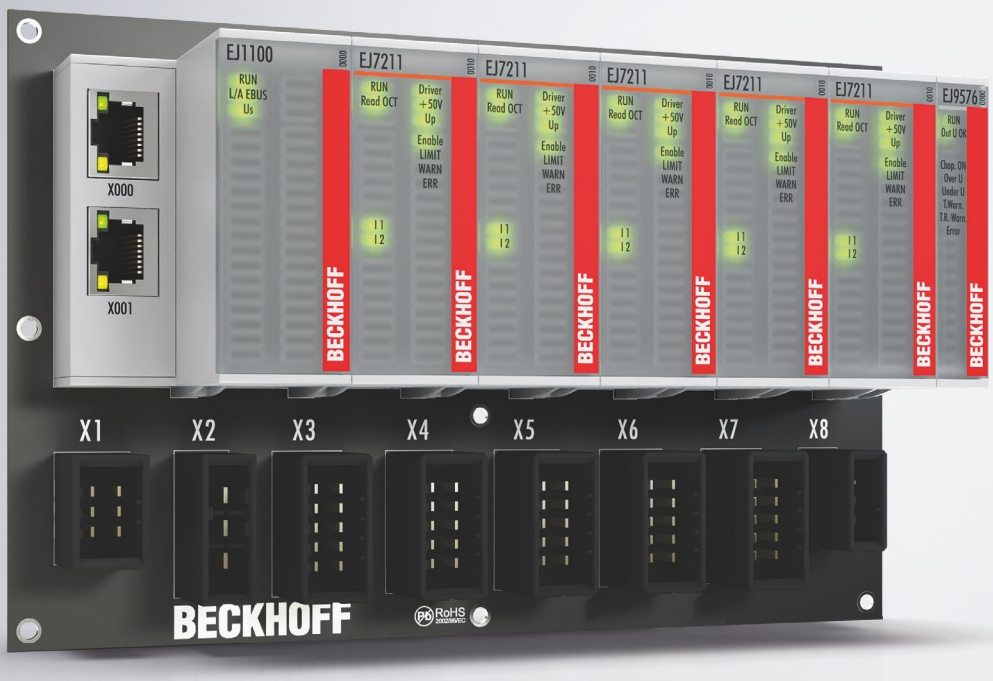
BECKHOFF New Automation Technology

Documentation | EN

EJ6080

Communikation interface, memory, 128 kByte, NOVRAM



2023-01-12 | Version: 1.0

Table of contents

1 Foreword

1.1 Notes on the documentation

Intended audience

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

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

Disclaimer

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

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

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

Trademarks

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

Patent Pending

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

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

Copyright

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

1.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Description of instructions

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

⚠ DANGER

Serious risk of injury!

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

⚠ WARNING

Risk of injury!

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

⚠ CAUTION

Personal injuries!

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

NOTE

Damage to environment/equipment or data loss

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

Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3 Intended use

⚠ WARNING

Caution - Risk of injury!

EJ components may only be used for the purposes described below!

1.4 Signal distribution board

NOTE

Signal distribution board

Make sure that the EtherCAT plug-in modules are used only on a signal distribution board that has been developed and manufactured in accordance with the Design Guide.

1.5 Documentation issue status

Version	Comment
1.0	• First publication EJ6080

1.6 Guide through documentation

NOTE

Further components of documentation

The documentations named in the following table are further components of the complete documentation. These documentations are required for the use of EtherCAT plug-in modules.

No.	Title	Description
[1]	EtherCAT System Documentation	• System overview • EtherCAT basics • Cable redundancy • Hot Connect • Distributed Clocks • Configuration of EtherCAT-Components
[2]	Infrastructure for EtherCAT/Ethernet	• Technical recommendations and notes for design, implementation an testing
[3]	Design Guide EJ8xxx - Signal distribution board for standard EtherCAT plug-in modules	Requirements for the design of a Signal-Distribution-Board for standard EtherCAT plug-in modules • Backplane mounting guidelines • Module placement • Routing guidelines
[4]	Documentation of the corresponding EtherCAT Terminal ELxxxx	• Notes on the principle of operation and • Descriptions for configuration and parameterization are transferable to the corresponding Module EJxxxx (s. note on documentation of ELxxxx).

1.7 Marking of EtherCAT plug-in modules

Designation

A Beckhoff EtherCAT device has a 14-digit **technical designation**, made up as follows (e.g. EJ1008-0000-0017)

- **Order identifier**
  - family key: EJ
  - product designation: The first digit of product designation is used for assignment to a product group (e.g. EJ2xxx = digital output module).
  - Version number: The four digit version number identifies different product variants.
- **Revision number:**
  It is incremented when changes are made to the product.

The Order identifier and the revision number are printed on the side of EtherCAT plug-in modules (s. following illustration (A and B).

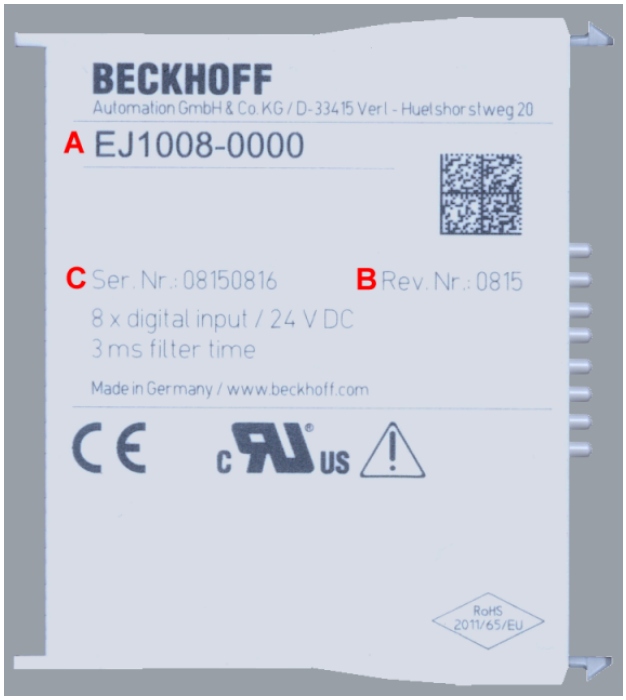


Fig. 1: Order identifier (A), Revision number (B) and serial number (C) using the example of EJ1008

Product group	Example		
	Product designation	Version	Revision
EtherCAT Coupler EJ11xx	EJ1101	-0022 (Coupler with external connectors, power supply module and optional ID switches	-0016
Digital input modules EJ1xxx	EJ1008 8-channel	-0000 (basic type)	-0017
Digital output modules EJ2xxx	EJ2521 1-channel	-0224 (2 x 24 V outputs)	-0016
Analog input modules EJ3xxx	EJ3318 8-channel thermocouple	-0000 (basic type)	-0017
Analog output modules EJ4xxx	EJ4134 4-channel	-0000 (basic type)	-0019
Special function modules EJ5xxx, EJ6xxx	EJ6224 IO-Link master	-0090 (with TwinSAFE SC)	-0016
Motion modules EJ7xxx	EJ7211 servomotor	-9414 (with ECT, STO and TwinSAFE SC)	-0029

Notes

- The elements mentioned above result in the **technical designation**. EJ1008-0000-0017 is used in the example below.
- EJ1008-0000 is the **order identifier**, in the case of “-0000” usually abbreviated to EJ1008.
- The **revision** -0017 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.
- The product designation, version and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

Serial number

The serial number for EtherCAT plug-in modules is usually the 8-digit number printed on the side of the module (see following illustration C). 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.

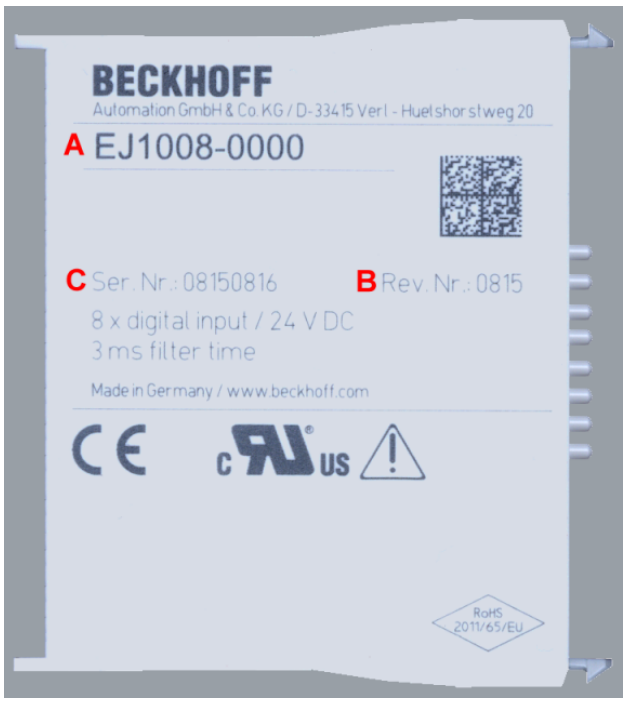


Fig. 2: Order identifier (A), revision number (B) and serial number (C) using the example of EJ1008

Serial number	Example serial number: 08 15 08 16
KK - week of production (CW, calendar week)	08 - week of production: 08
YY - year of production	15 - year of production: 2015
FF - firmware version	08 -f irmware version: 08
HH - hardware version	16 - hardware version: 16

1.7.1 Beckhoff Identification Code (BIC)

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

Fig. 3: 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, it shall be replaced by spaces. The data under positions 1-4 are always available.

The following information is contained:

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

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

Structure of the BIC

Example of composite information from items 1 - 4 and with the above given example value on positon 6. The data identifiers are marked in bold font for better display:

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

Accordingly as DMC:

Fig. 4: Example DMC **1P**072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

BTN

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

NOTE

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

1.7.2 Electronic access to the BIC (eBIC)

Electronic BIC (eBIC)

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

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

K-bus devices (IP20, IP67)

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

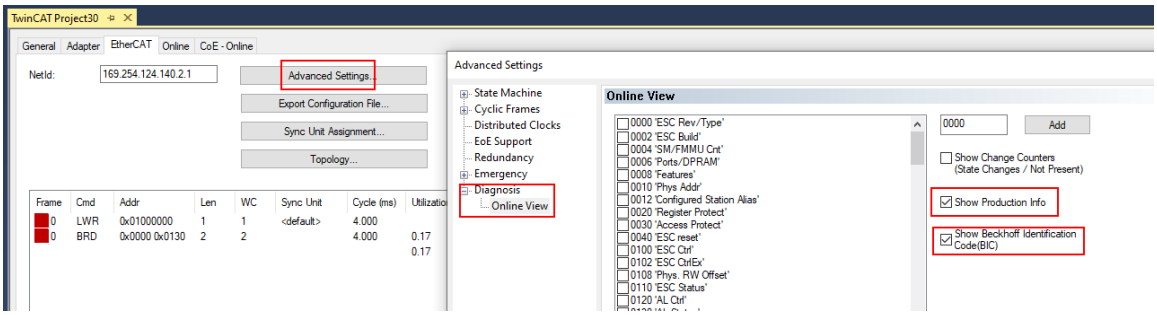
EtherCAT devices (IP20, IP67)

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

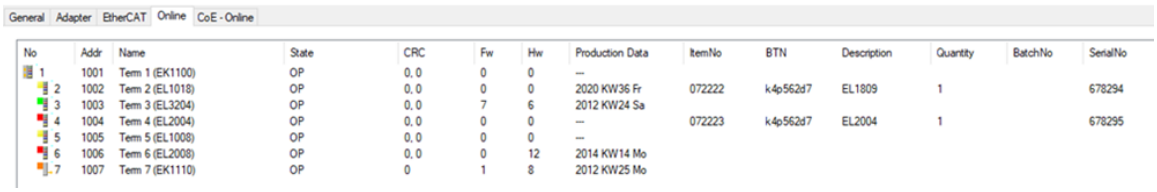
The eBIC is also stored in the ESI-EEPROM. The eBIC was introduced into the Beckhoff I/O production (terminals, box modules) from 2020; widespread implementation is expected in 2021.

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

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



  - The BTN and its contents are then displayed:



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

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

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

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

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

Profibus/Profinet/DeviceNet... Devices

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

1.7.3 Certificates

- The EhterCAT plug-in modules meet the requirements of the EMC and Low Voltage Directive. The CE mark is printed on the side of the modules.
- The cRUus imprint identifies devices that meet product safety requirements according to U.S. and Canadian regulations.
- The warning symbol is a request to read the corresponding documentation. The documentations for EtherCAT plug-in modules can be downloaded from the Beckhoff homepage.

Fig. 5: Marking for CE and UL using EJ1008 as an example

2 System overview

Electronically, the EJxxxx EtherCAT plug-in modules are based on the EtherCAT I/O system. The EJ system consists of the signal distribution board and EtherCAT plug-in modules. It is also possible to connect an IPC to the EJ system.
The EJ system is suitable for mass production applications, applications with small footprint and applications requiring a low total weight.
The machine complexity can be extended by means of the following:

- reserve slots,
- the use of placeholder modules,
- linking of EtherCAT Terminals and EtherCAT Boxes via an EtherCAT connection.

The following diagram illustrates an EJ system. The components shown are schematic, to illustrate the functionality.

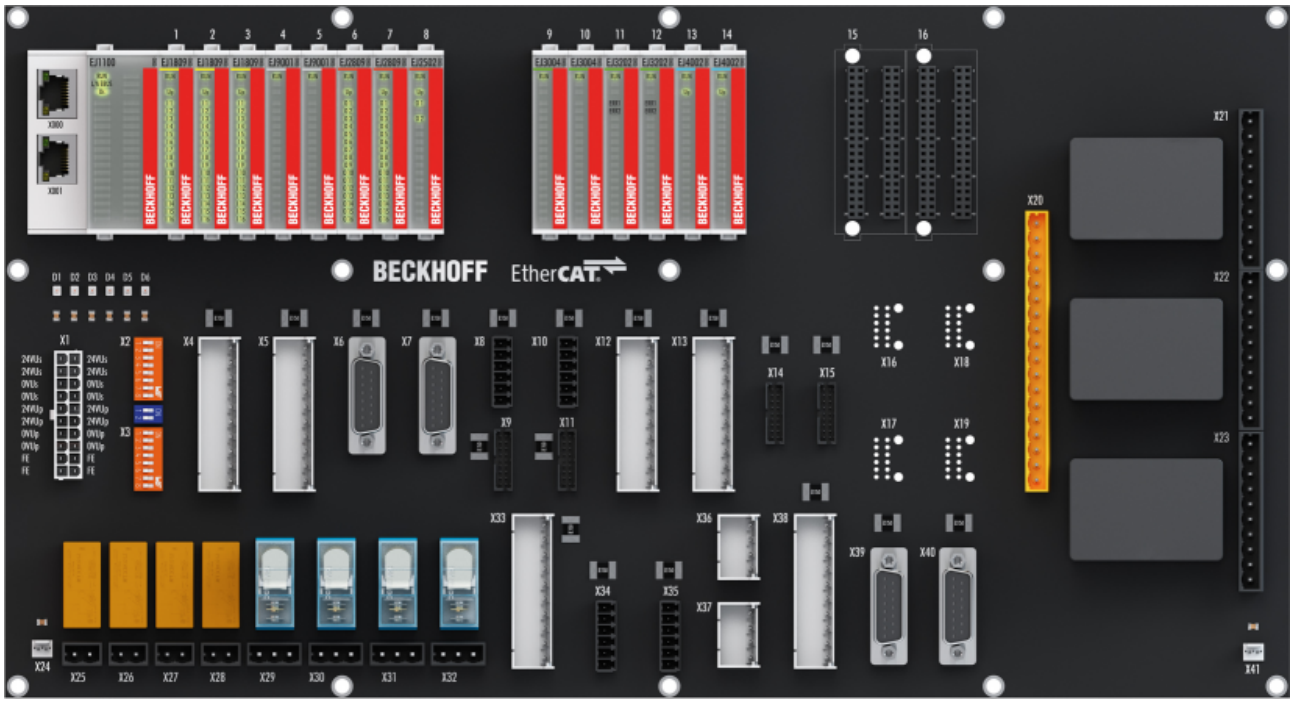


Fig. 6: EJ system sample

Signal distribution board

The signal distribution board distributes the signals and the power supply to individual application-specific plug connectors, in order to connect the controller to further machine modules. Using pre-assembled cable harnesses avoids the need for time-consuming connection of individual wires. Coded components reduce the unit costs and the risk of miswiring.
Beckhoff offers development of signal distribution boards as an engineering service. Customers have the option to develop their own signal distribution board, based on the design guide.

EtherCAT plug-in modules

Similar to the EtherCAT Terminal system, a module strand consists of a bus coupler and I/O modules. Almost all of the EtherCAT Terminals can also be manufactured in the EJ design as EtherCAT plug-in modules. The EJ modules are directly attached to the signal distribution board. The communication, signal distribution and supply take place via the contact pins at the rear of the modules and the PCB tracks of the signal distribution board. The coding pins at the rear serve as mechanical protection against incorrect connection. Color coding on the housing facilitates distinguishing of the modules.

3 EJ6080 - Product description

3.1 Introduction

Fig. 7: EJ6080

EtherCAT memory module 128 kbyte

The EtherCAT memory module has 128 kbyte of non-volatile memory (NOVRAM, non-volatile RAM). The module can be used for saving and reading parameters and recipes, and machine data such as operating hours counter or production counts. The module is used, for example, for storing module-related data in the machine module in modular machine concepts with a central controller.

Data are only stored in the RAM in the live module and would therefore not be stored permanently. If the event of a power failure, an internal buffer supplies the NOVRAM block until the entire contents of the RAM have been stored in a non-volatile memory.

The EJ6080 supports two methods of accessing the memory:

- with cyclic process data and
- via acyclic SDO/CoE access

The access time in both cases is dependent on the amount of data.

3.2 Technical data

Technical data	EJ6080
Total memory	128 kbyte (=131,072 byte) NOVRAM (non-volatile RAM)
Max. available storage space, cyclic access	1280 byte
Max. available storage space, acyclic access	up to 128 kByte in objects with a size of up to 8190 bytes, depending on variable structure, see chapter Calculation of the storage space [▶ 43]
Distributed Clocks (DC)	no
Current consumption via E-bus	typ. 120 mA
Configuration	with TwinCAT System Manager
Permissible ambient temperature range during operation	0 °C... + 55 °C
Permissible ambient temperature range during storage	-25 °C... + 85 °C
Permissible relative air humidity	95 %, no condensation
Operating altitude	max. 2,000 m
Dimensions (W x H x D)	approx. 12 mm x 66 mm x 55 mm
Weight	approx. 30 g
Mounting	on signal distribution board
Installation position	Standard [▶ 25]
Position of the coding pins [▶ 28]	2 and 5
Color coding	gray
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27 (with corresponding signal distribution board)
EMC immunity/emission	conforms to EN 61000-6-2 /EN 61000-6-4 (with corresponding signal distribution board)
Protection rating	EJ module: IP20 EJ system: dependent on the signal distribution board
Identification / approval*	CE, UKCA

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

CE approval

The CE Marking refers to the EtherCAT plug-in module mentioned above.
If the EtherCAT plug-in module is used in the production of a ready-to-use end product (PCB in conjunction with a housing), the manufacturer of the end product must check compliance of the overall system with relevant directives and CE certification.
To operate the EtherCAT plug-in modules, they must be installed in a housing.

3.3 Connection

EJ6080				
Pin#		Signal		
1	2	U_{EBUS}	U_{EBUS}	**E-Bus contacts** The power supply U_{EBUS} is provided by the coupler and supplied from the supply voltage U_S of the EtherCAT coupler.
3	4	GND	GND	
5	6	RX0+	TX1+	
7	8	RX0-	TX1-	
9	10	GND	GND	
11	12	TX0+	RX1+	
13	14	TX0-	RX1-	
15	16	GND	GND	
17	18	NC	NC	**Signals**
19	20	NC	NC	
21	22	NC	NC	
23	24	NC	NC	
25	26	NC	NC	
27	28	NC	NC	
29	30	NC	NC	
31	32	NC	NC	
33	34	NC	NC	**U_P-Contacts** The device has no U_P-contacts. The power is supplied exclusively via U_{EBUS}.
35	36	NC	NC	
37	38	NC	NC	
39	40	SGND	SGND	

Signal	Description
U_{EBUS}	E-Bus power supply 3.3 V
GND	E-Bus GND signal. Don`t connect with 0V Up!
RXn+	Positive E-Bus receive signal
RXn-	Negative E-Bus receive signal
TXn+	Positive E-Bus transmit signal
TXn-	Negative E-Bus transmit signal
NC	Do not connect
SGND	Shield Ground

Fig. 8: EJ6080 - Pinout

The PCB footprint can be downloaded from the Beckhoff homepage.

NOTE

Damage to devices possible!

- The pins named with “NC” must not be connected.
- Before installation and commissioning read the chapters Installation of EJ modules [▶ 21] and Commissioning [▶ 37]!

3.4 LEDs

LED No.	EJ6080
A	RUN
B	
C	CPUErr
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	

Fig. 9: EJ6080 - LEDs

LED	Color	Display	State	Description
RUN	green	off	Init	State of the EtherCAT State Machine: **INIT** = initialization of the plug-in module
		flashing	Pre-Operational	State of the EtherCAT State Machine: **PREOP** = function for mailbox communication and different standard-settings set
		Single flash	Safe-Operational	State of the EtherCAT State Machine: **SAFEOP** = verification of the Sync Manager channels and the distributed clocks. Outputs remain in safe state
		on	Operational	State of the EtherCAT State Machine: **OP** = normal operating state; mailbox and process data communication is possible
		flickering	Bootstrap	State of the EtherCAT State Machine: **BOOTSTRAP** = function for firmware updates of the plug-in module
CPUErr	red	Fault condition		

4 Installation of EJ modules

4.1 Power supply for the EtherCAT plug-in modules

⚠ WARNING

Power supply from SELV/PELV power supply unit!

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

Notes:

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

The signal distribution board should have a power supply designed for the maximum possible current load of the module string. Information on the current required from the E-bus supply can be found for each module in the respective documentation in section “Technical data”, online and in the catalog. The power requirement of the module string is displayed in the TwinCAT System Manager.

E-bus power supply with EJ1100 or EJ1101-0022 and EJ940x

The EJ1100 Bus Coupler supplies the connected EJ modules with the E-bus system voltage of 3.3 V. The Coupler can accommodate a load up to 2.2 A. If a higher current is required, a combination of the coupler EJ1101-0022 and the power supply units EJ9400 (2.5 A) or EJ9404 (12 A) should be used. The EJ940x power supply units can be used as additional supply modules in the module string.

Depending on the application, the following combinations for the E-bus supply are available:

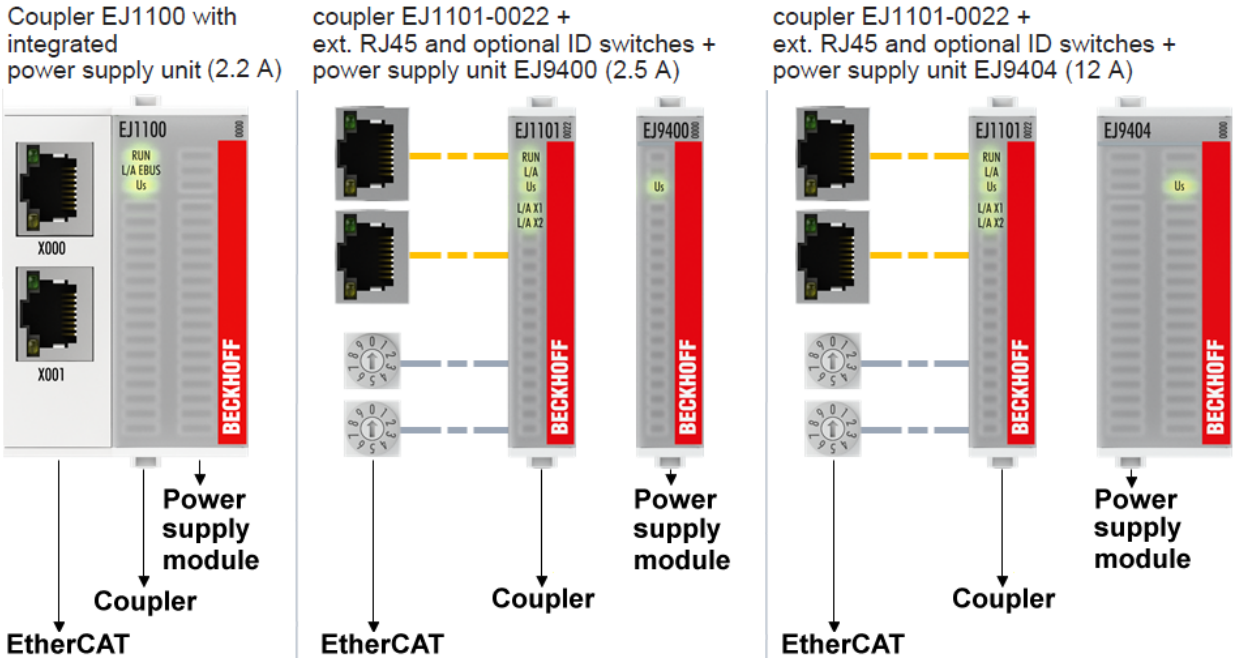


Fig. 10: E-bus power supply with EJ1100 or EJ1101-0022 + EJ940x

In the EJ1101-0022 coupler, the RJ45 connectors and optional ID switches are external and can be positioned anywhere on the signal distribution board, as required. This facilitates feeding through a housing.

The EJ940x power supply plug-in modules provide an optional reset function (see chapter Connection of the documentation for EJ9400 and EJ9404)

E-bus power supply with CXxxxx and EK1110-004x

The Embedded PC supplies the attached EtherCAT Terminals and the EtherCAT EJ coupler

- with a supply voltage Us of 24 V_{DC} (-15 %/+20%). This voltage supplies the E-bus and the bus terminal electronics.
  The CXxxxx units supply the E-bus with up to 2,000 mA E-bus current. If a higher current is required due to the attached terminals, power feed terminals or power supply plug-in modules must be used for the E-bus supply.
- with a peripheral voltage Up of 24 V_{DC} to supply the field electronics.

The EK1110-004x EtherCAT EJ couplers relay the following parameters to the signal distribution board via the rear connector:

- the E-bus signals,
- the E-bus voltage U_{EBUS} (3.3 V) and
- the peripheral voltage U_P (24 V_{DC}).

Fig. 11: PCB with Embedded PC, EK1110-0043 and EJxxxx, rear view EK1110-0043

4.2 EJxxxx - dimensions

The EJ modules are compact and lightweight thanks to their design. Their volume is approx. 50% smaller than the volume of the EL terminals. A distinction is made between four different module types, depending on the width and the height:

Module type	Dimensions (W x H x D)	Sample in figure below
Coupler	44 mm x 66 mm x 55 mm	EJ1100 (ej_44_2xrj45_coupler)
Single module	12 mm x 66 mm x 55 mm	EJ1809 (ej_12_16pin_code13)
Double module	24 mm x 66 mm x 55 mm	EJ7342 (ej_24_2x16pin_code18)
Single module (long)	12 mm x 152 mm x 55 mm	EJ1957 (ej_12_2x16pin_extended_code4747)

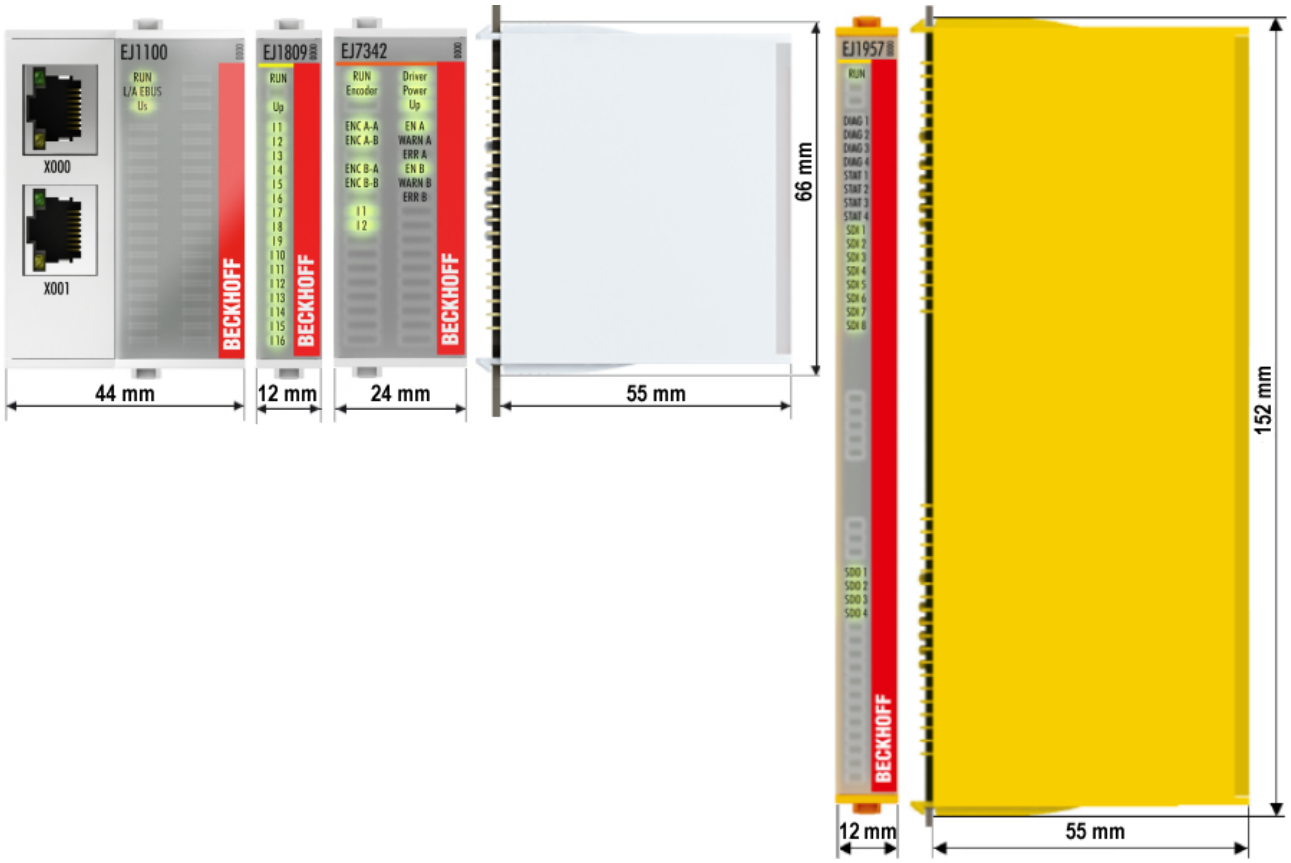


Fig. 12: EJxxxx - Dimensions

The technical drawings can be downloaded from the Beckhoff homepage. The drawings are named as described in the drawing below.

Fig. 13: Naming of the technical drawings

4.3 Installation positions and minimum distances

4.3.1 Minimum distances for ensuring installability

Note the dimensions shown in the following diagram for the design of the signal distribution board to ensure safe latching and simple assembly / disassembly of the modules.

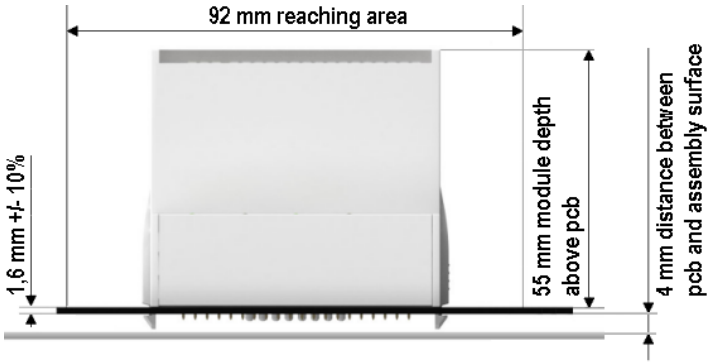


Fig. 14: Mounting distances EJ module - PCB

Observing the reaching area

A minimum reaching area of 92 mm is required for assembly / disassembly, in order to be able to reach the mounting tabs with the fingers.
Adherence to the recommended minimum distances for ventilation (see section Installation position [▶ 25]) ensures an adequate reaching area.

The signal distribution board must have a thickness of 1.6 mm and a minimum distance of 4 mm from the mounting surface, in order to ensure latching of the modules on the board.

4.3.2 Installation positions

NOTE

Constraints regarding installation position and operating temperature range

Please refer to the technical data [▶ 18] for the installed components to ascertain whether any restrictions regarding the mounting position and/or the operating temperature range have been specified. During installation of modules with increased thermal dissipation, ensure adequate distance above and below the modules to other components in order to ensure adequate ventilation of the modules during operation!

The standard installation position is recommended. If a different installation position is used, check whether additional ventilation measures are required.

Ensure that the specified conditions (see Technical data) are adhered to!

Optimum installation position (standard)

For the optimum installation position the signal distribution board is installed horizontally, and the fronts of the EJ modules face forward (see Fig. *Recommended distances for standard installation position*). The modules are ventilated from below, which enables optimum cooling of the electronics through convection. “From below” is relative to the acceleration of gravity.

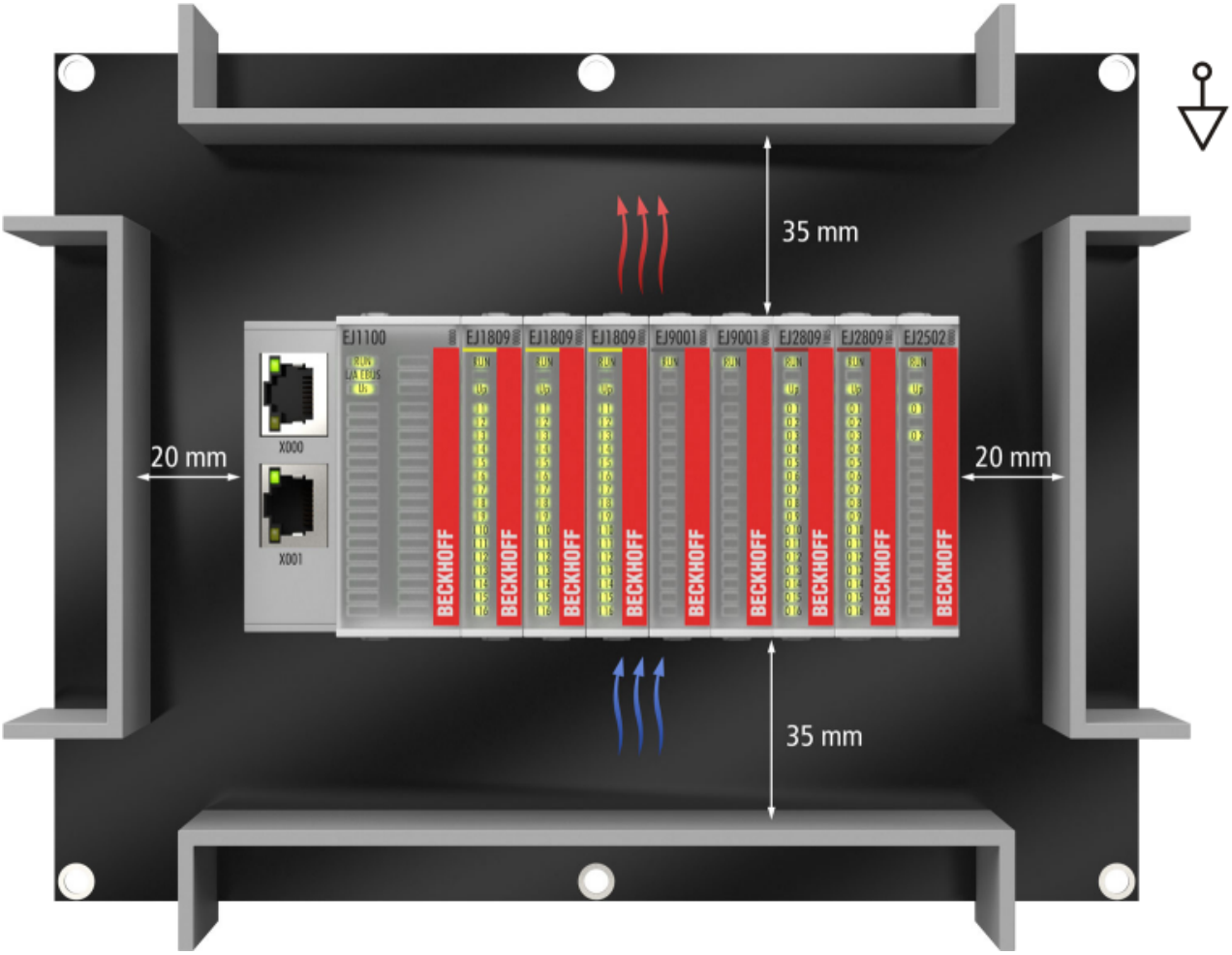


Fig. 15: Recommended distances for standard installation position

Compliance with the distances shown in Fig. *Recommended distances for standard installation position* is recommend. The recommended minimum distances should not be regarded as restricted areas for other components. The customer is responsible for verifying compliance with the environmental conditions described in the technical data. Additional cooling measures must be provided, if required.

Other installation positions

All other installation positions are characterized by a different spatial position of the signal distribution board, see Fig. *Other installation positions.*

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

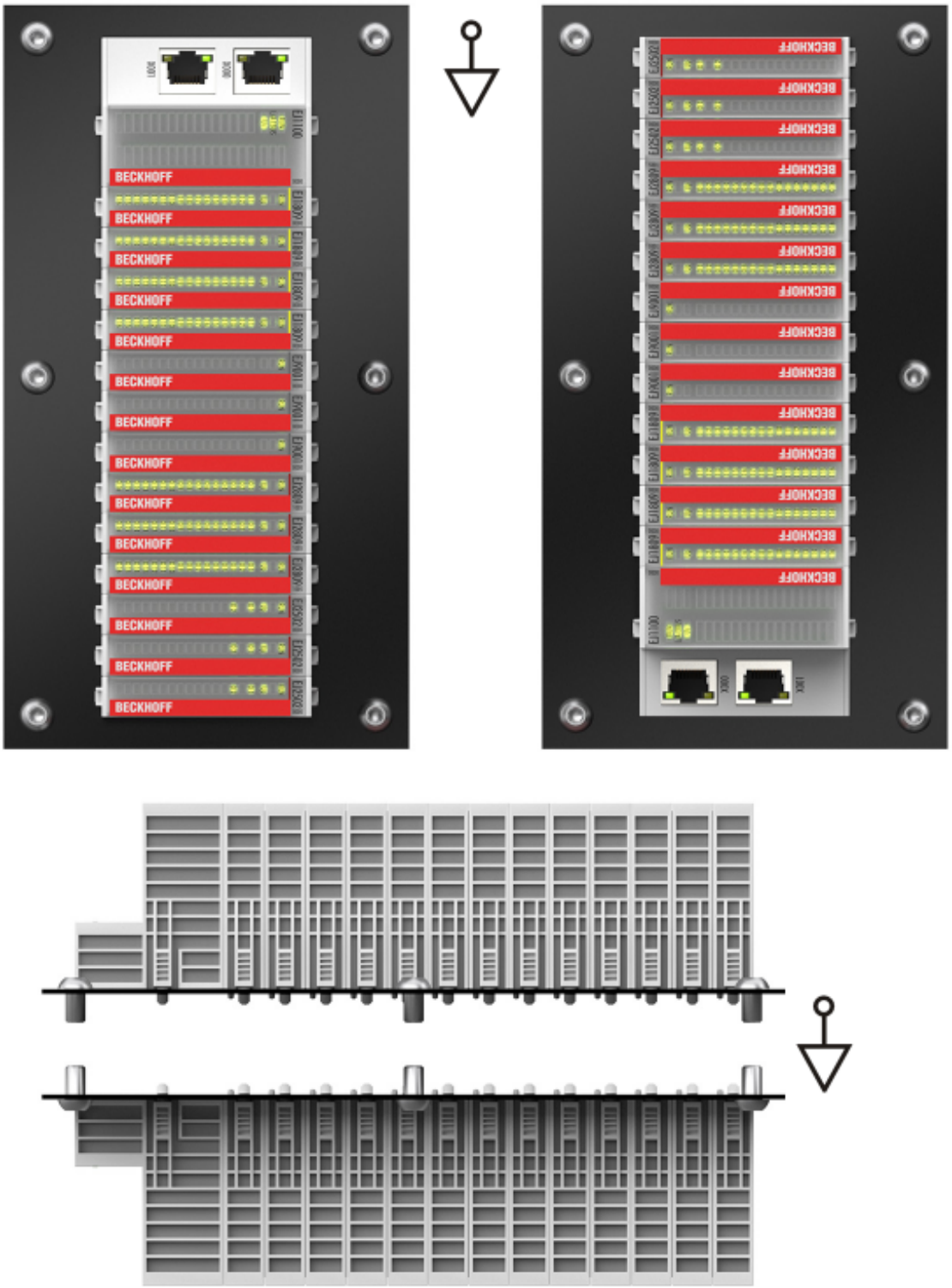


Fig. 16: Other installation positions

4.4 Codings

4.4.1 Color coding

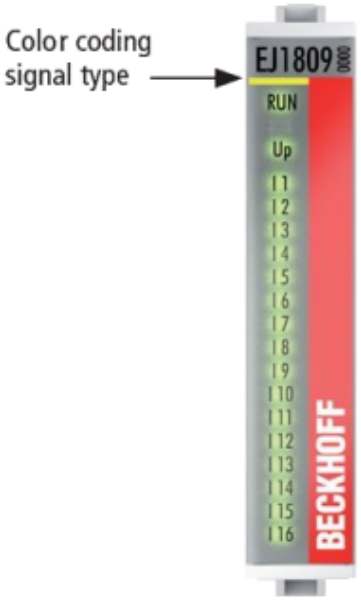


Fig. 17: EJ modules color code; sample: EJ1809

The EJ modules are color-coded for a better overview in the control cabinet (see diagram above). The color code indicates the signal type. The following table provides an overview of the signal types with corresponding color coding.

Signal type	Modules	Color
Coupler	EJ11xx	No color coding
Digital input	EJ1xxx	Yellow
Digital output	EJ2xxx	Red
Analog input	EJ3xxx	Green
Analog output	EJ4xxx	Blue
Position measurement	EJ5xxx	grey
Communication	EJ6xxx	grey
Motion	EJ7xxx	orange
System	EJ9xxx	grey

4.4.2 Mechanical position coding

The modules have two signal-specific coding pins on the underside (see Figs. B1 and B2 below). In conjunction with the coding holes in the signal distribution board (see Figs. A1 and A2 below), the coding pins provide an option for mechanical protection against incorrect connection. This significantly reduces the risk of error during installation and service.
Couplers and placeholder modules have no coding pins.

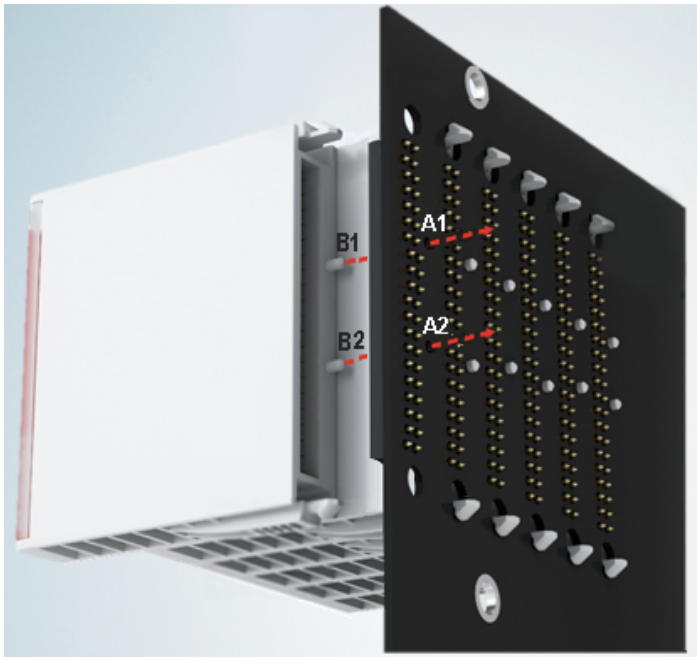


Fig. 18: Mechanical position coding with coding pins (B1 and B2) and coding holes (A1 and A2)

The following diagram shows the position of the position coding with position numbers on the left-hand side. Modules with the same signal type have the same coding. For sample, all digital input modules have the coding pins at positions one and three. There is no plug protection between modules with the same signal type. During installation the module type should therefore be verified based on the device name.

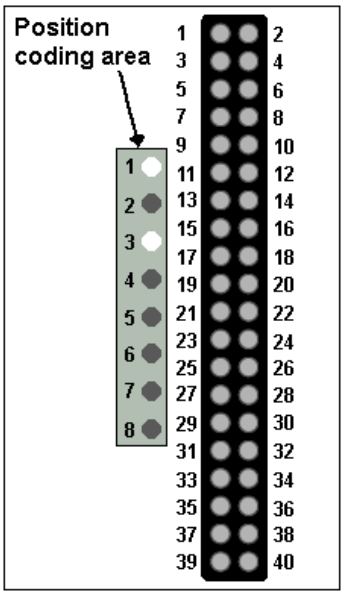


Fig. 19: Pin coding; sample: digital input modules

4.5 Installation on the signal distribution board

EJ modules are installed on the signal distribution board. The electrical connections between coupler and EJ modules are realized via the pin contacts and the signal distribution board.

The EJ components must be installed in a control cabinet or enclosure which must provide protection against fire hazards, environmental conditions and mechanical impact.

⚠ WARNING

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

Bring the module system into a safe, de-energized state before starting installation, disassembly or wiring of the modules.

NOTE

Risk of damage to components through electrostatic discharge!

Observe the regulations for ESD protection.

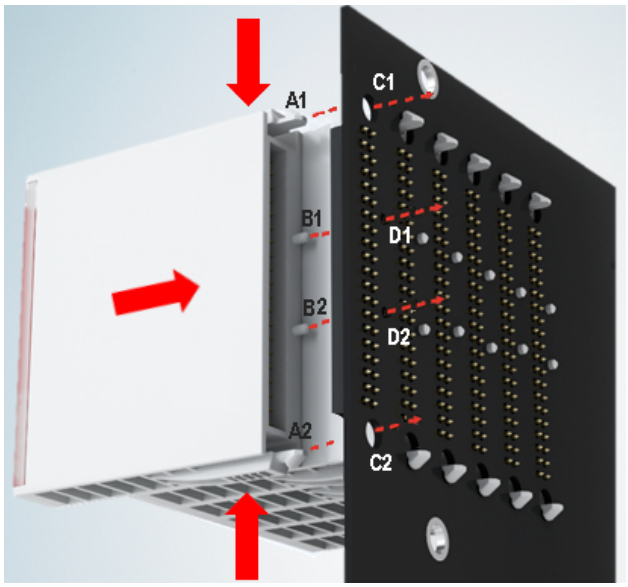


Fig. 20: Installation of EJ modules

A1 / A2	Latching lugs top / bottom	C1 / C2	Mounting holes
B1 / B2	Coding pins	D1 / D2	Coding holes

To install the modules on the signal distribution board proceed as follows:

1. Before the installation, ensure that the signal distribution board is securely connected to the mounting surface. Installation on an unsecured signal distribution board may result in damage to the board.
2. If necessary, check whether the positions of the coding pins (B) match the corresponding holes in the signal distribution board (D).
3. Compare the device name on the module with the information in the installation drawing.
4. Press the upper and the lower mounting tabs simultaneously and push the module onto the board while gently moving it up and down, until the module is latched securely.
   The required contact pressure can only be established and the maximum current carrying capacity ensured if the module is latched securely.
5. Use placeholder modules (EJ9001) to fill gaps in the module strand.

NOTE

- During installation ensure safe latching of the modules on the signal distribution board! The consequences of inadequate contact pressure include:

⇨ loss of quality of the transferred signals,

⇨ increased power dissipation of the contacts,

⇨ impairment of the service life.

4.6 Extension options

Three options are available for modifications and extensions of the EJ system.

- Replacing the placeholder modules with the function modules provided for the respective slot
- Assigning function modules specified for the respective slots for the reserve slots at the end of the module string
- Linking with EtherCAT Terminals and EtherCAT Box modules via an Ethernet/EtherCAT connection

4.6.1 Using placeholder modules for unused slots

The EJ9001 placeholder modules are used to close temporary gaps in the module strands (see Fig. A1 below). Gaps in the module strand cause interruption in EtherCAT communication and must be equipped with placeholder modules.
In contrast to the passive terminals of the EL series, the placeholder modules actively participate in the data exchange. Several placeholder modules can therefore be connected in series, without impairing the data exchange.
Unused slots at the end of the module strand can be left as reserve slots (see Fig. B1 below).

The machine complexity is extended (extended version) by allocating unused slots (see Figs. A2 below - Exchanging placeholder modules and B2 - Assigning reserve slots) according to the specifications for the signal distribution board.

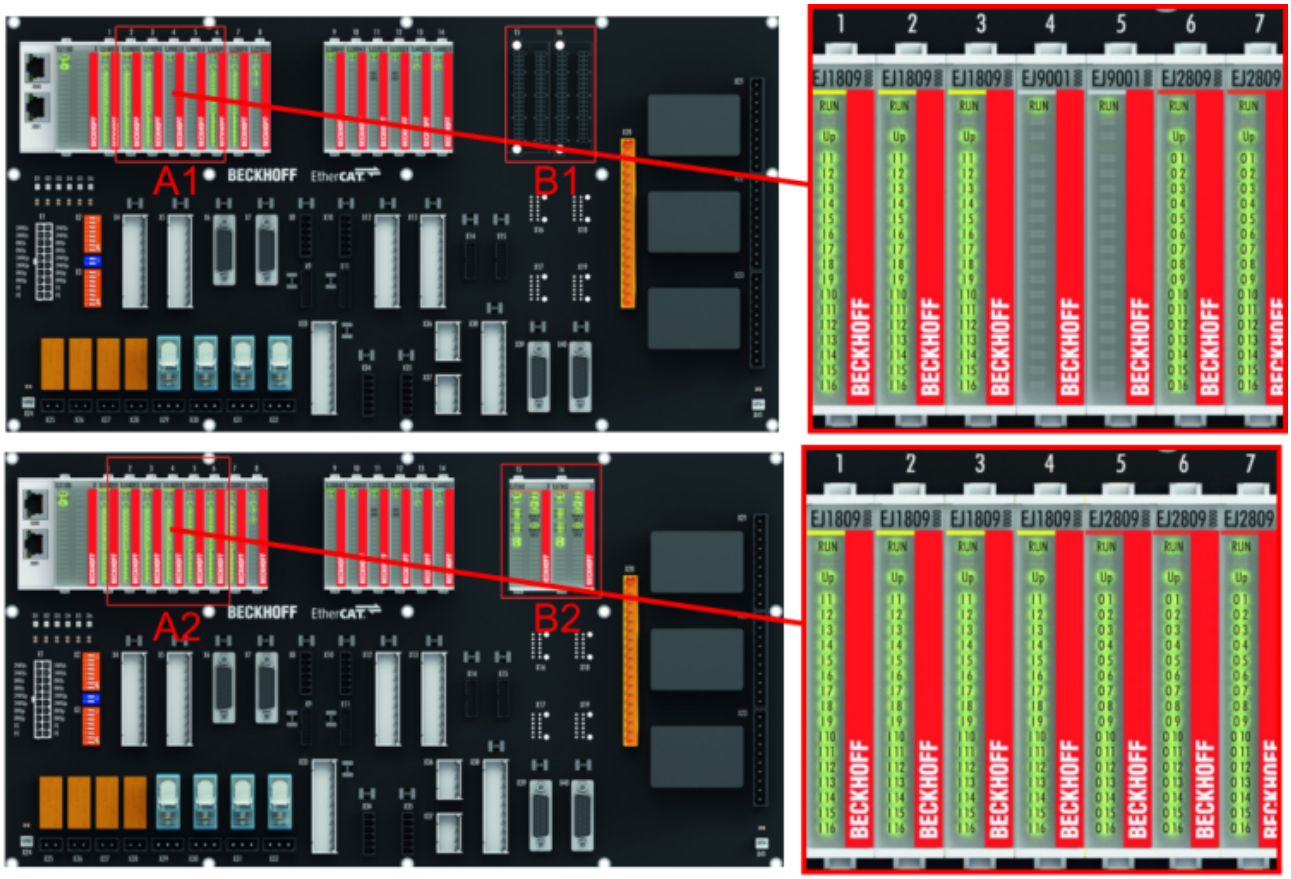


Fig. 21: Sample: Exchanging placeholder modules and assigning reserve slots

E-bus supply

Exchange the placeholder modules with other modules changes the current input from the E-Bus. Ensure that adequate power supply is provided.

4.6.2 Linking with EtherCAT Terminals and EtherCAT Box modules via an Ethernet/EtherCAT connection

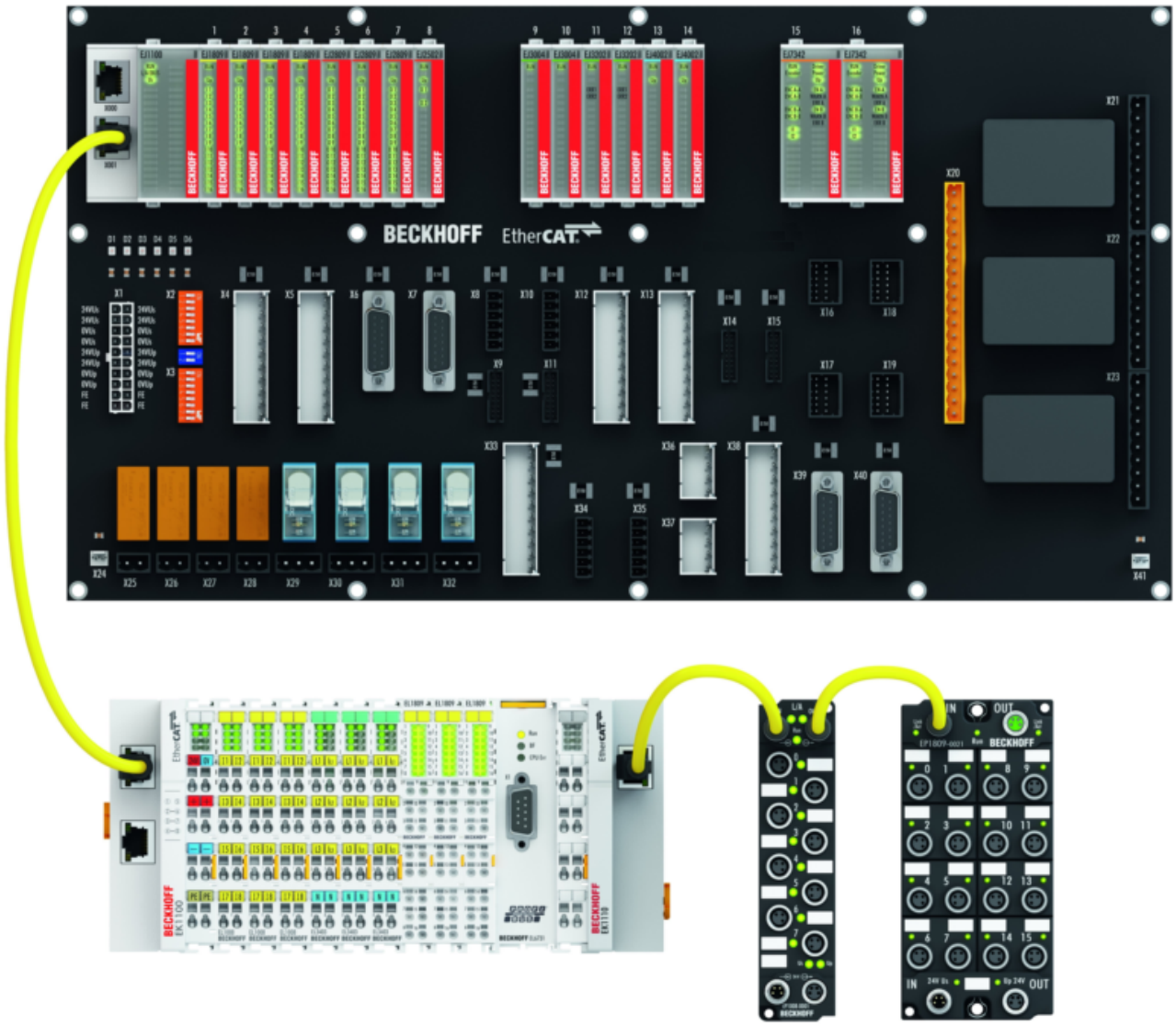


Fig. 22: Example of extension via an Ethernet/EtherCAT connection

4.7 IPC integration

Connection of CX and EL terminals via the EK1110-004x EtherCAT EJ coupler

The EK1110-0043 and EK1110-0044 EtherCAT EJ couplers connect the compact DIN-rail PCs of the CX series and attached EtherCAT Terminals (ELxxxx) with the EJ modules on the signal distribution board.

The EK1110-004x are supplied from the power supply unit of the Embedded PC.
The E-bus signals and the supply voltage of the field side U_P are routed directly to the PCB via a plug connector at the rear of the EtherCAT EJ couplers.

Due to the direct coupling of the Embedded PC and the EL terminals with the EJ modules on the PCB, no EtherCAT Extension (EK1110) or EtherCAT Coupler (EJ1100) is required.

The Embedded PC can be expanded with EtherCAT Terminals that are not yet available in the EJ system, for example.

Fig. 23: Example PCB with Embedded PC, EK1110-0043 and EJxxxx, rear view EK1110-0043

Connection of C6015 / C6017 via the EJ110x-00xx EtherCAT Coupler

Thanks to their ultra-compact design and versatile mounting options, the C6015 and C6017 IPCs are ideally suited for connection to an EJ system.

In combination with the ZS5000-0003 mounting set, it is possible to place the C6015 and C6017 IPCs compactly on the signal distribution board.
The EJ system is optimally connected to the IPC via the corresponding EtherCAT Cable (see following Fig. [A]).
The IPC can be supplied directly via the signal distribution board using the enclosed power plug (see Fig. [B] below).

NOTE

Positioning on the signal distribution board

The dimensions and distances for placement and other details can be found in the Design Guide and the documentation for the individual components.

The figure below shows the connection of a C6015 IPC to an EJ system as an example. The components shown are schematic, to illustrate the functionality.

Fig. 24: Example for the connection of a C6015 IPC to an EJ system

4.8 Disassembly of the signal distribution board

⚠ WARNING

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

Bring the module system into a safe, de-energized state before starting installation, disassembly or wiring of the modules.

NOTE

Risk of damage to components through electrostatic discharge!

Observe the regulations for ESD protection.

Each module is secured through latching on the distribution board, which has to be released for disassembly.

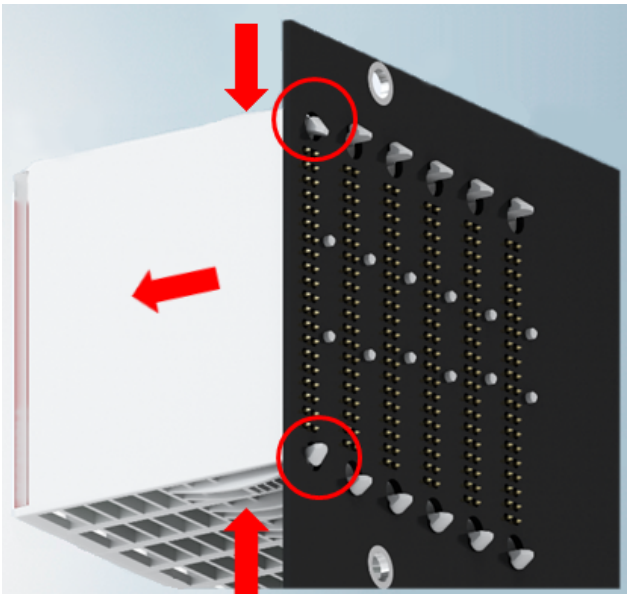


Fig. 25: Disassembly of EJ modules

To disassemble the module from the signal distribution board proceed as follows:

1. Before disassembly, ensure that the signal distribution board is securely connected to the mounting surface. Disassembly of an unsecured signal distribution board may result in damage to the board.
2. Press the upper and lower mounting tabs simultaneously and pull the module from board while gently moving it up and down.

4.9 Disposal

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

5 EtherCAT basics

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

6 Commissioning

6.1 Basic function principles

The EtherCAT plug-in module EJ6080 supports two access methods to the memory: with cyclic process data [▶ 41] and via acyclic SDO/CoE access [▶ 43]. Existing data can be deleted.

6.1.1 Delivery state

The EtherCAT plug-in module is delivered ready for operation with no predefined acyclic data structures.

General information on data consistency

The EJ6080 module can be used to cyclically store machine data in a fail-safe manner. A corresponding feedback in the status (cyclic operation) confirms that the data has been correctly transferred from the module. The data must appear correctly in the CoE in acyclic operation mode.

- It may happen that during a (a)cyclic write access by the task, the module is de-energized or the task is stopped.

  ⇨ In the EJ6080 module, shadow buffers or the NOVRAM ensure data consistency at the lowest level for this case, so that the last correctly written data set can always be accessed.

  ⇨ At the application layer, however, the user must ensure by appropriate application that the application adopts consistent data, e.g. at startup (e.g. continuous counters or ID identifiers in the data to be saved).

  Example:
  Several acyclic data objects 1 to 3 are defined, which are described continuously in sequence by the application (e.g. three axis positions determined at the same time in the application). The power fails during write access to object 2. Then object 1 contains the current axis position 1, but objects 2+3 contain obsolete ("Status" object 0xF100:01 = 0x0400 (Old Novram object restored). At restart, the application must not assume that it is receiving 3 axis positions originating from the same point in time.
- The same side effects must be taken into account in the case of simultaneous operation of the EtherCAT plug-in module EJ6080 and persistent/retain/other NOVRAM data (e.g. from FC cards or CX).

Readiness for operation of the EJ6080

- It is **imperative** that you ensure in your application that the module has a valid WorkingCounter WcState before beginning with writing and, in particular, reading communication!
- Process data delivered by an EtherCAT slave when the WcState is ≠ 0 (even if the data is ≠0) must be discarded as invalid!

CoE directory EJ6080

The functionality of the EJ6080 module results in CoE objects being deleted or created during reparameterization.

- Please refer to the notes in chapter Definition of the memory objects via the StartUp list [▶ 44] in order to ensure correct online CoE display.

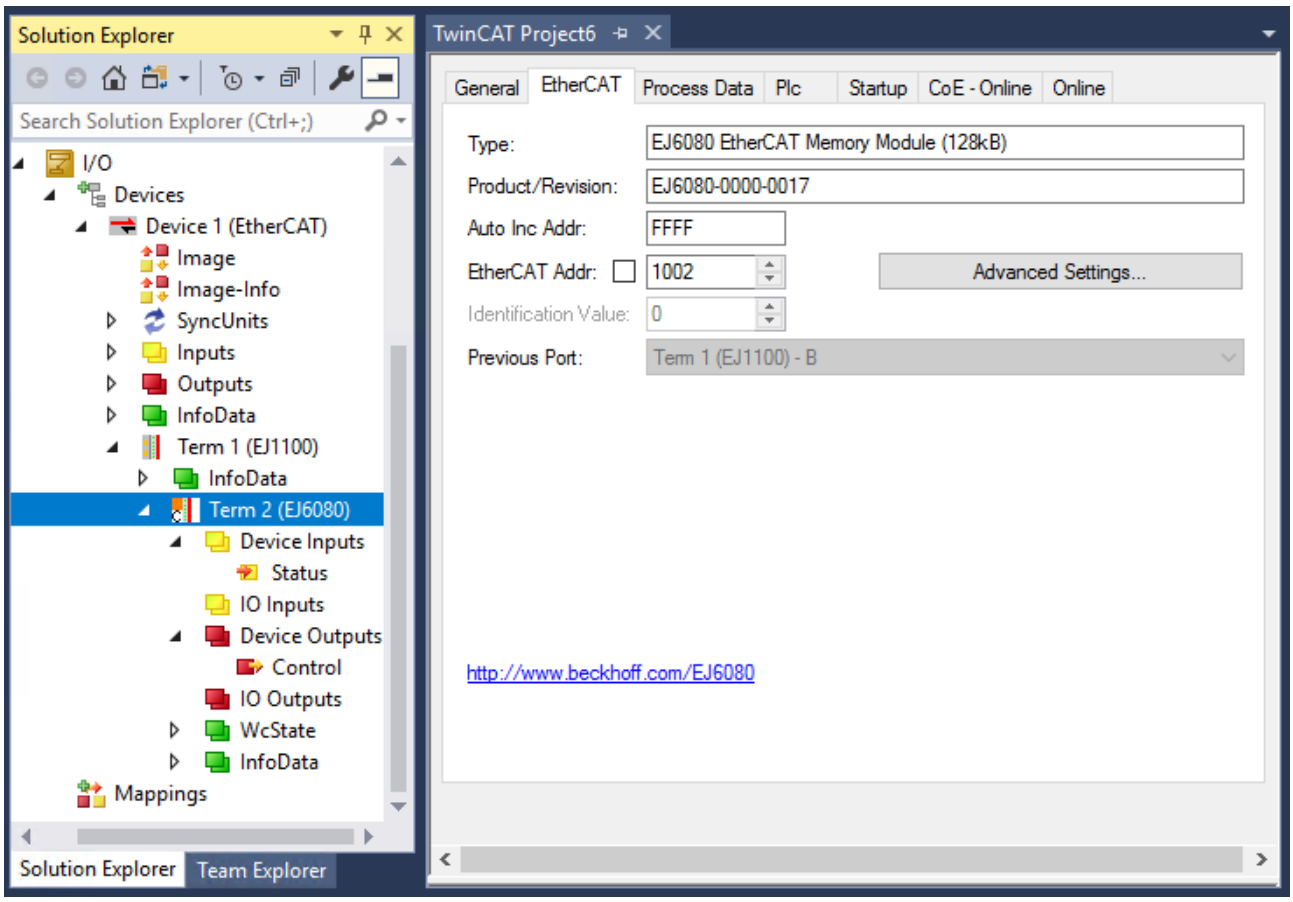


Fig. 26: Default process image of the EJ6080 after insertion

- Status: 16-bit feedback of the EJ6080
- Control: 16-bit control of the EJ6080

6.1.2 Status and control word

Status word

The status word (SW) is located in the input process image, and is transmitted from the module to the controller.

Bit	SW.15	SW.14	SW.13	SW.12	SW.11	SW.10	SW.9	SW.8
Name	Acyclic operation, 4 bit counter increments with each successful save				-	Old Novram object restored	Novram initial-ized	Novram objects locked

Bit	SW.7	SW.6	SW.5	SW.4	SW.3	SW.2	SW.1	SW.0
Name	-	-	-	-	No data written	-	-	Cyclic operation with handshake, "Data written"

The meaning of the status word is displayed in the TwinCAT System Manager in the "Comment" field of the "Variable" tab (see following figure).

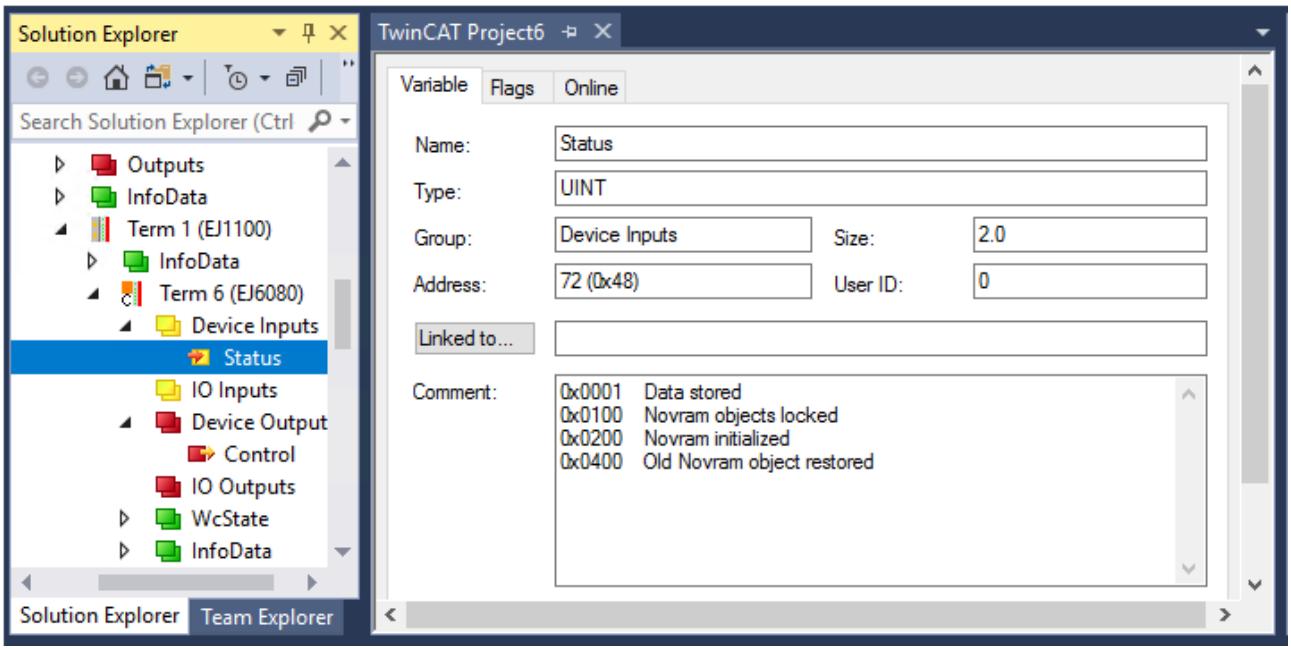


Fig. 27: Status word display in TwinCAT System Manager

Index 0xF100:01	Name	Meaning
0x0001	Data stored	Data was saved
0x0008	MASK_NO_DATA_WRITTEN	No data was saved (process data length is zero)
0x0100	Novram objects locked	The structure in 0x2F00 was locked against changes (see chapter Protect acyclic structure against changes [▶ 50])
0x0200	Novram initialized	NOVRAM was reinitialized at startup "Manufacturer configuration active (once after first boot process)"
0x0400	Old Novram object restored	NOVRAM was initialized at startup with values from a buffer (if the module was switched off during saving).

Control word

The control word (CW) is located in the output process image, and is transmitted from the controller to the module.

Bit	CW.15	CW.14	CW.13	CW.12	CW.11	CW.10	CW.9	CW.8
Name	-	-	-	-	-	-	-	-

Bit	CW.7	CW.6	CW.5	CW.4	CW.3	CW.2	CW.1	CW.0
Name	-	-	-	-	-	Unlock Novram objects	Lock Novram objects	Cyclic operation with handshake, "Start Writing"

The meaning of the control word is displayed in the TwinCAT System Manager in the "Comment" field of the "Variable" tab (see following figure).

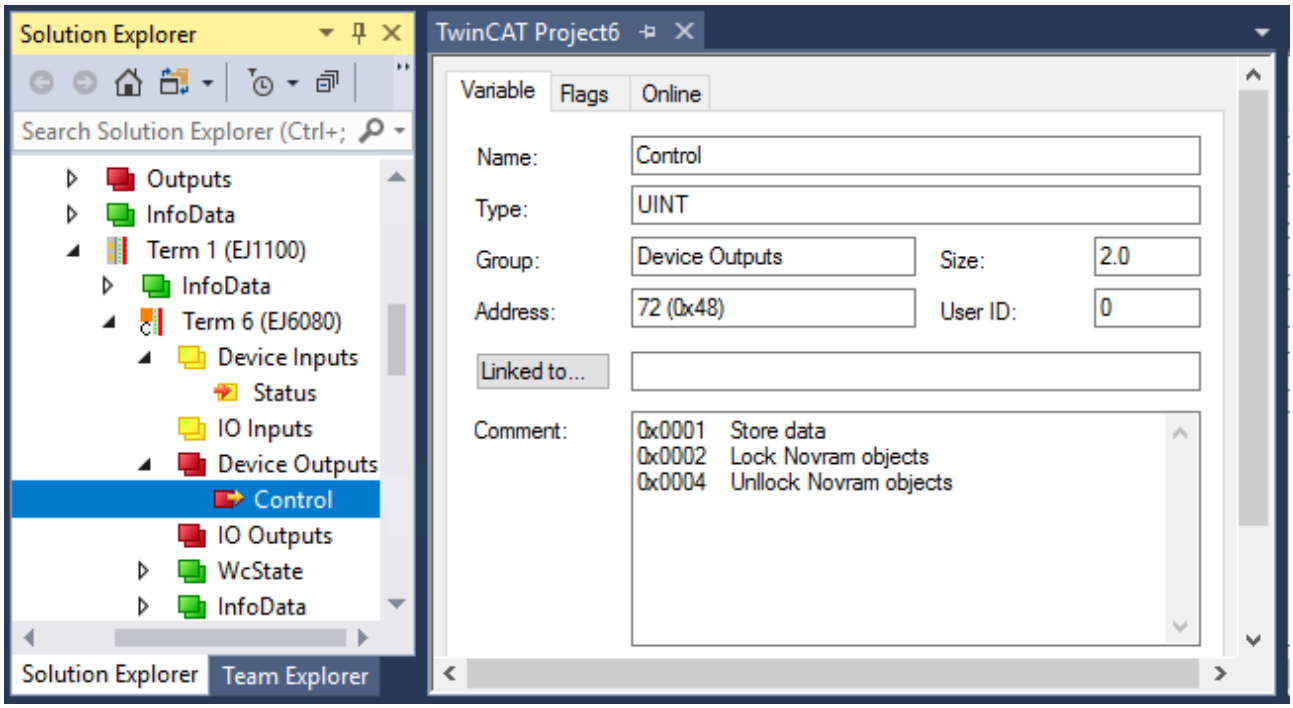


Fig. 28: Control word display in TwinCAT System Manager

Index 0xF200:01	Name	Meaning
0x0001	Store data	Data are saved
0x0002	Lock Novram objects	The structure in 0x2F00 is protected against changes.
0x0004	Unlock Novram objects	The protection against changes of the structure in 0x2F00 is removed.

6.1.3 Data traffic with cyclic process data

The user can create one set of process data with an arbitrary structure, max. 1280 bytes. This data set can be written cyclically in its entirety to the module or read by it. Individual access to the constituent parts of this data set is not possible (keyword: addressing). Control is performed by the task by means of a handshake via Control/Status Word, so that several task cycles may be necessary for saving or reading back, depending on the extent of the data and the cycle time.

When the module is switched on, the regularly saved data from the last operating session is offered immediately via the inputs for reading in.

Commissioning: the cyclic process image must be created identically at inputs and outputs, max. 1280 bytes. The creation is done via the dialog "Insert Variable". Open the dialog box by right-clicking on Inputs/Outputs -> selecting "Add New Item...".

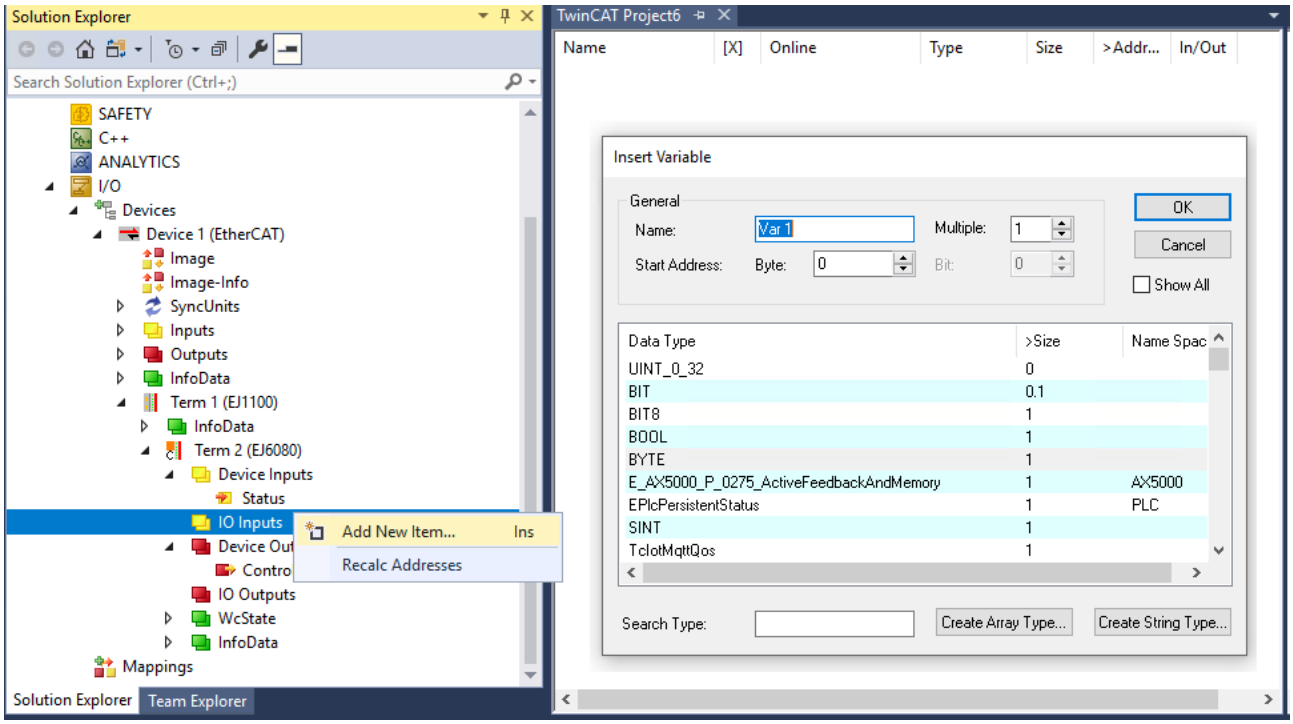


Fig. 29: Manual creation of the cyclic process data

If the input and output have different sizes, the module start fails: "PREOP to SAFEOP failed".
If more than 1280 bytes are created, the module start fails with "Invalid SM In/Out Cfg".

- The data to be written in the EtherCAT plug-in module EJ6080 is output to the module by the task at the outputs.
- The last valid data received by the EtherCAT plug-in module EJ6080 as read data arrives at the task at the inputs with no further assistance.

Structure of the process data in TwinCAT 2.10

The structure of the input/output data is subject to the following restriction up to TwinCAT 2.10 build 1330:

- All defined variables must consist of n*bytes, single bits or such compound structures are not allowed.

⇨ If such data is defined, the EJ6080 attains the OP state, but reports an error due to Working Counter = 1.

Examples:

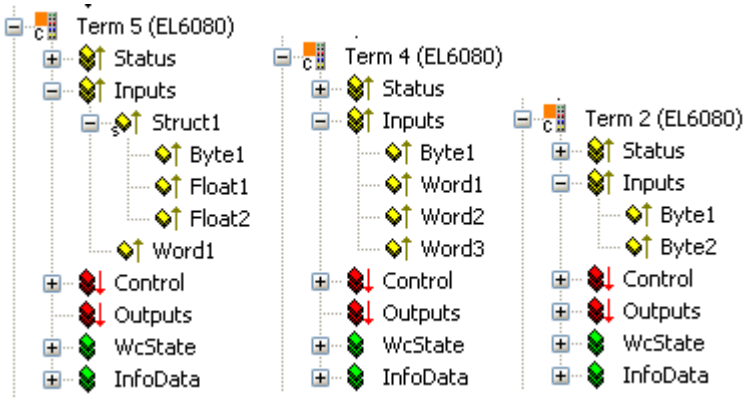


Fig. 30: Permissible configuration using the EL6080 as an example

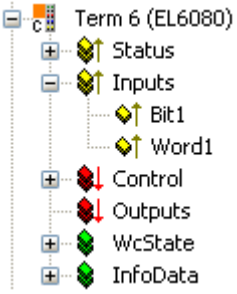


Fig. 31: Impermissible configuration using EL6080 to TwinCAT b2.10 b1330 as an example

Cyclic operation with handshake, recommended operating sequence:

1. Output the input data to the module; set ControlWord = 1.
2. Once the module has successfully adopted the data, StatusWord = 1 is returned.
3. Set ControlWord = 0.
4. Wait until StatusWord = 0 is returned; the module is then ready for a new write access operation.

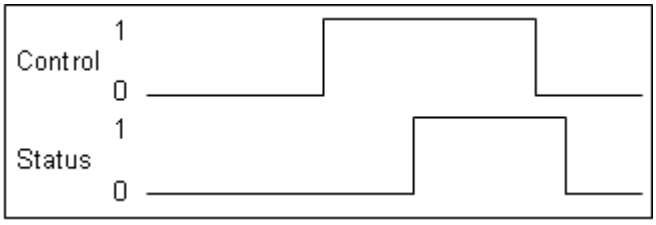


Fig. 32: Cyclic operation with handshake

Typical example of a measurement for a cyclic write process according to the sequence described above (Beckhoff reserves the right to make unannounced amendments):

- 20 bytes: 200 µs
- 1250 bytes: 2.5 ms

Process data image and frame length

In the case of short cycle times, an extensive process image for the EtherCAT plug-in module EJ6080 can result in an Ethernet frame that is longer than the cycle time.

Therefore, in order to use the entire 1280 bytes of the cyclic process image, the cycle time must be at least >= 200 µs.

6.1.4 Data traffic with acyclic CoE access

In acyclic access, the user can create up to 255 memory objects that may also be differently sized. These objects can be specifically accessed from the PLC task via acyclic SDO access (**S**ervice **D**ata **O**bjects) for writing or reading; see sample program in chapter Online access to the memory objects during operation [▶ 49]. This read/write access proceeds considerably slower than the above-described cyclic memory access.

The created structure can only be changed in the module status PREOP. It can also be generally locked to prevent further changes.

Background information:
All storage space is managed in the form of CoE objects (**C**AN **o**ver **E**therCAT). The administration data (number and byte size) is located in the Object 0x2F00 with its subindexes; the stored data itself in the CoE objects from 0x2000: 0x2000, 0x2008, 0x2010, 0x2018 etc.

A memory object in acyclic access can comprise 1 to 8190 bytes.

The following procedure is recommended in order to use acyclic access:

1. Calculation of the storage space [▶ 43]
2. Definition of the memory objects via the StartUp list [▶ 44]
3. Access to the memory objects during operation [▶ 49]

6.1.5 Calculation of the storage space

The available 128 KB (131,072 bytes) of memory are subdivided as follows:

- 1280 bytes for the cyclic data
- 2000 bytes for internal administration
- x bytes as defined by the user for the acyclic data in the following
- y bytes: a shadow buffer as large as the largest memory object is kept on standby in the background.

Example: the acyclic memory objects Obj1, Obj2 and Obj3 are defined with 1,000, 3,000 and 7,000 bytes respectively --> x = 11,000 bytes and y = 7,000 bytes. Thus 113,072 bytes of storage space remain.

6.1.6 Definition of the memory objects via the StartUp list

The desired structure of the memory objects must be created once only in the status PREOP in the CoE directory in CoE object 0x2F00. This is also possible offline (without connected module) in the System Manager. At startup, the EtherCAT plug-in module EJ6080 checks whether the memory structure has changed and if necessary creates the objects in the memory accordingly.

NOTE

Data loss in the event of a change in the data structure

If the data structure or the object 0x2F00 is changed, all existing data in the module EJ6080 is deleted.

- The data structure can be locked against change by setting the value in index 0xF200:01 = 2.

  ⇨ Observe the notes in the chapter Protect acyclic structure against changes [▶ 50]!

Procedure short form:

1. Unlock object 0x2F00 for changes to the structure
   Set index 0xF200:01 = 4 "Unlock Novram objects" (manually in the System Manager or via PLC).
2. Entering the StartUp command for the CoE object 0x2F00 [▶ 45]
   Content: number of objects + respective length in bytes.
   Note: complete access, byte alignment, no empty object possible, only possible in P -> S transition, 16-bit entries, max. 127 memory objects.
3. Reload the configuration. [▶ 46]
4. For control: reload the CoE directory [▶ 46].
5. Lock object 0x2F00 for changes of the structure
   Set index 0xF200:01 = 2 "Lock Novram objects" (manually in System Manager or via PLC).

Procedure in detail:

To define the desired structure of the memory objects, proceed as follows:

Following startup, the EtherCAT plug-in module EJ6080 contains either an already changed memory structure or the default memory structure. The CoE objects from 0x2000 and 0x2F00 are important in the following. In the following figure, an object of size 1 byte is already created in EJ6080:

- 0x2F00:0 (NOVRAM Size): "1" = 1 memory object exists.
- 0x2F00:01 (Subindex 001): "0x0001" = this one memory object has been defined with a size of 1 byte.
- 0x2000 (NOVRAM Data): 00 - this one memory object thus bears "00" as the date of use.

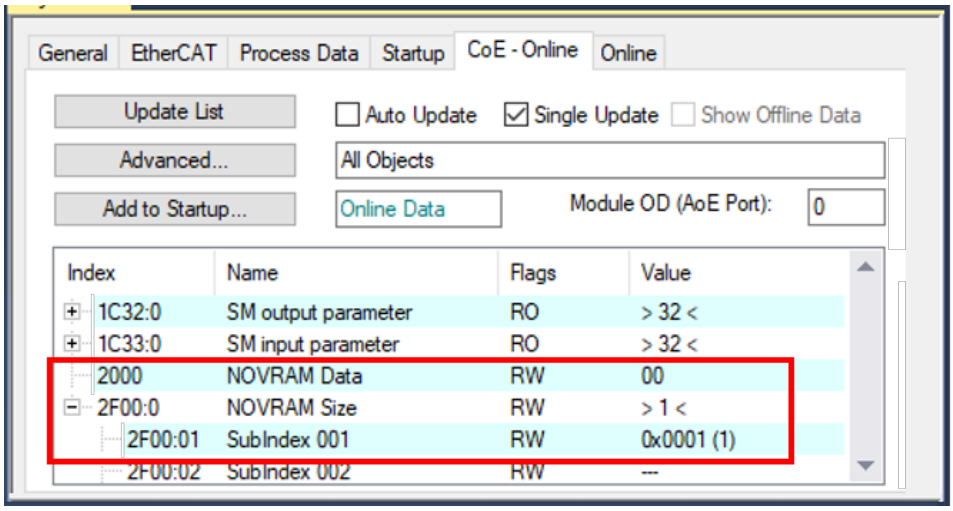


Fig. 33: Any start state of the EJ6080 module

Now the EtherCAT plug-in module EJ6080 is to be reconfigured to the memory structure:

- 1 byte
- 10 byte
- 256 byte
- 3 byte

i.e. a total of 270 bytes in four memory objects, which can each be accessed acyclically via the CoE.

Entry in the StartUp list

A corresponding entry for this must be made in the EJ6080's StartUp list. This list is empty for the EJ6080 module in the default state.

1. A new entry is added in the *StartUp* tab via the *New* button.
   ⇨ The "Edit CANopen Startup Entry" window opens, in which the new StartUp entry can be defined.

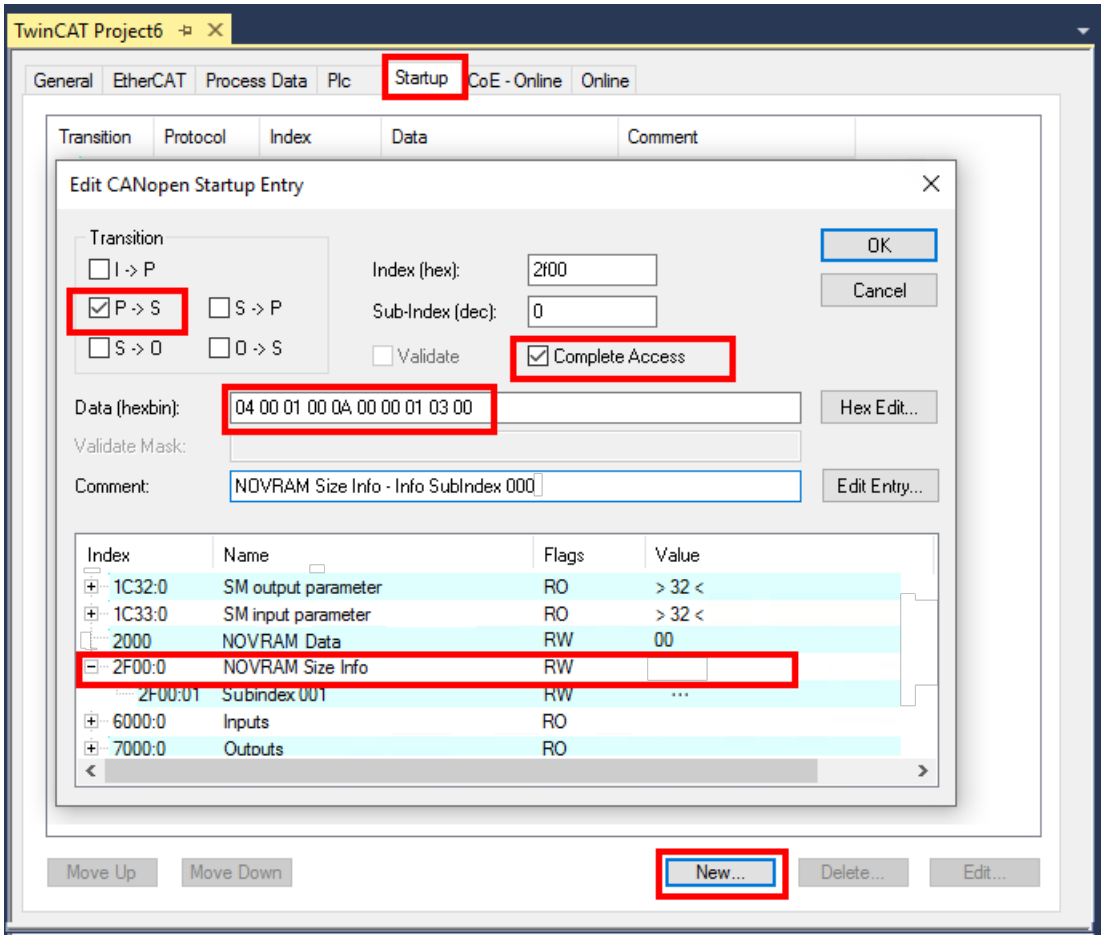


Fig. 34: Definition of a new StartUP entry

StartUp list selection dialog

If no CoE entries (see Fig. *Definition of the StartUp entry*) are offered for selection, this has the following reason: you are working offline (i.e. without a live connected module) and the ESI (EtherCAT Slave Information, XML description) of the EJ6080 EtherCAT plug-in module that you are using does not contain a dictionary.

- In this case you can also define StartUp entries entirely manually, i.e. enter Index and Subindex manually.

2. Selection of the object
   ⇨ Online:
   select object 0x2F00 from the list, so that the correct values already appear in *Index/SubIndex* and *Comment*.
   ⇨ Offline:
   manual entry via the fields *Index (hex)* and *Sub-index (dec)*
3. "P-->S" must be selected in *Transition* (the value will be loaded to the module EJ6080 during the state transition PREOP-->SAFEOP).
4. *CompleteAccess* must be activated.
5. In *Data (hexbin)* enter your desired structure in hex and reverse byte order (byte alignment), in the form "aa aa bb cc cc ...".
   ⇨ aa aa: number of desired memory objects
   ⇨ bb bb, cc cc, ...: respective size definition of the objects

   Example:
   - aa aa = 04 00: 4 memory objects are desired.
   - bb bb = 01 00: 1st memory object has a size of 1 byte (00 01_{hex} = 1_{dec}).
   - cc cc = 0A 00: 2nd memory object has a size of 10 bytes (00 $0A_{hex}$ = 10_{dec}).
   - dd dd = 00 01: 3rd memory object has a size of 256 bytes (01 00_{hex} = 256_{dec}).
   - ee ee = 03 00: 4th memory object has a size of 3 bytes (00 03_{hex} = 3_{dec}).

6. Confirm with *OK*. The new StartUp entry looks like this:

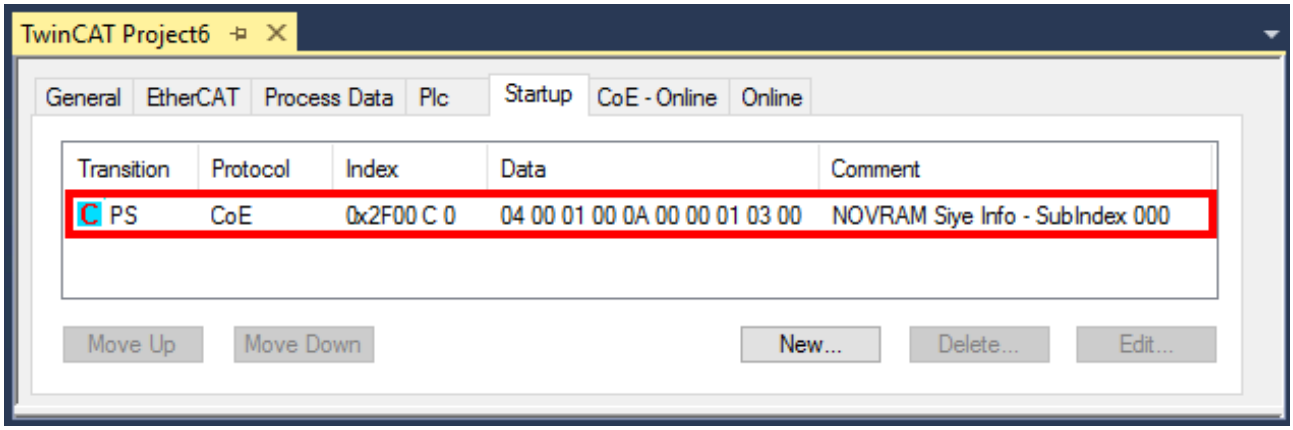


Fig. 35: New StartUp entry

Reloading the configuration

Now this configuration must be loaded to the EJ6080 module and in particular the PREOP-->SAFEOP state transition must be run through as defined in the StartUp entry.
Click the *Reload* button (Fig. *Reload the configuration*):

Fig. 36: Reload the configuration

Reload the CoE directory

The old entry is still visible in the *CoE-Online* tab.

The reason is that TwinCAT primarily only loads the values of CoE objects that are known to the System Manager. If the CoE structure changes on the device, the System Manager has to be instructed to load the new structure, which may differ from the default structure according to XML/Dictionary.

Load the reconfigured CoE directory (Fig. *Full reload of the CoE directory*):

- CoE-Online, *Advanced*...
- Double-click *AllObjects*
- then *OK*

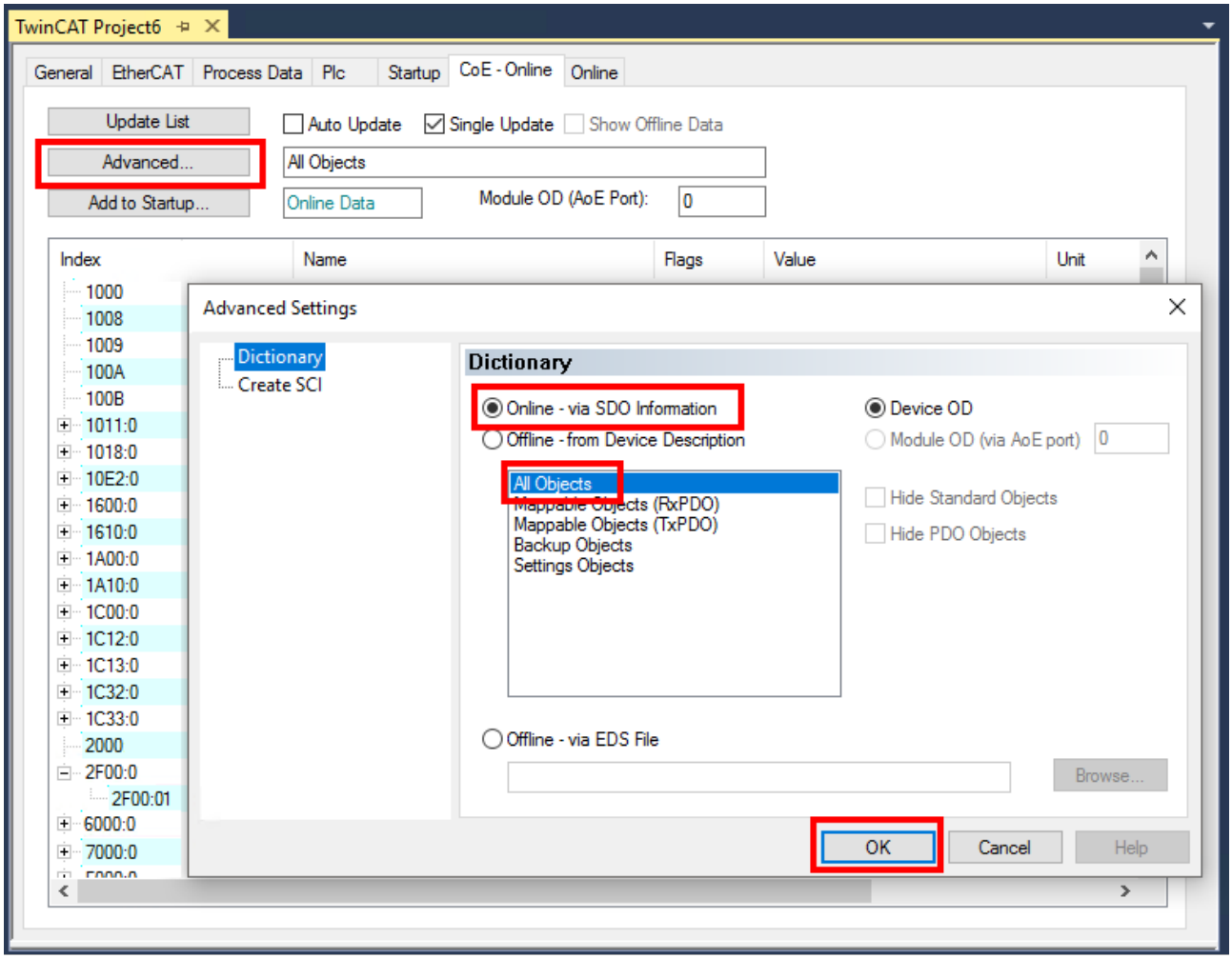


Fig. 37: Complete reload of the CoE directory

Representation of the new memory objects in the CoE directory

The newly defined memory objects are now visible and enabled, see Fig. *Modified CoE directory*.

- the structure information (byte size) of memory objects 1 to 4 is defined in CoE object 0x2F00, Subindex 001 to 004
- the current memory contents are displayed in the order according to associated CoE objects 0x2000, 0x2008, 0x2010 etc.

Updating the CoE directory

In order to update the display of the CoE directory, use AutoUpdate, UpdateList or the appropriate dialog under Advanced.

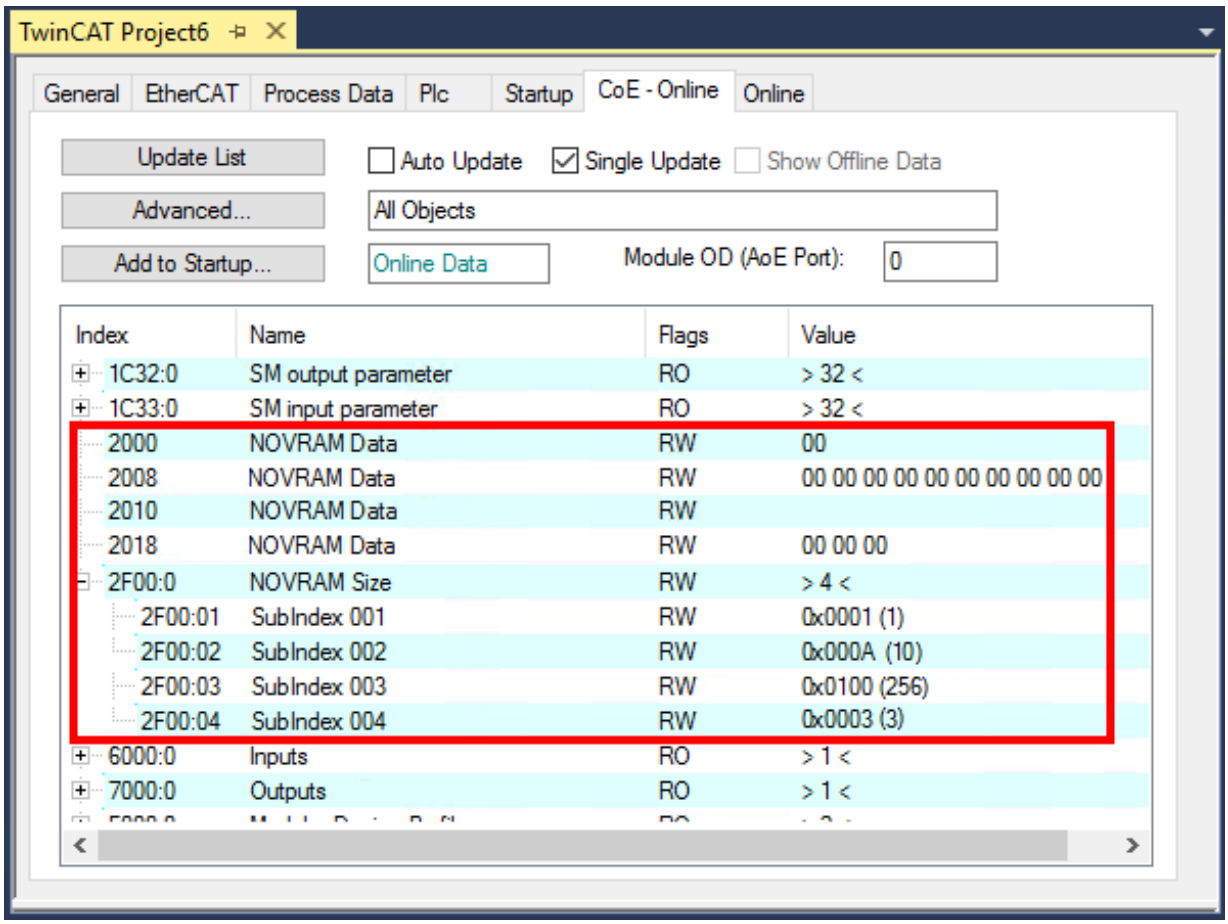


Fig. 38: Changed CoE directory

Displaying larger CoE objects

In the sample chosen here, the contents of the 3rd memory object in CoE object 0x2010 are not shown in the TwinCAT display, see fig. *Changed CoE directory*, because the contents are too large for the display. Regardless of that, the contents naturally exist.

6.1.7 Online access to the memory objects during operation

For access from the PLC, function blocks from the Beckhoff TcEtherCAT.lib such as FB_EcCoESdoWriteEx can be used. A write call in the form

```
fbWriteCoE(
sNetId:=sAmsNetId,
nSlaveAddr:= tAmsAddr.port ,
nSubIndex:= 0,
nIndex:= 16#2000 + ((byObjectNo - 1) * 8),
pSrcBuf:= pDataForWrite,
cbBufLen:= wSizeOfData,
bExecute:= TRUE,
tTimeout:= tAdsTimeOut,
bCompleteAccess:= FALSE,
bBusy=> ,
bError=> ,
nErrId=> );
```

is used in the attached sample program (https://infosys.beckhoff.com/content/1033/ej6080/Resources/2451364107/.zip).

6.1.8 Protecting an acyclic structure against changes

If the CoE object 0xF200:01 = 2 is set, the structure from CoE object 0x2F00 can no longer be changed.

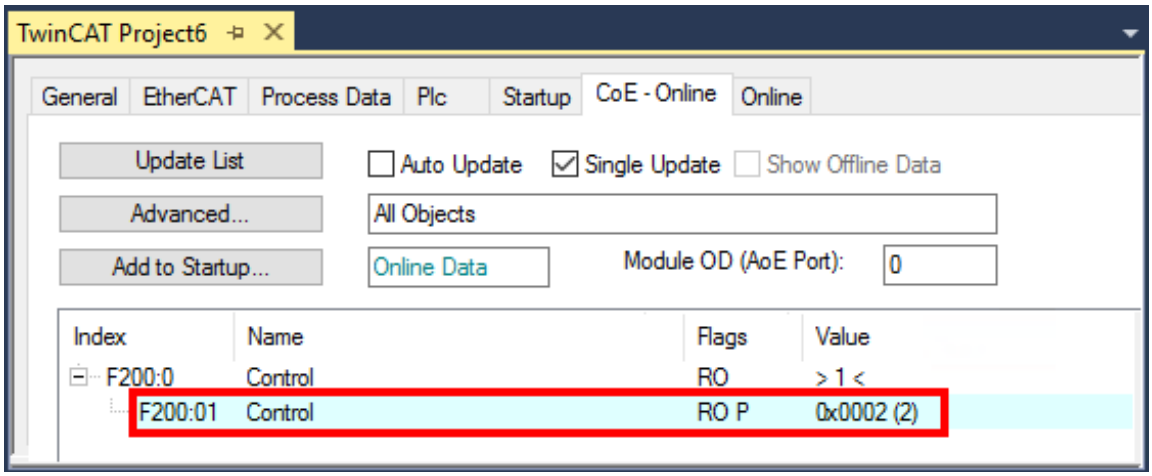


Fig. 39: Lock in the CoE

The protection of the data structure can be removed by setting the CoE object 0xF200:01 = 4.

NOTE

Data loss in the event of a change in the data structure

If the data structure in 0x2F00 is changed, all data of the module are deleted. The data structure can be protected from changes.

- To protect the acyclic data structure set the value of the "Control" object 0xF200:01 = 2 (default value?), "Lock Novram objects".
- Check the value of the status object 0xF100:01.

  ⇨ The structure in 0x2F00 is protected against changes:
  0xF100:01 = 0x0100: "Novram objects locked"

  ⇨ The structure in 0x2F00 is **not** protected against changes:
  0xF100:01 = 0x0001: "Data stored"
  0xF100:01 = 0x0008: "No Data written" (process data length is zero)
  0xF100:01 = 0x0200: "Novram initialized" (Novram was reinitialized at startup)
  0xF100:01 = 0x0400: "Old Novram objects restored" (Novram was initialized with values from the buffer at startup after the module was switched off during saving)

⇨ Make sure that the protection of the acyclic data structure is restored by setting the object 0xF200:01 = 2.

6.2 EJ6080 - object description and parameterization

EtherCAT XML Device Description

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

NOTE

Parameterization via the CoE list (CAN over EtherCAT)

The EtherCAT device is parameterized via the CoE - Online tab (with a double click on the respective object) or via the Process Data tab (assignment of PDOs). A detailed description can be found in the EtherCAT System-Documentation in chapter "EtherCAT subscriber configuration"

Please note the general CoE notes in the EtherCAT System Documentation in chapter "CoE-interface" when using/manipulating the CoE parameters:
- Keep a startup list if components have to be replaced
- Differentiation between online/offline dictionary,
- existence of current XML description
- use "CoE reload" for resetting changes

Introduction

The CoE overview contains objects for different intended applications:

- Objects required for parameterization during commissioning:
  - Restore object [▶ 51] index 0x1011
  - Configuration data [▶ 51] index 0x80n0
- Profile-specific objects:
  - Input data [▶ 52] index 0x60n0, 0x60n1
  - Output data [▶ 52] 0x70n0, 0x70n1
  - Information and diagnostic [▶ 53] data index 0xA0n0, 0xF000, 0xF008, 0xF010
- Standard objects [▶ 53]

The following section first describes the objects required for normal operation, followed by a complete overview of missing objects.

6.2.1 Restore object

Index 1011 Restore default parameters

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

6.2.2 Terminal-specific data

Index 2000 NOVRAM data

Index (hex)	Name	Meaning	Data type	Flags	Default
2000:0	NOVRAM data	Memory object 1 (enter size in index 0x2F00:01 [▶ 52])	-	RW	-

Index 2008 NOVRAM data

Index (hex)	Name	Meaning	Data type	Flags	Default
2008:0	NOVRAM data	Memory object 2 (enter size in index 0x2F00:02 [▶ 52])	-	RW	-

*

*

*

Index 23F8 NOVRAM Data

Index (hex)	Name	Meaning	Data type	Flags	Default
23F8:0	NOVRAM data	Memory object 127 (enter size in index 0x2F00:7F [▶ 52])	-	RW	-

Index 2F00 NOVRAM Size

Index (hex)	Name	Meaning	Data type	Flags	Default
2F00:0	NOVRAM size info	Maximum subindex	UINT8	RW	-
2F00:01	Subindex 001	Size of memory object 1 in bytes	-	RW	-
2F00:02	Subindex 002	Size of memory object 2 in bytes	-	RW	-
…	…	…	…	…	…
2F00:7F	Subindex 127	Size of memory object 127 in bytes	-	RW	-

Index F200 Control

Index (hex)	Name	Meaning		Data type	Flags	Default
F200:0	Control	Maximum subindex		UINT8	RW	0x01 (1_{dec})
F200:01	Control	Control		UINT16	RW	0x0000 (0_{dec})
		0x0001	"Store data" Store data			
		0x0002	"Lock Novram objects" Protect structure in index 0x2F00 against changes (see chapter Protect acyclic structure against changes [▶ 50])			
		0x0004	"Unlock Novram objects" Unprotect the structure in index 0x2F00			

6.2.3 Input data

Index 6000 Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
6000:0	Inputs	Length of this object	UINT8	RO	0x01 (1_{dec})
6000:01	Data	Process data	-	RO	-

6.2.4 Output data

Index 0x7000 Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
7000:0	Outputs	Max. subIndex (hex)	UINT8	RO	0x1 (1_{dec})
7000:01	Data	Process data	-	RO	-

6.2.5 Information and diagnostic data

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	0x0008 (8_{dec})
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x0001 (1_{dec})

Index F008 Code word

Index (hex)	Name	Meaning	Data type	Flags	Default
F008:0	Code word	Code word (see chapter Delete existing data)	UINT32	RW	0x00000000 (0_{dec})

6.2.6 Standard objects (0x1000-0x1FFF)

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

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default
1000:0	Device type	Device type of the EtherCAT slave: the LOW Word contains the used CoE profile (5001). The HIGH 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	EJ6080

Index 1009 Hardware version

Index (hex)	Name	Meaning	Data type	Flags	Default
1009:0	Hardware version	Hardware version of the EtherCAT slave	STRING	RO	00

Index 100A Software version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware version of the EtherCAT slave	STRING	RO	01

Index 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	0x17C02852 (398469202_{dec})
1018:03	Revision	Revision number of the EtherCAT slave; the Low Word (bit 0-15) indicates the special terminal number, the High Word (bit 16-31) refers to the device description	UINT32	RO	0x00000000 (0_{dec})
1018:04	Serial number	Serial number of the EtherCAT slave; the Low Byte (bit 0-7) of the Low Word contains the year of production, the High Byte (bit 8-15) of the Low Word contains the week of production, the High Word (bit 16-31) is 0	UINT32	RO	0x00000000 (0_{dec})

Index 1600 IO RxPDO-Map Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1600:0	IO RxPDO-Map Outputs	PDO Mapping RxPDO 1	UINT8	RO	0x00 (0_{dec})
1600:01	SubIndex 001		-	RW	-
1600:02	SubIndex 002		-	RW	-
…	….	…	-	RW	-
1600:2A	SubIndex 042		-	RW	-

Index 1610 Device RxPDO-Map Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1610:0	Device RxPDO-Map Outputs	PDO Mapping RxPDO 2	UINT8	RO	0x01 (01_{dec})
1610:01	SubIndex 001	1. PDO Mapping entry (object 0xF200 (Control), entry 0x01 (Control))	UINT32	RO	0xF200:01, 16

Index 1A00 IO TxPDO-Map Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00:0	IO TxPDO-Map Inputs	PDO Mapping TxPDO 1	UINT8	RO	0x00 (0_{dec})
1A00:01	SubIndex 001		-	RW	-
1A00:02	SubIndex 002		-	RW	-
…	…	…	-	RW	-
1A00:2A	SubIndex 042		-	RW	-

Index 1A10 Device TxPDO-Map Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1A10:0	Device TxPDO-Map Inputs	PDO Mapping TxPDO 2	UINT8	RO	0x01 (1_{dec})
1A10:01	SubIndex 001	1. PDO Mapping entry (object 0xF100 (Inputs), entry 0x01 (Status))	UINT32	RO	0xF100:01, 16

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x02 (2_{dec})
1C12:01	SubIndex 001	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1600 (5632_{dec})
1C12:02	SubIndex 002	2. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1610 (5648_{dec})

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RO	0x02 (2_{dec})
1C13:01	SubIndex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A00 (5656_{dec})
1C13:02	SubIndex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A10 (5672_{dec})

Index 1C32 SM output parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32_{dec})
1C32:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 3 event (no outputs available) • 2: DC - Synchron with SYNC0 Event • 3: DC - Synchron with SYNC1 Event • 34: Synchron with SM 2 event (outputs available)	UINT16	RW	0x0001 (1_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: cycle time of the local timer • Synchronous with SM 2 Event: cycle time of the master • DC-Mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x000F420 (1000000_{dec})
1C32:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0: Free Run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron with SM 3 Event is supported (no outputs available) • Bit 2-3 = 01: DC-Mode is supported • Bit 4-5 = 01: Input Shift through local event (outputs available) • Bit 4-5 = 10: Input Shift with SYNC1 event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C33:08)	UINT16	RO	0x0002 (2_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x000186A0 (100000_{dec})
1C32:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C32:07	Minimum delay time	Min. time between SYNC1 event and the reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:08	Command	With this entry the real required process data provision time can be measured. • 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 0x1C33:03, 0x1C33:06, 0x1C33:09 are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RW	0x0000 (0_{dec})
1C32:09	Maximum Delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C32:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0_{dec})
1C32:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	0x00 (0_{dec})

Index 1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32_{dec})
1C33:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 3 event (no outputs available) • 2: DC - Synchron with SYNC0 Event • 3: DC - Synchron with SYNC1 Event • 34: Synchron with SM 2 event (outputs available)	UINT16	RW	0x0022 (34_{dec})
1C33:02	Cycle time	Cycle time (in ns): • Free Run: cycle time of the local timer • Synchronous with SM 2 Event: cycle time of the master • DC-Mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x000F420 (1000000_{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: Free Run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron with SM 3 Event is supported (no outputs available) • Bit 2-3 = 01: DC-Mode is supported • Bit 4-5 = 01: Input Shift through local event (outputs available) • Bit 4-5 = 10: Input Shift with SYNC1 event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C33:08)	UINT16	RO	0x0002 (2_{dec})
1C33:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x0000186A0 (100000_{dec})
1C33:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:07	Minimum delay time	Min. time between SYNC1 event and the reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C33:08	Command	With this entry the real required process data provision time can be measured. • 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 0x1C33:03, 0x1C33:06, 0x1C33:09 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, only DC mode)	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	RO	0x00 (0_{dec})

Index F100 Status

Index (hex)	Name	Meaning		Data type	Flags	Default
F100:0	Status	Maximum subindex		UINT8	RW	0x03 (3_{dec})
F100:01	Status	Status		UINT16	RW	0x0000 (0_{dec})
		0x0001	"Data stored" Data stored			
		0x0008	"No data written" No data stored (process data length is zero)			
		0x0100	"Novram objects locked" the structure in 0x2F00 was locked against changes (see chapter Protecting an acyclic structure against changes [▶ 50])			
		0x0200	"Novram initialized" Novram was reinitialized at startup.			
		0x0400	"Old Novram objects restored" Novram was initialized at startup with values from a buffer (if the module was turned off while saving).			

7 Appendix

7.1 Support and Service

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

Beckhoff's branch offices and representatives

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

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages: https://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

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

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

Hotline:	+49 5246 963 157
Fax:	+49 5246 963 9157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963 460
Fax:	+49 5246 963 479
e-mail:	service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

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

More Information:
www.beckhoff.com/EJ6080

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