EtherCAT

User's manual Bus controllers

Version: **2.10 (February 2019)**
Model no.: **MAEC-ENG**

Translation of the original manual

1 General information

EtherCAT is an Ethernet-based fieldbus developed by the ETG user organization. This protocol is suitable for both hard and soft real-time requirements in automation technology.

EtherCAT slave devices take the data designated for them from a telegram as it is passing through the device. Input data is also added to the telegram as it is passing through. The bus controller allows X2X Link I/O modules to be coupled to EtherCAT and operated on any EtherCAT master system. A transition between IP20 and IP67 protection outside of the control cabinet is possible by arranging X20, X67 or XV modules one after the other as needed at distances up to 100 m.

Master systems without FoE support (File Access over EtherCAT) require an appropriate configuration tool or CoE support (CANopen over EtherCAT) to transfer the configuration (optional).

2 Technical description

2.1 X20 bus controller

2.1.1 Order data

Model number	Short description	Figure
	Bus controllers	
X20BC00G3	X20 bus controller, 1 EtherCAT interface, 2x RJ45, order bus base, power supply module and terminal block separately.	
	Expandable bus controllers	
X20BC80G3	X20 bus controller, 1 EtherNet/IP interface, supports expansion with X20 EtherCAT junction modules, 2x RJ45, order bus base, power supply module and terminal block separately.	
	Required accessories	
	System modules for bus controllers	
X20BB80	X20 bus base, for X20 base module (BC, HB, etc.) and X20 power supply module, X20 end plates (left and right) X20AC0SL1/X20AC0SR1 included	
X20PS9400	X20 power supply module, for bus controller and internal I/O power supply, X2X Link power supply	
X20PS9402	X20 power supply module, for bus controller and internal I/O power supply, X2X Link power supply, supply not electrically isolated	
	System modules for expandable bus controllers	
X20BB81	X20 bus base, for X20 base module (BC, HB, etc.) and X20 power supply module, with one expansion slot for an X20 add-on module (IF, HB, etc.), X20 locking plates (left and right) X20AC0SL1/X20AC0SR1 included	
	System modules for the X20 hub system	
X20HB28G0	X20 EtherCAT junction module, integrated 2-port EtherCAT junction, 2x RJ45	
	Terminal blocks	
X20TB12	X20 terminal block, 12-pin, 24 VDC keyed	

Table 1: X20BC00G3, X20BC80G3 - Order data

2.1.2 Technical data

Model number	X20BC00G3	X20BC80G3
Short description		
Bus controller	EtherCAT slave	EtherCAT bus controller with one slot for 1 junction module
General information		
B&R ID code	0xAC23	0xAEC2
Status indicators	Module status, bus function	
Diagnostics		
Module status	Yes, using status LED and software	
Bus function	Yes, using status LED and software	
Power consumption		
Bus	1.68 W	1.79 W
Additional power dissipation caused by actuators (resistive) [W]	-	
Certifications		
CE	Yes	
KC	Yes	
UL	cULus E115267 Industrial control equipment	
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5	
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X	
Interfaces		
Fieldbus	EtherCAT slave	
Variant	2x shielded RJ45	
Line length	Max. 100 m between 2 stations (segment length)	

Table 2: X20BC00G3, X20BC80G3 - Technical data

Model number	X20BC00G3	X20BC80G3
Transfer rate	100 Mbit/s	
Transfer		
Physical layer	100BASE-TX	
Half-duplex	Yes	
Full-duplex	Yes	
Autonegotiation	Yes	
Auto-MDI / MDIX	Yes	
Hub propagation delay	750 ns	750 ns 1)
Min. cycle time 2)		
Fieldbus	200 µs	
X2X Link	200 µs	
Synchronization between bus systems possible	Yes	
Electrical properties		
Electrical isolation	EtherCAT isolated from bus and I/O	
Operating conditions		
Mounting orientation		
Horizontal	Yes	
Vertical	Yes	
Installation elevation above sea level		
0 to 2000 m	No limitations	
>2000 m	Reduction of ambient temperature by 0.5°C per 100 m	
Degree of protection per EN 60529	IP20	
Ambient conditions		
Temperature		
Operation		
Horizontal mounting orientation	0 to 55°C	
Vertical mounting orientation	0 to 50°C	
Derating	-	
Storage	-25 to 70°C	
Transport	-25 to 70°C	
Relative humidity		
Operation	5 to 95%, non-condensing	
Storage	5 to 95%, non-condensing	
Transport	5 to 95%, non-condensing	
Mechanical properties		
Note	Order 1x terminal block X20TB12 separately Order 1x power supply module X20PS9400 or X20PS9402 separately Order 1x bus base X20BB80 separately	Order 1x terminal block X20TB12 separately Order 1x power supply module X20PS9400 or X20PS9402 separately Order 1x bus base X20BB81 separately
Spacing 3)	$37.5^{+0.2}$ mm	$62.5^{+0.2}$ mm

Table 2: X20BC00G3, X20BC80G3 - Technical data

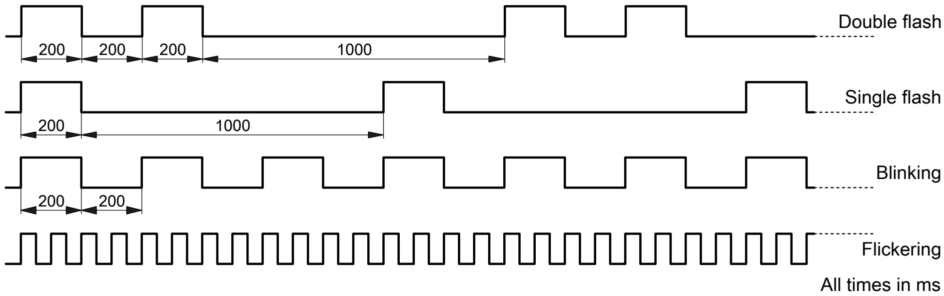
1) For the interfaces on the X20HB28G0 EtherCAT junction module, the hub propagation delay is 1.1 µs instead of 750 ns.
2) The minimum cycle time defines how far the bus cycle can be reduced without communication errors occurring.
3) Spacing is based on the width of bus base X20BB80. In addition, power supply module X20PS9400 or X20PS9402 is always required for the bus controller.

2.1.3 LED status indicators

Figure	LED	Color	Status	Description
	STATUS[1)]	Green	On	The bus controller is in state OPERATIONAL.
			Blinking	State PREOPERATIONAL
			Single flash	State SAFE-OPERATIONAL
			Flickering	The bus controller has started and is not yet in state INIT, or it is in state BOOTSTRAP (e.g. during firmware download).
			Off	State INIT
		Red	On	A critical communication or application error has occurred.
			Blinking	Invalid configuration data
			Single flash	The bus controller has an internal error and has changed the EtherCAT state on its own.
			Double flash	Watchdog timeout (process data watchdog or EtherCAT watchdog)
			Flickering	Error in startup procedure (state INIT achieved, but the error indicator bit in the AL status register is set)
			Off	No error
	L/A IN L/A OUT	Green	Blinking	The respective LED blinks when Ethernet activity is taking place (PORT OPEN) on the corresponding RJ45 interface (IN, OUT).
			On	A connection (link) is established, but no communication is taking place (PORT OPEN).
			Off	No physical Ethernet connection exists (PORT CLOSED).

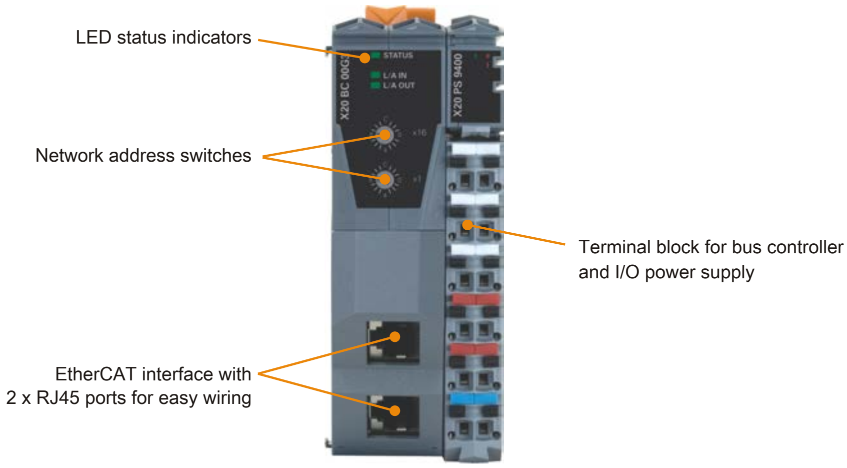
1) LED "STATUS" is a green/red dual LED used to indicate EtherCAT states ERROR and RUN.

LED status indicators - Blink times



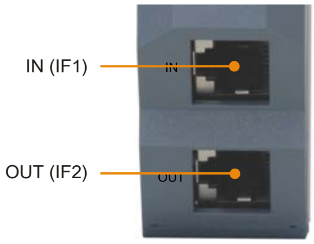
For error descriptions related to LED "Error", see also "LED "Error" - Error codes" on page 17.

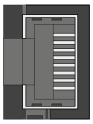
2.1.4 Operating and connection elements



2.1.5 Ethernet interface

For information about wiring X20 modules with an Ethernet interface, see section "Mechanical and electrical configuration - Wiring guidelines for X20 modules with Ethernet cables" of the X20 user's manual.



Interface	Pinout		
	Pin	Ethernet	
 1 Shielded RJ45	1	RXD	Receive data
	2	RXD\	Receive data\
	3	TXD	Transmit data
	4	Termination	
	5	Termination	
	6	TXD\	Transmit data\
	7	Termination	
	8	Termination	

2.1.6 EtherCAT network address switches

A slave alias address can be set using the 2 network address switches on the bus controller. During the initialization phase (during startup), the bus controller writes the value of the address switches to ESC register 0x12 or 0x13. However, this value is only applied in the corresponding registers if the switch value is between 0x00 and 0xFA (decimal 250).

Switch position	Description
0x00 to 0xFA	Writes the address switch value to the "Station alias" register.
0xFB to 0xFE	Does not apply the address switch value. ESC alias registers are not changed.
0xFF	Does not apply the address switch value. ESC alias registers are not changed. If the address switches are set to value 0xFF and the bus controller is restarted, then it will boot with the default values. All set parameters remain unchanged in flash memory.

The master determines whether the alias address is used to address the slave by setting the corresponding bit (bit 24) in the ESC DL control register.

2.1.6.1 Deleting parameters

Various parameters can be stored in the bus controller's flash memory. Deleting these parameters using switch position F0 returns the bus controller to its factory settings.

Deleting the parameters

1. Switch off the power supply to the bus controller.
2. Set the node number to F0.
3. Switch on the power supply to the bus controller.
4. Wait until LED "STATUS" flashes green.
5. Set the node number switch to 0 and then back to F0.
6. Wait until LED "STATUS" blinks (parameters have been deleted).
7. Switch the power supply to the bus controller off and then back on.
8. The bus controller boots with its default settings.

2.2 X67 bus controller

2.2.1 Order data

Model number	Short description	Figure
	Bus controller modules	
X67BCG321.L12	X67 bus controller, 1 EtherCAT interface, X2X Link power supply 15 W, 16 digital channels configurable as inputs or outputs, 24 VDC, 0.5 A, configurable input filter, 2 event counters 50 kHz, M12 connectors, high-density module	

Table 3: X67BCG321.L12 - Order data

Required accessories
For a general overview, see section "Accessories - General overview" of the X67 system user's manual.

2.2.2 Technical data

Model number	X67BCG321.L12
Short description	
Bus controller	EtherCAT
General information	
Inputs/Outputs	16 digital channels, configurable as inputs or outputs using Automation Studio or data point, inputs with additional functions
Isolation voltage between channel and bus	500 V_{eff}
Nominal voltage	24 VDC
B&R ID code	
Bus controller	0xACF8
Internal I/O module	0xB402
Sensor/Actuator power supply	0.5 A summation current
Status indicators	I/O function for each channel, supply voltage, bus function
Diagnostics	
Outputs	Yes, using status LED and software
I/O power supply	Yes, using status LED and software
Connection type	
Fieldbus	M12, D-keyed
X2X Link	M12, B-keyed
Inputs/Outputs	8x M12, A-keyed
I/O power supply	M8, 4-pin
Power output	15 W X2X Link supply for I/O modules
Power consumption	
Fieldbus	2.5 W
Internal I/O	0.5 W
X2X Link power supply	15% of the power output for X2X Link
Additional power dissipation caused by actuators (resistive) [W]	0.6
Certifications	
CE	Yes
KC	Yes
UL	cULus E115267 Industrial control equipment
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5
ATEX	Zone 2, II 3G Ex nA IIA T5 Gc IP67, Ta = 0 - Max. 60°C TÜV 05 ATEX 7201X
Interfaces	
Fieldbus	EtherCAT slave
Variant	M12 interface (female) 2x on the module
Line length	Max. 100 m between 2 stations (segment length)
Transfer rate	100 Mbit/s

Table 4: X67BCG321.L12 - Technical data

Model number	X67BCG321.L12
Transfer	
Physical layer	100BASE-TX
Half-duplex	Yes
Full-duplex	Yes
Autonegotiation	Yes
Auto-MDI / MDIX	Yes
Hub propagation delay	750 ns
Min. cycle time [1)]	
Fieldbus	200 µs
X2X Link	200 µs
Synchronization between bus systems possible	Yes
I/O power supply	
Nominal voltage	24 VDC
Voltage range	18 to 30 VDC
Integrated protection	Reverse polarity protection
Power consumption	
Sensor/Actuator power supply	Max. 12 W [2)]
Sensor/Actuator power supply	
Voltage	I/O power supply minus voltage drop for short circuit protection
Voltage drop for short-circuit protection at 0.5 A	Max. 2 VDC
Summation current	Max. 0.5 A
Short-circuit proof	Yes
Digital inputs	
Input voltage	18 to 30 VDC
Input current at 24 VDC	Typ. 4 mA
Input characteristics per EN 61131-2	Type 1
Input filter	
Hardware	≤10 µs (channels 1 to 4) / ≤70 µs (channels 5 to 16)
Software	Default 0 ms, configurable between 0 and 25 ms in 0.2 ms intervals
Input circuit	Sink
Additional functions	50 kHz event counting, gate measurement
Input resistance	Typ. 6 kΩ
Switching threshold	
Low	<5 VDC
High	>15 VDC
Event counter	
Quantity	2
Signal form	Square wave pulse
Evaluation	Each falling edge, cyclic counter
Input frequency	Max. 50 kHz
Counter 1	Input 1
Counter 2	Input 3
Counter frequency	Max. 50 kHz
Counter size	16-bit
Gate measurement	
Quantity	1
Signal form	Square wave pulse
Evaluation	Rising edge - Falling edge
Counter frequency	
Internal	48 MHz, 3 MHz, 187.5 kHz
Counter size	16-bit
Length of pause between pulses	≥100 µs
Pulse length	≥20 µs
Supported inputs	Input 2 or input 4
Digital outputs	
Variant	FET positive switching
Switching voltage	I/O power supply minus residual voltage
Nominal output current	0.5 A
Total nominal current	8 A
Output circuit	Source
Output protection	Thermal cutoff on overcurrent or short circuit, integrated protection for switching inductances, reverse polarity protection for output power supply
Diagnostic status	Output monitoring with 10 ms delay
Leakage current when switched off	5 µA
Switching on after overload cutoff	Approx. 10 ms (depends on the module temperature)
$R_{DS(on)}$	150 mΩ
Residual voltage	<0.15 V at 0.5 A nominal current
Peak short-circuit current	<12 A
Switching delay	
0 → 1	<400 µs
1 → 0	<400 µs

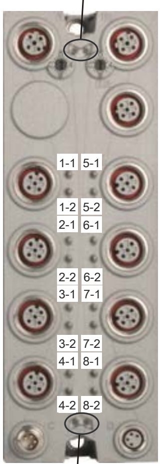
Table 4: X67BCG321.L12 - Technical data

Model number	X67BCG321.L12
Switching frequency	
Resistive load	Max. 100 Hz
Inductive load	See section "Switching inductive loads"
Braking voltage when switching off inductive loads	50 VDC
Electrical properties	
Electrical isolation	Bus isolated from EtherCAT and channel Channel not isolated from channel
Operating conditions	
Mounting orientation	
Any	Yes
Installation elevation above sea level	
0 to 2000 m	No limitation
>2000 m	Reduction of ambient temperature by 0.5°C per 100 m
Degree of protection per EN 60529	IP67
Ambient conditions	
Temperature	
Operation	-25 to 60°C
Derating	-
Storage	-40 to 85°C
Transport	-40 to 85°C
Mechanical properties	
Dimensions	
Width	53 mm
Height	155 mm
Depth	42 mm
Weight	370 g
Torque for connections	
M8	Max. 0.4 Nm
M12	Max. 0.6 Nm

Table 4: X67BCG321.L12 - Technical data

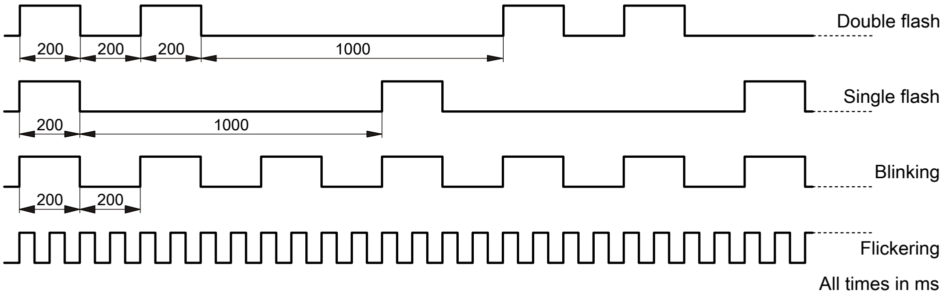
1) The minimum cycle time defines how far the bus cycle can be reduced without communication errors occurring.
2) The power consumption of the sensors and actuators connected to the module is not permitted to exceed 12 W.

2.2.3 LED status indicators

Figure	LED	Color	Status	Description
Status indicator 1: Left: green, right: red  Status indicator 2: Left: green, right: red	**Status indicator 1:** Status indicator for Ethernet activity			
	L/A IF[1]	Status indicator for Ethernet activity.		
		Green	Blinking	There is Ethernet activity (PORT OPEN) taking place on at least one of the EtherCAT connections.
			On	A connection has been established on at least one of the EtherCAT connections. However, there is no communication taking place (PORT OPEN).
			Off	An Ethernet connection has not been established on any of the EtherCAT connections (PORT CLOSED)
	STATUS[2]	Status indicator for the EtherCAT bus controller.		
		Green (RUN)	On	State OPERATIONAL
			Blinking	State PRE-OPERATIONAL
			Single flash	State SAFE-OPERATIONAL
			Flickering	The bus controller has started and is not yet in state INIT or it is in state BOOTSTRAP (e.g. during firmware download).
			Off	State INIT
		Red (ERROR)	On	A critical communication or application error has occurred.
			Blinking	Invalid configuration data
			Single flash	The bus controller has an internal error and has changed the EtherCAT state on its own.
			Double flash	Watchdog timeout (process data watchdog or EtherCAT watchdog)
			Flickering	Error in the start procedure (state INIT achieved but the error indicator bit in the AL status register is set)
			Off	No error
	I/O LEDs			
	1-1/2 to 8-1/2	Orange	-	Input/Output state of the corresponding channel.
	Status indicator 2: Status indicator for module functionality			
	Left	Green	Off	No power supplied to the module
			Single flash	Mode RESET
			Blinking	Mode PREOPERATIONAL
			On	Mode RUN
	Right	Red	Off	No power to module or everything OK
			On	Error or reset state
			Single flash	Warning/Error on an I/O channel. Level monitoring for digital outputs has been triggered.
			Double flash	Supply voltage not in the valid range

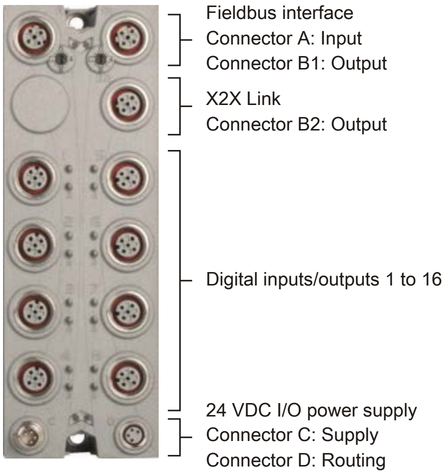
1) LED "L/A IF" shows the signals of the 2 EtherCAT interfaces combined (IN and OUT).
2) LED "STATUS" is a green/red dual LED used to indicate EtherCAT states ERROR (red) and RUN (green).

LED status indicators - Blink times



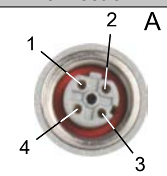
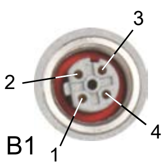
For error descriptions related to LED "Status indicator 1", see also "LED "Error" - Error codes" on page 17.

2.2.4 Operating and connection elements



2.2.5 EtherCAT interfaces

The module is connected to the network using pre-assembled cables. The connection is made using M12 circular connectors.

Connection	Pinout		
	Pin	Name	
 	1	TXD	Transmit data
	2	RXD	Receive data
	3	TXD\	Transmit data\
	4	RXD\	Receive data\
	Shield connection made via threaded insert in the module A → D-keyed (female), input B1 → D-keyed (female), output		

Information:

The color of the wires used in field-assembled cables for connecting to the fieldbus interface may deviate from the standard.

It is extremely important to make sure that the pinout is correct (see section "Accessories - POWERLINK cables" in the X67 user's manual).

2.2.5.1 Cabling guidelines for bus controllers with Ethernet cables

Some X67 system bus controllers are based on Ethernet technology. POWERLINK cables supplied by B&R can be used for wiring.

Model number	Connection type
X67CA0E41.xxxx	Attachment cables - RJ45 to M12
X67CA0E61.xxxx	Connection cables - M12 to M12

The following cabling guidelines must be observed:

- Use Cat 5 SFTP cables.
- Observe the minimum cable bend radius (see data sheet for the cable).

Information:

Using POWERLINK cables supplied by B&R (X67CA0E61.xxxx and X67CA0E41.xxxx) satisfies product standard EN 61131-2.

The customer must implement additional measures in the event of further requirements.

2.2.6 EtherCAT network address switches

Information:

The network address switches on this bus controller have no function.

3 Basic information

3.1 General information

The EtherCAT bus controller enables modular X20, X67 and XV I/O systems from B&R to be integrated into EtherCAT. Up to 253 I/O modules can be connected to a bus controller using the X2X Link network. The X2X Link network is synchronized with the EtherCAT fieldbus in order to achieve optimal performance.

During bus controller startup, the connected I/O modules are detected and their I/O data points are mapped to a corresponding modular I/O mapping.
All mapping information is represented in a CANopen-compatible object dictionary. This gives the user or master system the ability to read all relevant information about the CoE mechanism (CANopen over EtherCAT) and to then automatically assemble the I/O mapping on the master.

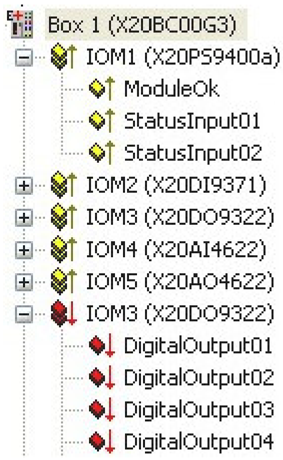


Figure 1: Module and register overview with a user-defined configuration

ESI file

To connect and operate the bus controller in an EtherCAT master environment, an ESI file is required. This is an XML file that contains all the parameters needed to configure the bus controller and communicate with it.

The ESI file can be downloaded from the B&R website (www.br-automation.com) in the Downloads section for the respective bus controller. Due to the large number of possible EtherCAT master environments, the ESI file cannot be installed automatically. Instead, it must usually be copied manually to the corresponding installation directory of the master environment (e.g. \TwinCAT\3.1\Config\Io\EtherCAT).

3.1.1 EtherCAT interface

From an Ethernet standpoint, the EtherCAT bus is simply a single large Ethernet station. This "station" receives and transmits Ethernet telegrams. The station does not include an Ethernet controller with downstream microprocessor, however, but rather a large number of EtherCAT slaves. These slaves process the incoming telegrams during the cycle and extract the relevant payload data or add it and then forward the telegram to the next EtherCAT slave. The last EtherCAT slave then returns the now fully processed telegram so that it will be sent back to the master by the first slave as a kind of response telegram. This procedure utilizes the fact that Ethernet deals separately with transfers in separate directions (Tx and Rx lines) and operates in full-duplex mode.

The telegrams are processed directly "on the fly". While the telegrams are delayed by only a few bits before being forwarded, the slave recognized relevant commands and executes them accordingly. Processing is handled within the hardware and is therefore independent of slave response times. Each station has an addressable memory of 64 kB that can be read or written, either consecutively or simultaneously. Several EtherCAT commands can be embedded within an Ethernet telegram, each addressing individual stations and/or memory areas.

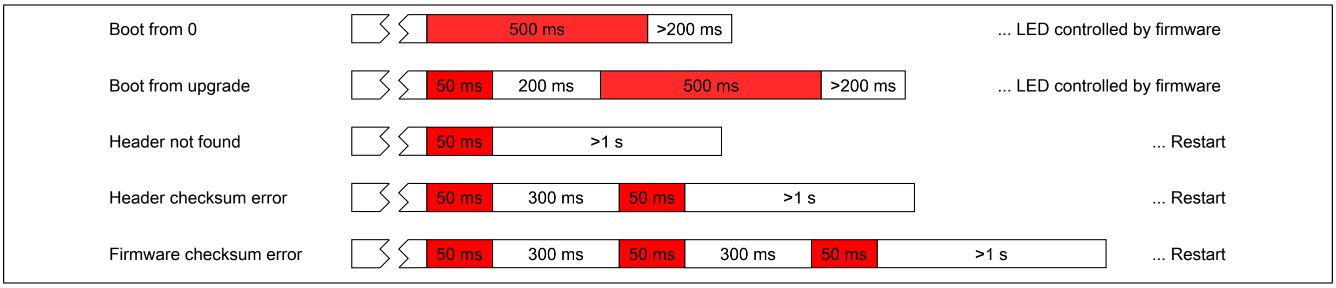
3.2 Bootup procedure

Initialization takes place after the operating voltage has been switched on. The bus controller determines the input and output data of the individual I/O modules, takes any saved configurations into account and generates the process image.
If there is a problem during startup, this is indicated by the blink code output by the LED status indicator on the bus controller (see product data sheet).

> ## Information:
>
> **An existing configuration can be deleted by clearing the flash memory with CoE object "2100:01" (see section "Bus controller services (0x2100)" on page 26). This will reset the bus controller to its factory settings.**
>
> **On the X20 bus controller, parameters can also be deleted using the node number switches.**

3.2.1 Blink codes during startup

The boot loader indicates the following states on module status LED "STATUS":



If faulty firmware in flash memory causes an error during booting, then the system will attempt to reboot using the factory default boot block.
This means that if an error occurs in the firmware upgrade sector, the module will automatically revert to the factory default sector (boot from 0).

3.2.2 Forcing a boot from the factory default sector

This is necessary if firmware has been stored in the upgrade sector and operates the watchdog correctly but does not allow the booting process to occur without errors. The boot loader would start the defective firmware, no longer providing a way to perform a subsequent update.

To force a boot from the factory default sector, one of the network address switches must be moved continuously during booting. This is detected by the boot loader, which causes module status LED "STATUS" to begin flickering rapidly in red. After 1 second passes in which the network address switch is no longer changed, the bus controller restarts using the factory default boot sector and the current value of the network address switches.

3.3 LED "Error" - Error codes

In addition to indicating runtime monitoring (expired watchdog) by flashing red twice, constant red blinking indicates an internal error on the bus controller and steadily flickering red indicates configuration problems.

Possible AL status code errors

LED signal	Error
Double flash	"WATCHDOG_EXPIRED"
Blinking	"IOM_FAILED" "IOM_MISMATCH" "INVALID_CFG_DATA" "INVALID_ESI_DATA"
Flickering	"OUT_OF_RESOURCES" "FLASH_READ_FAILED" "INVALID_FIRMWARE" "ECAT_EEPROM_FAILED" "IOM_UNSUPPORTED"

For information about the status of the bus controller, see the EtherCAT AL status code or CoE object "Bus controller state (0xF100)" on page 26.

A missing I/O module (error state "MODULE_MISSING") is not indicated via LED. This state occurs if the bus controller is being operated with a user-defined configuration (see section "I/O configuration" on page 18) and a configured I/O module is not physically present.

Information:

CoE object "IOM failed action" 0x2000:05 or parameter "I/O module failed action" in the configuration dialog box for the bus controller in Automation Studio V4.3 or later can be used to select whether the bus controller should switch to state SAFE-OPERATIONAL (default behavior) or remain in state OPERATIONAL (no action) when an I/O module fails during operation.

If a configured I/O module is not already present during startup (after the initialization delay time has expired, see section "Bus controller system parameters (0x2000)" on page 25), the bus controller can still be switched to state OPERATIONAL. The state of each I/O module is indicated by its "ModuleOK" input data point (see section "Additional information - General data points - ModuleOK" in the X20 user's manual).

4 I/O configuration

The bus controller supports the following operating modes for configuring I/O modules.

- Automatic configuration
- Configuration in Automation Studio

In all operating modes, it is possible to change the module configuration at runtime using CoE object 0xF801.

4.1 Automatic configuration

With this type of configuration, after the initialization delay time expires the bus controller queries all modules on the X2X Link network for their cyclic input and output data and automatically creates a corresponding image in the CoE object dictionary. I/O modules that are started later are not taken into account in the process image.

This operating mode uses function model 254 "Bus controller" on the I/O module.

4.2 Configuration in Automation Studio

The bus controller and all connected I/O modules can be configured using the FieldbusDESIGNER integrated in Automation Studio.

Automation Studio can be downloaded at no cost from the B&R website www.br-automation.com. The evaluation license is permitted to be used to create complete configurations for fieldbus bus controllers at no cost.

The user-defined configuration can be transferred to the bus controller using the configuration files generated by Automation Studio. The bus controller saves the configuration to its flash memory and may restart if necessary to allow the changed configuration to become active.

> **Information:**
>
> **The bus controller is automatically restarted if changes are made to the configuration data.**

To operate the bus controller, the ESI file must be integrated in the EtherCAT master environment (see "ESI file" on page 15) and the applicable Automation Studio configuration must be transferred.

The following options are available for transferring the configuration to the bus controller:

- "Downloading the binary file using FoE" on page 19
- "Importing the XML device description file" on page 19
- "Configuration via CoE objects" on page 20
- "Configuration using an ESI file" on page 22

> **Information:**
>
> **If the bus controller has been configured using a configuration file, then the bus controller system parameters (see "Bus controller system parameters (0x2000)" on page 25) can no longer be edited via CoE.**
>
> **In order to use the Automatic configuration operating mode on a bus controller to which a user-defined configuration has previously been loaded, the steps in section "Bus controller services (0x2100)" on page 26 must be completed first.**

4.2.1 Downloading the binary file using FoE

The BIN file created by Automation Studio can be loaded to the bus controller using the "File Access over EtherCAT" (FoE) mechanism.

In master environments such as TwinCAT, the respective bus controller must be scanned for first, i.e. the master needs an online connection to the bus controller.

As long as the name of this bus controller is unknown in TwinCAT, it is identified as a "box". Tab "Online" for this box contains button "Download" that allows any files to be selected.

The path to the binary file is specified here, and a dialog box for downloading the configuration appears. If the binary file has been successfully downloaded and the included configuration contains changes from the current configuration, the bus controller and all connected I/O modules are restarted to enable the configuration. The box should now be deleted and searched for again so that the changed CoE object dictionary for the bus controller can be uploaded again according to the specifications in the ESI file (PdoUpload = True). This also causes the master to update its I/O mapping.

4.2.2 Importing the XML device description file

The XML device description file from Automation Studio contains all information about the structure of the process data (I/O data) as well as the configuration data for the bus controller and I/O modules.

This device description file can be imported into master environments such as TwinCAT or EtherCAT Configurator using menu option "*TwinCAT → Selected item →* Import XML description". To do this, the slave that should receive this configuration must first be added to the hardware tree and selected. This is done by right-clicking on the master and selecting "Add new item".

The XML device description file must not be confused with the ESI file (also an XML file). The ESI file serves to generally describe an EtherCAT slave and can be downloaded from the B&R website (see "General information" on page 15).

The advantage of this method is that all I/O data points are already known (name, type) in the development environment without actually having to have a physical device present. The alternative would be to download the configuration to the device over an existing online connection and then to upload all information from the CoE object dictionary.

In addition, the XML device description file specifies that configuration data must be transmitted to the bus controller during startup (state transition from PRE-OPERATIONAL to SAFE-OPERATIONAL). This ensures that the master always backs up the configuration automatically.

4.2.3 Configuration via CoE objects

In the XML device description file generated by Automation Studio, the data under "Mailbox → CoE" can be used to configure the B&R bus controller using CoE objects.

The number of X2X modules configured in Automation Studio determines how many CFG startup streams are listed here. This depends on the amount of configuration data and the complexity of the X2X modules.

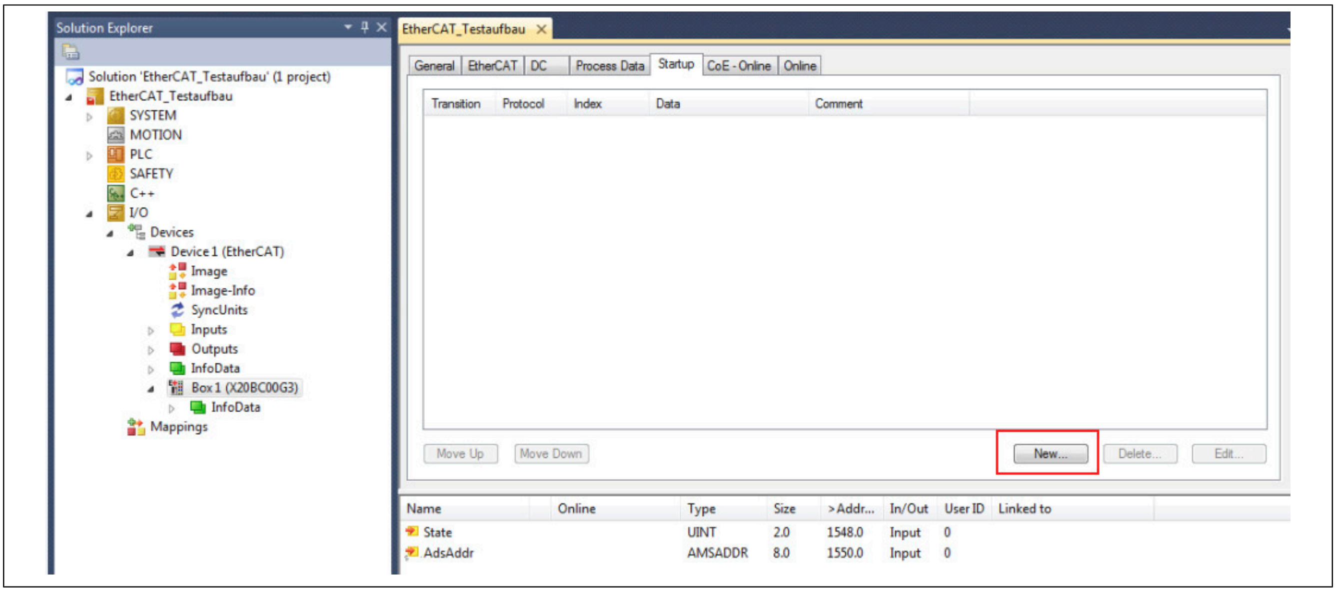
```
</ProcessData>
<Mailbox>
    <CoE>
        <InitCmds>
            <InitCmd>
                <Transition>PS</Transition>
                <Comment><![CDATA[ST1.X20BC00G3 CFG Startup Stream(1)]]></Comment>
                <Timeout>0</Timeout>
                <Ccs>1</Ccs>
                <Index>8193</Index>
                <SubIndex>0</SubIndex>
                <Data>0000FC3594BD0000017B000000D20000000000000000000000000000000001
            </InitCmd>
            <InitCmd>
                <Transition>PS</Transition>
                <Comment><![CDATA[ST1.X20BC00G3 CFG Startup Stream(2)]]></Comment>
                <Timeout>0</Timeout>
                <Ccs>1</Ccs>
                <Index>8193</Index>
                <SubIndex>0</SubIndex>
                <Data>FFFF75743031004469676974616C4F75747075743032004469676974616C4F
            </InitCmd>
        </InitCmds>
    </CoE>
</Mailbox>
```

In this example, there are only 2 CFG startup streams that are written via 2 CoE objects.

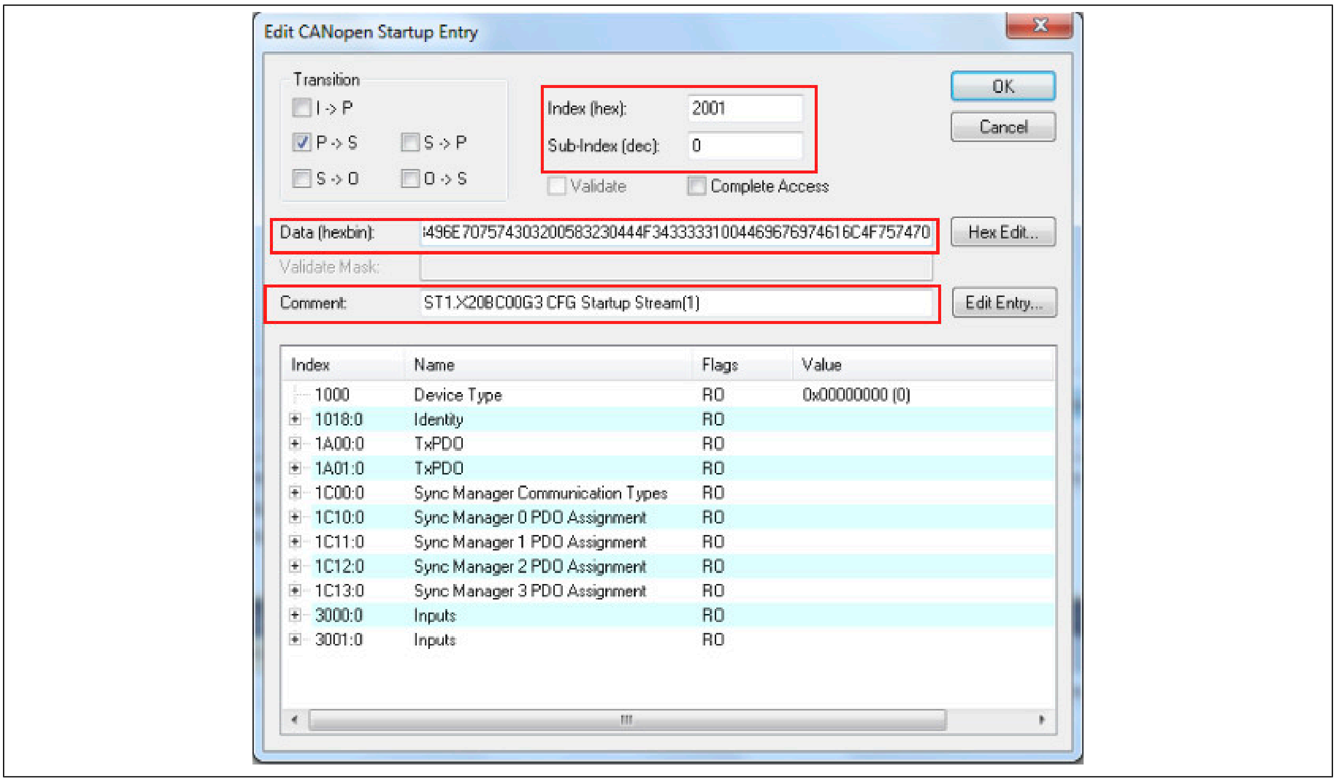
4.2.3.1 Example

In this example, TwinCAT is used as the master environment. First, the ESI file from the B&R slave is copied to the TwinCAT folder and the B&R bus controller is imported into the project in the TwinCAT environment by right-clicking and selecting "Insert new item".

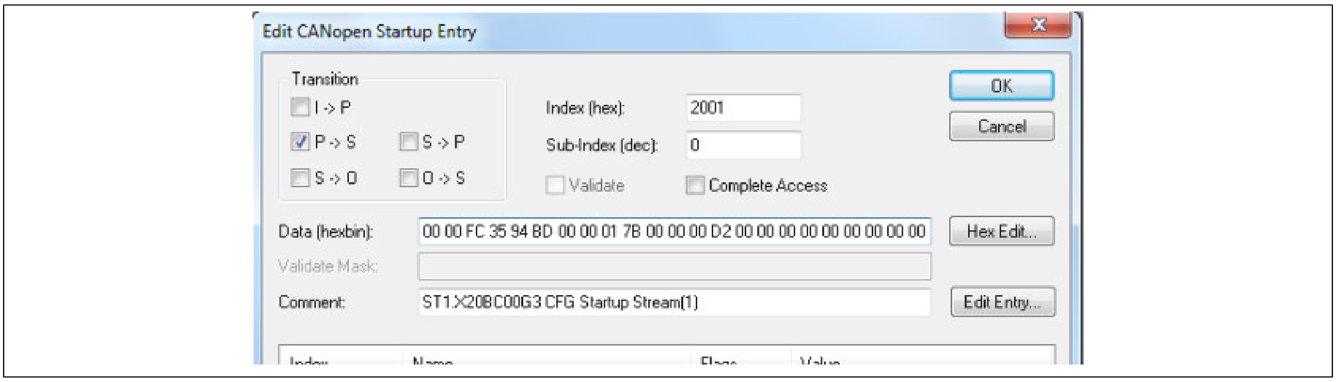
● Select the bus controller and create a new CoE object by clicking on "New" under "Startup".



● In window "Edit CANopen startup entry", enter the "CFG startup stream(1)" from the XML device description file generated by Automation Studio. This is done by entering the index, subindex, data and comment from the device description file in the startup window.

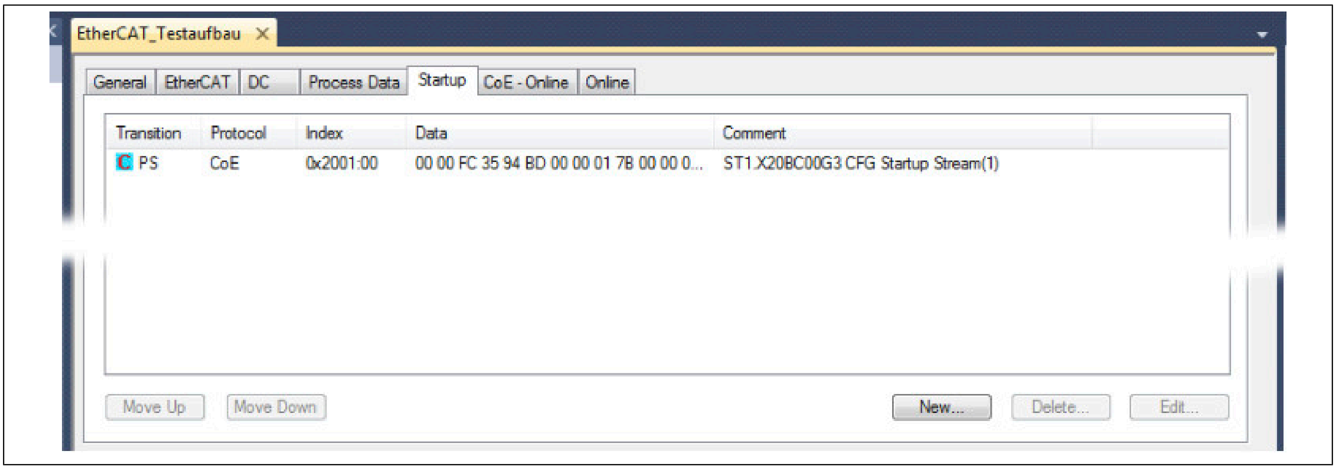


To specify the stream correctly, there must be a space between each of the bytes.



Click "OK" to confirm the entry and close the dialog box.

● The newly created stream is displayed in the TwinCAT environment. Click "New" to create additional streams.



● Once all necessary streams have been created, transfer the entire project to the bus controller.

4.2.4 Configuration using an ESI file

In Automation Studio V4.4.1 and later, a complete ESI file is additionally saved in the project's output folder when the project is generated.

This ESI file is based on the general ESI file available on the website and has been expanded to include the set configuration and process image. This one file is all that is needed to provide the master with the complete device description. It is therefore no longer necessary to download the ESI file from the B&R website. See "ESI file" on page 15

Information:

With this type of configuration, the general ESI file on the website is no longer needed.

5 The object dictionary

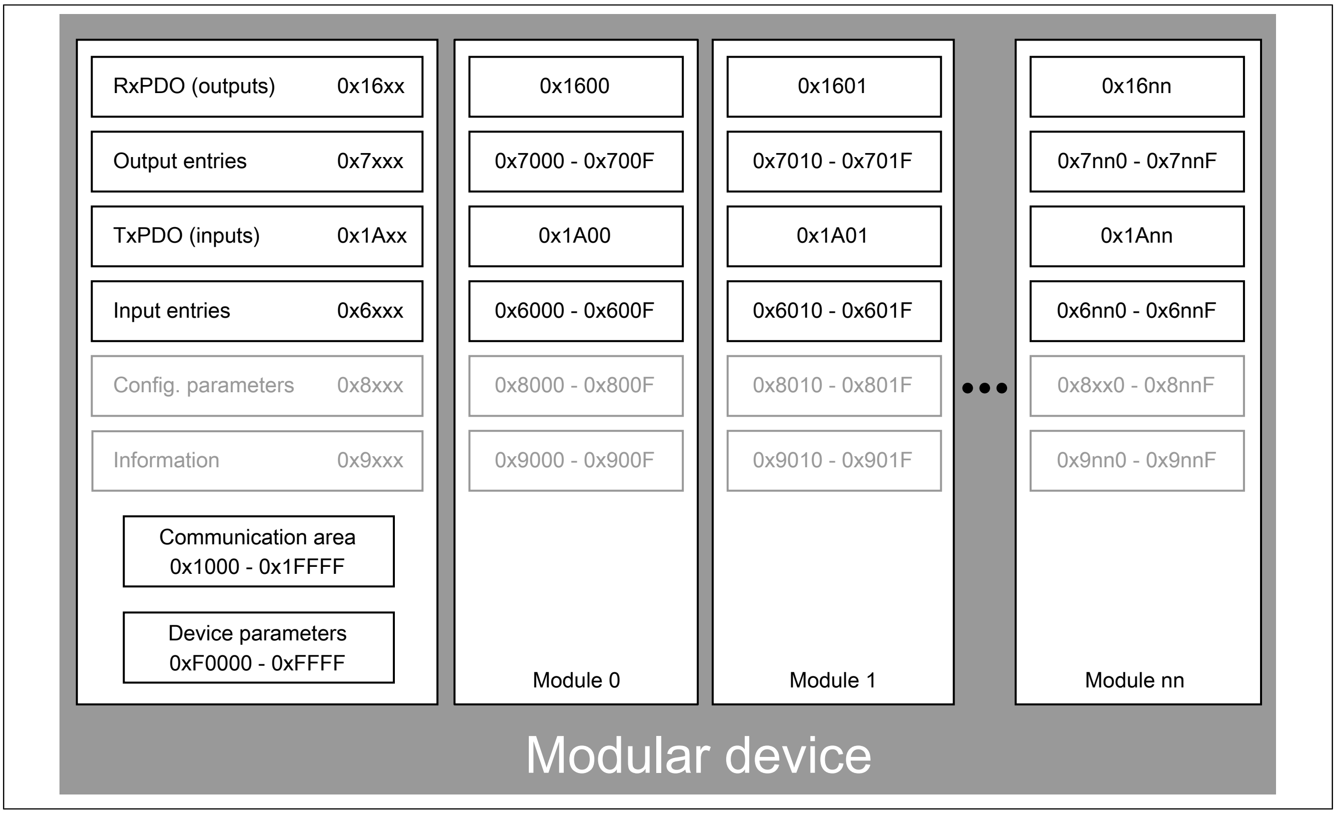
The EtherCAT object dictionary is heavily based on CANopen; for this reason, it is also referred to as a "CANopen over EtherCAT" or CoE object dictionary. The objects are filled with current values (e.g. PDO mapping 0x16... and 0x1a..., input data 0x6..., output data 0x7...) by the bus controller in both operating modes. In addition to these specified objects, manufacturer-specific objects are also available for configuration and operation.

Index (hex)		Object dictionary area
0x0000 - 0x0FFF		Data type area
0x1000 - 0x1FFF		Communication area
0x2000 - 0x5FFF		Manufacturer-specific area
	0x2000 Bus controller system parameters	
	0x2100 Bus controller services	
0x6000 - 0x6FFF		Input area
0x7000 - 0x7FFF		Output area
0x8000 - 0x8FFF		Configuration area
0x9000 - 0x9FFF		Information area
0xA000 - 0xAFFF		Diagnostics area
0xB000 - 0xBFFF		Service transfer area
0xC000 - 0xEFFF		Reserved area
0xF000 - 0xFFFF		Device area
	0xF100 Bus controller status	
	0xF120 Bus controller mapping information	
	0xF800 I/O module register read	
	0xF810 I/O module register write	
	0xF810 I/O module information	

For information about objects outside "Manufacturer-specific area" and "Device area", please see the EtherCAT specification.

5.1 The modular object model

During bus controller startup, the connected I/O modules are configured and their I/O data points mapped to a corresponding modular I/O mapping. All of the mapping information is then reflected in the dynamically created object dictionary and can be queried by the master via CoE. The object dictionary is structured according to the modular I/O system and appears as follows:



Master environments usually begin the numbering of I/O modules with 1 instead of 0. TwinCAT calls the first module "IOM1", for example. The associated RxPDOs and TxPDOs here are located at 0x1600 and 0x1A00. For "IOM2", they are located at 0x1601 and 0x1A01.

The areas shown in gray in the image are not supported by the bus controllers.
For I/O configuration options, see "I/O configuration" on page 18.

5.2 Description of B&R-specific objects

5.2.1 Bus controller system parameters (0x2000)

Access rights for the bus controller system parameters

The bus controller can be configured using a configuration file from Automation Studio or CoE objects.

Information:

If the bus controller was configured using a configuration file, then all bus control system parameters will be set to "Read-only" (RO). In this case, parameters cannot be changed via CoE!

Index number: 0x2000:xx

<table>
<tr><th>Subindex (xx)</th><th>Name</th><th>Access</th><th>Function</th><th>Default value</th></tr>
<tr><td>0</td><td>Bus controller
System parameters</td><td>RO</td><td>Object identifier</td><td>5</td></tr>
<tr><td>01</td><td>X2X bus configuration[1]</td><td>RO/RW</td><td>Values: 0 to 9
The X2X Link cycle time and the data width achieved are 2 system parameters that cannot be configured separately in any area.
This value can be optimized based on the required cycle time and number of connected I/O modules.
<table>
<tr><th>Value</th><th>Cycle time</th><th>Description</th></tr>
<tr><td>0</td><td>4.0 ms</td><td>Max. 253 I/O modules, max. 1400 bytes sync data</td></tr>
<tr><td>1</td><td>3.5 ms</td><td>Max. 253 I/O modules, max. 1150 bytes sync data</td></tr>
<tr><td>2</td><td>3.0 ms</td><td>Max. 253 I/O modules, max. 900 bytes sync data</td></tr>
<tr><td>3</td><td>2.5 ms</td><td>Max. 200 I/O modules, max. 800 bytes sync data</td></tr>
<tr><td>4</td><td>2.0 ms</td><td>Max. 200 I/O modules, max. 500 bytes sync data</td></tr>
<tr><td>5</td><td>1.5 ms</td><td>Max. 100 I/O modules, max. 450 bytes sync data</td></tr>
<tr><td>6</td><td>1.0 ms</td><td>Max. 100 I/O modules, max. 260 bytes sync data</td></tr>
<tr><td>7</td><td>500 µs</td><td>Max. 40 I/O modules, max. 120 bytes sync data</td></tr>
<tr><td>8</td><td>250 µs</td><td>Max. 15 I/O modules, max. 40 bytes sync data</td></tr>
<tr><td>9</td><td>200 µs</td><td>Max. 10 I/O modules, max. 20 bytes sync data</td></tr>
</table></td><td>0x0006 (6)</td></tr>
<tr><td>02</td><td>X2X bus cable length[1]</td><td>RO/RW</td><td>Values: 0 to 25300 m
Used to optimize X2X Link timing with respect to low ESD emissions. If set to the default value of zero, no optimization takes place.
The actual total length of the X2X Link network line starting from the bus controller must be specified in meters. The maximum length is determined by the maximum distance between two X2X Link stations (100 m) and the maximum number of stations (253 modules), which equals in total 25.3 km.</td><td>0x0000 (0)</td></tr>
<tr><td>03</td><td>X2X bus reserved
Parameter</td><td>RO/RW</td><td>Not used</td><td>0x0000 (0)</td></tr>
<tr><td>04</td><td>IOM initialization
Delay[1]</td><td>RO/RW</td><td>Values: 3000 to 60000 ms
This object can be used to configure or read out the module initialization delay. This delay is specified in milliseconds.
After restarting, the system enters a module initialization phase. It is not possible to change to state PRE-OPERATIONAL during this phase. LED "Status" blinks green quickly to indicate that the initialization phase is in progress.
This phase is extended by the value set for the initialization delay.
This allows the system to compensate for variations in the time it takes for connected modules to be initialized. The bus controller is thus forced to wait longer for module initialization to be completed. If a value less than 3000 ms has been set, then the default value of 3000 ms will be used internally.
The total duration of the initialization phase is the sum of the boot durations of the I/O modules being used and the specified I/O module initialization value.</td><td>0x0BB8 (3000)</td></tr>
<tr><td>5</td><td>IOM failed action</td><td>RO/RW</td><td>Defines whether the bus controller should switch to state SAFE-OPERATIONAL (value 1, default behavior) or stay in state OPERATIONAL (value 0, no action) when an I/O module fails during operation.
If a configured I/O module is not already present during startup (after "IOM initialization delay" has expired), the bus controller can still be switched to state OPERATIONAL.</td><td>0x0001 (1)</td></tr>
</table>

1) **The software/hardware must be restarted in order to apply changes that have been made.**

5.2.2 Bus controller services (0x2100)

This object can be used to trigger a wide variety of service functions on the bus controller.

The respective service function is triggered by writing value "0x0BCA" to the respective index (e.g. 0x2100:02).

In order to use the automatic configuration operating mode on a bus controller to which a user-defined configuration has previously been loaded, the following 3 steps must be completed:

1. Delete the configuration file (0x2100:01).
2. Reset the system settings (0x2100:02).
3. Restart the system (0x2100:03).

Index number: 0x2100:xx

Subindex (xx)	Name	Access	Function	Default
0	Bus controller services	RO	Object identifier	3
01	Delete config file	RW	Deletes the configuration data in the bus controller's flash memory.	0x0000
02	Delete system parameters	RW	Resets the system parameters in flash memory (see section "Bus controller system parameters (0x2000)" on page 25) to their default values.	0x0000
03	Restart bus controller	RW	Forces an immediate restart of the bus controller.	0x0000

5.2.3 Bus controller state (0xF100)

This object indicates the bus controller's current operating state.
The value is made up of 4 bytes. The "upper" word indicates potential error states, while the "lower" word indicates the operating state (OK state). The value is bit-coded.

$$0x\underbrace{0000}_{\text{Error state}}\underbrace{0000}_{\text{OK state}}$$

OK state

Bit	Code	Function
0	0xxxxx0000	Everything OK
	0xxxxx0001	The bus controller was configured using configuration files from Automation Studio.

Error state

Bit	Code	Function
-	0x0000xxxx	OK, no error.
16	0x0001xxxx	**Watchdog expired** This error state is reset by transitioning from SAFE-OPERATIONAL to PRE-OPERATIONAL.
17	0x0002xxxx	**Bus controller flash memory read/write error** This error state is reset by restarting the bus controller. Before restarting, the bus controller system parameters should be deleted from flash memory (by setting CoE object "2100:02", see section "Bus controller services (0x2100)" on page 26).
18	0x0004xxxx	**Faulty or missing I/O module detected during operation** This error state is reset by correcting the error (e.g. connecting the missing I/O module to the corresponding slot).
19	0x0008xxxx	**Missing I/O module detected during boot phase** This error state indicates that the bus controller is being operated with a user-defined configuration and a configured module is missing. This error state can be reset by connecting the configured I/O modules or deleting the existing configuration (setting CoE object "2100:01", see section "Bus controller services (0x2100)" on page 26) and then restarting the bus controller (power cycle or setting CoE object "2100:03").
20	0x0010xxxx	**Incorrect I/O module detected during boot phase** Similar to state "Missing I/O module detected during boot phase". Instead of an empty or missing I/O slot, an I/O module was detected that does not match the configured module type. This error state is reset by correcting the error and then restarting the system (for details, see above under "Missing I/O module detected during the boot phase").
21	0x0020xxxx	**Error in bus controller configuration data** An error was detected in the configuration file from Automation Studio. In order to reset the error state, it is necessary to delete the existing configuration (by setting CoE object "2100:01", see section "Bus controller services (0x2100)" on page 26) as well as to delete the bus controller system parameters (by setting CoE object "2100:02"). A restart is then required (power cycle or setting CoE object "2100:03").
22	0x0040xxxx	**Faulty settings in the ESI file** The error state is reset by transitioning from INIT to PRE-OPERATIONAL.
23	0x0080xxxx	**No more bus controller resources available (memory problem)** This error state is reset by deleting the existing configuration (setting CoE object "2100:01", see section "Bus controller system parameters (0x2000)" on page 25). The system must then be restarted (power cycle or setting CoE object "2100:03").
24	0x0100xxxx	**Firmware error** This error can occur during a firmware update or when restarting after the update. If this error is indicated after restarting, a faulty firmware update was detected and the system was booted using the factory default firmware. For additional details, see section "Bootup procedure" on page 16. This error state is reset by transitioning from PRE-OPERATIONAL to SAFE-OPERATIONAL.
25	0x0200xxxx	**Error writing or reading ECAT EEPROM** This error state is reset by transitioning from PRE-OPERATIONAL to SAFE-OPERATIONAL.
26	0x0400xxxx	**Non-supported I/O module connected** This error state is reset by correcting the error (i.e. by replacing the I/O module and then performing a system restart).

See also "AL status code" on page 37.

5.2.4 Bus controller mapping information (0xF120)

The bus controller mapping provides information about the active I/O modules and their process data. These objects are read-only and cannot be written to.

Index number: 0xF120:xx

Subindex (xx)	Name	Access	Function	Default value
0	Bus controller Mapping information	RO	Object identifier	3
01	I/O module number	RO	Number of successfully linked or booted I/O modules. The I/O module does not have to be physically present if I/O modules were configured via a user-defined configuration file from Automation Studio. It will still be added to the mapping and counted in this object.	0x0000 (0)
02	PD input size	RO	Size of the input process data in bytes	0x0000 (0)
03	PD output size	RO	Size of the output process data in bytes	0x0000 (0)

5.2.5 Read I/O module register (0xF800)

This object allows direct read access to individual I/O module registers.
For a description of individual I/O registers, see the module documentation. The slot (X2X station number) of the affected I/O module must first be specified via subindex 1. The register is then read by setting the respective register address on subindex 2. The read value can be then be taken from subindex 3.

Index number: 0xF800:xx

Subindex (xx)	Name	Access	Function	Default value
0	Read I/O module register	RO	Object identifier	3
01	Slot ID	RW	I/O module slot ID 1 to x	0x0001 (1)
02	Register address	RW	I/O register address **The I/O register's read procedure is triggered by writing the register address.**	0x0000 (0)
03	Register value	RO	I/O register value	0x00000000 (0)

5.2.6 Write I/O module register (0xF801)

Object "0xF801" allows direct write access to individual I/O module registers.
This function is typically used for changing module parameters at runtime (state OPERATIONAL). For a description of individual I/O registers, see the module documentation.
The slot (X2X station number) of the affected I/O module must first be specified using subindex 1 and the register address specified using subindex 2. The respective register value is then set on subindex 3, after which the bus controller has write access to that register.

Information:

If this mechanism is used for writing to cyclic registers that are part of the I/O data (i.e. mapped as process data), then these registers will immediately be overwritten with cyclic I/O data in the next X2X cycle.

Index number: 0xF801:xx

Subindex (xx)	Name	Access	Function	Default value
0	Write I/O module register	RO	Object identifier	3
01	Slot ID	RW	I/O module slot ID 1 to x	0x0001 (1)
02	Register address	RW	I/O register address	0x0000 (0)
03	Register value	RW	I/O register value **The I/O register's write procedure is triggered by writing the register value.**	0x00000000 (0)

Procedure for I/O module configuration

1. Object "F801:01": I/O module station number (write to slot ID x)
2. Object "F801:02": Write to I/O module register address
3. Object "F801:03": Write to I/O module register value

Information about configuration via CoE objects

EtherCAT master systems often offer the ability to automate I/O register configurations. All of the necessary I/O register configurations can then be defined in CoE startup lists. The master processes these startup lists sequentially and transmits the I/O register data to the respective slave when a defined EtherCAT state transition occurs.

5.2.7 I/O module information (0xF810)

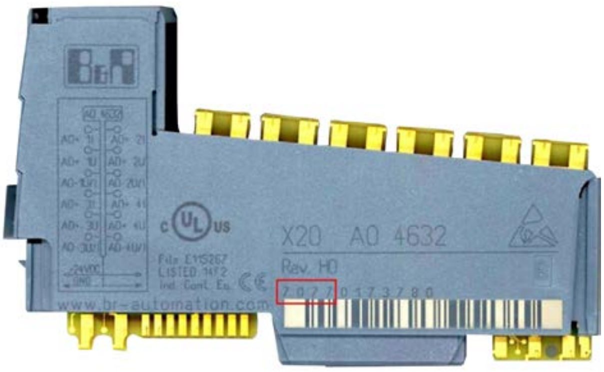
This object provides information from the electronic data sheets of all linked I/O modules. Writing the module slot (X2X Link station number) of the desired I/O module to subindex 1 will read all parameters and update all subindex values.

Index number: 0xF810:xx

Subindex (xx)	Name	Access	Function	Default value
0	I/O module information	RO	Object identifier	7
01	Slot ID	RW	I/O module slot ID 1 to x **All I/O module parameters are read when writing the slot ID.**	0x0001 (1)
02	Vendor code	RO	For B&R I/O modules: "0x006C"	0x006C (108)
03	Device ID	RO	I/O module hardware ID	0x1F8C (8076)
04	Hardware revision	RO	I/O module hardware variants	0x0005 (5)
05	Software version	RO	I/O module firmware version	0x0300 (768)
06	Serial number	RO	I/O module serial number	0x0002A332 (172850)
07	Status	RO	I/O module status (bit-coded) **Status code** – **Function** 0x0000 – Everything OK, no error 0x0001 – Missing I/O module detected during boot phase (warning) 0x0002 – I/O module mismatch detected during boot phase 0x0004 – I/O module not supported 0x0008 – Faulty or missing I/O module detected during operation	0x0000 (0)

Information regarding module device ID (hardware ID) and serial number

For the module hardware ID, see the corresponding module documentation (B&R ID code in the technical data). In addition, a serial number is printed on each electronics module; the module hardware ID corresponds to the first 4 positions of the serial number (see figure: the hardware ID is also colored black.)



X20 system

X67 system

Information:

IDs beginning from 9999 are printed as hexadecimal numbers and must be converted to their decimal values for comparison!

The other 7 digits correspond to the serial number (in decimal format).

5.3 Description of input and output data

The following image shows an X20 example configuration illustrating the composition of the objects in the communication area and input/output data.

Input data	
Offset	Register type
0	NWS
1	Alignment
2	AI
3	AI
4	AI
5	AI
6	AI
7	AI
8	NWS
9	NWS
10	Alignment
11	Alignment
12	AI
13	AI
14	AI
15	AI
16	AI
17	AI
18	AI
19	AI
20	AI
21	AI
22	AI
23	AI
24	Length in bytes

Output data	
Offset	Register type
0	AO
1	AO
2	AO
3	AO
4	AO
5	AO
6	AO
7	AO
8	AO
9	AO
10	AO
11	AO
12	AO
13	AO
14	Length in bytes

These assignments result from the automatic configuration of the I/O modules, i.e. without using a configuration file from Automation Studio:

The power supply module on slot 1 delivers a total of 8 bytes of input data:
1 byte network status, 1 alignment byte (because the following data point has a width of 16 bits and therefore has to be on an even byte offset) and 3 word registers.

Meaning of register types:

- NWS: Network status, see section "Network status" on page 31.
- Alignment: See section "Alignment rules" on page 34.
- AI: Analog input
- AO: Analog output

5.3.1 Network status

The network status provides information about the operating state of the individual X2X Link stations (i.e. the bus modules of the respective I/O modules).

Information:

Any value unequal to 245 (0xF5) means that the I/O data of the corresponding module is invalid. This circumstance is usually taken into account or processed accordingly in the application.

The operating state of the I/O modules (i.e. the electronics modules, in contrast to the network status of the bus modules described here) can be queried via object "Status" on "F810:07" in the electronic data sheet (see section "I/O module information (0xF810)" on page 29).

Each X2X Link station is equipped with a hardware component (ASIC) that reports its state to the X2X Link master – in this case, the bus controller – during each X2X Link cycle.

Each network status byte is structured as follows:

Bit	Value	Description
0	0x01	X2X Link power supply voltage OK
1	0x02	Reserved (always 0)
2	0x04	Communication between ASIC and electronics module OK (required to validate bits 3 to 7)
3	0x08	I/O data invalid (void)
4	0x10	Reserved (always 1)
5	0x20	Reserved (always 1)
6	0x40	Reserved (always 1)
7	0x80	Reserved (always 1)

This results in the following values:

Description	Value (decimal)	Value (hex)
X2X Link station inactive (e.g. no X2X Link power supply)	0	0x00
Everything OK (I/O data valid)	245	0xF5
No communication with the electronics module (bits 7 to 3 invalid)	249	0xF9
I/O data invalid, communication between X2X ASIC and electronics module OK (ASIC carried out a valid "Sync in" transfer with the electronics module in the previous X2X Link cycle)	253	0xFD

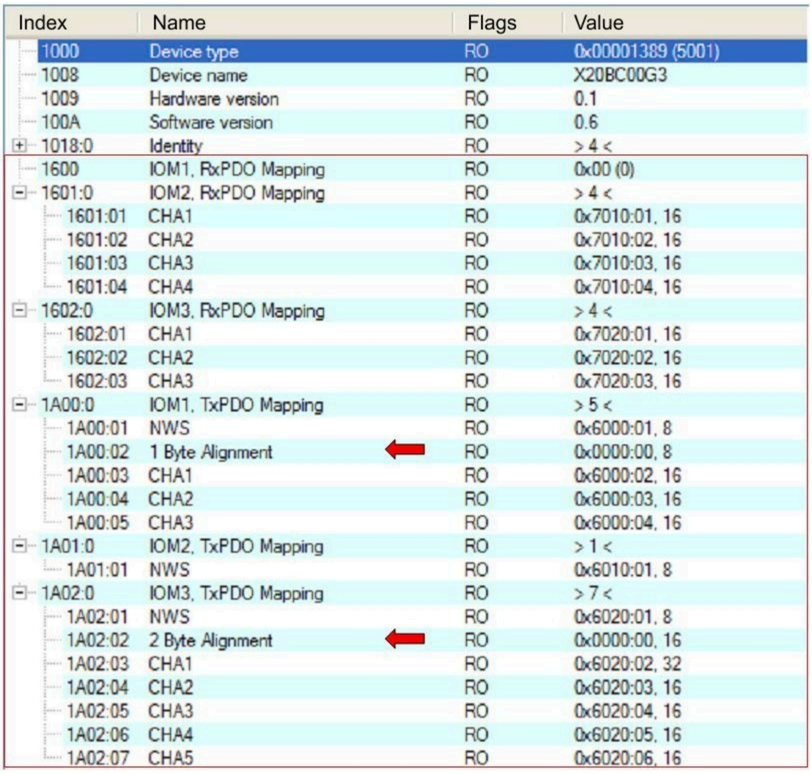
5.4 RxPDO/TxPDO objects

These objects correspond to process data objects (as with CANopen) and are used for describing and mapping the input and output data.
RxPDO stands for "Receive PDO" and means the receipt of data from the standpoint of the bus controller. In this case, this is the output data for the I/O modules.

RxPDO = Output data, objects "16**"
TxPDO = Input data, objects "1A**"

Unlike CANopen, where the length of a PDO is limited to 8 bytes, EtherCAT uses a separate object index for each I/O module. This means that the I/O data for the first module is located on objects "1600" and "1A00", while the I/O data for the second module is located on "1601" and "1A01", etc.

The following objects result from the mapping example in the last chapter (see section "Description of input and output data" on page 30):



Index	Name	Flags	Value
1000	Device type	RO	0x00001389 (5001)
1008	Device name	RO	X20BC00G3
1009	Hardware version	RO	0.1
100A	Software version	RO	0.6
1018:0	Identity	RO	> 4 <
1600	IOM1, RxPDO Mapping	RO	0x00 (0)
1601:0	IOM2, RxPDO Mapping	RO	> 4 <
1601:01	CHA1	RO	0x7010:01, 16
1601:02	CHA2	RO	0x7010:02, 16
1601:03	CHA3	RO	0x7010:03, 16
1601:04	CHA4	RO	0x7010:04, 16
1602:0	IOM3, RxPDO Mapping	RO	> 4 <
1602:01	CHA1	RO	0x7020:01, 16
1602:02	CHA2	RO	0x7020:02, 16
1602:03	CHA3	RO	0x7020:03, 16
1A00:0	IOM1, TxPDO Mapping	RO	> 5 <
1A00:01	NWS	RO	0x6000:01, 8
1A00:02	1 Byte Alignment	RO	0x0000:00, 8
1A00:03	CHA1	RO	0x6000:02, 16
1A00:04	CHA2	RO	0x6000:03, 16
1A00:05	CHA3	RO	0x6000:04, 16
1A01:0	IOM2, TxPDO Mapping	RO	> 1 <
1A01:01	NWS	RO	0x6010:01, 8
1A02:0	IOM3, TxPDO Mapping	RO	> 7 <
1A02:01	NWS	RO	0x6020:01, 8
1A02:02	2 Byte Alignment	RO	0x0000:00, 16
1A02:03	CHA1	RO	0x6020:02, 32
1A02:04	CHA2	RO	0x6020:03, 16
1A02:05	CHA3	RO	0x6020:04, 16
1A02:06	CHA4	RO	0x6020:05, 16
1A02:07	CHA5	RO	0x6020:06, 16

Figure 2: RxPDO/TxPDO mapping example

Information:

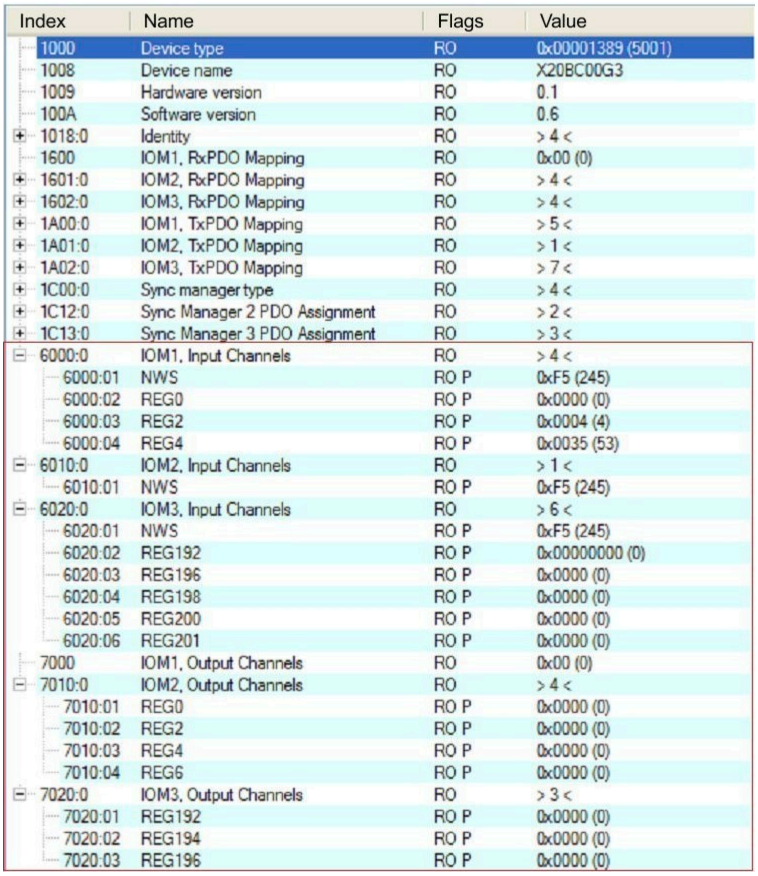
Alignment must be kept in mind (see section "Alignment rules" on page 34).

"IOM1" stands for the first I/O module after the bus controller in slot 1 (slot ID = 1), "IOM2" the module in slot 2 and so on.
The mapped channels "CHA*" are listed in the subindexes of these objects. The mapped object from the input ("0x6***") and output data area ("0x7***") is specified in column "Value". Each I/O module provides its network status ("NWS") as the first byte of the input data (TxPDO, "0x1A**:01"). In this example, the power supply module ("IOM1") does not receive any output data. This is why the value for "0x1600" (subindex 0) is 0.

5.5 Input/Output data in the object dictionary

These objects reflect the data points of the corresponding "IOM*" I/O modules. The input data is in the range "0x6***" and the output data at "0x7***".
"REG" followed by the register address is given as the name for the (cyclic) input and output registers. For a description of each register address, see the I/O module documentation. The value is an indicator of the data width of the individual registers.

These objects are used for mapping the input and output data objects (see section "Description of input and output data" on page 30).



Index	Name	Flags	Value
1000	Device type	RO	0x00001389 (5001)
1008	Device name	RO	X20BC00G3
1009	Hardware version	RO	0.1
100A	Software version	RO	0.6
1018:0	Identity	RO	> 4 <
1600	IOM1, RxPDO Mapping	RO	0x00 (0)
1601:0	IOM2, RxPDO Mapping	RO	> 4 <
1602:0	IOM3, RxPDO Mapping	RO	> 4 <
1A00:0	IOM1, TxPDO Mapping	RO	> 5 <
1A01:0	IOM2, TxPDO Mapping	RO	> 1 <
1A02:0	IOM3, TxPDO Mapping	RO	> 7 <
1C00:0	Sync manager type	RO	> 4 <
1C12:0	Sync Manager 2 PDO Assignment	RO	> 2 <
1C13:0	Sync Manager 3 PDO Assignment	RO	> 3 <
6000:0	IOM1, Input Channels	RO	> 4 <
6000:01	NWS	RO P	0xF5 (245)
6000:02	REG0	RO P	0x0000 (0)
6000:03	REG2	RO P	0x0004 (4)
6000:04	REG4	RO P	0x0035 (53)
6010:0	IOM2, Input Channels	RO	> 1 <
6010:01	NWS	RO P	0xF5 (245)
6020:0	IOM3, Input Channels	RO	> 6 <
6020:01	NWS	RO P	0xF5 (245)
6020:02	REG192	RO P	0x00000000 (0)
6020:03	REG196	RO P	0x0000 (0)
6020:04	REG198	RO P	0x0000 (0)
6020:05	REG200	RO P	0x0000 (0)
6020:06	REG201	RO P	0x0000 (0)
7000	IOM1, Output Channels	RO	0x00 (0)
7010:0	IOM2, Output Channels	RO	> 4 <
7010:01	REG0	RO P	0x0000 (0)
7010:02	REG2	RO P	0x0000 (0)
7010:03	REG4	RO P	0x0000 (0)
7010:04	REG6	RO P	0x0000 (0)
7020:0	IOM3, Output Channels	RO	> 3 <
7020:01	REG192	RO P	0x0000 (0)
7020:02	REG194	RO P	0x0000 (0)
7020:03	REG196	RO P	0x0000 (0)

Figure 3: Input and output data in the object dictionary - Example

5.6 Alignment rules

Alignment of individual registers in the input and output data image (written by the "RxPDO/TxPDO objects") makes it easier to access data via the application. Base data type alignment is performed, with the **entire** process data stream always being taken into account. Keep in mind that the number of alignment bytes can change based on how the I/O modules are arranged. The alignment bytes between registers of the same module are also different depending on which offset a module's first channel ends up on.

Alignment example with 3 module types:

The first module "AI" has 2 (input) registers. The first of these registers (the register with the lower address) has a width of one byte (SINT); the second has a width of one word (INT).

AI	DI	AI32
SINT	SINT	DINT
INT		

SINT: 1 byte
INT: 2 bytes
DINT: 4 bytes

Different I/O module combinations with corresponding alignment:

Example A:

AI	SINT
Alignment	SINT
AI	INT
DI	SINT
Alignment	SINT
Alignment	SINT
Alignment	SINT
AI32	DINT

Example B:

DI	SINT
AI	SINT
AI	INT
AI32	DINT

Example C:

DI	SINT
Alignment	SINT
Alignment	SINT
Alignment	SINT
AI32	DINT
AI	SINT
Alignment	SINT
AI	INT

Example D:

AI	SINT
Alignment	SINT
AI	INT
AI32	DINT
DI	SINT

Example B is the only example in which the I/O modules are arranged in a way that does not require alignment bytes. Every other example combination results in different alignments and therefore has different (input) data offsets.

6 The ESI device description file

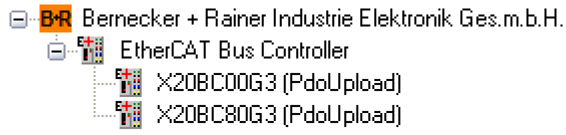


Figure 4: View of available ESI files

A (generic) device description of each EtherCAT bus controller is provided on the B&R website in the form of ESI device description files that define all sync managers and mailbox parameters. Mailbox parameter "PdoUpload" is defined here with "TRUE". This is the instruction for the master to read all of the process data information from the bus controller via CoE (CANopen over EtherCAT) and to create a corresponding I/O mapping.

Information:

I/O modules added afterwards are not taken into account in this case and are not incorporated in the process data until another upload takes place.

Faulty ESI parameters are indicated by the blinking of LED "Error" as well as bit 22 being set in bus controller status object "0xF100" (see section "Bus controller state (0xF100)" on page 26).

6.1 Sync manager settings

The following entries are defined in the ESD file:

- Output and input mailbox. This is needed for asynchronous communication between the EtherCAT master and bus controller.
- Properties of output and input process data for synchronous communication

```
<!-- SM0 MBX OUT -->
<Sm MinSize="#x40" MaxSize="#x500" DefaultSize="#x100" StartAddress="#x3400"
ControlByte="#x26" Enable="1">MBoxOut</Sm>

<!-- SM1 MBX INP -->
<Sm MinSize="#x40" MaxSize="#x500" DefaultSize="#x100" StartAddress="#x3A00"
ControlByte="#x22" Enable="1">MBoxIn</Sm>

<!-- Sm2 PD OUT -->
<Sm StartAddress="#x1000" ControlByte="#x24" Enable="1">Outputs</Sm>

<!-- SM3 PD INP -->
<Sm StartAddress="#x2200" ControlByte="#x20" Enable="1">Inputs</Sm>
```

6.2 Mailbox settings

Mailbox configuration:

```
<Mailbox DataLinkLayer="true">
        <CoE
                SdoInfo = "true"
                PdoAssign = "false"
                PdoConfig = "false"
                PdoUpload = "true"
                CompleteAccess = "false">
        </CoE>
        <FoE/>
</Mailbox>
```

The size of the mailbox can be freely selected within limits MinSize="#x40" and MaxSize= "#x500".

Information:

Only even values are allowed.

6.3 EEPROM settings

EEPROM configuration:

```
<Eeprom>
        <ByteSize>256</ByteSize>
        <ConfigData>08000200</ConfigData>
        <BootStrap>00340005003A0005</BootStrap>
</Eeprom>
```

Explanation of bootstrap settings 0034, 0005, 003A, 0005

Information:

The position of the high and low bytes is switched around by the bus controller (e.g. 0x0034 → 0x3400).

0x3400:	Start address of mailbox "OUT" for FoE in mode "Bootstrap"
0x0500:	Size of mailbox "OUT" in mode "Bootstrap" ("0x500" = 1280 bytes)
0x3A00:	Start address of mailbox "INP" for FoE in mode "Bootstrap"
0x0500:	Size of mailbox "INP" in mode "Bootstrap" ("0x500" = 1280 bytes)

The bootstrap parameters are used for setting the sync manager during state BOOT and are valid for functions such as updating the firmware via FoE.

7 AL status code

The "AL status code" is located in ESC registers "0x134" and "0x135" and can be read out by the master. This code reflects the current slave error state.

List of standard EtherCAT AL status codes

Code	Description	Current state[1)] (or state change)	Resulting state[2)]
0x0000	No error	Any	Current state
0x0001	Unspecified error	Any	Any + E
0x0011	Invalid requested state change	I → S, I → O, P → O, O → B, S → B, P → B	Current state + E
0x0012	Unknown requested state	-	Current state + E
0x0013	Bootstrap not supported	I → B	I + E
0x0014	No valid firmware	I → P	I + E
0x0015	Invalid mailbox configuration	I → B	I + E
0x0016	Invalid mailbox configuration	I → P	I + E
0x0017	Invalid sync manager configuration	P → S, S → O	Current state + E
0x0018	No valid inputs available	O, S, P → S	P + E
0x0019	No valid outputs available	O, S → O	S + E
0x001A	Synchronization error	O, S → O	S + E
0x001B	Sync manager watchdog	O, S	S + E
0x001C	Invalid sync manager type	O, S, P → S	S + E
0x001D	Invalid output configuration	O, S, P → S	S + E
0x001E	Invalid input configuration	O, S, P → S	P + E
0x001F	Invalid watchdog configuration	O, S, P → S	P + E
0x0020	Slave needs cold restart	Any	Current state + E
0x0021	Slave needs "INIT"	B, P, S, O	Current state + E
0x0022	Slave needs "PRE OPERATIONAL"	S, O	S + E, O + E
0x0023	Slave needs "SAFE OPERATIONAL"	O	O + E
0x002D	Invalid output "FMMU" configuration	O, S, P → S	S + E
0x002E	Invalid input "FMMU" configuration	O, S, P → S	P + E
0x0030	Invalid DC "SYNCH" configuration	O, S	S + E
0x0031	Invalid DC latch configuration	O, S	S + E
0x0032	"PLL" error	O, S	S + E
0x0033	Invalid DC I/O error	O, S	S + E
0x0034	Invalid DC timeout error	O, S	S + E
0x0042	"MBX_EOE"	B, P, S, O	Current state + E
0x0043	"MBX_COE"	B, P, S, O	Current state + E
0x0044	"MBX_FOE"	B, P, S, O	Current state + E
0x0045	"MBX_SOE"	B, P, S, O	Current state + E
0x004F	"MBX_VOE"	B, P, S, O	Current state + E
0x0050 to 0x8000	Reserved		
0x8000 to 0xFFFF	Manufacturer-specific		

1) I = State INIT
S = State SAFE OPERATIONAL
P = State PRE OPERATIONAL
O = State OPERATIONAL
B = State BOOTSTRAP
2) E = State ERROR

AL status code

If the standard EtherCAT error code does not sufficiently describe the current error, then the bus controller status error code will be written to the "AL status code register" with an offset of "0x8000" (manufacturer-specific area).

The following error states are entered in the "AL status code register":

Bus controller Status code	Error	"AL status code"
$0x\underbrace{0000}_{\text{Error state}}\underbrace{0000}_{\text{OK state}}$		
0x00010000	"WATCHDOG_EXPIRED"	0x1B (covered by EtherCAT-specific "AL status code")
0x00020000	"FLASH_READ_FAILED"	0x8002
0x00040000	"IOM_FAILED"	0x8004
0x00080000	"IOM_MISSING"	0x8008
0x00100000	"IOM_MISMATCH"	0x8010
0x00200000	"INVALID_CFG_DATA"	0x8020
0x00400000	"INVALID_ESI_DATA"	0x15, 0x16, 0x17 (covered by EtherCAT-specific AL status code)
0x00800000	"OUT_OF_RESOURCES"	0x8080
0x01000000	"INVALID_FIRMWARE"	0x14 (covered by EtherCAT-specific "AL status code")
0x02000000	"ECAT_EEPROM_FAILED"	This error is not written to the AL status code register.
0x04000000	"IOM_UNSUPPORTED"	0x8400

7.1 Difference between bus controller status code and AL status code

AL status code

The AL status code is provided in an ESC register.
The AL status code register shows EtherCAT-specific errors as well as errors specific to the bus controller (manufacturer-specific errors).

Bus controller status code

The bus controller status code is set independently from the bus controller status and does not change the state of the bus controller.
The bus controller status code can only be read using the mailbox and corresponding CoE object.

8 Firmware update

New functions and improved versions of the bus controller can be implemented by updating the firmware. Firmware files can be obtained from the B&R support team.
There are 2 ways to update the firmware:

- "Firmware update via FoE" on page 39
- "Firmware update via RS232" on page 40

8.1 Firmware update via FoE

An FoE-capable EtherCAT master or an FoE tool is required in order to update the firmware via FoE (File Access over EtherCAT). Firmware can be updated in states BOOTSTRAP, PRE OPERATIONAL, SAFE OPERATIONAL and OPERATIONAL.
Firmware updates are generally fastest in state BOOTSTRAP because the full mailbox capacity is available.

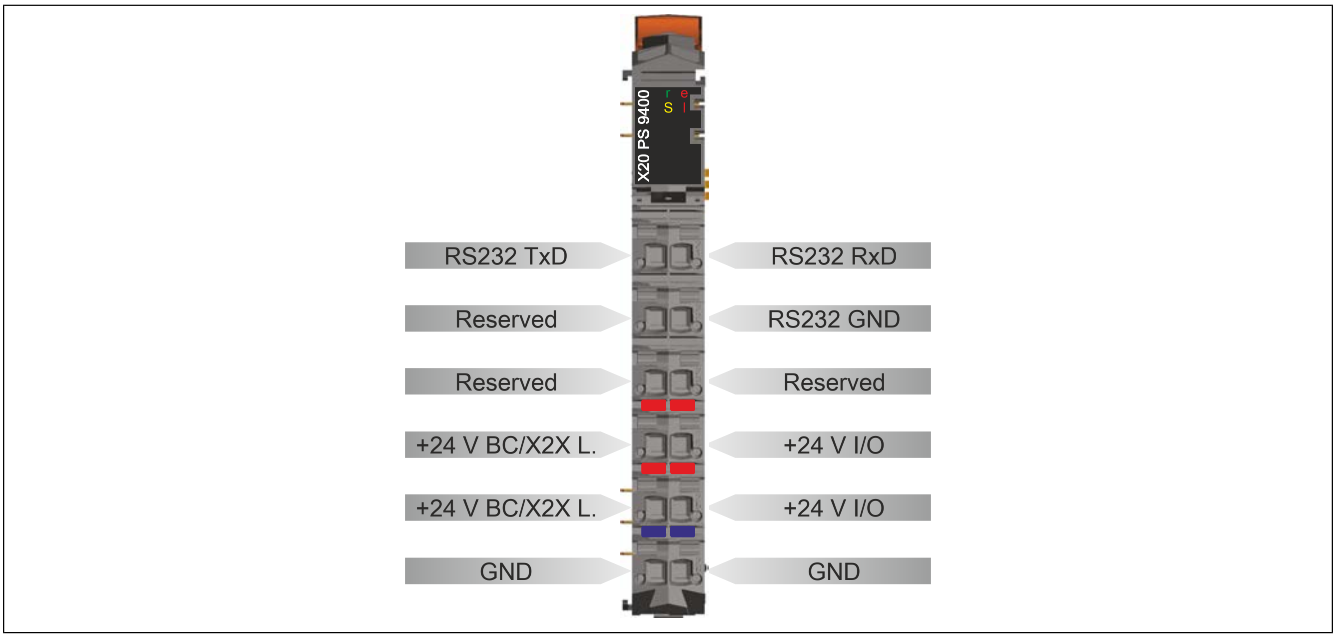
Information:

If a firmware update is performed during the current power-on phase, then the bus controller will be rebooted when changing from state BOOTSTRAP to state INIT.

8.2 Firmware update via RS232

The firmware for the X20 bus controller can be updated via an RS232 interface. This is accessed via the bus controller's X20PS9400 power supply module. LED status indicator "S" indicates when communication is taking place.

The following image shows the wiring connections. Terminals 11, 21 and 22 of the serial interface are used on the software side.



8.2.1 Firmware updates in Windows XP and earlier

Any terminal program with "1K-Xmodem" emulation can be used to update the firmware. In Windows XP and earlier, there is a suitable program installed by default ("HyperTerminal", located under Windows "Start" → All programs → Accessories → Communication → HyperTerminal).

It must be configured correctly before a connection can be established. Use the following settings:

- 115200 baud (bits per second)
- 8 data bits
- No parity bit
- 1 stop bit
- No flow control

Command "Help" in the terminal provides an overview of supported commands.

The firmware update is performed using command **"D 0 2"** (**D** for download, **0** for module no. 0, which represents the bus controller itself, and **2** for firmware update).

A brief description is displayed after transmitting the command (by pressing the "Enter" key). Select "Transfer" in the HyperTerminal menu bar and then "Send file" to open the following dialog box:

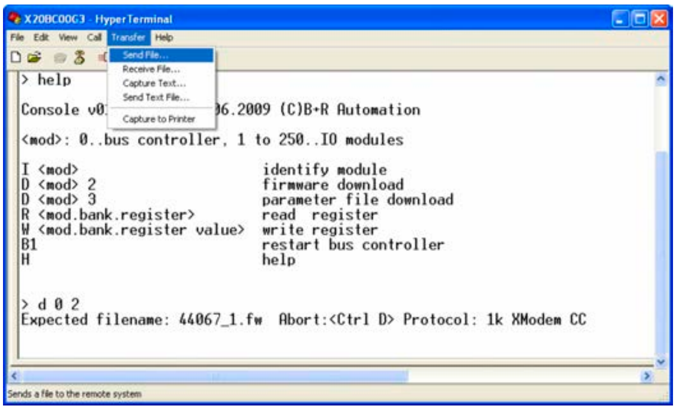


Figure 5: HyperTerminal "Transfer" - Example

Enter "1K-Xmodem" as the protocol and select the path to the location of the firmware file:

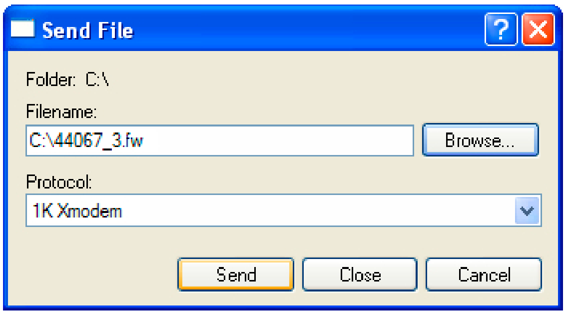


Figure 6: Selecting the firmware file - Example

The file transfer is started by pressing the "Send" button.

The message "Program done" appears in the main HyperTerminal window as soon as the firmware update is completed.
The bus controller must be restarted to enable the new firmware (switch power supply off/on or use command "B1").

A new line containing version information about the bus controller console will appear in HyperTerminal after restarting.

The new firmware version can be queried as "FW Rev" using command **"i 0"** (**i** for information about the module and **0** for module no. 0):

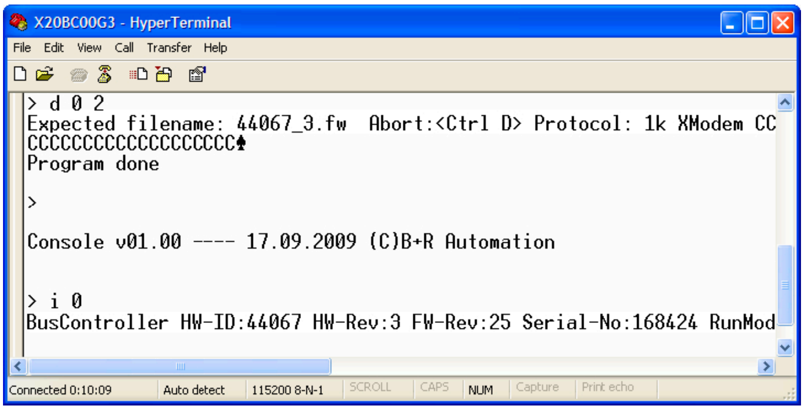


Figure 7: Querying the firmware version - Example

8.2.2 Firmware update in Windows Vista and later

Since Windows Vista and later versions no longer include a terminal program, a good alternative is the open source tool **Tera Term**. This can be downloaded for free from the Internet.

1. Settings

Before a connection can be established, the serial interface must be configured for access.

To do this, open the setup dialog box under *Setup* → *Serial port* and select the appropriate parameters.

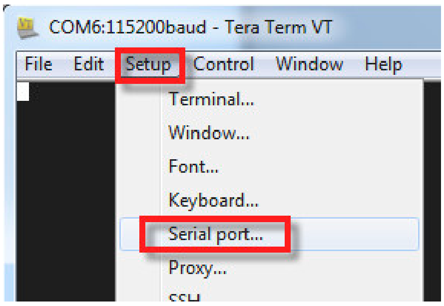


Figure 8: Opening the serial interface dialog box

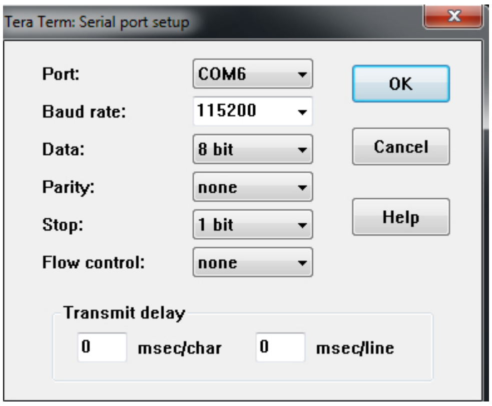


Figure 9: Serial interface parameters

2. Establish the connection

Once setup is complete, a connection can be established under *File* → *New connection* using the serial interface just configured.

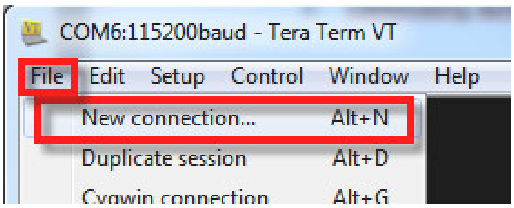


Figure 10: Selecting a new connection

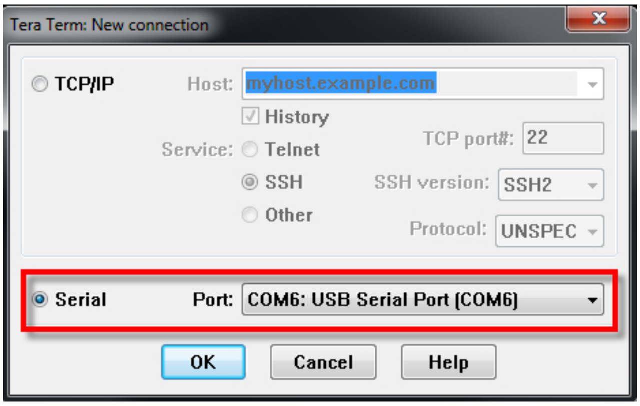


Figure 11: Selecting the serial interface

3. Transfer the firmware

Use command "Help" to open an overview of all available commands.

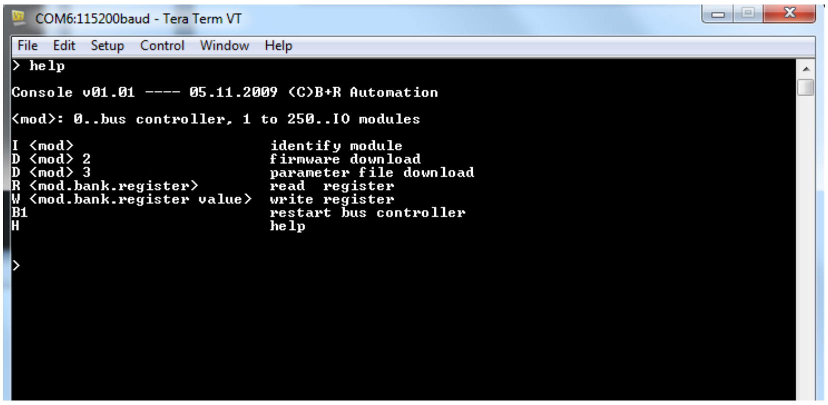


Figure 12: Available commands in VT

Update the firmware using command "**D 0 2**" (D for download, 0 for module no. 0, which represents the bus controller itself, and 2 for firmware update).

A brief description of the command is displayed after the command is transmitted (by pressing the "Enter" key).

Next, select the desired firmware file under *File → Transfer → XMODEM→ Send*.

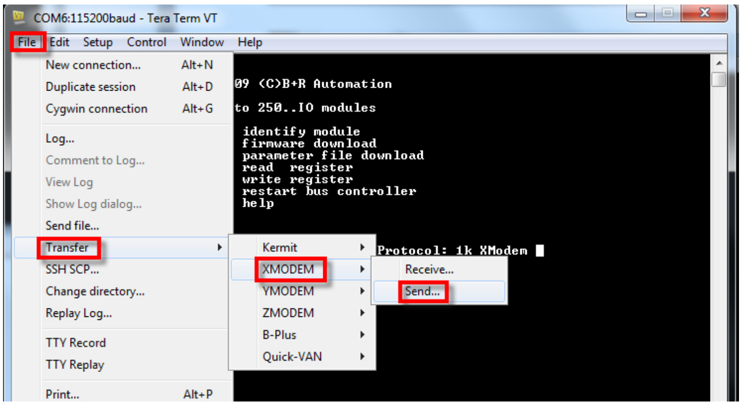


Figure 13: Initiating the data transfer

It is important to use transfer option 1K (1K XMODEM).

The file transfer is started by clicking on button *Open*.

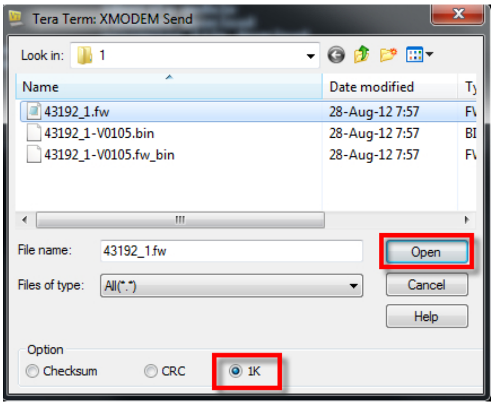


Figure 14: Selecting the firmware file

"Program done" appears in the window as soon as the firmware download is completed.

The new firmware becomes active after the bus controller is restarted (power cycle of power supply or command "B1").