	EL6821-0000-0016	05/2024
	02		08/2024
	03		12/2024
01*	04*	EL6821-0000-0017	09/2024

*) 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.

8.3 Firmware Update EL/ES/EM/ELM/EP/EPP/ERPxxxx

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

NOTICE

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.

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:

- Each EtherCAT slave has a device description, consisting of identity (name, product code), timing specifications, communication settings, etc.
  This device description (ESI; EtherCAT Slave Information) can be downloaded from the Beckhoff website in the download area as a zip file and used in EtherCAT masters for offline configuration, e.g. in TwinCAT.
  Above all, each EtherCAT slave carries its device description (ESI) electronically readable in a local memory chip, the so-called **ESI EEPROM**. When the slave is switched on, this description is loaded locally in the slave and informs it of its communication configuration; on the other hand, the EtherCAT master can identify the slave in this way and, among other things, set up the EtherCAT communication accordingly.

NOTICE

Application-specific writing of the ESI-EEPROM

The ESI is developed by the device manufacturer according to ETG standard and released for the corresponding product.
- Meaning for the ESI file: Modification on the application side (i.e. by the user) is not permitted.
- Meaning for the ESI EEPROM: Even if a writeability is technically given, the ESI parts in the EEPROM and possibly still existing free memory areas must not be changed beyond the normal update process. Especially for cyclic memory processes (operating hours counter etc.), dedicated memory products such as EL6080 or IPC's own NOVRAM must be used.

- 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.

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

NOTICE

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.

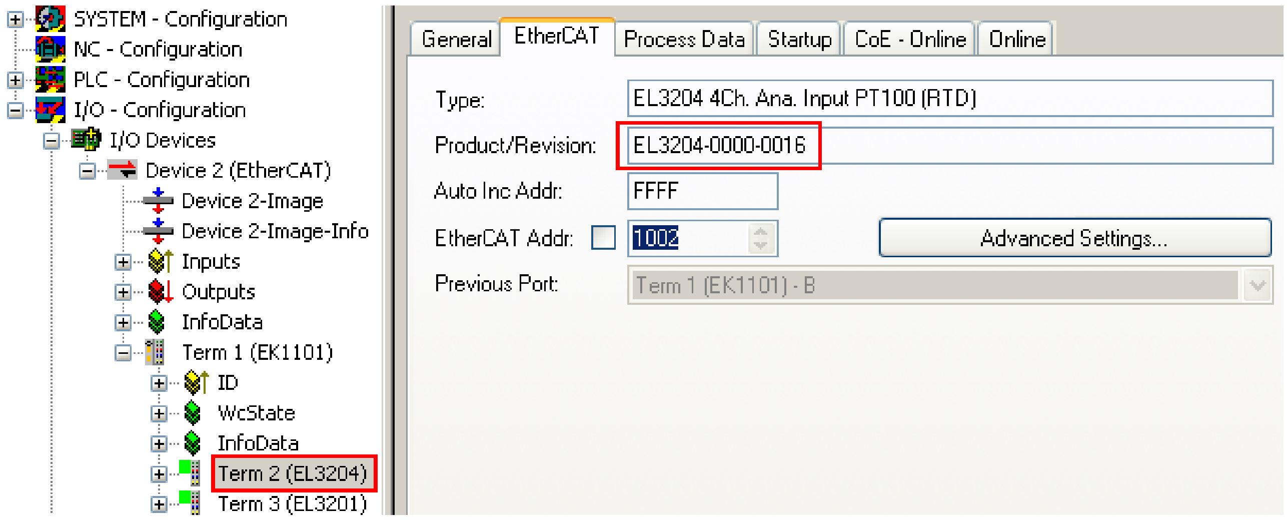
8.3.1 Device description ESI file/XML

NOTICE

Attention regarding update of the ESI description/EEPROM

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

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



Type: EL3204 4Ch. Ana. Input PT100 (RTD)
Product/Revision: EL3204-0000-0016
Auto Inc Addr: FFFF
EtherCAT Addr: 1002
Previous Port: Term 1 (EK1101) - B

Fig. 99: 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:

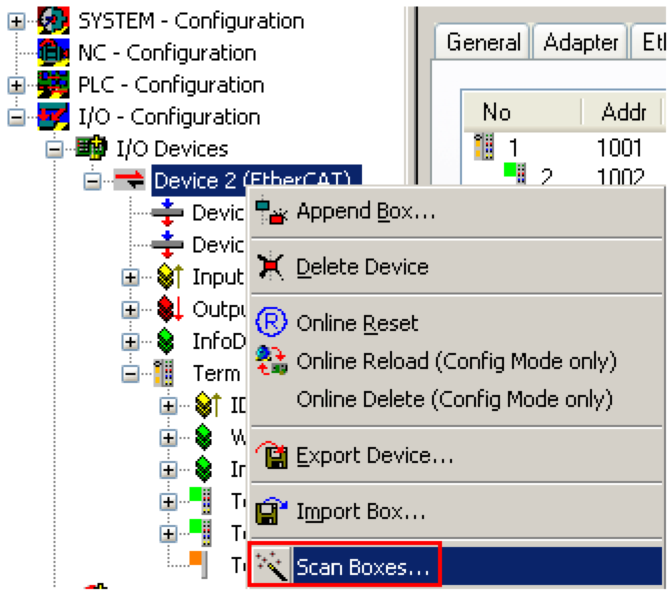
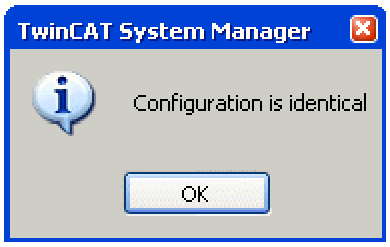


Fig. 100: Scan the subordinate field by right-clicking on the EtherCAT device

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


TwinCAT System Manager

Configuration is identical

OK

Fig. 101: Configuration is identical

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

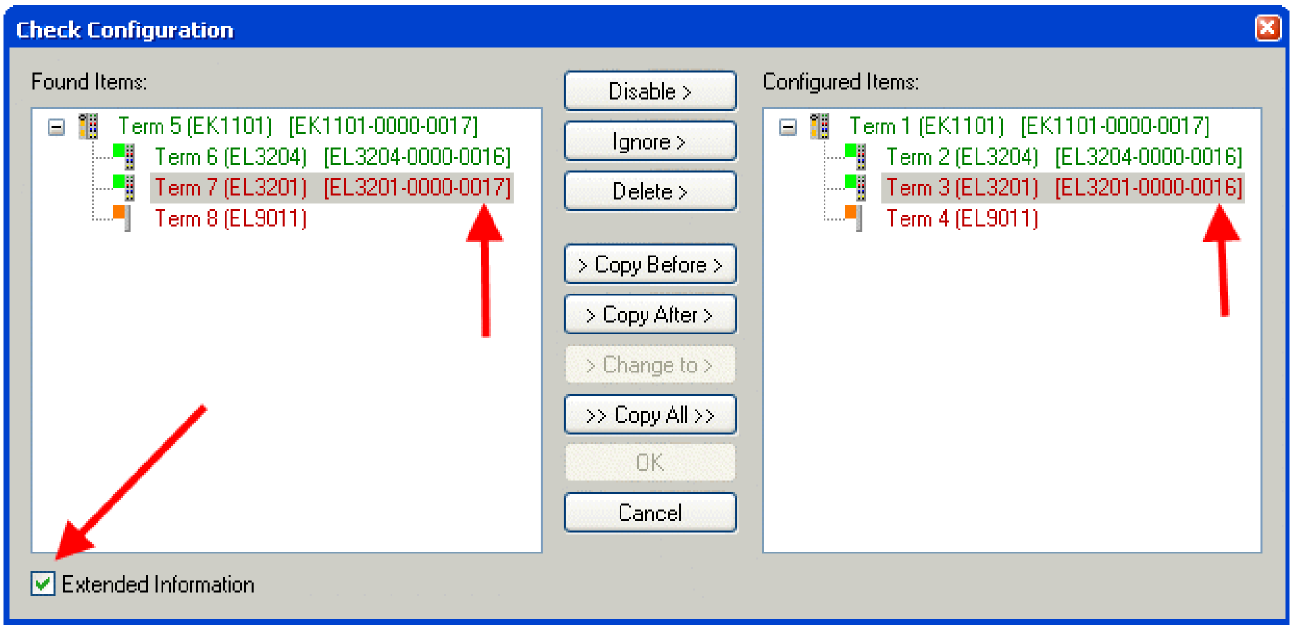

Check Configuration
Found Items:
Term 5 (EK1101) [EK1101-0000-0017]
Term 6 (EL3204) [EL3204-0000-0016]
Term 7 (EL3201) [EL3201-0000-0017]
Term 8 (EL9011)
Disable >
Ignore >
Delete >
> Copy Before >
> Copy After >
> Change to >
>> Copy All >>
OK
Cancel
Configured Items:
Term 1 (EK1101) [EK1101-0000-0017]
Term 2 (EL3204) [EL3204-0000-0016]
Term 3 (EL3201) [EL3201-0000-0016]
Term 4 (EL9011)
Extended Information

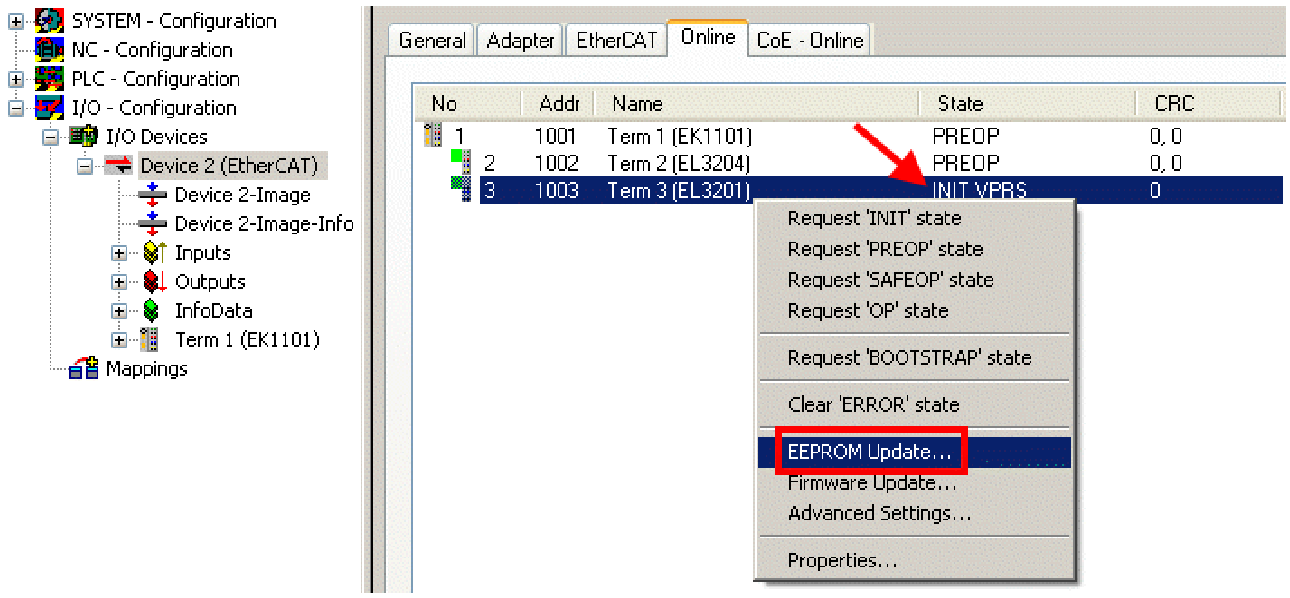
Fig. 102: Change dialog

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

Changing the ESI slave identifier

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

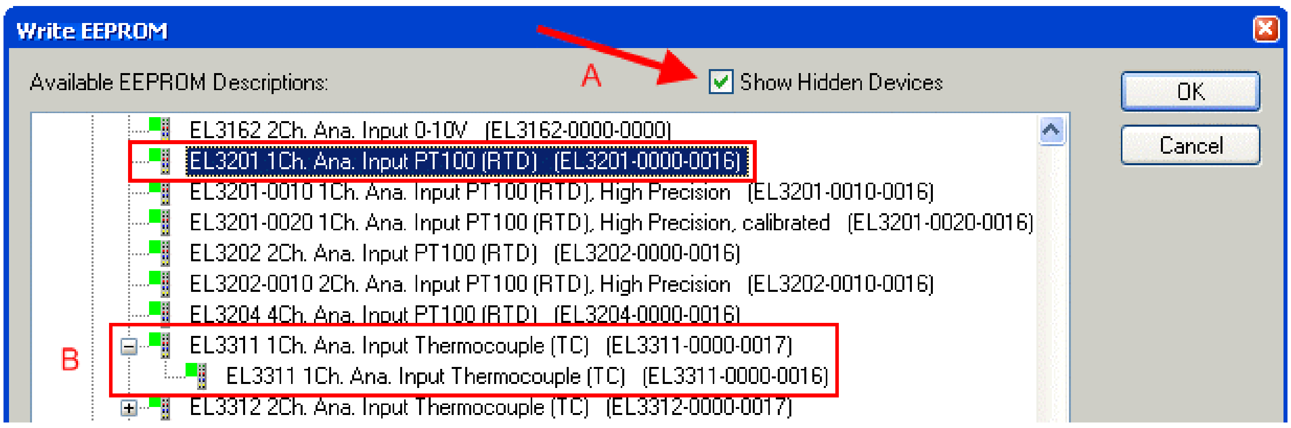
- Trouble-free EtherCAT communication must be established with the slave.
- The state of the slave is irrelevant.
- Right-clicking on the slave in the online display opens the *EEPROM Update* dialog, Fig. *EEPROM Update*



No	Addr	Name	State	CRC
1	1001	Term 1 (EK1101)	PREOP	0, 0
2	1002	Term 2 (EL3204)	PREOP	0, 0
3	1003	Term 3 (EL3201)	INIT VPRS	0

Fig. 103: 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.



EL3162 2Ch. Ana. Input 0-10V (EL3162-0000-0000)
EL3201 1Ch. Ana. Input PT100 (RTD) (EL3201-0000-0016)
EL3201-0010 1Ch. Ana. Input PT100 (RTD), High Precision (EL3201-0010-0016)
EL3201-0020 1Ch. Ana. Input PT100 (RTD), High Precision, calibrated (EL3201-0020-0016)
EL3202 2Ch. Ana. Input PT100 (RTD) (EL3202-0000-0016)
EL3202-0010 2Ch. Ana. Input PT100 (RTD), High Precision (EL3202-0010-0016)
EL3204 4Ch. Ana. Input PT100 (RTD) (EL3204-0000-0016)
EL3311 1Ch. Ana. Input Thermocouple (TC) (EL3311-0000-0017)
EL3311 1Ch. Ana. Input Thermocouple (TC) (EL3311-0000-0016)
EL3312 2Ch. Ana. Input Thermocouple (TC) (EL3312-0000-0017)

Fig. 104: 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.

8.3.2 Firmware explanation

Determining the firmware version

Determining the version via the TwinCAT System Manager

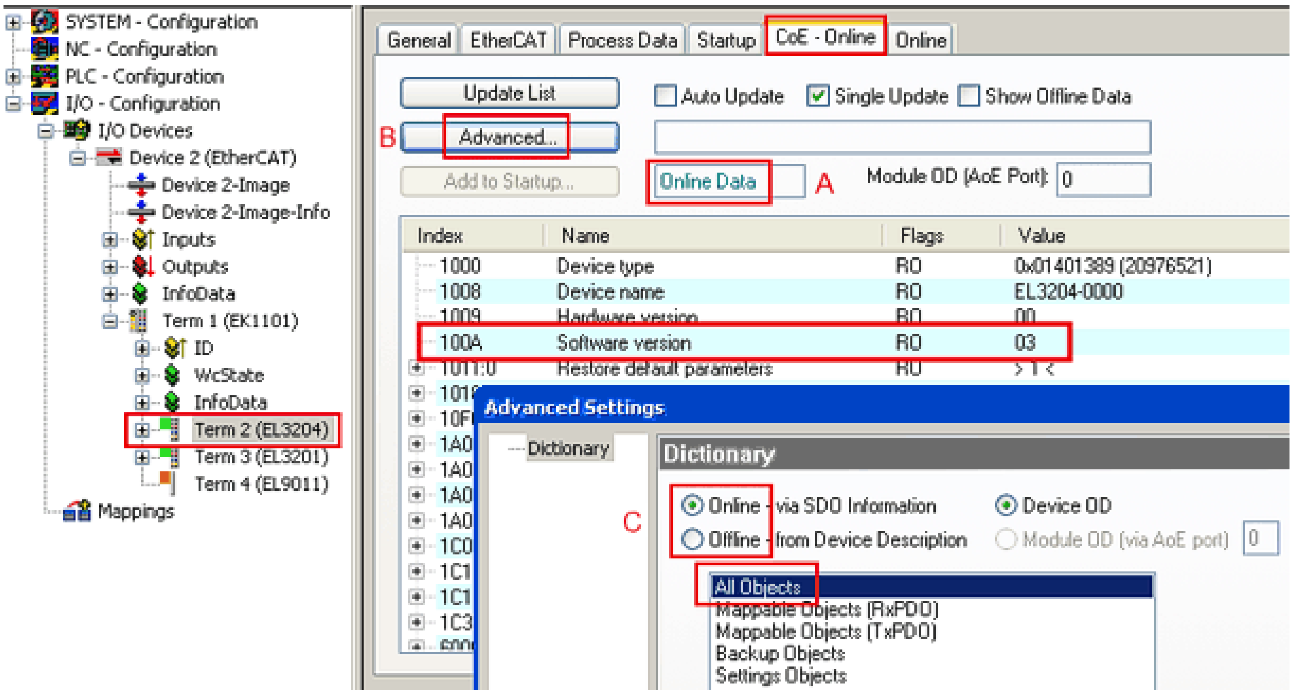
The TwinCAT System Manager shows the version of the controller firmware if the master can access the slave online. Click on the E-Bus Terminal whose controller firmware you want to check (in the example terminal 2 (EL3204)) and select the tab *CoE Online* (CAN over EtherCAT).

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.



[✓ Single Update]

Index	Name	Flags	Value
1000	Device type	RO	0x01401389 (20976521)
1008	Device name	RO	EL3204-0000
1009	Hardware version	RO	00
100A	Software version	RO	03
1011:0	Restore default parameters	RO	> 1 <

[✓ Online - via SDO Information] Offline - from Device Description

[✓ Device OD] Module OD (via AoE port) 0

Fig. 105: Display of EL3204 firmware version

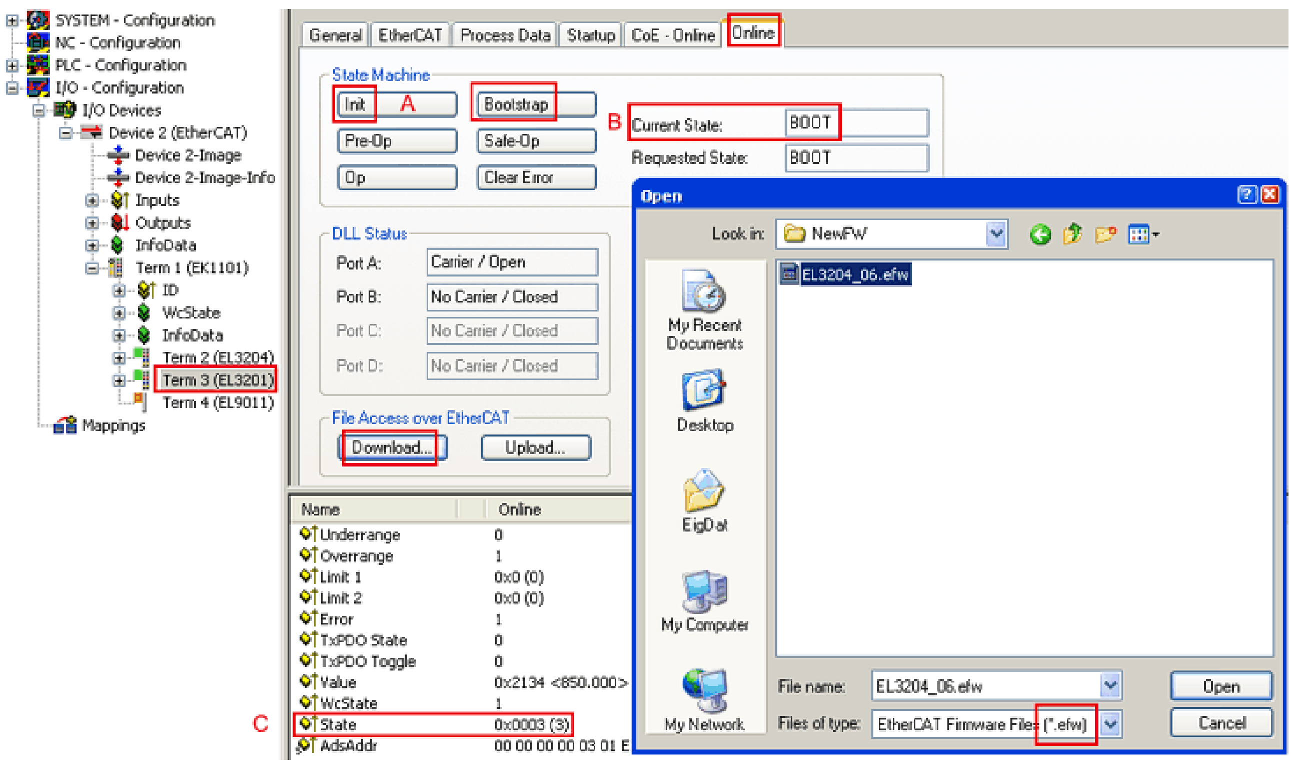
In (A) TwinCAT 2.11 shows that the Online CoE directory is currently displayed. If this is not the case, the Online directory can be loaded via the *Online* option in Advanced Settings (B) and double-clicking on *AllObjects*.

8.3.3 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.*



Current State: BOOT
Requested State: BOOT

Port	DLL Status
Port A:	Carrier / Open
Port B:	No Carrier / Closed
Port C:	No Carrier / Closed
Port D:	No Carrier / Closed

Name	Online
Underrange	0
Overrange	1
Limit 1	0x0 (0)
Limit 2	0x0 (0)
Error	1
TxPDO State	0
TxPDO Toggle	0
Value	0x2134 <850.000>
WcState	1
State	0x0003 (3)
AdsAddr	00 00 00 00 03 01 E

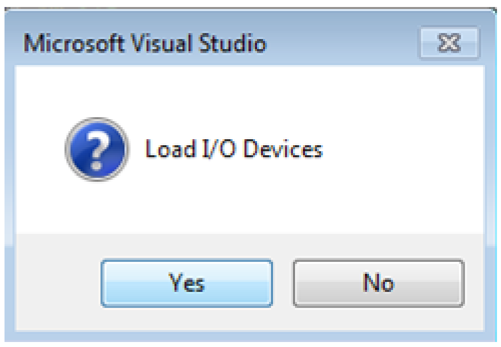
EL3204_06.efw

File name: EL3204_06.efw
Files of type: EtherCAT Firmware Files (*.efw)

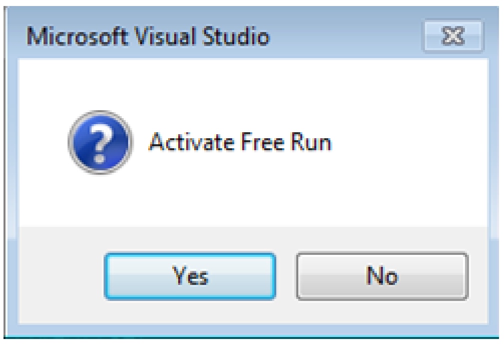
Fig. 106: Firmware Update

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.

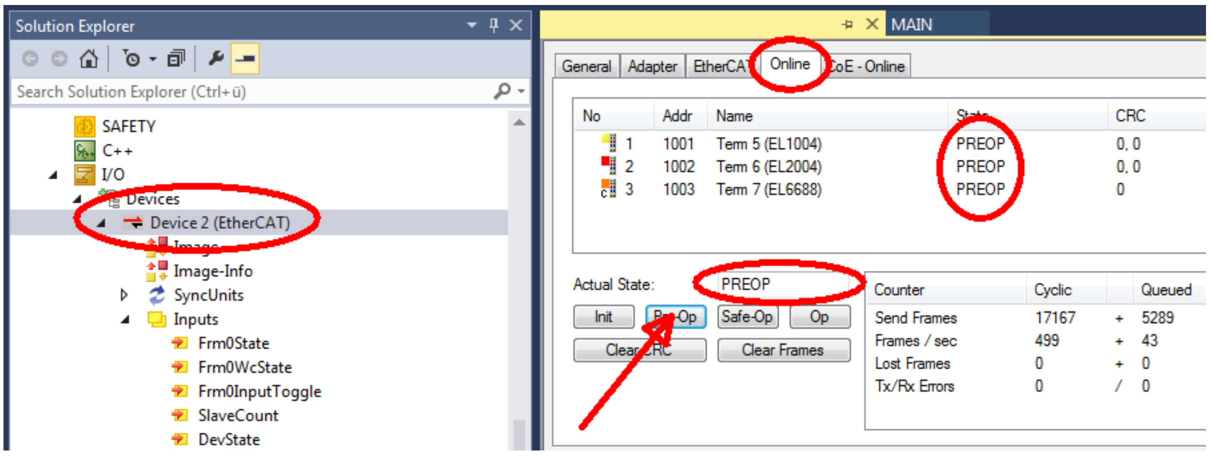


Load I/O Devices



Activate Free Run

- Switch EtherCAT Master to PreOP



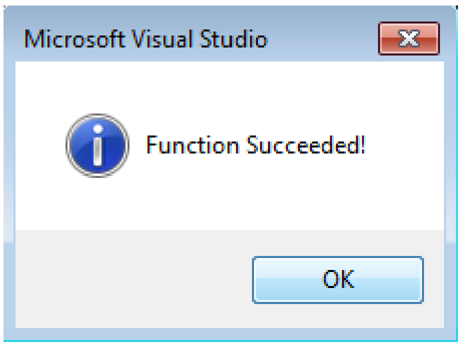
No	Addr	Name	State	CRC
1	1001	Term 5 (EL1004)	PREOP	0, 0
2	1002	Term 6 (EL2004)	PREOP	0, 0
3	1003	Term 7 (EL6688)	PREOP	0

Actual State: PREOP

Counter	Cyclic		Queued
Send Frames	17167	+	5289
Frames / sec	499	+	43
Lost Frames	0	+	0
Tx/Rx Errors	0	/	0

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

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


Microsoft Visual Studio

Function Succeeded!

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

8.3.4 FPGA firmware *.rbf

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

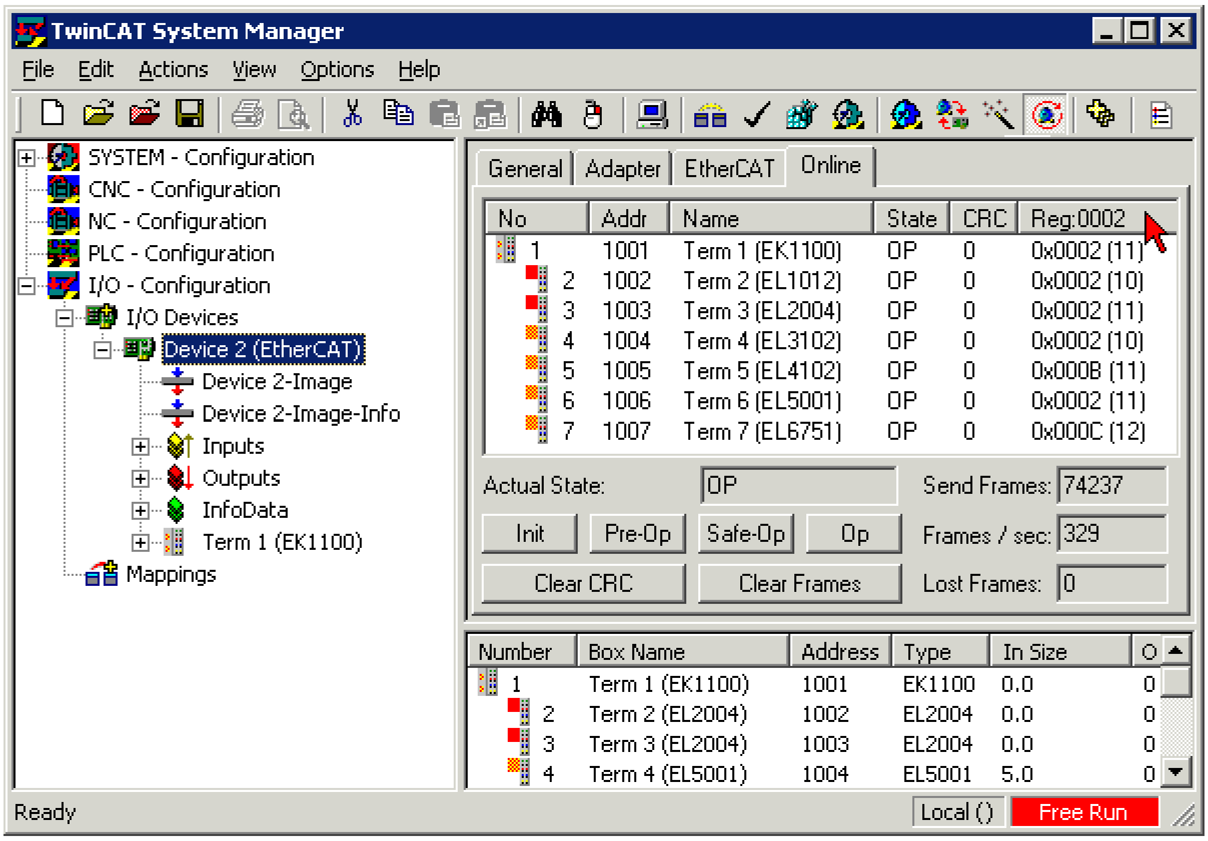
- Controller firmware for processing I/O signals
- FPGA firmware for EtherCAT communication (only for terminals with FPGA)

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 TwinCAT System Manager

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

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



No	Addr	Name	State	CRC	Reg:0002
1	1001	Term 1 (EK1100)	OP	0	0x0002 (11)
2	1002	Term 2 (EL1012)	OP	0	0x0002 (10)
3	1003	Term 3 (EL2004)	OP	0	0x0002 (11)
4	1004	Term 4 (EL3102)	OP	0	0x0002 (10)
5	1005	Term 5 (EL4102)	OP	0	0x000B (11)
6	1006	Term 6 (EL5001)	OP	0	0x0002 (11)
7	1007	Term 7 (EL6751)	OP	0	0x000C (12)

Actual State: OP
Send Frames: 74237
Frames / sec: 329
Lost Frames: 0

Number	Box Name	Address	Type	In Size	O
1	Term 1 (EK1100)	1001	EK1100	0.0	0
2	Term 2 (EL2004)	1002	EL2004	0.0	0
3	Term 3 (EL2004)	1003	EL2004	0.0	0
4	Term 4 (EL5001)	1004	EL5001	5.0	0

Fig. 107: FPGA firmware version definition

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

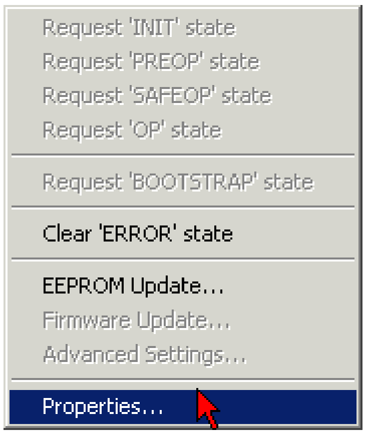


Fig. 108: Context menu *Properties*

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.

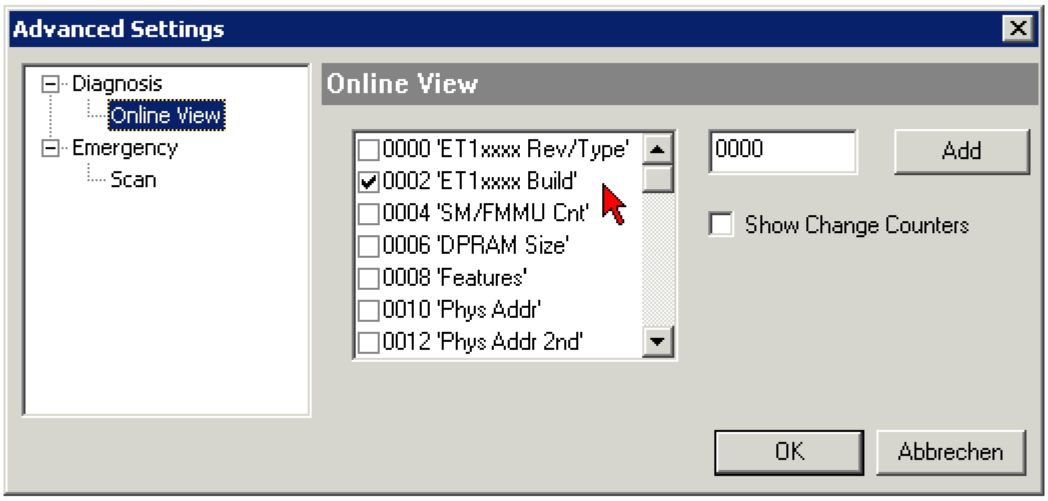


Fig. 109: Dialog *Advanced Settings*

Update

For updating the FPGA firmware

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

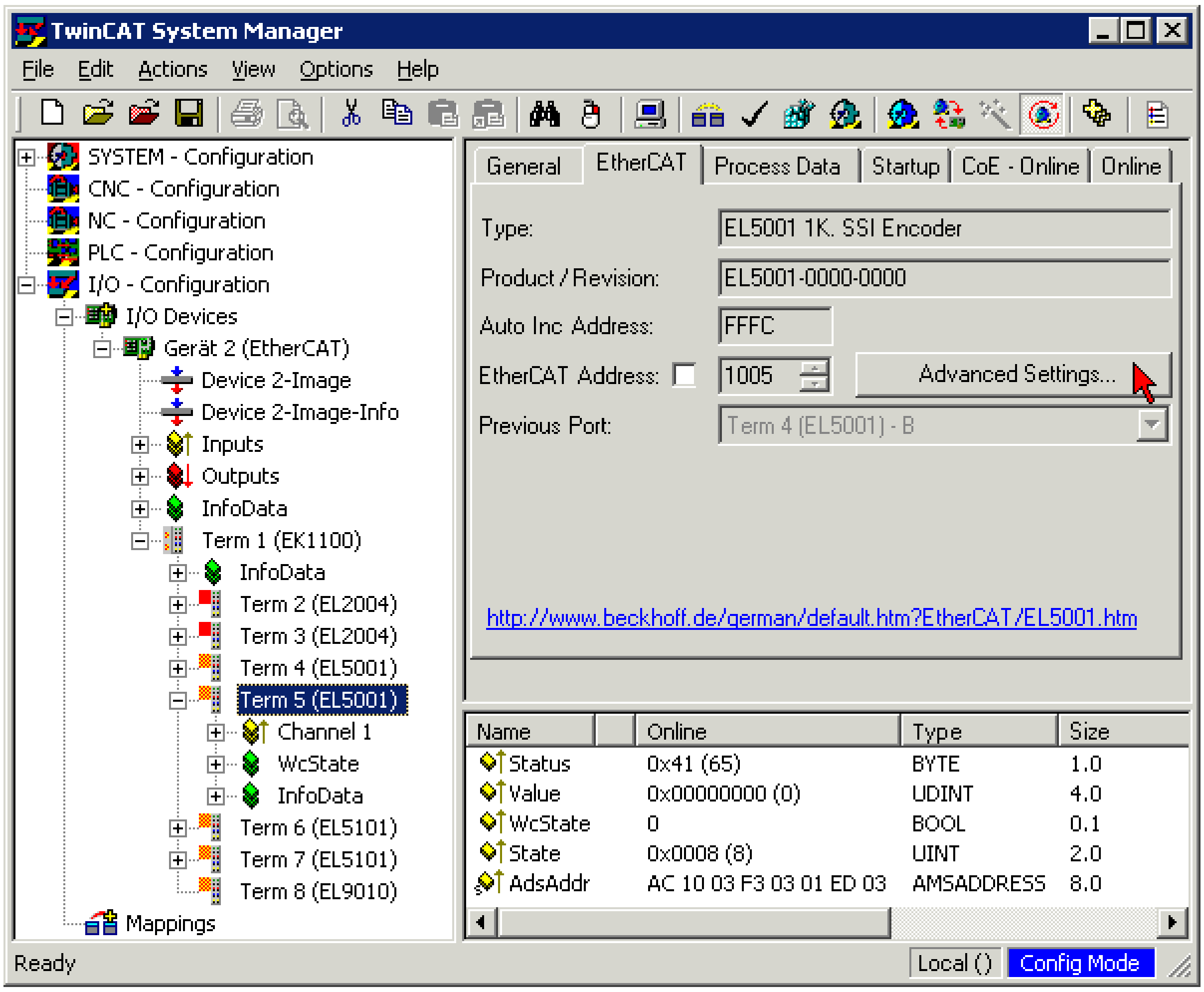
Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

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

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

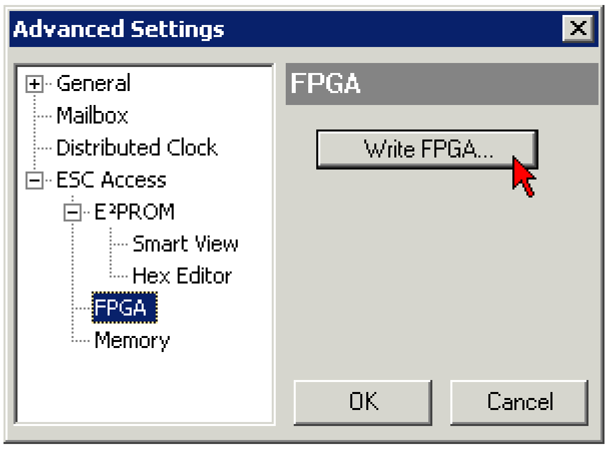
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and
click the *Advanced Settings* button in the *EtherCAT* tab:



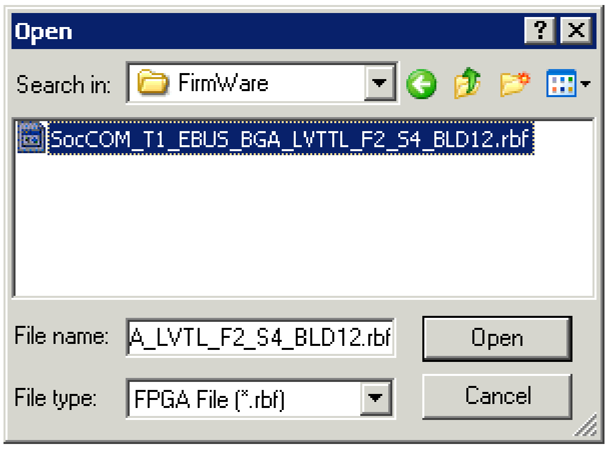
Type: EL5001 1K. SSI Encoder
Product / Revision: EL5001-0000-0000
Auto Inc Address: FFFC
EtherCAT Address: 1005
Previous Port: Term 4 (EL5001) - B
http://www.beckhoff.de/german/default.htm?EtherCAT/EL5001.htm

Name	Online	Type	Size
Status	0x41 (65)	BYTE	1.0
Value	0x00000000 (0)	UDINT	4.0
WcState	0	BOOL	0.1
State	0x0008 (8)	UINT	2.0
AdsAddr	AC 10 03 F3 03 01 ED 03	AMSADDRESS	8.0

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



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


Open
Search in: FirmWare
SocCOM_T1_EBUS_BGA_LVTTL_F2_S4_BLD12.rbf
File name: A_LVTL_F2_S4_BLD12.rbf
File type: FPGA File (*.rbf)
Open
Cancel

- 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

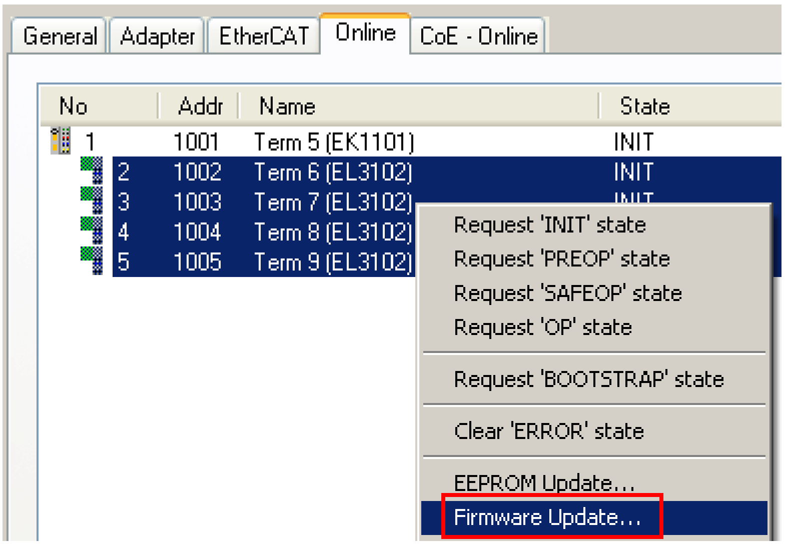
NOTICE

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!

8.3.5 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.



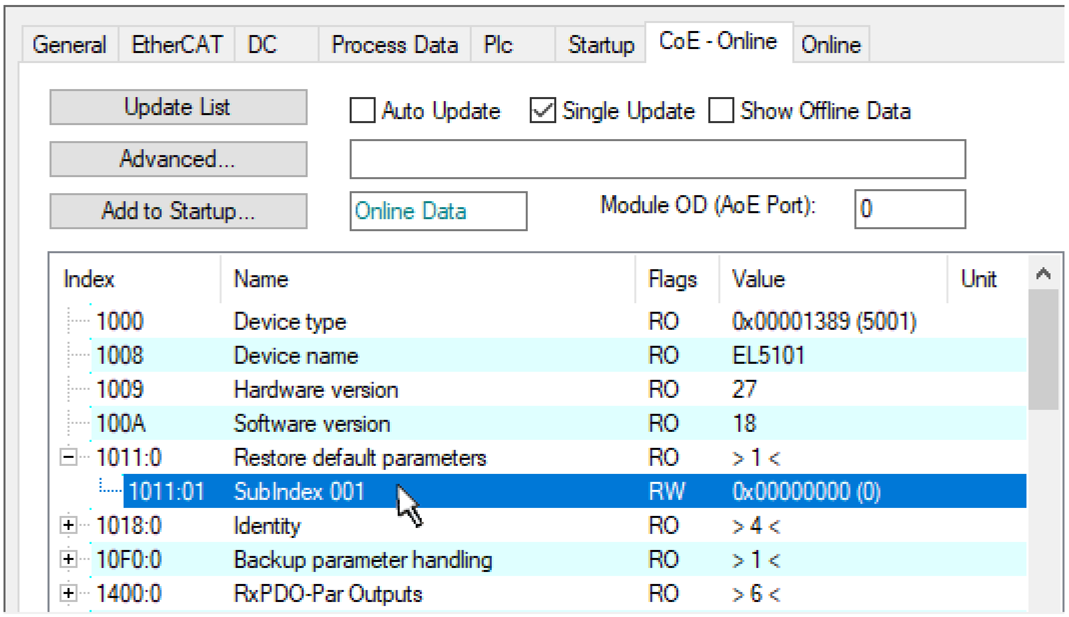
No	Addr	Name	State
1	1001	Term 5 (EK1101)	INIT
2	1002	Term 6 (EL3102)	INIT
3	1003	Term 7 (EL3102)	INIT
4	1004	Term 8 (EL3102)	
5	1005	Term 9 (EL3102)	

Fig. 110: Multiple selection and firmware update

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

8.4 Restoring the delivery state

To restore the delivery state (factory settings) of CoE objects for EtherCAT devices (“slaves”), the CoE object *Restore default parameters*, SubIndex 001 can be used via EtherCAT master (e.g. TwinCAT) (see Fig. *Selecting the Restore default parameters PDO*).



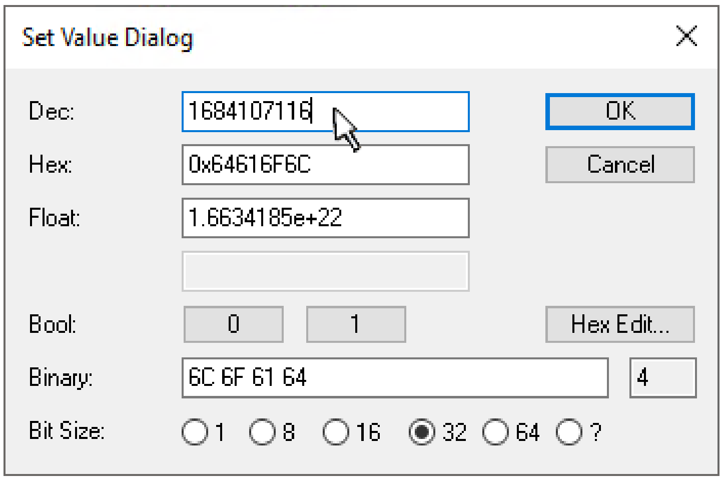
[ ] Auto Update [✓ Single Update] [ ] Show Offline Data

Online Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
1000	Device type	RO	0x00001389 (5001)	
1008	Device name	RO	EL5101	
1009	Hardware version	RO	27	
100A	Software version	RO	18	
1011:0	Restore default parameters	RO	> 1 <	
1011:01	SubIndex 001	RW	0x00000000 (0)	
1018:0	Identity	RO	> 4 <	
10F0:0	Backup parameter handling	RO	> 1 <	
1400:0	RxPDO-Par Outputs	RO	> 6 <	

Fig. 111: Selecting the *Restore default parameters* PDO



Dec:	1684107116
Hex:	0x64616F6C
Float:	1.6634185e+22
Bool:	0 1
Binary:	6C 6F 61 64 4
Bit Size:	1 8 16 [✓ 32] 64 ?

Fig. 112: Entering a restore value in the Set Value dialog

Double-click on *SubIndex 001* to enter the Set Value dialog. Enter the reset value **1684107116** in field *Dec* or the value **0x64616F6C** in field *Hex* (ASCII: “load”) and confirm with *OK* (Fig. *Entering a restore value in the Set Value dialog*).

- All changeable entries in the slave are reset to the default values.
- The values can only be successfully restored if the reset is directly applied to the online CoE, i.e. to the slave. No values can be changed in the offline CoE.
- TwinCAT must be in the RUN or CONFIG/Freerun state for this; that means EtherCAT data exchange takes place. Ensure error-free EtherCAT transmission.
- No separate confirmation takes place due to the reset. A changeable object can be manipulated beforehand for the purposes of checking.
- This reset procedure can also be adopted as the first entry in the startup list of the slave, e.g. in the state transition PREOP->SAFEOP or, as in Fig. *CoE reset as a startup entry*, in SAFEOP->OP.

All backup objects are reset to the delivery state.

Alternative restore value

In some older terminals (FW creation approx. before 2007) the backup objects can be switched with an alternative restore value: Decimal value: 1819238756, Hexadecimal value: 0x6C6F6164.

An incorrect entry for the restore value has no effect.

8.5 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/EL6821

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