Modbus TCP

User's manual

Version: **2.30 (December 2023)**
Order no.: **Modbus TCP**

Translation of the original documentation

Publishing information

B&R Industrial Automation GmbH
B&R Strasse 1
5142 Eggelsberg
Austria
Telephone: +43 7748 6586-0
Fax: +43 7748 6586-26
office@br-automation.com

Disclaimer

All information in this document is current as of its creation. The contents of this document are subject to change without notice. B&R Industrial Automation GmbH assumes unlimited liability in particular for technical or editorial errors in this document only (i) in the event of gross negligence or (ii) for culpably inflicted personal injury. Beyond that, liability is excluded to the extent permitted by law. Liability in cases in which the law stipulates mandatory unlimited liability (such as product liability) remains unaffected. Liability for indirect damage, consequential damage, business interruption, loss of profit or loss of information and data is excluded, in particular for damage that is directly or indirectly attributable to the delivery, performance and use of this material.

B&R Industrial Automation GmbH notes that the software and hardware designations and brand names of the respective companies used in this document are subject to general trademark, brand or patent protection.

Hardware and software from third-party suppliers referenced in this document is subject exclusively to the respective terms of use of these third-party providers. B&R Industrial Automation GmbH assumes no liability in this regard. Any recommendations made by B&R Industrial Automation GmbH are not contractual content, but merely non-binding information for which no liability is assumed. When using hardware and software from third-party suppliers, the relevant user documentation of these third-party suppliers must additionally be consulted and, in particular, the safety guidelines and technical specifications contained therein must be observed. The compatibility of the products from B&R Industrial Automation GmbH described in this document with hardware and software from third-party suppliers is not contractual content unless this has been separately agreed in individual cases; in this respect, warranty for such compatibility is excluded in any case, and it is the sole responsibility of the customer to verify this compatibility in advance.

1 General information

Established in 1979, the Modbus protocol has approved the use of Ethernet with both Modbus TCP and Modbus/UDP. Today, Modbus TCP is an open Internet draft standard introduced by Schneider Automation to the Internet Engineering Task Force (IETF), the organization responsible for Internet standardization. The Modbus services and object model have been preserved since the original version and left unchanged for use with the TCP/IP transmission medium.

Modbus/UDP differs from Modbus TCP in that it uses connectionless communication via UDP/IP. The advantages of faster and easier communication with UDP/IP also brings with it the disadvantage of requiring error detection and correction in the application layer.

This bus controller makes it possible to connect X2X Link I/O nodes to Modbus via Ethernet. The bus controller can be operated on B&R controllers through the use of Automation Studio or on third-party systems with Modbus TCP or -UDP master functionality.

1.1 Organization of notices

Safety notices

Contain **only** information that warns of dangerous functions or situations.

Signal word	Description
Danger!	Failure to observe these safety guidelines and notices will result in death, severe injury or substantial damage to property.
Warning!	Failure to observe these safety guidelines and notices can result in death, severe injury or substantial damage to property.
Caution!	Failure to observe these safety guidelines and notices can result in minor injury or damage to property.
Notice!	Failure to observe these safety guidelines and notices can result in damage to property.

General notices

Contain **useful** information for users and instructions for avoiding malfunctions.

Signal word	Description
Information:	Useful information, application tips and instructions for avoiding malfunctions.

2 Technical description

2.1 X20 bus controller

2.1.1 X20 - Order data

Order number	Short description	Figure
	Bus controllers	
X20BC0087	X20 bus controller, 1 Modbus TCP or Modbus UDP interface, integrated 2-port switch, 2x RJ45, order bus base, power supply module and terminal block separately!	
X20cBC0087	X20 bus controller, coated, Modbus TCP or Modbus UDP interface, integrated 2-port switch, 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 cover 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 supply, supply not galvanically isolated	
X20cBB80	X20 bus base, coated, for X20 base module (BC, HB, etc.) and X20 power supply module, X20 end cover plates (left and right) X20AC0SL1/X20AC0SR1 included	
X20cPS9400	X20 power supply module, coated, for bus controller and internal I/O power supply X2X Link power supply	
	Terminal blocks	
X20TB12	X20 terminal block, 12-pin, 24 VDC keyed	

Table 1: X20BC0087, X20cBC0087 - Order data

2.1.2 X20 - Technical data

Order number	X20BC0087	X20cBC0087
Short description		
Bus controller	Modbus TCP/UDP slave	
General information		
B&R ID code	0x227C	0xD577
Status indicators	Module status, bus function	
Diagnostics		
Module status	Yes, using LED status indicator and software	
Bus function	Yes, using LED status indicator and software	
Power consumption		
Bus	2 W	
Additional power dissipation caused by actuators (resistive) [W]	-	

Table 2: X20BC0087, X20cBC0087 - Technical data

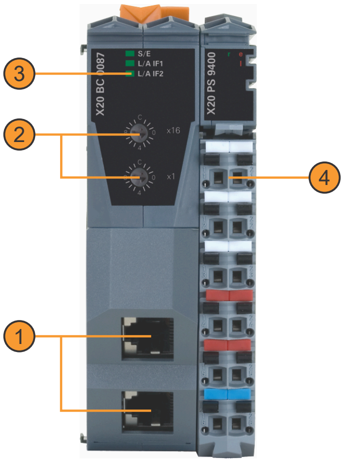
Order number	X20BC0087	X20cBC0087
Certifications		
CE	Yes	
UKCA	Yes	
ATEX	Zone 2, II 3G Ex nA nC IIA T5 Gc IP20, Ta (see X20 user's manual) FTZÚ 09 ATEX 0083X	
UL	cULus E115267 Industrial control equipment	
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5	
DNV	Temperature: **B** (0 to 55°C) Humidity: **B** (up to 100%) Vibration: **B** (4 g) EMC: **B** (bridge and open deck)	
LR	ENV1	
KR	Yes	
ABS	Yes	
BV	**EC33B** Temperature: 5 - 55°C Vibration: 4 g EMC: Bridge and open deck	
EAC	Yes	
KC	Yes	-
Interfaces		
Fieldbus	Modbus TCP/UDP slave	
Variant	2x shielded RJ45 (switch)	
Line length	Max. 100 m between 2 stations (segment length)	
Transfer rate	10/100 Mbit/s	
Transfer		
Physical layer	10BASE-T/100BASE-TX	
Half-duplex	Yes	
Full-duplex	Yes	
Autonegotiation	Yes	
Auto-MDI/MDIX	Yes	
Min. cycle time [1]		
Fieldbus	1 ms	
X2X Link	500 µs	
Synchronization between bus systems possible	No	
Electrical properties		
Electrical isolation	Modbus isolated from bus and I/O	
Operating conditions		
Mounting orientation		
Horizontal	Yes	
Vertical	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	IP20	
Ambient conditions		
Temperature		
Operation		
Horizontal mounting orientation	-25 to 60°C	
Vertical mounting orientation	-25 to 50°C	
Derating	-	
Starting temperature	-	Yes, -40°C
Storage	-40 to 85°C	
Transport	-40 to 85°C	
Relative humidity		
Operation	5 to 95%, non-condensing	Up to 100%, 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 X20cPS9400 separately. Order 1x bus base X20cBB80 separately.
Pitch [2]	$37.5^{+0.2}$ mm	

Table 2: X20BC0087, X20cBC0087 - Technical data

[1] The minimum cycle time specifies how far the bus cycle can be reduced without communication errors occurring.

[2] Pitch is based on the width of bus base X20BB80. In addition, power supply module X20PS9400 or X20PS9402 is always required for the bus controller.

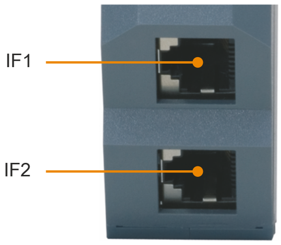
2.1.3 Operating and connection elements

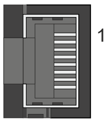


1	Modbus TCP connection with 2x RJ45 for simple wiring	2	Network address switches
3	LED status indicators	4	Terminal block for bus controller and I/O supply

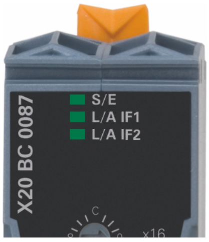
2.1.4 Ethernet interface

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



Interface	Pinout		
	Pin	Ethernet	
 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	

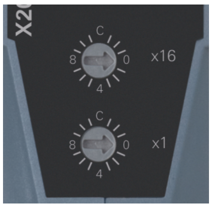
2.1.5 LED status indicators

Figure	LED	Color	Status	Description
	S/E[1]	Green	On	Indicates that there is at least one client connection
			2 pulses	Indicates that there are no client connections
			4 pulses	Indicates that the controller is waiting for an address from the DHCP server
			Blinking	Initialization of connected I/O modules
		Red	2 pulses	Watchdog timeout
			3 pulses	Faulty I/O module configuration data
			4 pulses	Indicates that the controller has detected an IP address being used twice
			5 pulses	Indicates a missing, defective or incorrect I/O module
			6 pulses	Error reading flash memory. Last write operation was incomplete or contained errors.[2]
			On	Indicates a major unrecoverable fault
	L/A IFx	Green	Blinking	Ethernet activity taking place on the RJ45 port (IF1, IF2) indicated by the respective LED
			On	Indicates an established connection (link), but no communication is taking place
			Off	Indicates that no physical Ethernet connection exists

[1] The Status/Error LED "S/E" is a green/red dual LED. The LED blinks red several times immediately after startup. This is a boot message, however, and not an error.

[2] Possible cause: The bus controller received a command to save, but was switched off before saving was complete. In this case, the bus controller continues to use the old configuration and indicates the failed write operation with a blink code.

2.1.6 Network address switches



The network address switches have multiple functions:

- Uses the bus controller parameters stored in flash memory or preset at the factory (0x00)
- Sets the default IP address (in the range 0x01 to 0x7F)
- Enables operation with a DHCP server (in the range 0x80 to 0xEF)
- Automatically saves modified parameters (0xF0)
- Initializes all bus controller parameters with their default values (0xFE)
- Initializes the communication parameters with their default values (0xFF)

For an overview of network address switch functions, see "Commissioning" on page 21.

Information:

Please note that the IP address configured in the bus controller is not used or only used partially (in the range 0x01 to 0xF) for all switch positions other than 0x00.

Information:

Changes to the network address switches are only applied after a restart. A restart can also be carried out from the Telnet interface (command "restart") or via the fieldbus (fc6 0x1143 0xC0).

2.2 X67 bus controller

2.2.1 X67 - Order data

Order number	Short description	Figure
	Bus controller modules	
X67BCJ321.L12	X67 bus controller, 1 Modbus TCP/UDP 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: X67BCJ321.L12 - Order data

2.2.2 X67 - Technical data

Order number	X67BCJ321.L12
Short description	
Bus controller	Modbus TCP/UDP slave
General information	
Inputs/Outputs	16 digital channels, configurable as inputs or outputs using Automation Studio or data point, inputs with additional functions
Insulation voltage between channel and bus	500 V_{eff}
Nominal voltage	24 VDC
B&R ID code	
Bus controller	0xAD3C
Internal I/O module	0xBD76
Sensor/Actuator power supply	0.5 A summation current
Status indicators	I/O function per channel, supply voltage, bus function
Diagnostics	
Outputs	Yes, using LED status indicator and software
I/O power supply	Yes, using LED status indicator and software
Connection type	
Fieldbus	M12, D-coded
X2X Link	M12, B-coded
Inputs/Outputs	8x M12, A-coded
I/O power supply	M8, 4-pin
Power output	15 W X2X Link power supply for I/O modules
Power consumption	
Fieldbus	4.2 W
Internal I/O	2.5 W
X2X Link power supply	24.3 W at maximum power output for connected I/O modules
Certifications	
CE	Yes
UKCA	Yes
ATEX	Zone 2, II 3G Ex nA IIA T5 Gc IP67, Ta = 0 - Max. 60°C TÜV 05 ATEX 7201X
UL	cULus E115267 Industrial control equipment
HazLoc	cCSAus 244665 Process control equipment for hazardous locations Class I, Division 2, Groups ABCD, T5
EAC	Yes
KC	Yes
Interfaces	
Fieldbus	Modbus TCP/UDP slave
Variant	2x M12 interface (switch), 2x female connector on the module
Line length	Max. 100 m between 2 stations (segment length)
Transfer rate	10/100 Mbit/s

Table 4: X67BCJ321.L12 - Technical data

Order number	X67BCJ321.L12
Transfer	
Physical layer	10BASE-T/100BASE-TX
Half-duplex	Yes
Full-duplex	Yes
Autonegotiation	Yes
Auto-MDI/MDIX	Yes
Min. cycle time [1]	
Fieldbus	1 ms
X2X Link	500 µs
Synchronization between bus systems possible	No
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 characteristics per EN 61131-2	Type 1
Input voltage	18 to 30 VDC
Input current at 24 VDC	Typ. 4 mA
Input circuit	Sink
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 resistance	Typ. 6 kΩ
Additional functions	50 kHz event counting, gate measurement
Switching threshold	
Low	<5 VDC
High	>15 VDC
Event counters	
Quantity	2
Signal form	Square wave pulse
Evaluation	Each negative 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	Positive edge - Negative 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	Current-sourcing FET
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 shutdown in the event of overcurrent or short circuit, integrated protection for switching inductive loads, reverse polarity protection of the output power supply
Diagnostic status	Output monitoring with 10 ms delay
Leakage current when the output is switched off	5 µA
Switching on after overload shutdown	Approx. 10 ms (depends on the module temperature)
Residual voltage	<0.3 V at 0.5 A nominal current
Peak short-circuit current	<12 A
Switching delay	
0 → 1	<400 µs
1 → 0	<400 µs
Switching frequency	
Resistive load	Max. 100 Hz
Inductive load	See section "Switching inductive loads".
Braking voltage when switching off inductive loads	50 VDC

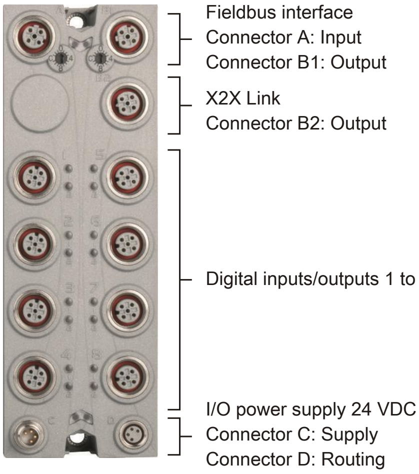
Table 4: X67BCJ321.L12 - Technical data

Order number	**X67BCJ321.L12**
Electrical properties	
Electrical isolation	Bus isolated from channel Modbus not isolated from bus and 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	350 g
Torque for connections	
M8	Max. 0.4 Nm
M12	Max. 0.6 Nm

Table 4: X67BCJ321.L12 - Technical data

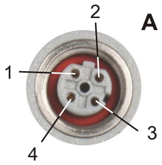
1) The minimum cycle time specifies how far the bus cycle can be reduced without communication errors occurring.
2) The power consumption of the sensors and actuators connected to the module is not permitted to exceed 12 W.

2.2.3 Operating and connection elements



2.2.4 Fieldbus interface

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-coded (female), input		

Information:

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

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

2.2.4.1 Wiring guidelines for bus controllers with Ethernet cable

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

Order 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 bend radius of the cable (see the data sheet of the cable)

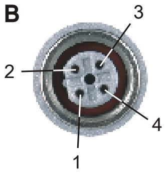
Information:

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

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

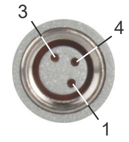
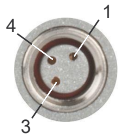
2.2.5 X2X Link

Pre-assembled cables can be used to connect up to 250 additional modules to the module via X2X Link. The connection is made using a circular connector (1x M12, 4-pin).

Connection	Pinout	
	Pin	Description
	1	X2X+
	2	X2X
	3	X2X⊥
	4	X2X\
	B ... B-keyed female connector on the module, output SHLD ... Shielding provided by threaded insert in the module	

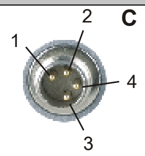
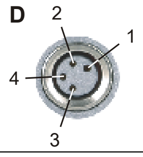
2.2.6 Digital inputs/outputs

Digital inputs/outputs are connected using pre-assembled cables with circular connectors (8x M8, 3-pin).

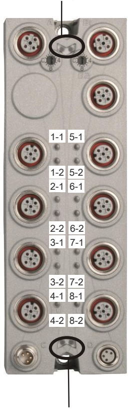
Connection	Pinout	
	Pin	Description
 	1	24 VDC sensor/actuator power supply[1]
	3	GND
	4	Input/Output x
	1) Sensors/Actuators are not permitted to be supplied externally.	

2.2.7 24 VDC module supply

The module power supply connection is made using circular connectors (2x M8, 4-pin). The power supply is connected via the male C connector. The female D connector is used to route the supply voltage to other modules (see also the general description of BC modules in the "Power supply" section of the X67 system user's manual).

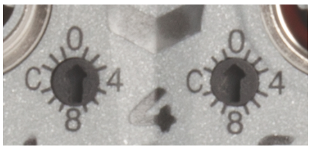
Connection	Pinout		
	Pin	**Male connector C**	**Female connector D**
 	1	24 VDC fieldbus	24 VDC I/O
	2	24 VDC I/O	24 VDC I/O
	3	GND	GND
	4	GND	GND
	C ... Male connector on the module, power supply D ... Female connector on the module, supply routing		

2.2.8 LED status indicators

Figure	LED	Color	Status	Description
Status indicator 1: Left: L/A IF; right: S/E  1-1 5-1 1-2 5-2 2-1 6-1 2-2 6-2 3-1 7-1 3-2 7-2 4-1 8-1 4-2 8-2 Status indicator 2: Left: Green, Right: Red	**Status indicator 1**: Status indicator for Modbus TCP bus controller			
	L/A IF	Green	Blinking	The LED flashes when there is Ethernet activity on one or both Ethernet connections.
			Permanently on	There is a connection (link) on one or both Ethernet connections, but there is no communication.
			Off	There is no physical Ethernet connection.
	S/E[1]	Green	Permanently on	At least one client connection exists.
			2 pulses	No client connection exists.
			4 pulses	The controller is waiting for the address assignment of a DHCP server.
			Blinking	Initialization of connected I/O modules
		Red	Permanently on	Major unrecoverable hardware fault
			2 pulses	Watchdog timeout
			3 pulses	Faulty I/O module configuration data
			4 pulses	The controller has detected a duplicate IP address.
			5 pulses	Missing, defective or incorrect I/O module detected
			6 pulses	Faulty reading or writing of flash memory.
	I/O LEDs			
	1-1 to 8-2	Orange	-	Input/Output state of the corresponding channel
	Status indicator 2: Status indicator for module functionality			
	Left	Green	Off	No power to module
			Single flash	Mode RESET
			Blinking	Mode PREOPERATIONAL
			On	Mode RUN
	Right	Red	Off	Module not supplied with power 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 within the valid range

[1] LED "Status/Error" is a green/red dual LED. Several red blinking signals are displayed immediately after the device is switched on. These are startup messages, however, and not errors.

2.2.9 Network address switches



The network address switches have multiple functions:

- Uses the bus controller parameters stored in flash memory or preset at the factory (0x00)
- Sets the default IP address (in the range 0x01 to 0x7F)
- Enables operation with a DHCP server (in the range 0x80 to 0xEF)
- Automatically saves modified parameters (0xF0)
- Initializes all bus controller parameters with their default values (0xFE)
- Initializes the communication parameters with their default values (0xFF)

For an overview of network address switch functions, see "Commissioning" on page 21.

Information:

Please note that the IP address configured in the bus controller is not used or only used partially (in the range 0x01 to 0xF) for all switch positions other than 0x00.

Information:

Changes to the network address switches are only applied after a restart. A restart can also be carried out from the Telnet interface (command "restart") or via the fieldbus (fc6 0x1143 0xC0).

3 Basic information

3.1 Automatic configuration

After the Modbus TCP bus controller is started, it will detect all I/O modules in the node (X2X Link modules, terminals) and generate a local process image using this information.

Depending on the data type, I/O data is split up among different address ranges:

- All analog and more complex X2X Link modules are word-oriented. The most significant byte is transferred in the first position when data is exchanged (big-endian format, see "Communication protocol" on page 76). Data from analog X2X Link modules is mapped in the 16-bit process image according to the position of the modules after the bus controller.
- All digital X2X Link modules and status data is byte-oriented and mapped in the process image in order.

The local process image is divided into input and output data areas. For more detailed information and examples, see "Bus controller process image" on page 25.

A copy of digital I/O data is also mapped in a separate discrete bit-oriented image. This bit-oriented area is split up into an input and output area, each of which starts at address 0x0000. Bit-oriented Modbus functions can be used to access this area.

Information:

The "Modbus TCP Mapping Tool" is available for download from the bus controller's Downloads section on the B&R web portal www.br-automation.com. It can be used to see which I/O data points are available on individual modules and how they are arranged in the process image when an automatic configuration takes place.

Information:

The process image parameters as well as the module parameters for each I/O module can be used to query the number and length (see "Process image data" on page 50) or starting addresses (index) of both analog and digital input/output data (see "I/O module register configuration" on page 70).

3.2 Multifunction modules

Only standard function model "254" is supported when the bus controller is used to automatically configure X2X Link multifunction I/O modules. All other function models are supported when these modules are configured manually (see "I/O module register configuration" on page 70). For additional information about module configuration, see chapter "Configuration of the I/O modules" on page 31.

3.3 Automation Studio

The Automation Studio is recommended for configuring the Modbus TCP bus controller and connected I/O modules.

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.

All supported I/O modules can be easily integrated on the bus controller and configured using the selection menus. Variables can be defined in the I/O mapping as usual.
When the project is compiled, configuration files are generated that can either be directly implemented in a 3rd-party development environment, transferred to the bus controller or used in other solutions, such as the B&R Modbus PVI line.
Automation Studio always creates a Full configuration.

3.4 Execution check

Modbus command execution is a serial process. As such, it is possible that some parts of a command can be executed without errors but that other areas inside of the same command will cause an error. One example of this is fc16 "Write to multiple registers" to an address range that is only partially writable.

In this case, the command would only be carried out up to an undefined part. In order to avoid this undefined state, make sure that no partial actions are carried out on the B&R Modbus TCP bus controller when an error occurs. This means either the command is executed completely and without errors or all partial actions already executed are discarded.

3.5 ModbusTCP Toolbox

In addition to the Telnet service, the ModbusTCP Toolbox is available for managing and troubleshooting the bus controller.

This tool is available at no cost as a download from the B&R website (www.br-automation.com) and offers extended diagnostic options.

3.6 Deleting an existing configuration

The following Modbus commands can be used to delete a configuration:

- Erasing flash memory using Modbus function code 6:
  Write 0xC1 to address 0x1144 (fc6 0x1144 0xC1)
- Deleting the module configuration data and saving all settings to flash memory:
  fc6 0x1146 0xC0, then fc6 0x1140 0xC1

In the event that configuration data should remain in flash memory, a restart can be carried out in boot mode 0xC2 (Load factory default values: fc6 0x1143 0xC2).

> ### Information:
>
> **Flash memory can also be erased from the Telnet interface (command "flash erase"), ModbusTCP Toolbox or via the fieldbus. This will reset the bus controller to its factory settings.**

4 Commissioning

4.1 General information

An IP address must be assigned in order to communicate with the bus controller.
2 options are possible here:

- Permanent IP address
- Operation with a DHCP server

The network address switches are used for configuring both options.

If the network address switch is set to 0xFF, the bus controller is assigned the static IP address 192.168.100.1 after a restart. A new IP address can be assigned as follows:

1. Via the fieldbus (see "Changing the IP address with the network address switches" on page 23)
2. Via the Telnet interface (see "Assigning an IP address" on page 86)
3. Via the "ModbusTCP Toolbox" on page 20

> ### Information:
>
> **For operation with a DHCP server, the network address switches must be assigned a value between 0x80 and 0xEF, with the hostname of the controller depending on the value of the network address switches. It is therefore important to make sure that 2 bus controllers are not being operated in the same network with the same network address switch settings.**

4.2 Connecting to the bus controller via Ethernet

The connection between the Modbus client (master) and the bus controller (slave) can be established as follows:

- Direct connection via patch cable between the PC's network interface and the bus controller
- Over an Ethernet network

Straight-through or crossover Ethernet cables can be used. Only Ethernet interface IF1 or IF2 is permitted to be used for the slot on the bus controller.

Since the default subnet mask of the bus controller is 255.255.255.0, the first 3 bytes of the IP address for the PC must match that of the bus controller.

Example

The bus controller has the default IP address of 192.168.100.1. In this case, the PC must be set to 192.168.100.xxx, with xxx representing a number between 2 and 254.

The Modbus TCP bus controller can be accessed in 2 different ways:

- Via its IP address
- Via its hostname

The IP address of the controller can be altered using its network address switches. The (configured) IP address and port number stored in the controller's flash memory are used in position 0x00.

If the network address switches are set to 0xFF, then the controller is assigned an IP address of 192.168.100.1 and default Modbus TCP port 502 after restarting.
For additional details about address switches, see "Setting the IP address (default value)" on page 23.

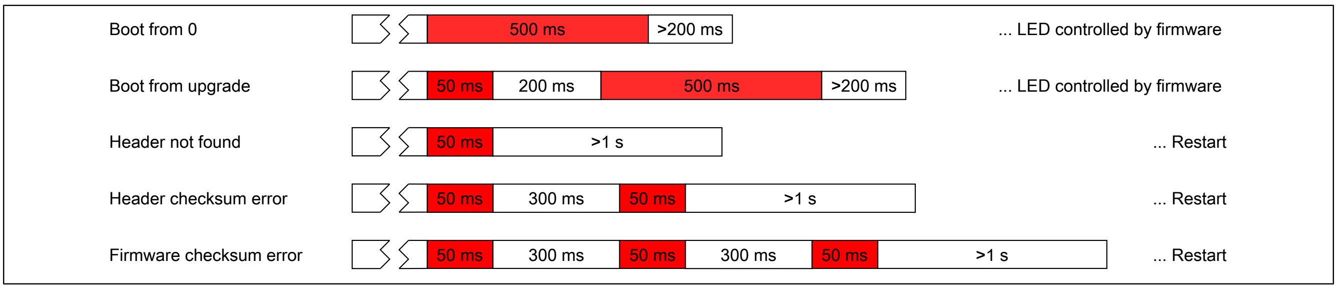
4.3 Startup procedure

Initialization takes place after the operating voltage has been switched on. The bus controller determines the input and output data size of the individual I/O modules, accounts for any saved configurations and generates the process image.

LED "S/E" on the bus controller indicates any problems during startup by blinking in a certain pattern.

4.3.1 Blink codes during startup

The bootloader indicates the following states via the module's "S/E" status LED:



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).

4.3.2 Forcing a boot from the factory default sector

This is necessary if firmware has been stored in the upgrade sector, operates the watchdog correctly but does not allow the booting process to occur without errors. The bootloader 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 bootloader, which causes the module's "S/E" status LED to begin flashing red very rapidly. 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.

5 Setting the IP address (default value)

Changes to the network address switches only become active after a restart. If the bus controller is restarted with the address switch value 0xFF, it is initialized with the IP address 192.168.100.1. This address is also the factory default setting. The interface number is set to 502 (reserved for Modbus).
This IP address can be used to establish a connection to the bus controller. The internationally unique MAC address is listed on the housing side of the bus controller. The combination of "br" and the MAC address results in a unique name (primary NetBIOS name) that also makes it possible to access the bus controller.

Example of the primary NetBIOS name:

MAC address:	00-60-65-00-49-02
Resulting NetBIOS name:	br006065004902

This means that, without additional parameter modifications, either the default IP address (192.168.100.1) or NetBIOS name "br+MAC" can be used to communicate with the bus controller.

Since NetBIOS is being used, the bus controller can only be accessed via this name if there are no intermediary routers or gateways in the way.

5.1 Automatic IP assignment by a DHCP server

If a network address switch setting between 0x80 and 0xEF is configured, the bus controller will attempt to request an IP address from the DHCP server. The assigned IP address can be queried with command "ping" together with the hostname. The bus controller registers this hostname on the DHCP server, which should forward it to a DNS server.

Example The hostname (DNS name) is made up of 3 elements:
"br" + "mb" + Address switch value (3 decimal places)
This means, for example, that the following hostname is generated with address switch value 0xD7 (dec. 215): "brmb215".

If DNS service is not available on the network, the bus controller's two NetBIOS names can also be used for access. The secondary NetBIOS name is identical to the hostname. If the address switches are set to 0x00, it is identical to the primary NetBIOS name. The bus controller can only be reached via its NetBIOS name if no other routers or gateways are in the way.

5.2 Changing the IP address with the network address switches

The address switches can be used to change the last byte in the IP address configured on the bus controller. The IP address saved in flash memory is not changed. If the address switches are set to 0x00, the bus controller applies the IP address last saved to flash memory. Switch positions between 0x01 and 0x7F cause the last position of the IP address (the lowest byte) to be overwritten by the value of the address switch. This provides the user a quick and easy way to address a large number of bus controllers. In short, an IP address between 192.168.100.1 and 192.168.100.127 can be selected for a bus controller using the address switches without requiring any additional software configuration.

5.3 Overview of network address switch values

Switch position	Description
0x00	This switch position is the factory default setting. In this position, the address switches have no effect on system parameters. The bus controller parameters in flash memory are used (IP address or interface number). The bus controller is started with factory default values if valid flash data is not present.
0x01 - 0x7F	The last position of the IP address saved in flash memory is changed to the address switch value. The IP address saved in flash memory is not changed. The interface number is read from flash memory.
0x80 - 0xEF	Sets the bus controller to DHCP mode for this range. The DNS server is informed of the current hostname. A hostname is generated according to the setting of the address switch. **Example** The generated hostname is made up of 3 elements: "br" + "mb" + Address switch value (3 decimal places) This means, for example, that the following hostname is generated for address switch setting 0xD7 (dec. 215): "brmb215".
0xF0	Auto-store mode: The IP settings are obtained from the DHCP or BooTP server. If the IP settings are different than the values stored in flash memory, then the current IP parameters are saved. This function is available in firmware version 1.39 and later.
0xF1 - 0xFD	Reserved (same function as position 0xFF).
0xFE	Initializes all bus controller parameters with default values during booting. No values are read from flash memory. The communication parameters correspond to the values assigned with switch setting 0xFF.
0xFF	Initializes all communication parameters with default values. All other bus controller parameters are read from flash memory. Default parameters: • IP address: 192.168.100.1 • Subnet mask: 255.255.255.0 • Gateway: 192.168.100.254 • Primary NetBIOS name: "br" + MAC address • Secondary NetBIOS name: "br" + "mb" + address switch value (decimal) • Interface number: 502 • X2X Link configuration: 4 ms cycle time • X2X Link cable length: 0 m

5.4 Information about NetBIOS names

In addition to the hostname used to register on the DHCP server, the bus controller also has so-called NetBIOS names. These are used to access the bus controller from a PC using its name (as opposed to its IP address). This is only possible if no routers or gateways are in the way, however.

The primary NetBIOS name is always composed of the prefix "br" and the MAC address from the bus controller (see "Automatic IP assignment by a DHCP server" on page 23).

The secondary NetBIOS name corresponds to the primary NetBIOS name at address switch position 0x00. This is necessary because several bus controllers with address switch value 0x00 are permitted to be located in one network segment. In this case, the IP address from flash memory is used.

For all other address switch positions, the secondary NetBIOS name is generated from the network address switch value (as in DHCP mode): "br" + "mb" + Address switch value (3 decimal places).
A hostname defined explicitly by the user will be used for the secondary NetBIOS name regardless of the address switch value.

This makes it possible to access the bus controller with the NetBIOS name configured using the address switches. This is also possible if the controller was not configured for use with a DHCP server (address switch setting between 0x01 and 0x7F).

5.5 Saving an IP address to flash memory

The IP parameters in flash memory can be changed via the Modbus protocol, the ModbusTCP Toolbox or the Telnet interface. The ModbusTCP Toolbox can be downloaded from the B&R website.
The IP address, subnet and gateway are all defined in the address range 0x1003 to 0x100E. The data has a length of 4 words in each case. The data is applied by writing constant 0xC1 to address 0x1140 ("Write single register" fc6, addr. 0x1140, data 0xC1). The new settings are applied after the next startup of the bus controller.

6 Bus controller process image

6.1 General information

After it is booted, the bus controller detects and starts all connected I/O modules and creates an internal image of the input and output data.
If configuration data for the I/O modules is stored in flash memory on the bus controller, the respective modules will be configured at startup.

In the event that additional I/O modules are enabled during operation and the bus controller parameter "I/O module configuration mode" on page 56 is set to the value 0xC0 (incomplete configuration), then the process image is updated automatically .

All data from the I/O modules is then mapped in a vector with a width of 16 bits. Depending on the data type, I/O data is split up among different address ranges: All analog and more complex X2X Link modules are word-oriented. Data exchange takes place on a 16-bit basis, with the most significant byte transferred in the first position (big-endian). All digital X2X Link modules and status data is byte-oriented and mapped in order in the 16-bit process image. An empty spacer byte is used when there is an odd number of bytes.

A copy of digital I/O data is also mapped in a separate discrete bit-oriented image. This bit-oriented area is split up into an input and output area, each of which starts at address 0x0000. Bit-oriented Modbus functions can be used to access this area.

> ### Information:
>
> **The number and length of the various input and output data can be requested using the process image parameters (see "Process image data" on page 50).**
>
> **The starting addresses (index) of the analog as well as digital input and output data can be queried using the module parameters of the respective I/O module (see "Module parameter overview" on page 65).**

6.2 Structure of the process image

6.2.1 Word-oriented

Word Address range	Number of word objects	Description	Access methods	Permitted Modbus functions
0x0000 - 0x07FF	2048	Analog inputs	Read	3, 4, 23
0x0800 - 0x0FFF	2048	Analog outputs	Read/Write	3, 4, 6, 16, 23
0x1000 - 0x1FFF	4096	System parameters	Read/Write	3, 4, 6, 16, 23
0x2000 - 0x23FF	1024	Digital inputs	Read	3, 4, 23
0x2400 - 0x27FF	1024	Digital outputs	Read/Write	3, 4, 6, 16, 23
0x2800 - 0x29FF	512	X2X Link network status	Read	3, 4, 23
0x2A00 - 0x2BFF	512	Analog or digital output status	Read	3, 4, 23
0x2C00 - 0x9FFF	29696	Reserved	Read	3, 4, 23
0xA000 - 0xAFCF	4048	Module data organized by slot index	Read/Write	3, 4, 6, 16, 23
0xAFD0 - 0xAFFF	48	Reserved (data for 3 modules)	Read	3, 4, 23
0xB000 - 0xBFFF	4096	Module data organized by parameter	Read/Write	3, 4, 6, 16, 23
0xC000 - 0xDFFF	8192	Module configuration data	Read/Write	3, 4, 6, 16, 23
0xE000 - 0xFFFF	16384	Reserved	Read	3, 4, 23

6.2.2 Bit-oriented

Bit address range	Number of bit objects	Description	Access methods	Permitted Modbus functions
0x0000 - 0x3FFF	16384	Digital input data	Read	2

Bit address range	Number of bit objects	Description	Access methods	Permitted Modbus functions
0x0000 - 0x3FFF	16384	Digital output data	Read/Write	1, 5, 15

Information:

If the number of digital I/O channels of a module does not completely fill a byte, then the missing bits are completed with zeros, i.e. the smallest mapped data unit per module is one byte.

6.3 Example of an X20 process image

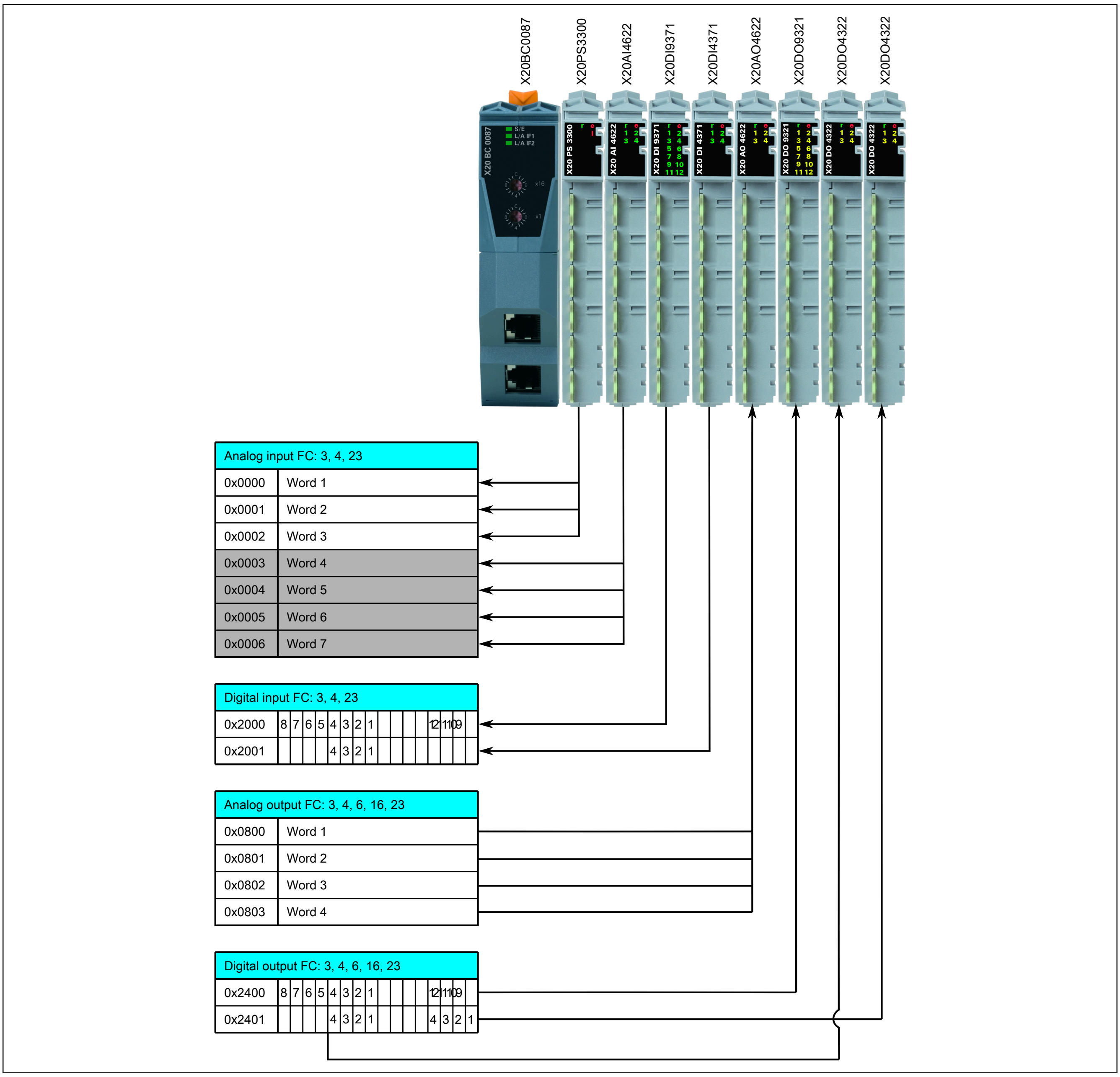
Module name	Module type	Input	Output
X20PS9400	Supply module	3 analog channels (6 bytes)	-
X20AI4622	Analog input	4 analog channels (8 bytes)	-
X20DI9371	Digital input	12 digital channels (2 bytes)	-
X20DI4371	Digital input	4 digital channels (1 byte)	-
X20AO4622	Analog output	-	4 analog channels (8 bytes)
X20DO9321	Digital output	-	12 digital channels (2 bytes)
X20DO4322	Digital output	-	4 digital channels (1 byte)

Excel file "Modbus TCP mapping tool" can be downloaded from the B&R website (www.br-automation.com). It shows which I/O data points are available for the individual modules and how they are located in the process image with an automatic configuration.

Information:

The process image parameters as well as the module parameters for each I/O module can be used to query the number and length (see "Process image data" on page 50) or starting addresses (index) of both analog and digital input/output data (see "Module parameter overview" on page 65).

6.3.1 Word-oriented mapping



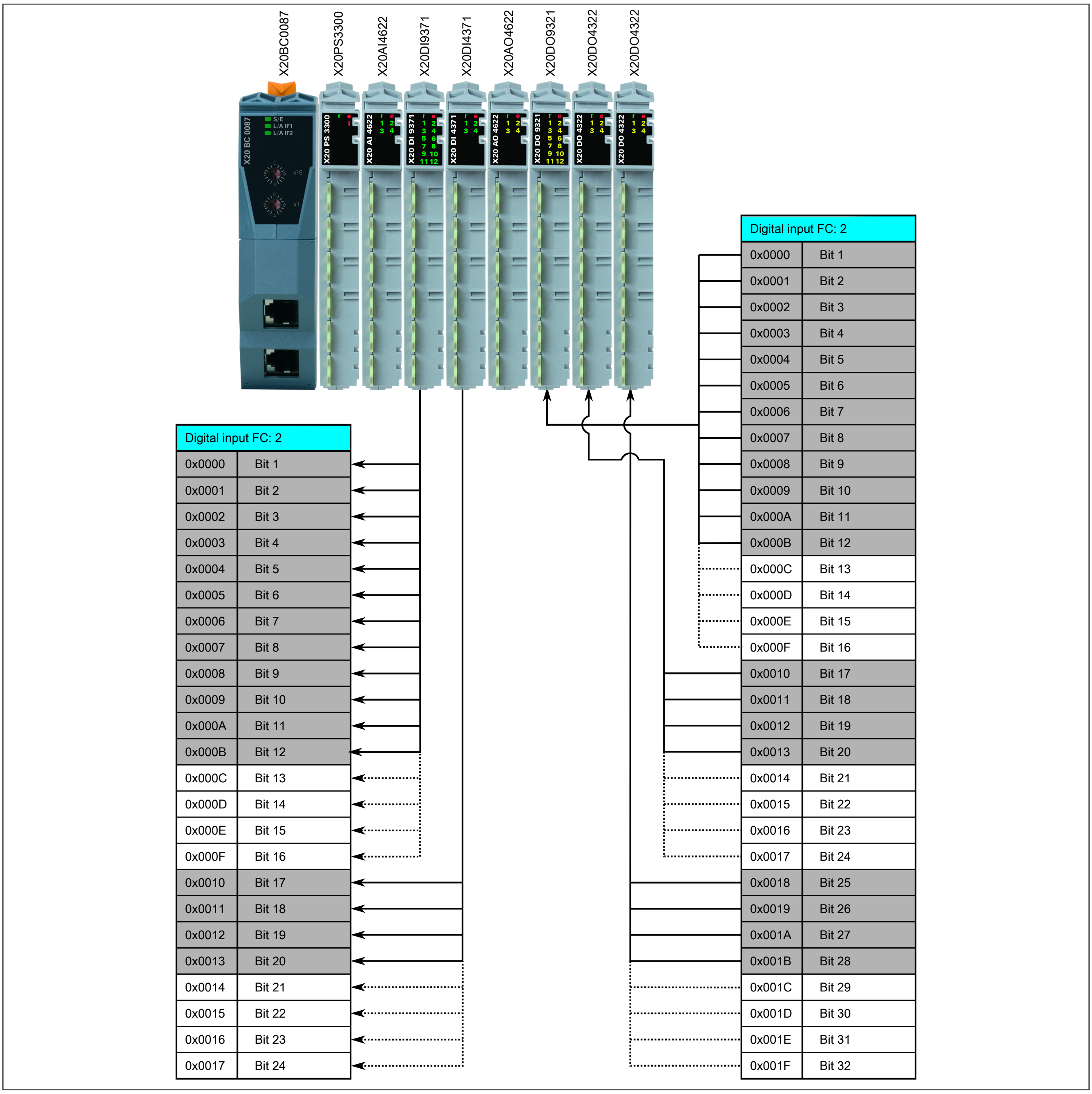
Analog input FC: 3, 4, 23	
0x0000	Word 1
0x0001	Word 2
0x0002	Word 3
0x0003	Word 4
0x0004	Word 5
0x0005	Word 6
0x0006	Word 7

Digital input FC: 3, 4, 23																
0x2000	8	7	6	5	4	3	2	1					12	11	10	9
0x2001					4	3	2	1								

Analog output FC: 3, 4, 6, 16, 23	
0x0800	Word 1
0x0801	Word 2
0x0802	Word 3
0x0803	Word 4

Digital output FC: 3, 4, 6, 16, 23																
0x2400	8	7	6	5	4	3	2	1					12	11	10	9
0x2401					4	3	2	1					4	3	2	1

6.3.2 Bit-oriented mapping



Digital input FC: 2	
0x0000	Bit 1
0x0001	Bit 2
0x0002	Bit 3
0x0003	Bit 4
0x0004	Bit 5
0x0005	Bit 6
0x0006	Bit 7
0x0007	Bit 8
0x0008	Bit 9
0x0009	Bit 10
0x000A	Bit 11
0x000B	Bit 12
0x000C	Bit 13
0x000D	Bit 14
0x000E	Bit 15
0x000F	Bit 16
0x0010	Bit 17
0x0011	Bit 18
0x0012	Bit 19
0x0013	Bit 20
0x0014	Bit 21
0x0015	Bit 22
0x0016	Bit 23
0x0017	Bit 24

Digital input FC: 2	
0x0000	Bit 1
0x0001	Bit 2
0x0002	Bit 3
0x0003	Bit 4
0x0004	Bit 5
0x0005	Bit 6
0x0006	Bit 7
0x0007	Bit 8
0x0008	Bit 9
0x0009	Bit 10
0x000A	Bit 11
0x000B	Bit 12
0x000C	Bit 13
0x000D	Bit 14
0x000E	Bit 15
0x000F	Bit 16
0x0010	Bit 17
0x0011	Bit 18
0x0012	Bit 19
0x0013	Bit 20
0x0014	Bit 21
0x0015	Bit 22
0x0016	Bit 23
0x0017	Bit 24
0x0018	Bit 25
0x0019	Bit 26
0x001A	Bit 27
0x001B	Bit 28
0x001C	Bit 29
0x001D	Bit 30
0x001E	Bit 31
0x001F	Bit 32

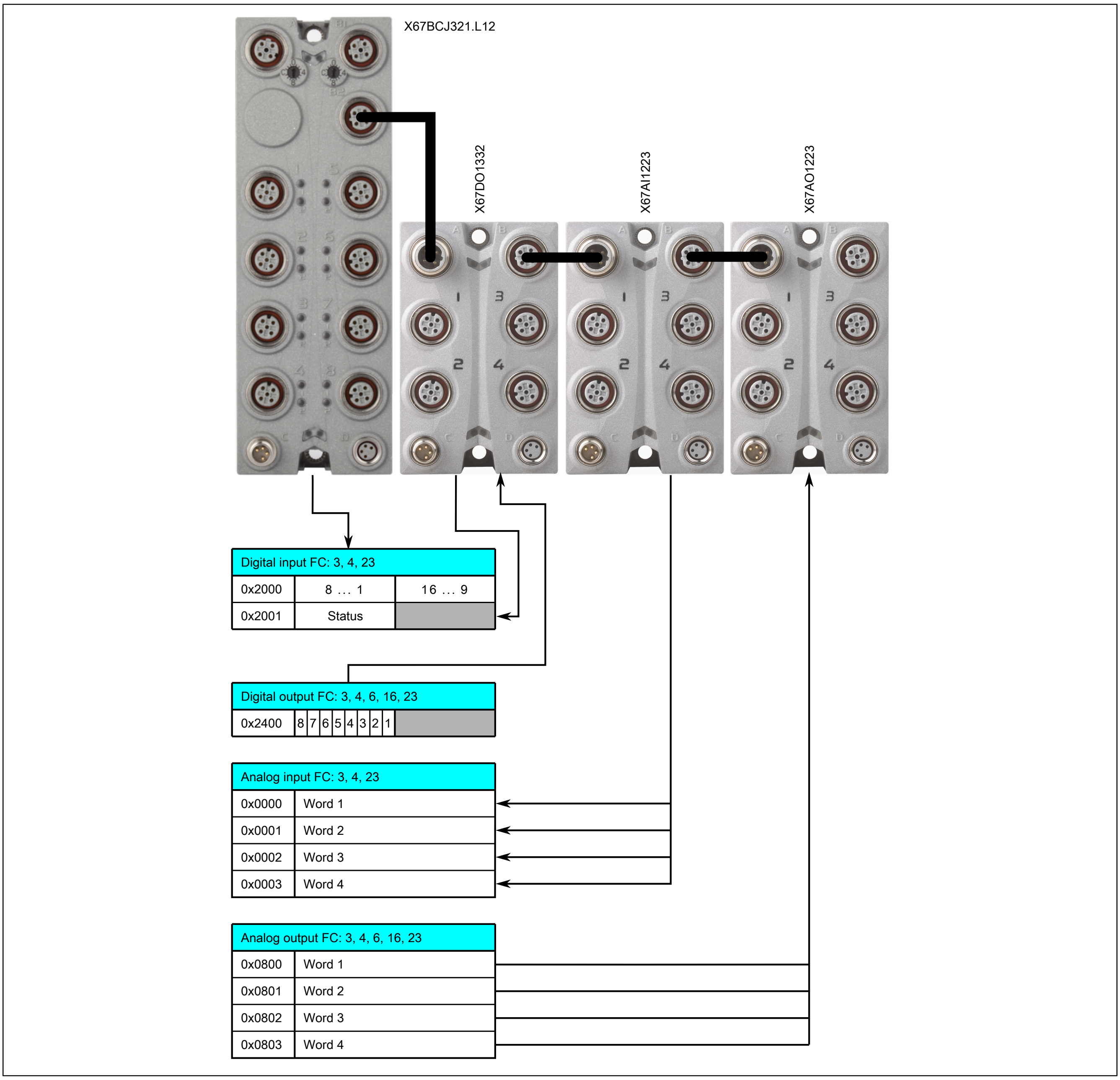
6.4 Example of an X67 process image

Module name	Module type	Input	Output
X67BCJ321.L12	Bus controller	16 digital channels	-
X67DO1332	Digital output	-	8 digital channels
X67AI1223	Analog inputs	4 analog inputs	-
X67AO1223	Digital input	-	4 analog outputs

Excel file "Modbus TCP mapping tool" can be downloaded from the B&R website (www.br-automation.com). It shows which I/O data points are available for the individual modules and how they are located in the process image with an automatic configuration.

> ## Information:
>
> **The process image parameters as well as the module parameters for each I/O module can be used to query the number and length (see "Process image data" on page 50) or starting addresses (index) of both analog and digital input/output data (see "Module parameter overview" on page 65).**

6.4.1 Word-oriented mapping



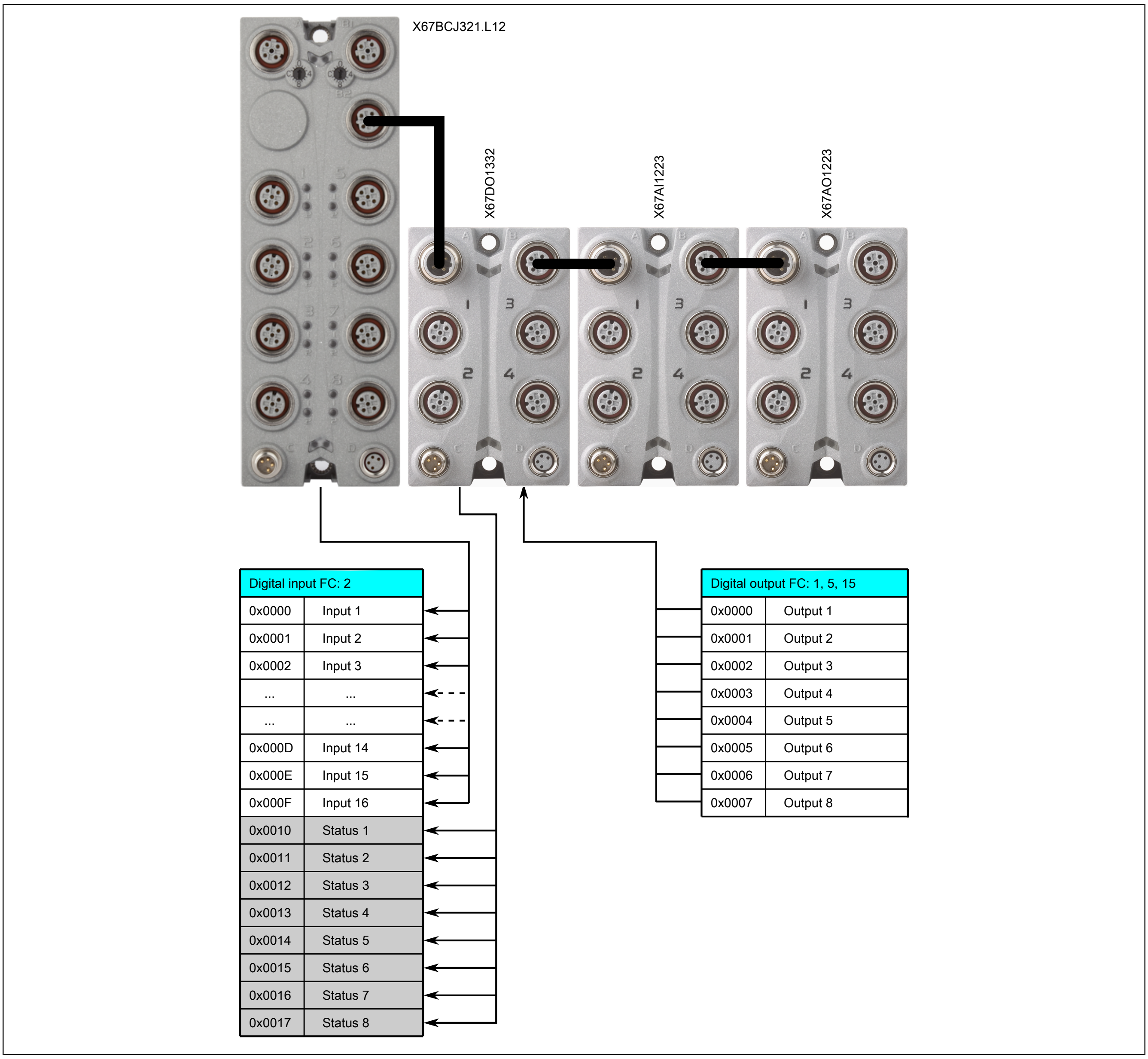
Digital input FC: 3, 4, 23		
0x2000	8 ... 1	16 ... 9
0x2001	Status	

Digital output FC: 3, 4, 6, 16, 23									
0x2400	8	7	6	5	4	3	2	1	

Analog input FC: 3, 4, 23	
0x0000	Word 1
0x0001	Word 2
0x0002	Word 3
0x0003	Word 4

Analog output FC: 3, 4, 6, 16, 23	
0x0800	Word 1
0x0801	Word 2
0x0802	Word 3
0x0803	Word 4

6.4.2 Bit-oriented mapping



Digital input FC: 2	
0x0000	Input 1
0x0001	Input 2
0x0002	Input 3
...	...
...	...
0x000D	Input 14
0x000E	Input 15
0x000F	Input 16
0x0010	Status 1
0x0011	Status 2
0x0012	Status 3
0x0013	Status 4
0x0014	Status 5
0x0015	Status 6
0x0016	Status 7
0x0017	Status 8

Digital output FC: 1, 5, 15	
0x0000	Output 1
0x0001	Output 2
0x0002	Output 3
0x0003	Output 4
0x0004	Output 5
0x0005	Output 6
0x0006	Output 7
0x0007	Output 8

7 Configuration of the I/O modules

7.1 General information

The B&R Modbus TCP bus controller recognizes several different types of configuration for connected X2X Link I/O modules:

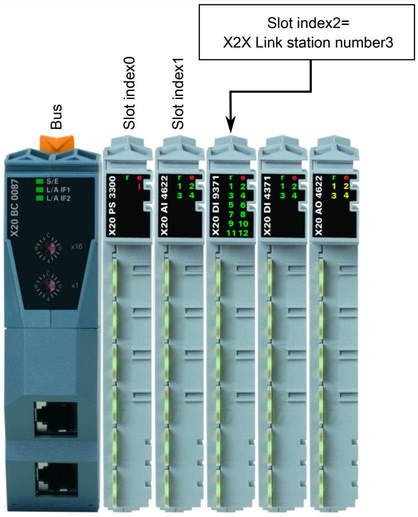
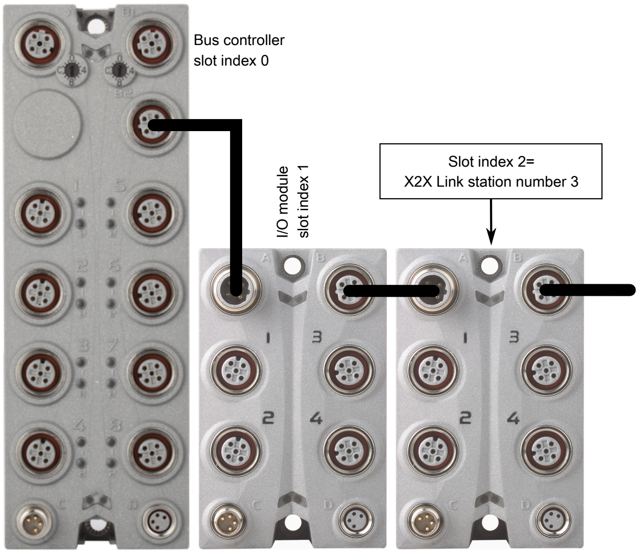
- Automatic configuration
- Mixed configuration
- Full configuration (manual configuration)

The bus controller parameters relevant for these configuration options are listed below:

Parameter name	Modbus address	For description, see
I/O module configuration mode	0x1188	"Miscellaneous" on page 53
Required module hardware ID	0xA**8 or 0xB8**	"I/O module register configuration" on page 70
Module start mode	0xA**9 or 0xB9**	"I/O module register configuration" on page 70
Module configuration data index	0xA**A or 0xBA**	"I/O module register configuration" on page 70
Module configuration data length	0xA**B or 0xBB**	"I/O module register configuration" on page 70
Module configuration data	0xC000 to 0xDFFF	"Example of a register configuration" on page 71

Information:

The * symbols of module-specific parameter addresses correspond to the slot index, with the first module after the power supply or supply module using index "1". The power supply has slot index "0".

X20	X67
 Slot index2= X2X Link station number3 Bus Slot index0 Slot index1	 Bus controller slot index 0 I/O module slot index 1 Slot index 2= X2X Link station number 3

Information:

There is a direct relationship between the slot index and the X2X Link network address switch: X2X Link network switch value = Slot index + 1

In Automation Studio the X2X Link network address switch is always displayed.

Automation Studio can be used to configure the Modbus TCP bus controller and connected I/O modules. This generates an XML file containing a full configuration that can be transferred to the bus controller using the ModbusTCP Toolbox.

In addition, the module register configuration and addresses for accessing I/O data points are written to text files and an HTML file. This information simplifies configuration via a third-party master.

7.2 Automatic configuration

If there are no references to the module configuration data on startup (i.e. configuration data length 0xA**B is set to 0), then the connected I/O modules are configured automatically. In this case, any configuration data present in the range 0xC000 to 0xDFFF is ignored.

> ### Information:
>
> **This requires that the I/O module configuration mode be set to the value 0xC0. If the value 0xC1 is configured, then valid module configuration data is required!**

To delete all reference entries, the module configuration header data can be initialized by writing the constant 0xC0 to the address 0x1145. This function sets the parameters "Module configuration data index" on page 69, "Module configuration data length" on page 69 and "Required module hardware ID" on page 68 to the value 0. The "Module start mode" on page 68 parameter (function model) is initialized with 0xFE (decimal 254). This change must then be enabled by saving the new parameter to the flash memory.

With automatic configuration, each module is operated in standard function model "254" . On startup, each module reports the length of their cyclic input and output registers to the bus controller, which then uses this information to create the I/O process image.

> ### Information:
>
> **It is not possible to use bus modules with node number switches (e.g. X20BM15, X67DM9321) in the "Automatic configuration" operating mode (see "Empty module slots" on page 34).**

7.3 Mixed configuration

This type of configuration is used if the configuration of individual modules is different than that provided by their default parameters.
It is also possible to configure certain slots within X2X Link in a flexible manner, i.e different modules can be used in the same slot. This is referred to as a "wildcard" configuration.

> **Information:**
>
> **In a mixed configuration, the I/O module configuration mode must be set to the value 0xC0 (default value). If set to 0xC1, the bus controller uses only the specified module configuration data and only registers the module registers that have been configured there. This is referred to as a Full configuration.**

An I/O module can be referenced to one or more consecutive register configurations. To do this, the starting address of a configuration data block must be specified with the "Module configuration data index" on page 69 parameter in the module-specific parameters for the respective slot index; the length of the block must be specified with the "Module configuration data length" on page 69 parameter. The length corresponds to the number of configuration entries, i.e. an entry with 4 words has a length of 1.
For an example, see "Example of a register configuration" on page 71.

> **Information:**
>
> **If an I/O module references a blank or deleted configuration data range, then an error occurs since the bus controller is attempting to set register address 0, type 0, length 0 and value 0.**

In a mixed configuration, the bus controller initially behaves in a way that is similar to an automatic configuration. Following startup, the registers from all I/O modules are requested. In addition, every module is checked for existing configuration data via module-specific parameters (index and length). A combination of default module data and user-defined configuration data is used.

Since each entry uses 4 words, the bus controller's address range of 0xC000 to 0xDFFF can be used for 2048 registers of configuration data. The configuration parameters and the default configuration can be taken from the description of the corresponding module.

> **Information:**
>
> **Modules with identical configuration data are permitted to reference the same block in order to save space.**

The "Required module hardware ID" on page 68 and "Module start mode" on page 68 parameters are optional, i.e. they do not have to be specified in a mixed configuration.

7.3.1 Configuration of multi-function modules

Some I/O modules support other function models in addition to standard function model "254".

> **Information:**
>
> **Either a partial or full configuration must be made in order to operate this type of module in a different function model.**

In principle, it is sufficient to set the "Module start mode" on page 68 parameter to the desired value for the corresponding slots. This can be taken from the respective module description.

If a module that does not support this function model is located in one of these X2X Link slots, then this module will not be started. This is indicated by an LED directly on the module and can also be read from the bus controller via the Module status parameter.

7.3.2 "Wildcard" configuration

It is also possible to configure certain slots within X2X Link in a flexible manner, i.e different modules can be used in the same slot.

> ## Information:
>
> **This configuration corresponds to the factory default setting. A "wildcard" configuration is only possible if a mixed configuration is configured in I/O module configuration mode (0xC0).**

To do so, the "Required module hardware ID" on page 68 parameter for the corresponding slots must be set to the value 0x0000.

This will accept all I/O modules in addition to configuring and booting them with the corresponding configuration data (i.e. a combination of module default data and the configuration data defined by the user in the address range 0xC000 to 0xDFFF). To boot subsequent I/O modules, an I/O module must be physically present in this slot, or module-specific cyclic registers must have been defined in the configuration data for this slot.

7.3.3 Empty module slots

In order to leave bus modules empty or to use bus modules with node number switches, the "Required module hardware ID" on page 68 parameter must be set to the value 0xFFFF for the X2X Link slots not being used. No mapping entries are generated for this slot regardless of whether an actual I/O module is inserted. Subsequent I/O modules are not affected by one or more empty slots.

> ## Information:
>
> **If parameter "Required module hardware ID" on page 68 remains set to the factory default value of 0x0000 with empty slots, the I/O modules following this slot will not be started!**

7.3.4 Specifying the I/O module hardware ID

If the I/O module hardware ID for one or more X2X Link slots should be specified – as is also done in a full configuration – then the module-specific "Required module hardware ID" on page 68 parameter must be set accordingly for the respective slots.

For the hardware ID of an X2X Link module, see the module documentation or the first 4 digits of the serial number printed on the module.

The module is only booted if the specified I/O module hardware ID matches the physical I/O module in this slot. An error is reported if an I/O module is missing or the hardware ID is different.

To boot subsequent I/O modules, either the configured I/O module must be physically present or the module-specific cyclic registers must have been defined for the missing module in the configuration data. This is because the bus controller requires information about the I/O data width of each module in order to configure X2X Link. If this information is not available for a module, then none of the modules connected to it will be started.

7.4 Full configuration

In a full configuration, the bus controller configures the I/O modules using only the module configuration data stored in the flash (address range 0xC000 to 0xDFFF). Corresponding reference entries (module header data) are needed for each module. No register information is queried from the modules. Each of these configuration entries uses 4 words (see "Structure of the configuration data block" on page 36).

An I/O module can be referenced to one or more consecutive register configurations. To do this, the starting address of a configuration data block must be specified with the "Module configuration data index" on page 69 parameter in the module-specific parameters for the respective slot index; the number of configuration entries for this block must be specified with the "Module configuration data length" on page 69 parameter. For an example, see "Example of a register configuration" on page 71.

Modules with identical configuration data are permitted to reference the same block in order to save space.

> **Information:**
>
> **In a full configuration, the I/O module configuration mode must be set to the value 0xC1 (default value).**

If an I/O module references a blank or deleted configuration data range, then an error occurs since the bus controller is attempting to set register address 0, type 0, length 0 and value 0.

It is absolutely mandatory to specify the "Required module hardware ID" on page 68 parameter for a full configuration. A "Wildcard" configuration is not possible.
Module start mode is optional, i.e. it does not have to be specified in a full configuration. Its default value is 0xFE (decimal 254). This enables the standard function model for the respective module.

7.4.1 Auto mode

Auto mode refers to situations where additional modules are connected to the bus controller together with the I/O modules configured in a full configuration. These additional modules must have a higher slot ID (i.e. the network address switch values are higher in X2X Link) than those that are configured.
These modules are configured automatically as described in "Automatic configuration" on page 19.
This type of configuration requires that all modules with lower X2X Link network address switch values be configured in a uniform manner (i.e. together in a block).

7.4.2 Structure of the configuration data block

A configuration data block is made up of the following entries:

Modbus address starting at 0xC000	Explanation
Word 1	Register number (register address)
Word 2	Register type (high byte) + Register size (low byte)
Word 3	Register value high word
Word 4	Register value low word

Word 1 (register number) must contain the hexadecimal equivalent of the module's register address. The register numbers can be taken from the respective module description.
Word 2 contains the register type in the higher-value byte and the register length (in bytes) in the lower-value byte. Both values must be specified in hexadecimal.

Register type	Explanation
0	Cyclic dynamic input register (read)
1	Cyclic dynamic output register (write)
2	Cyclic fixed input register (read)
3	Cyclic fixed output register (write)
4	Reserved
5	Acyclic output register (read), normally used for configurations

Example

In function model 1, a counter module has a register (register number 2064) of data type DINT (4 bytes long) called "CfO_PresetABR01_1_32-bit" for setting (initializing) the counter state for a homing procedure.
This is an acyclic output register (type 5). The correct value for word 1 is 0x0810 (dec. 2064). For word 2, it is 0x0504 (type 5 and 4 bytes long).

If the counter from our example is initialized with the lowest possible value (dec. -2147483648), then word 3 = 0x8000 and word 4 = 0x0000 (0x80000000 is the two's complement representation of decimal -2147483648).

For additional examples, see "Example of a register configuration" on page 71.

8 System parameters

8.1 Overview of system parameters

Communication	
Address range	**Description**
0x1000 - 0x1002	MAC address
0x1003 - 0x1006	IP address
0x1007 - 0x100A	Subnet mask
0x100B - 0x100E	Default gateway
0x100F	Modbus port number
0x1010	Lifespan of the TCP connection [sec.]
0x1011	IP maximum transmission unit
0x1012	X2X Link configuration
0x1013 - 0x1016	IP address currently being used
0x1017	X2X Link cable length
0x1018 - 0x101E	Hostname
0x101F - 0x1025	TelnetPasswort
0x1027 - 0x1029	Controller for the interfaces
0x102B - 0x102E	Network mask currently being used
0x102F - 0x1032	Gateway currently being used

Watchdog	
Address range	**Description**
0x1040	Watchdog threshold [ms]
0x1041	Current value of the watchdog timer in ms
0x1042	Watchdog status
0x1043	Watchdog mode
0x1044	Watchdog reset

Product data	
Address range	**Description**
0x1080 - 0x1082	Serial number
0x1083	Product code
0x1084	Hardware major revision
0x1085	Hardware minor revision
0x1086	Active firmware major revision
0x1087	Active firmware minor revision
0x1088	FPGA hardware revision
0x1089	Active boot block
0x108A	Default firmware major revision
0x108B	Default firmware minor revision
0x108C	Update firmware major revision
0x108D	Update firmware minor revision
0x108E	Default FPGA software revision
0x108F	Update FPGA software revision

Modbus protocol statistics	
Address range	**Description**
0x10C0	Number of client connections
0x10C1 - 0x10C2	Global telegram counter
0x10C3 - 0x10C4	Local telegram counter
0x10C5 - 0x10C6	Global protocol error counter
0x10C7 - 0x10C8	Local protocol error counter
0x10C9 - 0x10CA	Global maximum command execution time in µs
0x10CB - 0x10CC	Local maximum command execution time in µs
0x10CD - 0x10CE	Global minimum command execution time in µs
0x10CF - 0x10D0	Local minimum command execution time in µs
0x10D1 - 0x10D2	Global protocol fragment counter
0x10D3 - 0x10D4	Local protocol fragment counter

Process image data	
Address range	**Description**
0x1100	Number of modules
0x1101	Number of analog input registers
0x1102	Size of the analog input registers in bytes
0x1103	Number of analog output registers
0x1104	Size of the analog output registers in bytes
0x1105	Number of digital input registers
0x1106	Size of the digital input registers in bytes
0x1107	Number of digital output registers

Process image data	
Address range	**Description**
0x1108	Size of the digital output registers in bytes
0x1109	Number of analog and digital output status registers
0x110A	Size of the analog and digital output status registers in bytes
0x110B	Number of X2X Link network status registers
0x110C	Size of the X2X Link network status registers in bytes

Controller	
Address range	**Description**
0x1140	Save all system data to flash memory
0x1141	Read all system data from flash memory
0x1142	Delete entire flash memory
0x1143	Restart system
0x1144	Close all TCP connections
0x1145	Initialize module configuration header data
0x1146	Initialize module configuration data
0x1147	Initialize user data

Miscellaneous	
Address range	**Description**
0x1180	Reading network address switches
0x1181	Module initialization delay in ms
0x1182	Verification mode for I/O access limits
0x1183	Enable/Disable Telnet password
0x1184	Modified configuration flag
0x1185	Default configuration flag
0x1186	Bus controller operating status (error-free state)
0x1187	Bus controller error status (error state)
0x1188	I/O module configuration mode
0x1189	Bus controller Error/Status LED signal mask
0x118A	Process data byte order

X2X Link statistics	
Address range	**Description**
0x11C0	X2X Link cycle counter
0x11C1	Number of X2X Link off cycles
0x11C2	Cyclic errors
0x11C3	Cyclic: Bus timing errors
0x11C4	Cyclic: Frame timing errors
0x11C5	Cyclic: Frame checksum errors
0x11C6	Cyclic: Frame pending errors
0x11C7	Cyclic: Buffer underrun
0x11C8	Cyclic: Buffer overflow
0x11C9	Acyclic errors
0x11CA	Acyclic: Bus timing errors
0x11CB	Acyclic: Frame timing errors
0x11CC	Acyclic: Frame checksum errors
0x11CD	Acyclic: Frame pending errors
0x11CE	Acyclic: Buffer underrun
0x11CF	Acyclic: Buffer overflow

Network statistics	
Address range	**Description**
0x1200	IF1: Ethernet frames received
0x1201	IF1: Frames lost due to high load
0x1202	IF1: Oversized frames
0x1203	IF1: CRC error
0x1204	IF1: Frames lost
0x1205	IF1: Frames lost due to high load
0x1206	IF1: Collisions
0x1207	IF1: Frames lost due to switch overflow
0x1208	IF1: Frames lost due to switch errors
0x1210	IF2: Ethernet frames received
0x1211	IF2: Frames lost due to high load
0x1212	IF2: Oversized frames
0x1213	IF2: CRC error
0x1214	IF2: Frames lost
0x1215	IF2: Frames lost due to high load
0x1216	IF2: Collisions
0x1217	IF2: Frames lost due to switch overflow
0x1218	IF2: Frames lost due to switch errors

User data	
Address range	**Description**
0x1240 - 0x1241	Configuration data checksum
0x1242 - 0x127F	User data block

Acyclic I/O register configuration	
Address range	**Description**
0x1280 - 0x1283	Write to acyclic I/O registers
0x1284 - 0x1285	Read from acyclic I/O registers
0x1286 - 0x1287	Result of the I/O register read operation

8.2 Description of individual module parameters

8.2.1 Communication

8.2.1.1 MAC address

MAC address	
Address or address range	0x1000 - 0x1002
Data length in words	3
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	00-60-65-xx-yy-zz
Description	Internationally unique physical MAC (Media Access Control) address. This address is permanently assigned and can only be read. The MAC address is also printed on the bus controller housing next to the B&R logo and used for addressing purposes in a network (see "Information about NetBIOS names" on page 24). Transmission methods:

Word 1		Word 2		Word 3	
0x1000		0x1001		0x1002	
00	60	65	xx	yy	zz

8.2.1.2 IP address

IP address	
Address or address range	0x1003 - 0x1006
Data length in words	4
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	192.168.100.1
Description	Freely configurable IP address. The default value is 192.168.100.1. **To be able to use this configured IP address, the network address switches must be set to the value 0x00 (see "Changing the IP address with the network address switches" on page 23).** Changes are only applied after a restart. Transmission methods:

Word 1	Word 2	Word 3	Word 4
0x1003	0x1004	0x1005	0x1006
192	168	100	1

8.2.1.3 Subnet mask

Subnet mask	
Address or address range	0x1007 - 0x100A
Data length in words	4
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	255.255.255.0
Description	Freely configurable subnet mask. The default value is 255.255.255.0. Changes are only applied after a restart. Transmission methods:

Word 1	Word 2	Word 3	Word 4
0x1007	0x1008	0x1009	0x100A
255	255	255	0

8.2.1.4 Default gateway

Default gateway	
Address or address range	0x100B - 0x100E
Data length in words	4
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	192.168.100.254
Description	Freely configurable default gateway. The default value is 192.168.100.254. Changes are only applied after a restart. Transmission methods:

Word 1	Word 2	Word 3	Word 4
0x100B	0x100C	0x100D	0x100E
192	168	100	254

8.2.1.5 Modbus port number

Modbus port number	
Address or address range	0x100F
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	502
Description	The default Modbus TCP port number is 502. The Modbus server can also be operated with a different port number, however. Changes are only applied after a restart.

8.2.1.6 Lifespan of the TCP connection [sec.]

Lifespan of the TCP connection [sec.]	
Address or address range	0x1010
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	0 [sec]
Description	Period of inactivity for TCP communication. The server or bus controller closes the TCP connection if no TCP requests are received during this specified period. Times are not monitored if the parameter value is 0. In this case, the connection is never closed by the server. This parameter is specified in seconds.

8.2.1.7 IP maximum transmission unit

IP maximum transmission unit	
Address or address range	0x1011
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	1500 [bytes]
Description	The maximum transmission unit (MTU) specifies the maximum size of the complete TCP/IP packet. The smaller the packet size, the more the payload data is fragmented. Values between 100 and 1500 are permitted. **The Modbus master must be able to process fragmented telegrams, if necessary.** Changes are only applied after a restart.

8.2.1.8 X2X Link configuration

X2X Link configuration	
Address or address range	0x1012
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	0xC0 (4 ms)
Description	The X2X Link cycle time and the resulting data width can only be configured together. The following values are possible depending on the required cycle time and number of connected I/O modules:

Value	Cycle time	Description
0xC0	4 ms	Max. 253 I/O modules, max. 1400 bytes of cyclic data
0xC1	3.5 ms	Max. 253 I/O modules, max. 1150 bytes of cyclic data
0xC2	3 ms	Max. 253 I/O modules, max. 900 bytes of cyclic data
0xC3	2.5 ms	Max. 200 I/O modules, max. 800 bytes of cyclic data
0xC4	2 ms	Max. 200 I/O modules, max. 500 bytes of cyclic data
0xC5	1.5 ms	Max. 100 I/O modules, max. 450 bytes of cyclic data
0xC6	1 ms	Max. 80 I/O modules, max. 300 bytes of cyclic data
0xC7	0.5 ms	Max. 40 I/O modules, max. 120 bytes of cyclic data

Changes are only applied after a restart.

8.2.1.9 IP address currently being used

IP address currently being used	
Address or address range	0x1013 - 0x1016
Data length in words	4
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	192.168.100.1
Description	Contains the IP address currently being used by the Modbus TCP bus controller (server). Transmission methods:

Word 1	Word 2	Word 3	Word 4
0x1013	0x1014	0x1015	0x1016
192	168	100	1

8.2.1.10 X2X Link cable length

X2X Link cable length	
Address or address range	0x1017
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	0 [m]
Description	This parameter is used to optimize the X2X Link timing with respect to low ESD emissions. If set to the default value (0), no optimization takes place. The actual total length (in meters) of the X2X Link line starting from the bus controller must be specified. The maximum length is determined by the maximum distance between 2 X2X Link stations (100 m) and the maximum number of stations (253 modules), which equals in total 25.3 km. Changes are only applied after a restart.

8.2.1.11 Hostname

Hostname	
Address or address range	0x1018 - 0x101E
Data length in words	7
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	0
Description	This parameter range is used to define a hostname. 2 ASCII characters are packed into one word (parameter). The resulting maximum length for the hostname is 14 characters. Only alphanumeric characters are permitted. The hostname is **not** case-sensitive. The first null character is interpreted as the end of the string. If the string has a length of 0 bytes, then the default hostname will be used during initialization (see "Automatic IP assignment by a DHCP server" on page 23). Changes are only applied after a restart.

8.2.1.12 TelnetPasswort

IP address	
Address or address range	0x101F - 0x1025
Data length in words	7
Access methods	Read/Write
Permissible Modbus functions	3, 4, 6, 16, 23
Default value	"BcModBus"
Description	This register range is used to define a Telnet password. 2 ASCII characters are packed into one word. The following must be taken into account when defining the password: • The maximum length of the password is 14 characters. • Only alphanumeric characters are permitted. • The password is case-sensitive. • The first null byte is interpreted as the end of the password. • If the length is 0 characters, Telnet must be used without a login. A changed password is effective immediately but not automatically saved in flash memory. Function "Enable/Disable Telnet password" on page 54 must be called to apply the password. **Data is saved to remanent memory only after the Save all system data to flash memory command is executed.** This function is available in firmware version 1.46 and later.

8.2.1.13 Controller for the interfaces

<table>
<tr><th colspan="2">Controller for the interfaces</th></tr>
<tr><td>Address or address range</td><td>0x1027 - 0x1029</td></tr>
<tr><td>Data length in words</td><td>3</td></tr>
<tr><td>Access methods</td><td>Read/Write</td></tr>
<tr><td>Permissible Modbus functions</td><td>3, 4, 16</td></tr>
<tr><td>Default value</td><td>
<table>
<tr><th>Parameter</th><th>Default value</th><th>Description</th></tr>
<tr><td>PIN</td><td>0, 0, 0, 0</td><td>The PIN is not active. The interface controller can be written to with any PIN.</td></tr>
<tr><td>cmd</td><td>0x00</td><td>No command active.</td></tr>
<tr><td>state</td><td>0xFF</td><td>All interfaces are enabled or open.</td></tr>
</table>
</td></tr>
<tr><td>Description</td><td>
The interface controller is used to manage the communication interfaces.
It gives the user the possibility of switching off unwanted interfaces. These are the UDP service channel and Telnet interfaces, which are not absolutely necessary for basic Modbus functionality.
Changes are effective immediately but not automatically saved in flash memory.
<b>Data is saved to remanent memory only after the Save all system data to flash memory command is executed.</b>
This function is available in firmware version 1.46 and later.

<b>Structure of the interface controller</b>
Writing is only possible with Modbus function 16 Write multiple registers. The length of the data must be 6 bytes.
<table>
<tr><th colspan="6">Interface controller (6-byte array)</th></tr>
<tr><td colspan="4">PIN</td><td colspan="2">ICP</td></tr>
<tr><td>Byte 1</td><td>Byte 2</td><td>Byte 3</td><td>Byte 4</td><td>cmd (byte)</td><td>state (byte)</td></tr>
</table>
<b>Explanation of parameters</b>
<table>
<tr><th>Parameter</th><th>Values</th><th>Description</th></tr>
<tr><td>PIN</td><td>x, x, x, x</td><td>Protection for the interface settings. After successful initialization, a change is only possible with a valid pin.</td></tr>
<tr><td rowspan="2">cmd</td><td>0</td><td>No command active.</td></tr>
<tr><td>1</td><td>Resets the interface controller to its default values.</td></tr>
<tr><td>state</td><td>0xFF</td><td>State of the interfaces. The following interfaces can be switched off:
<table>
<tr><th>Interface</th><th>State</th><th>Description</th></tr>
<tr><td>UDP service channel</td><td>Bit 0</td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>Interface is available.</td></tr>
<tr><td>0</td><td>Interface is blocked.</td></tr>
</table>
</td></tr>
<tr><td>Telnet</td><td>Bit 1</td><td>
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>Interface is available.</td></tr>
<tr><td>0</td><td>Interface is blocked.</td></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>
Possible errors
<table>
<tr><th>Name</th><th>Code</th><th>Description</th></tr>
<tr><td>MB_PEC_ILLEGAL_FUNCTION</td><td>1</td><td>Invalid Modbus function</td></tr>
<tr><td>MB_PEC_ILLEGAL_DATA_ADDRESS</td><td>2</td><td>Invalid data length.
Partial access is not supported. The interface controller must be written to as a coherent block.</td></tr>
<tr><td>MB_PEC_ILLEGAL_DATA_VALUE</td><td>3</td><td>Invalid PIN or "cmd" parameter.</td></tr>
</table>
<b>Using the PIN</b>
• With the PIN default value (0, 0, 0, 0), transmit any PIN together with the interface settings to the bus controller. The settings are applied immediately without restarting the bus controller.
• If the PIN is set, the bus controller is locked after 10 write attempts with an incorrect PIN. A new write is possible only after the bus controller is restarted.
• To change the PIN, parameter "cmd" with value 1 "Reset to default values" must be used. A transferred "state" parameter is not taken into account, i.e. ALL parameters must be set again afterwards.

<b>Network address switch function</b>
A changed network address switch is evaluated without restarting the bus controller.
At switch position 0xFF, the interface controller has no influence on the bus controller. All interfaces can be used, and the interface controller can be written to or reset without a valid PIN.
</td></tr>
</table>

8.2.1.14 Network mask currently being used

<table>
<tr><th colspan="2">Network mask currently being used</th></tr>
<tr><td>Address or address range</td><td>0x102B - 0x102E</td></tr>
<tr><td>Data length in words</td><td>4</td></tr>
<tr><td>Access methods</td><td>Read</td></tr>
<tr><td>Permissible Modbus functions</td><td>3, 4, 23</td></tr>
<tr><td>Default value</td><td>255.255.255.0</td></tr>
<tr><td>Description</td><td>
Contains the network mask currently being used by the Modbus TCP bus controller (server).

Transfer methods:
<table>
<tr><th>Word 1</th><th>Word 2</th><th>Word 3</th><th>Word 4</th></tr>
<tr><td>0x102B</td><td>0x102C</td><td>0x102D</td><td>0x102E</td></tr>
<tr><td>255</td><td>255</td><td>255</td><td>0</td></tr>
</table>
</td></tr>
</table>

8.2.1.15 Gateway currently being used

Gateway currently being used	
Address or address range	0x102F - 0x1032
Data length in words	4
Access methods	Read
Permissible Modbus functions	3, 4, 23
Default value	192.168.100.254
Description	Contains the gateway currently being used by the Modbus TCP bus controller (server). Transfer methods:

Word 1	Word 2	Word 3	Word 4
0x102F	0x1030	0x1031	0x1032
192	168	100	254

8.2.2 Watchdog

8.2.2.1 Watchdog threshold [ms]

Watchdog threshold [ms]	
Address or address range	0x1040
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	3000 [ms]
Description	The watchdog is used to monitor data transfers between the Modbus client and server. Depending on the selected Watchdog mode, the watchdog is reset either by any type of communication or by write access only. The monitoring function is enabled with the first telegram and triggered by additional telegrams. The watchdog is reset to 0 each time it is triggered. If the watchdog times out, then the server responds to each write command with default error code 0x0004 (slave device failure). Write commands include writing to analog or digital outputs. Read access takes place regardless of whether the watchdog has timed out. The time is specified in milliseconds.

8.2.2.2 Current value of the watchdog timer in ms

Current value of the watchdog timer in ms	
Address or address range	0x1041
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	This query can be used to determine the watchdog time that has already elapsed. This is the amount of time that has passed since the last trigger (i.e. read or write access depending on the configured mode). The watchdog begins at 0 and ends with the specified Watchdog threshold. The watchdog is reset to 0 when triggered or with the Watchdog reset command. The value is returned in milliseconds.

8.2.2.3 Watchdog status

Watchdog status	
Address or address range	0x1042
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	The watchdog status allows the user to determine the current state of the watchdog function. This can involve the following values.

Constant	Description
0xC0	Watchdog not in service
0xC1	Watchdog active
0xC2	Watchdog timeout

8.2.2.4 Watchdog mode

<table>
<tr><th colspan="2">Watchdog mode</th></tr>
<tr><td>Address or address range</td><td>0x1043</td></tr>
<tr><td>Data length in words</td><td>1</td></tr>
<tr><td>Access methods</td><td>Read/Write</td></tr>
<tr><td>Permitted Modbus functions</td><td>3, 4, 6, 16, 23</td></tr>
<tr><td>Default value</td><td>0xC1 (watchdog triggered with each record)</td></tr>
<tr><td>Description</td><td>This parameter can be used to define how the watchdog works.
<table>
<tr><th>Constant</th><th>Description</th></tr>
<tr><td>0xC0</td><td>Watchdog disabled or being disabled</td></tr>
<tr><td>0xC1</td><td>Watchdog triggered with each record</td></tr>
<tr><td>0xC2</td><td>Watchdog only triggered by write access</td></tr>
</table>
In 0xC1 mode, the watchdog is triggered by each read procedure. This is also the case if the current value of the Watchdog timer is being read. As a result, this query always produces the timer value 0.
<b>Changing the watchdog resets the watchdog, i.e. a previously expired watchdog is reset.</b></td></tr>
</table>

8.2.2.5 Watchdog reset

Watchdog reset	
Address or address range	0x1044
Data length in words	1
Access methods	Write
Permitted Modbus functions	6, 16
Default value	-
Description	Writing the value 0xC1 to this parameter resets a timed-out watchdog back to 0.

8.2.3 Product data

8.2.3.1 Serial number

<table>
<tr><th colspan="2">Serial number</th></tr>
<tr><td>Address or address range</td><td>0x1080 - 0x1082</td></tr>
<tr><td>Data length in words</td><td>3</td></tr>
<tr><td>Access methods</td><td>Read</td></tr>
<tr><td>Permitted Modbus functions</td><td>3, 4, 23</td></tr>
<tr><td>Default value</td><td>-</td></tr>
<tr><td>Description</td><td>This parameter can be used to read the serial number of the bus controller.
The decimal serial number is subdivided into 3 groups of four numbers and transferred in 3 words. The serial number already contains the hardware ID. This is different from the I/O module data where the hardware/module ID and serial number are handled separately.
This can also be read as the Product code.
<b>Example</b>:
Serial number: 0882.8016.8593

Transmission methods:
<table>
<tr><th>Word 1</th><th>Word 2</th><th>Word 3</th></tr>
<tr><td>0x1080</td><td>0x1081</td><td>0x1082</td></tr>
<tr><td>0882</td><td>8016</td><td>8593</td></tr>
</table>
Composition on the client side:
Serial number = (Word 1 * 1E+8) + (Word 2 * 1E+4) + Word 3 = 88280168593</td></tr>
</table>

8.2.3.2 Product code

Product code	
Address or address range	0x1083
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	This parameter can be used to query the hardware ID (B&R product code).

8.2.3.3 Hardware major revision

Hardware major revision	
Address or address range	0x1084
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Hardware major revision (number before the decimal point, e.g. V1.02 → 1) The hardware revision provides information about the hardware generation and, like the firmware version, is associated with the revision information (e.g. C0) printed on the bus controller.

8.2.3.4 Hardware minor revision

Hardware minor revision	
Address or address range	0x1085
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Hardware minor revision (number after the decimal point, e.g. V1.02 → 2)

8.2.3.5 Active firmware major revision

Active firmware major revision	
Address or address range	0x1086
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Active firmware major revision (number before the decimal point, e.g. v1.24 → 1)

8.2.3.6 Active firmware minor revision

Active firmware minor revision	
Address or address range	0x1087
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Active firmware minor revision (number after the decimal point, e.g. v1.24 → 24)

8.2.3.7 FPGA hardware revision

FPGA hardware revision	
Address or address range	0x1088
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	FPGA hardware revision Specifies the hardware revision of the installed FPGA chip.

8.2.3.8 Active boot block

Active boot block	
Address or address range	0x1089
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	This parameter can be used to determine the flash memory block from which the firmware or FPGA software was loaded.

Flash block	Explanation
0	Default firmware
1	Update firmware

8.2.3.9 Default firmware major revision

Default firmware major revision	
Address or address range	0x108A
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Default firmware major revision

8.2.3.10 Default firmware minor revision

Default firmware minor revision	
Address or address range	0x108B
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Default firmware minor revision

8.2.3.11 Update firmware major revision

Update firmware major revision	
Address or address range	0x108C
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Update firmware major revision

8.2.3.12 Update firmware minor revision

Update Firmware minor revision	
Address or address range	0x108D
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Update firmware minor revision

8.2.3.13 Default FPGA software revision

Default FPGA software revision	
Address or address range	0x108E
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Factory default FPGA software revision (default block, see "Active boot block" on page 46)

8.2.3.14 Update FPGA software revision

Update FPGA software revision	
Address or address range	0x108F
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	FPGA software revision of the update block (see "Active boot block" on page 46)

8.2.4 Modbus protocol statistics

8.2.4.1 Number of client connections

Number of client connections	
Address or address range	0x10C0
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	This parameter can be used to determine the current number of TCP connections. A maximum of 16 connections can be established simultaneously.

8.2.4.2 Global telegram counter

Global telegram counter	
Address or address range	0x10C1 - 0x10C2
Data length in words	2
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	These parameters can be used to read the sum of the telegrams from all connections since the controller was last restarted. The value is transferred as a 32-bit integer (big-endian). It is also possible to write to these registers in order to reset the counter.

8.2.4.3 Local telegram counter

Local telegram counter	
Address or address range	0x10C3 - 0x10C4
Data length in words	2
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	These parameters can be used to read the number of telegrams from the current connection since the controller was last restarted. The value is transferred as a 32-bit integer (big-endian). It is also possible to write to these parameters in order to reset the counter.

8.2.4.4 Global protocol error counter

Global protocol error counter	
Address or address range	0x10C5 - 0x10C6
Data length in words	2
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	These parameters can be used to read the sum of the telegram errors from all connections since the controller was last restarted. The value is transferred as a 32-bit integer (big-endian). It is also possible to write to these parameters in order to reset the counter.

8.2.4.5 Local protocol error counter

Local protocol error counter	
Address or address range	0x10C7 - 0x10C8
Data length in words	2
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	These parameters can be used to read the number of telegram errors from the current connection since the controller was last restarted. The value is transferred as a 32-bit integer (big-endian). It is also possible to write to these parameters in order to reset the counter.

8.2.4.6 Global maximum command execution time in µs

Global maximum command execution time in µs	
Address or address range	0x10C9 - 0x10CA
Data length in words	2
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	These parameters can be used to read the maximum command execution time of all connections since the controller was last restarted. The value is measured in microseconds [µs] and transferred as a 32-bit integer (big-endian). It is also possible to write to these parameters in order to reset the value.

8.2.4.7 Local maximum command execution time in µs

Local maximum command execution time in µs	
Address or address range	0x10CB - 0x10CC
Data length in words	2
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	These parameters can be used to read the maximum command execution time of the current connections since the controller was last restarted. The value is measured in microseconds [µs] and transferred as a 32-bit integer (big-endian). It is also possible to write to these parameters in order to reset the value.

8.2.4.8 Global minimum command execution time in µs

Global minimum command execution time in µs	
Address or address range	0x10CD - 0x10CE
Data length in words	2
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	These parameters can be used to read the minimum command execution time of all connections since the controller was last restarted. The value is measured in microseconds [µs] and transferred as a 32-bit integer (big-endian). It is also possible to write to these parameters in order to reset the value.

8.2.4.9 Local minimum command execution time in µs

Local minimum command execution time in µs	
Address or address range	0x10CF - 0x10D0
Data length in words	2
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	These parameters can be used to read the minimum command execution time of the current connections since the controller was last restarted. The value is measured in microseconds [µs] and transferred as a 32-bit integer (big-endian). It is also possible to write to these parameters in order to reset the value.

8.2.4.10 Global protocol fragment counter

Global protocol fragment counter	
Address or address range	0x10D1 - 0x10D2
Data length in words	2
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	These parameters can be used to read the number of fragmented records from all existing connections. The value is transferred as a 32-bit integer. It is also possible to write to these parameters in order to reset the value.

8.2.4.11 Local protocol fragment counter

Local protocol fragment counter	
Address or address range	0x10D3 - 0x10D4
Data length in words	2
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	These parameters can be used to read the number of fragmented records from the current connection. The value is transferred as a 32-bit integer. It is also possible to write to these parameters in order to reset the value.

8.2.5 Process image data

8.2.5.1 Number of modules

Number of modules	
Address or address range	0x1100
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Number of successfully started I/O modules

8.2.5.2 Number of analog input registers

Number of analog input registers	
Address or address range	0x1101
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Number of analog input registers

8.2.5.3 Size of the analog input registers in bytes

Size of the analog input registers in bytes	
Address or address range	0x1102
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Size of the analog input registers in bytes

8.2.5.4 Number of analog output registers

Number of analog output registers	
Address or address range	0x1103
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Number of analog output registers

8.2.5.5 Size of the analog output registers in bytes

Size of the analog output registers in bytes	
Address or address range	0x1104
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Size of the analog output registers in bytes

8.2.5.6 Number of digital input registers

Number of digital input registers	
Address or address range	0x1105
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Number of digital input registers

8.2.5.7 Size of the digital input registers in bytes

Size of the digital input registers in bytes	
Address or address range	0x1106
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Size of the digital input registers in bytes

8.2.5.8 Number of digital output registers

Number of digital output registers	
Address or address range	0x1107
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Number of digital output registers

8.2.5.9 Size of the digital output registers in bytes

Size of the digital output registers in bytes	
Address or address range	0x1108
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Size of the digital output registers in bytes

8.2.5.10 Number of analog and digital output status registers

Number of analog and digital output status registers	
Address or address range	0x1109
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Number of analog and digital output status registers

8.2.5.11 Size of the analog and digital output status registers in bytes

Size of the analog and digital output status registers in bytes	
Address or address range	0x110A
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Size of the analog and digital output status registers in bytes

8.2.5.12 Number of X2X Link network status registers

Number of X2X Link network status registers	
Address or address range	0x110B
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Number of X2X Link network status registers (see "X2X Link network status" on page 64)

8.2.5.13 Size of the X2X Link network status registers in bytes

Size of the X2X Link network status registers in bytes	
Address or address range	0x110C
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Size of the X2X Link network status registers in bytes

8.2.6 Controller

8.2.6.1 Save all system data to flash memory

Save all system data to flash memory	
Address or address range	0x1140
Data length in words	1
Access methods	Write
Permitted Modbus functions	6
Default value	-
Description	Writing the constant 0xC1 to this address saves all current system data to flash memory.

8.2.6.2 Read all system data from flash memory

Read all system data from flash memory	
Address or address range	0x1141
Data length in words	1
Access methods	Write
Permitted Modbus functions	6
Default value	-
Description	Writing the constant 0xC1 to this address reads all system data from flash memory. The system is **not** reinitialized in this process! Temporary configuration data in RAM is lost.

8.2.6.3 Delete entire flash memory

Delete entire flash memory	
Address or address range	0x1142
Data length in words	1
Access methods	Write
Permitted Modbus functions	6
Default value	-
Description	Writing the constant 0xC1 to this address deletes all of the data in flash memory. When the system is restarted, the system parameters are automatically initialized with their factory default values.

8.2.6.4 Restart system

Restart system	
Address or address range	0x1143
Data length in words	1
Access methods	Write
Permitted Modbus functions	6
Default value	-
Description	This parameter can be used to restart the system. The following boot modes are available:

Constant	Boot mode
0xC0	Reboots with current flash data. Any changes that have not been saved to flash memory will be lost.
0xC1	Reboots with current temporary configuration data
0xC2	Reboots with factory default values
0xC3	Reboots with current flash data and loads new firmware from flash memory to RAM

8.2.6.5 Close all TCP connections

Close all TCP connections	
Address or address range	0x1144
Data length in words	1
Access methods	Write
Permitted Modbus functions	6
Default value	-
Description	Writing the constant 0xC1 to this address closes all client connections.

8.2.6.6 Initialize module configuration header data

<table>
<tr><th colspan="2">Initialize module configuration header data</th></tr>
<tr><td>Address or address range</td><td>0x1145</td></tr>
<tr><td>Data length in words</td><td>1</td></tr>
<tr><td>Access methods</td><td>Write</td></tr>
<tr><td>Permitted Modbus functions</td><td>6</td></tr>
<tr><td>Default value</td><td>-</td></tr>
<tr><td>Description</td><td>Initialization values of the 4 parameters in the configuration header structure when using the constants 0xC0 and 0xC1:
<table>
<tr><th colspan="2">Initialization value for constant</th><th rowspan="2">Header structure</th></tr>
<tr><th>0xC0</th><th>0xC1</th></tr>
<tr><td>0</td><td>Value of the respective slot index</td><td>Module configuration data index</td></tr>
<tr><td>0</td><td>Value of the respective slot index</td><td>Module configuration data length</td></tr>
<tr><td>0</td><td>0</td><td>Required module hardware ID</td></tr>
<tr><td>0</td><td>254 (standard function model)</td><td>Module start mode</td></tr>
</table>
<b>This functionality is only for test purposes. It may cause an INVALID_CONFIG_DATA error depending on the currently connected I/O modules!</b>
Data is only initialized temporarily for the time being.
<b>Data is saved to remanent memory only after the Save all system data to flash memory command is executed.</b></td></tr>
</table>

8.2.6.7 Initialize module configuration data

<table>
<tr><th colspan="2">Initialize module configuration data</th></tr>
<tr><td>Address or address range</td><td>0x1146</td></tr>
<tr><td>Data length in words</td><td>1</td></tr>
<tr><td>Access methods</td><td>Write</td></tr>
<tr><td>Permitted Modbus functions</td><td>6</td></tr>
<tr><td>Default value</td><td>-</td></tr>
<tr><td>Description</td><td>
<table>
<tr><th>Constant</th><th>Initialization mode</th></tr>
<tr><td>0xC0</td><td>Writes 0 to the module configuration data</td></tr>
<tr><td>0xC1</td><td>Writes the respective slot index to the module configuration data (used for test purposes only)</td></tr>
</table>
Module configuration data is stored in address range 0xC000 to 0xDFFF. Data is only initialized temporarily for the time being.
<b>Data is saved to remanent memory only after the Save all system data to flash memory command is executed.</b></td></tr>
</table>

8.2.6.8 Initialize user data

<table>
<tr><th colspan="2">Initialize user data</th></tr>
<tr><td>Address or address range</td><td>0x1147</td></tr>
<tr><td>Data length in words</td><td>1</td></tr>
<tr><td>Access methods</td><td>Write</td></tr>
<tr><td>Permitted Modbus functions</td><td>6</td></tr>
<tr><td>Default value</td><td>-</td></tr>
<tr><td>Description</td><td>
<table>
<tr><th>Constant</th><th>Initialization mode</th></tr>
<tr><td>0xC0</td><td>Writes 0 to the user data block</td></tr>
<tr><td>0xC1</td><td>Writes a sequential ID to the user data block</td></tr>
</table>
Data is only initialized temporarily for the time being.
This function also overwrites the checksum of the configuration data!
<b>Data is saved to remanent memory only after the Save all system data to flash memory command is executed.</b></td></tr>
</table>

8.2.7 Miscellaneous

8.2.7.1 Reading network address switches

Reading network address switches	
Address or address range	0x1180
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	This parameter can be read to determine the network address switch value.

8.2.7.2 Module initialization delay in ms

Module initialization delay in ms	
Address or address range	0x1181
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	3000
Description	This parameter can be used to configure or read the module initialization delay. This delay is specified in [ms]. After a restart, the system enters a module initialization phase where all client queries are answered with the Modbus error "Slave device busy". 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. **Data is saved to remanent memory only after the Save all system data to flash memory command is executed.**

8.2.7.3 Verification mode for I/O access limits

<table>
<tr><th colspan="2">Verification mode for I/O access limits</th></tr>
<tr><td>Address or address range</td><td>0x1182</td></tr>
<tr><td>Data length in words</td><td>1</td></tr>
<tr><td>Access methods</td><td>Read/Write</td></tr>
<tr><td>Permitted Modbus functions</td><td>3, 4, 6, 16, 23</td></tr>
<tr><td>Default value</td><td>0xC0 (limits not checked)</td></tr>
<tr><td>Description</td><td>The amount of data that can be input/output is determined by the number of connected I/O modules and their I/O data points, i.e. the input and output address limits are defined by the number of I/O data points.
This parameter is used to set whether these limits are checked.
<table>
<tr><th>Constant</th><th>Description</th></tr>
<tr><td>0xC0</td><td>Limits not checked</td></tr>
<tr><td>0xC1</td><td>Limits checked</td></tr>
</table>
If checking is enabled and reading/writing takes place that extends beyond the physically existing module data, then the controller will abort the procedure with the error Illegal data address.
If checking is not enabled, reading/writing beyond the physical module data is managed as follows:
• Read: Missing data is filled with zeros.
• Write: Excess data is ignored.
<b>Data is saved to remanent memory only after the Save all system data to flash memory command is executed.</b></td></tr>
</table>

8.2.7.4 Enable/Disable Telnet password

<table>
<tr><th colspan="2">Enable/Disable Telnet password</th></tr>
<tr><td>Address or address range</td><td>0x1183</td></tr>
<tr><td>Data length in words</td><td>1</td></tr>
<tr><td>Access methods</td><td>Read/Write</td></tr>
<tr><td>Permitted Modbus functions</td><td>3, 4, 6, 16, 23</td></tr>
<tr><td>Default value</td><td>0xC0 (password disabled)</td></tr>
<tr><td>Description</td><td>This parameter can be used to enable or disable the Telnet password.
<table>
<tr><th>Constant</th><th>Description</th></tr>
<tr><td>0xC0</td><td>Password disabled</td></tr>
<tr><td>0xC1</td><td>Password enabled</td></tr>
</table>
<b>Data is saved to remanent memory only after the Save all system data to flash memory command is executed.</b></td></tr>
</table>

8.2.7.5 Modified configuration flag

<table>
<tr><th colspan="2">Modified configuration flag</th></tr>
<tr><td>Address or address range</td><td>0x1184</td></tr>
<tr><td>Data length in words</td><td>1</td></tr>
<tr><td>Access methods</td><td>Read/Write</td></tr>
<tr><td>Permitted Modbus functions</td><td>3, 4, 6, 16, 23</td></tr>
<tr><td>Default value</td><td>0xC0 (data not modified)</td></tr>
<tr><td>Description</td><td>This flag is automatically set to the value 0xC1 whenever system data is modified. This provides a way for the user to check for unintended data modifications. This flag is also stored along with the other system data in flash memory. The user can delete or set this flag by writing the constant 0xC0 or 0xC1, respectively.
<table>
<tr><th>Constant</th><th>Description</th></tr>
<tr><td>0xC0</td><td>Data not modified</td></tr>
<tr><td>0xC1</td><td>Data modification found</td></tr>
</table>
<b>Data is saved to remanent memory only after the Save all system data to flash memory command is executed.</b></td></tr>
</table>

8.2.7.6 Default configuration flag

<table>
<tr><th colspan="2">Default configuration flag</th></tr>
<tr><td>Address or address range</td><td>0x1185</td></tr>
<tr><td>Data length in words</td><td>1</td></tr>
<tr><td>Access methods</td><td>Read</td></tr>
<tr><td>Permitted Modbus functions</td><td>3, 4, 23</td></tr>
<tr><td>Default value</td><td>0xC1 (all system parameters correspond to their default values.)</td></tr>
<tr><td>Description</td><td>This parameter can be used to check whether the bus controller has already been configured. It is only possible for the user to read this flag.
If the bus controller starts up with default values, the flag receives the value 0xC1. This flag is automatically set to the value 0xC0 if system parameters are modified.
A restart with the constant "0xC2" is needed to reset all parameters to their default values (see "Restart system" on page 52).
Write access to the "Modified configuration flag" (0x1184) also results in a change to 0xC0.
<table>
<tr><th>Constant</th><th>Description</th></tr>
<tr><td>0xC0</td><td>The bus controller has already been configured.</td></tr>
<tr><td>0xC1</td><td>All system parameters correspond to their default values.</td></tr>
</table></td></tr>
</table>

8.2.7.7 Bus controller operating status (error-free state)

<table>
<tr><th colspan="2">Bus controller operating status (error-free state)</th></tr>
<tr><td>Address or address range</td><td>0x1186</td></tr>
<tr><td>Data length in words</td><td>1</td></tr>
<tr><td>Access methods</td><td>Read</td></tr>
<tr><td>Permitted Modbus functions</td><td>3, 4, 23</td></tr>
<tr><td>Default value</td><td>-</td></tr>
<tr><td>Description</td><td>Bus controller operating state
<table>
<tr><th>Bit</th><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>0x0001</td><td>Bus controller no longer in its default state, i.e. settings and configurations have already been made</td></tr>
<tr><td>1</td><td>0x0002</td><td>At least one master connection exists</td></tr>
<tr><td>2</td><td>0x0004</td><td>System boot or I/O module initialization active</td></tr>
<tr><td>3</td><td>0x0008</td><td>Bus controller waiting for an IP address from the DHCP server</td></tr>
</table></td></tr>
</table>

8.2.7.8 Bus controller error status (error state)

<table>
<tr><th colspan="2">Bus controller error status (error state)</th></tr>
<tr><td>Address or address range</td><td>0x1187</td></tr>
<tr><td>Data length in words</td><td>1</td></tr>
<tr><td>Access methods</td><td>Read</td></tr>
<tr><td>Permitted Modbus functions</td><td>3, 4, 23</td></tr>
<tr><td>Default value</td><td>0 (no error)</td></tr>
<tr><td>Description</td><td>Error status of the bus controller
<table>
<tr><th>Bit</th><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>0x0001</td><td>Watchdog timeout</td></tr>
<tr><td>1</td><td>0x0002</td><td>Flash memory read error</td></tr>
<tr><td>2</td><td>0x0004</td><td>Faulty or missing module detected during runtime</td></tr>
<tr><td>3</td><td>0x0008</td><td>Missing module detected during boot phase</td></tr>
<tr><td>4</td><td>0x0010</td><td>Incorrect module detected during boot phase</td></tr>
<tr><td>5</td><td>0x0020</td><td>Faulty I/O module configuration data</td></tr>
<tr><td>6</td><td>0x0040</td><td>IP address conflict. An IP address conflict is only detected during the bus controller's startup phase.</td></tr>
</table></td></tr>
</table>

8.2.7.9 I/O module configuration mode

<table>
<tr><th colspan="2">I/O module configuration mode</th></tr>
<tr><td>Address or address range</td><td>0x1188</td></tr>
<tr><td>Data length in words</td><td>1</td></tr>
<tr><td>Access methods</td><td>Read/Write</td></tr>
<tr><td>Permitted Modbus functions</td><td>3, 4, 6, 16, 23</td></tr>
<tr><td>Default value</td><td>0xC0</td></tr>
<tr><td>Description</td><td>
<table>
<tr><th>Constant</th><th>Description</th></tr>
<tr><td>0xC0</td><td>The I/O module configuration consists of the specified configuration data and the additional data provided by the I/O module.
This makes it possible to configure individual I/O module registers. It is also possible to implement a "wildcard" configuration in this mode (see "I/O module register configuration" on page 70).</td></tr>
<tr><td>0xC1</td><td>The I/O module configuration only uses the configuration data provided by the user. The hardware ID of physically present I/O modules must match the specified I/O module hardware IDs. A "wildcard" insertion is not possible within a configured I/O module group.</td></tr>
</table>
Note about 0xC1 mode:
Data is not exchanged between the bus controller and I/O modules if configuration data is missing. No cyclic registers are provided.
It is possible to combine a grouped and completely configured I/O module group together with a number of non-configured I/O modules. In this case, the non-configured I/O modules are booted with default data. The grouped, configured I/O module group must start with the first I/O module (slot index 0) (see "Auto mode" on page 35).
A special case would be a configured I/O module group of size zero, i.e. all connected modules are automatically booted with default settings.
Data is saved to remanent memory only after the Save all system data to flash memory command is executed.
</td></tr>
</table>

8.2.7.10 Bus controller Error/Status LED signal mask

<table>
<tr><th colspan="2">Bus controller Error/Status LED signal mask</th></tr>
<tr><td>Address or address range</td><td>0x1189</td></tr>
<tr><td>Data length in words</td><td>1</td></tr>
<tr><td>Access methods</td><td>Read/Write</td></tr>
<tr><td>Permitted Modbus functions</td><td>3, 4, 6, 16, 23</td></tr>
<tr><td>Default value</td><td>0xFFFF</td></tr>
<tr><td>Description</td><td>
This parameter allows the user to control the behavior of LED "Error".
LED "Error" can be checked by turning the corresponding bits on and off. In the default state, all errors are indicated accordingly.
Bus controller status should be indicated by LED: Respective bit set to 1
Bus controller status should not be indicated by LED: Respective bit set to 0
The following Error LED states can be controlled.
<table>
<tr><th>Bit</th><th>Controllable</th><th>Description</th></tr>
<tr><td>0</td><td>No</td><td>Watchdog timeout</td></tr>
<tr><td>1</td><td>No</td><td>Flash memory read error</td></tr>
<tr><td>2</td><td>Yes</td><td>Faulty or missing module detected during runtime</td></tr>
<tr><td>3</td><td>Yes</td><td>Missing module detected during boot phase</td></tr>
<tr><td>4</td><td>No</td><td>Incorrect module detected during boot phase</td></tr>
<tr><td>5</td><td>No</td><td>Faulty I/O module configuration data</td></tr>
<tr><td>6</td><td>No</td><td>IP address conflict</td></tr>
</table>
Data is saved to remanent memory only after the Save all system data to flash memory command is executed.
</td></tr>
</table>

8.2.7.11 Process data byte order

<table>
<tr><th colspan="2">Process data byte order</th></tr>
<tr><td>Address or address range</td><td>0x118A</td></tr>
<tr><td>Data length in words</td><td>1</td></tr>
<tr><td>Access methods</td><td>Read/Write</td></tr>
<tr><td>Permitted Modbus functions</td><td>3, 4, 6, 16, 23</td></tr>
<tr><td>Default value</td><td>0x0000</td></tr>
<tr><td>Description</td><td>
In line with the Modbus specification, big-endian format is used by default for communication. This Modbus function can be used to change the byte order of I/O process data. When the bit is set, the byte order of the corresponding Modbus address range is reversed.
This function is available in firmware version 1.39 or later.
<table>
<tr><th>Bit</th><th>Frame</th><th>Address range</th><th>Description</th></tr>
<tr><td>0</td><td>AI</td><td>0x0000 - 0x07FF</td><td>Analog input</td></tr>
<tr><td>1</td><td>DI</td><td>0x2000 - 0x23FF</td><td>Digital input</td></tr>
<tr><td>2</td><td>NS</td><td>0x2800 - 0x29FF</td><td>X2X Link network status (input)</td></tr>
<tr><td>3</td><td>OS</td><td>0x2A00 - 0x2BFF</td><td>Analog or digital output status (input)</td></tr>
<tr><td>4 - 7</td><td></td><td></td><td>Reserved</td></tr>
<tr><td>8</td><td>AO</td><td>0x0800 - 0x0FFF</td><td>Analog output</td></tr>
<tr><td>9</td><td>DO</td><td>0x2400 - 0x27FF</td><td>Digital output</td></tr>
</table>
</td></tr>
</table>

8.2.8 X2X Link statistics

8.2.8.1 X2X Link cycle counter

X2X Link cycle counter	
Address or address range	0x11C0
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	This cycle counter is incremented after each completed X2X Link I/O cycle.

8.2.8.2 Number of X2X Link off cycles

Number of X2X Link off cycles	
Address or address range	0x11C1
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	This counter is incremented if the system is restarted in order to restart X2X Link.

8.2.8.3 Cyclic errors

Cyclic errors	
Address or address range	0x11C2
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	This counter is incremented each time an error occurs in the cyclic part of X2X Link communication.

8.2.8.4 Cyclic: Bus timing errors

Cyclic: Bus timing errors	
Address or address range	0x11C3
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of frames that could not be sent because the X2X Link transmitter was not ready

8.2.8.5 Cyclic: Frame timing errors

Cyclic: Frame timing errors	
Address or address range	0x11C4
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of expected response frames that could not be received because of timeouts

8.2.8.6 Cyclic: Frame checksum errors

Cyclic: Frame checksum errors	
Address or address range	0x11C5
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of frames received with a checksum error

8.2.8.7 Cyclic: Frame pending errors

Cyclic: Frame pending errors	
Address or address range	0x11C6
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of frames that could not be sent because the input frame was still active

8.2.8.8 Cyclic: Buffer underrun

Cyclic: Buffer underrun	
Address or address range	0x11C7
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Not used: Only exists to remain compatible with the Modbus standard

8.2.8.9 Cyclic: Buffer overflow

Cyclic: Buffer overflow	
Address or address range	0x11C8
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Not used: Only exists to remain compatible with the Modbus standard

8.2.8.10 Acyclic errors

Acyclic errors	
Address or address range	0x11C9
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	This counter is incremented each time an error occurs in the acyclic part of X2X Link communication.

8.2.8.11 Acyclic: Bus timing errors

Acyclic: Bus timing errors	
Address or address range	0x11CA
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of frames that could not be sent because the X2X Link transmitter was not ready

8.2.8.12 Acyclic: Frame timing errors

Acyclic: Frame timing errors	
Address or address range	0x11CB
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of expected response frames that could not be received because of timeouts

8.2.8.13 Acyclic: Frame checksum errors

Acyclic: Frame checksum errors	
Address or address range	0x11CC
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of frames received with a checksum error

8.2.8.14 Acyclic: Frame pending errors

Acyclic: Frame pending errors	
Address or address range	0x11CD
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of frames that could not be sent because the input frame was still active

8.2.8.15 Acyclic: Buffer underrun

Acyclic: Buffer underrun	
Address or address range	0x11CE
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Not used: Only exists to remain compatible with the Modbus standard

8.2.8.16 Acyclic: Buffer overflow

Acyclic: Buffer overflow	
Address or address range	0x11CF
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Not used: Only exists to remain compatible with the Modbus standard

8.2.9 Network statistics

8.2.9.1 IF1: Ethernet frames received

IF1: Ethernet frames received	
Address or address range	0x1200
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of Ethernet frames received on IF1

8.2.9.2 IF1: Frames lost due to high load

IF1: Frames lost (performance problem)	
Address or address range	0x1201
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of frames discarded by the bus controller's integrated switch due to high load

8.2.9.3 IF1: Oversized frames

IF1: Oversized frames	
Address or address range	0x1202
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of oversized frames received

8.2.9.4 IF1: CRC error

IF1: CRC error	
Address or address range	0x1203
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of frames detected with CRC errors (disruptions)

8.2.9.5 IF1: Frames lost

IF1: Frames lost	
Address or address range	0x1204
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Internal error

8.2.9.6 IF1: Frames lost due to high load

IF1: Frames lost (performance problem)	
Address or address range	0x1205
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of frames discarded by the bus controller due to high load

8.2.9.7 IF1: Collisions

IF1: Collisions	
Address or address range	0x1206
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of collisions. Can only occur in half-duplex mode, e.g. when using hubs.

8.2.9.8 IF1: Frames lost due to switch overflow

IF1: Frames lost due to switch overflow	
Address or address range	0x1207
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of frames lost due to a switch overload

8.2.9.9 IF1: Frames lost due to switch errors

IF1: Frames lost due to switch errors	
Address or address range	0x1208
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of frames lost due to internal errors in the switch

8.2.9.10 IF2: Ethernet frames received

IF2: Ethernet frames received	
Address or address range	0x1210
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of Ethernet frames received on IF2

8.2.9.11 IF2: Frames lost due to high load

IF2: Frames lost (performance problem)	
Address or address range	0x1211
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of frames discarded by the bus controller's integrated switch due to high load

8.2.9.12 IF2: Oversized frames

IF2: Oversized frames	
Address or address range	0x1212
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of oversized frames received

8.2.9.13 IF2: CRC error

IF2: CRC error	
Address or address range	0x1213
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of frames detected with CRC errors (disruptions)

8.2.9.14 IF2: Frames lost

IF2: Frames lost	
Address or address range	0x1214
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Internal error

8.2.9.15 IF2: Frames lost due to high load

IF2: Frames lost (performance problem)	
Address or address range	0x1215
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of frames discarded by the bus controller due to high load

8.2.9.16 IF2: Collisions

IF2: Collisions	
Address or address range	0x1216
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of collisions. Can only occur in half-duplex mode, e.g. when using hubs.

8.2.9.17 IF2: Frames lost due to switch overflow

IF2: Frames lost due to switch overflow	
Address or address range	0x1217
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of frames lost due to a switch overload

8.2.9.18 IF2: Frames lost due to switch errors

IF2: Frames lost due to switch errors	
Address or address range	0x1218
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of frames lost due to internal errors in the switch

8.2.10 User data

See also "Initialize user data" on page 53.

8.2.10.1 Configuration data checksum

Configuration data checksum	
Address or address range	0x1240 - 0x1241
Data length in words	2
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	0x00000000
Description	These 4 bytes are used to store a checksum for the configuration data. This checksum is calculated with the configuration data from Automation Studio. In the event of a restart, the application on the master or a configuration tool can be used to check if the configuration on the bus controller is current or if it necessary to transfer new register data.

8.2.10.2 User data block

User data block	
Address or address range	0x1242 - 0x127F
Data length in words	1 - 62
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	0
Description	This data block (size: 62 words, or 124 bytes) can be used by the user for private data. **Data is saved to remanent memory only after the Save all system data to flash memory command is executed.**

8.2.11 Acyclic I/O register configuration

8.2.11.1 Write to acyclic I/O registers

<table>
<tr><th colspan="2">Write to acyclic I/O registers</th></tr>
<tr><td>Address or address range</td><td>0x1280 - 0x1283</td></tr>
<tr><td>Data length in words</td><td>4</td></tr>
<tr><td>Access methods</td><td>Read/Write</td></tr>
<tr><td>Permitted Modbus functions</td><td>3, 4, 16</td></tr>
<tr><td>Default value</td><td>-</td></tr>
<tr><td>Description</td><td>These 4 Modbus parameters can be used to write to acyclic I/O registers. This can be done to change I/O module configurations during runtime, for example.
<table>
<tr><th>Modbus object address</th><th>Function</th></tr>
<tr><td>0x1280</td><td>Slot index of the I/O module (X2X Link network address switch value minus 1)</td></tr>
<tr><td>0x1281</td><td>I/O register address</td></tr>
<tr><td>0x1282</td><td>I/O register value high word</td></tr>
<tr><td>0x1283</td><td>I/O register value low word</td></tr>
</table>
Write access can only take place using Modbus command FC16: Write multiple registers. The number of Modbus parameters to be written must be 4.
<b>If module registers that are subject to cyclic data exchange between the bus controller and the I/O module are written to acyclically in this manner, then they will be overwritten with cyclic data again in the next X2X Link cycle.</b></td></tr>
</table>

8.2.11.2 Read from acyclic I/O registers

<table>
<tr><th colspan="2">Read from acyclic I/O registers</th></tr>
<tr><td>Address or address range</td><td>0x1284 - 0x1285</td></tr>
<tr><td>Data length in words</td><td>2</td></tr>
<tr><td>Access methods</td><td>Read/Write</td></tr>
<tr><td>Permitted Modbus functions</td><td>3, 4, 23</td></tr>
<tr><td>Default value</td><td>-</td></tr>
<tr><td>Description</td><td>These 2 Modbus parameters can be used to access acyclic I/O registers to perform a read operation.
The result of this read operation is available at Modbus addresses 0x1286 and 0x1287.
<table>
<tr><th>Modbus object address</th><th>Function</th></tr>
<tr><td>0x1284</td><td>Slot index of the I/O module (X2X Link network address switch value minus 1)</td></tr>
<tr><td>0x1285</td><td>I/O register address</td></tr>
</table>
Write access can only take place using Modbus command FC23: Read/Write multiple registers. The number of Modbus parameters to be written must be 2.
This combined read/write command ensures data consistency between the read access (write procedure to 0x1284 and 0x1285) and the result (read procedure from 0x1286 and 0x1287).</td></tr>
</table>

8.2.11.3 Result of the I/O register read operation

<table>
<tr><th colspan="2">Result of the I/O register read operation</th></tr>
<tr><td>Address or address range</td><td>0x1286 - 0x1287</td></tr>
<tr><td>Data length in words</td><td>2</td></tr>
<tr><td>Access methods</td><td>Read</td></tr>
<tr><td>Permitted Modbus functions</td><td>3, 4, 23</td></tr>
<tr><td>Default value</td><td>-</td></tr>
<tr><td>Description</td><td>These two Modbus addresses contain the result of the current I/O register read procedure (see "Read from acyclic I/O registers" on page 62).
<table>
<tr><th>Modbus object address</th><th>Function</th></tr>
<tr><td>0x1286</td><td>I/O register value high word</td></tr>
<tr><td>0x1287</td><td>I/O register value low word</td></tr>
</table>
<b>Because I/O register communication involves acyclic write and read operations and the Modbus server can be operated simultaneously by several client devices, data consistency between the I/O register access and the result is only ensured by executing Modbus command FC23: Read/Write multiple registers.</b></td></tr>
</table>

9 X2X Link network status

9.1 General information

The X2X Link network status provides information about the operating state of individual X2X Link stations. These are the bus modules for the respective I/O modules. The operating state of the I/O modules themselves can be queried using module-specific parameters (see "Operating status" on page 73).

Each X2X Link bus module occupies 1 byte of data.
The Modbus TCP bus controller can access up to 253 X2X Link modules (index 0X00 to 0xFC).
The following assignments are derived from this for the addresses of the X2X Link network status (addresses 0x2800 to 0x29FF):

Address	X2X Link network address switch value
0x2800	Module 1 and 2 (slot index 0 and 1)
0x2801	Module 3 and 4
0x2802	Module 5 and 6
0x2803	Module 7 and 8
...	...
0x287D	Module 251 and 252 (slot index 0xFA and 0xFB)
0x287E	Module 253 (slot index 0xFC)
0x287F	Reserved
...	...
0x29FF	Reserved

The network status of the first module is stored in the higher-value byte; the lower-value byte contains the status of the second module.

Example:
At address 0x2800, the data 0xAABB (1 word) indicates the following:

- AA: Network status of module 1 (slot index 0)
- BB: Network status of module 2 (slot index 1)

Each X2X Link station is equipped with a hardware component (ASIC) that reports its status to the X2X Link master – in this case, the bus controller – during every 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 electronic module OK (required for bits 3 to 7 to be valid)
3	0x08	I/O data invalid
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:

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

10 Module-specific parameters

10.1 Module parameter overview

10.1.1 Module-oriented access

This access method makes it possible to read all available parameters sequentially from an individual I/O module in addition to writing module configuration data.
The parameters 0 to D are supported here. Accessing E and F returns only null bytes in response.
A module is accessed using the middle two hexadecimal digits of the address (marked in red). The Modbus TCP bus controller can access up to 253 I/O modules (index 0x00 to 0xFC). Access outside of this range, i.e. from 0xAFD0 to 0xAFFF, is reserved.

> **Information:**
>
> **The module with slot index 0 corresponds to the power supply (e.g. power supply module X20PS9400). For details, see "Configuration of the I/O modules" on page 31.**

Module parameters are accessed using the digits with the lowest values (marked in blue).

0xAMMP:	MM:	Module access	[0x0 to 0xFC or 0 to 252]
	P:	Parameter access	[0x0 to 0xD or 0 to 13]

Address range	Description	Access types	Group
0xA000 - 0xAFC0	Read module status	Read	Module data Module-oriented access
0xA001 - 0xAFC1	Read module product code (hardware ID)	Read	
0xA002 - 0xAFC2	Read module serial number (high word)	Read	
0xA003 - 0xAFC3	Read module serial number (low word)	Read	
0xA004 - 0xAFC4	Read index of analog input data (AI)	Read	
0xA005 - 0xAFC5	Read index of analog output data (AO)	Read	
0xA006 - 0xAFC6	Read index of digital input data (DI)	Read	
0xA007 - 0xAFC7	Read index of digital output data (DO)	Read	
0xA008 - 0xAFC8	Module configuration: Required module hardware ID	Read/Write	
0xA009 - 0xAFC9	Module configuration: Module start mode (function model)	Read/Write	
0xA00A - 0xAFCA	Module configuration: Module configuration data index	Read/Write	
0xA00B - 0xAFCB	Module configuration: Module configuration data length	Read/Write	
0xA00C - 0xAFCC	Read module firmware version	Read	
0xA00D - 0xAFCD	Read module hardware variant	Read	

10.1.2 Parameter-oriented access

This access method makes it possible to read identical module parameters sequentially from some or all I/O modules.

One example of this would be querying the module status of the first 4 modules using command "Read input register" fc4 (starting address 0xB000, number of addresses to be read: 0x4).

A module is accessed using the two digits with the lowest values. The Modbus TCP bus controller can access up to 253 I/O modules (index 0x00 to 0xFC). Access outside of this range is reserved.

> **Information:**
>
> **With X20 bus controllers, the module with index 0 corresponds to the power supply (e.g. power supply module X20PS9400).**

The parameter is accessed with the low-value nibble of the first byte (marked in blue). Only the parameters 0x0 to 0xD are supported. Accessing 0xE and 0xF returns only null bytes in response.

Permitted access types for individual parameters are listed in the descriptions of the respective parameters.

0xBPMM:	P:	Parameter access	[0x0 to 0xD or 0 to 15]
	MM:	Module access	[0x0 to 0xFC or 0 to 252]

Address	Description	Group
0xB000	Read module status of slot index 0	Module data Parameter-oriented Access
0xB001	Read module status of slot index 1	
0xB002	Read module status of slot index 2	
0xB003	Read module status of slot index 3	
...		
0xB100	Read module product code of slot index 0	
0xB101	Read module product code of slot index 1	
0xB102	Read module product code of slot index 2	
0xB103	Read module product code of slot index 3	
...		

10.2 Description of individual module parameters

10.2.1 Module status

Module-specific parameters: Module status	
Address or address range	0xA000 - 0xAFC0: Module-oriented access (e.g. all of a module's parameters) 0xB000 - 0xB0FC: Parameter-oriented access (e.g. one parameter type on all modules)
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Reads the module status of a connected module

Constant	Description
0x00 "0"	No module connected
0x4E "N"	Bus module present but electronics module not starting. Cause: Faulty I/O power supply, or the electronics module is not connected to the bus module.
0x42 "B"	Boot procedure (OS loader test)
0x55 "U"	Boot procedure (uploading IDs)
0x70 or 0x50 "p" / "P"	Preoperational (module ready to start)
0x53 "S"	Synchronization based on the bus controller's time
0x43 "C"	Module being configured
0x52 "R"	Module active and functioning without errors
0x44 "D"	Firmware download active
0xE0	Error: Module without I/O firmware detected
0xE1	Error: Module with invalid firmware detected
0xE2	Error: Module cannot be activated, e.g. configuration error (incorrect function model, etc.)
0xE3	Error: Registers could not be registered, e.g. faulty module configuration data
0xE4	Error: Internal error, I/O module cannot be started
0xE5	Error: Module cannot be started, X2X Link frame too small
0xE6	Module not started, different module type configured for this slot

10.2.2 Module product code (hardware ID)

Module-specific parameters: Module product code (hardware ID)	
Address or address range	0xA001 - 0xAFC1: Module-oriented access (e.g. all of a module's parameters) 0xB100 - 0xB1FC: Parameter-oriented access (e.g. one parameter type on all modules)
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	This parameter can be used to read the hardware ID of a connected module. The hardware ID is the first 4 digits of the module's serial number. This parameter specifies the current ID being used for this slot. This may deviate from the configured ID. To see how the complete serial number is put together, see "Composition of the module serial number" on page 67.

10.2.3 High word of the module serial number

Module-specific parameters: High word of the module serial number	
Address or address range	0xA002 - 0xAFC2: Module-oriented access (e.g. all of a module's parameters) 0xB200 - 0xB2FC: Parameter-oriented access (e.g. one parameter type on all modules)
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	This parameter can be used to read the high word of the serial number. This parameter specifies the serial number currently found on this slot. To see how the complete serial number is put together, see "Composition of the module serial number" on page 67.

10.2.4 Low word of the module serial number

Module-specific parameters: Low word of the module serial number	
Address or address range	0xA003 - 0xAFC3: Module-oriented access (e.g. all of a module's parameters) 0xB300 - 0xB3FC: Parameter-oriented access (e.g. one parameter type on all modules)
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	This parameter can be used to read the low word of the serial number. This parameter specifies the current serial number being used for this slot. To see how the complete serial number is put together, see "Composition of the module serial number" on page 67.

10.2.5 Composition of the module serial number

Every B&R module has a unique serial number. The complete serial number is made up of the module hardware ID, the high word and low word of the serial number as follows:

Serial number = (Hardware ID * 1E+7) + (High word * 1E+4) + Low word

The serial number is printed in decimal form on the module's housing.

Example
Hardware ID = (decimal) 1213
High word of the module's serial number = (decimal) 67
Low word of the module's serial number = (decimal) 1339

Serial number = 1213 * 10000000 + 67 * 10000 + 1339 = 12130671339

10.2.6 Index of analog input data

Module-specific parameters: Index of analog input data	
Address or address range	0xA004 - 0xAFC4: Module-oriented access (e.g. all of a module's parameters) 0xB400 - 0xB4FC: Parameter-oriented access (e.g. one parameter type on all modules)
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Byte index that can be used to access analog input process data The byte index must be converted to a Modbus-specific word index to make data access possible. If the respective module fails to return any analog input data, the query will result in 0xFFFF.

10.2.7 Index of analog output data

Module-specific parameters: Index of analog output data	
Address or address range	0xA005 - 0xAFC5: Module-oriented access (e.g. all of a module's parameters) 0xB500 - 0xB5FC: Parameter-oriented access (e.g. one parameter type on all modules)
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Byte index that can be used to access analog output process data The byte index must be converted to a Modbus-specific word index to make data access possible. If the respective module fails to return any analog output data, the query will result in 0xFFFF.

10.2.8 Index of digital input data

Module-specific parameters: Index of digital input data	
Address or address range	0xA006 - 0xAFC6: Module-oriented access (e.g. all of a module's parameters) 0xB600 - 0xB6FC: Parameter-oriented access (e.g. one parameter type on all modules)
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Byte index that can be used to access digital input process data The byte index must be converted to a Modbus-specific word index to make data access possible. If the respective module fails to return any digital input data, the query will result in 0xFFFF.

10.2.9 Index of the digital output data

Module-specific parameters: Index of digital output data	
Address or address range	0xA007 - 0xAFC7: Module-oriented access (e.g. all of a module's parameters) 0xB700 - 0xB7FC: Parameter-oriented access (e.g. one parameter type on all modules)
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Byte index that can be used to access digital output process data. The byte index must be converted to a Modbus-specific word index to make data access possible. If the respective module fails to return any digital output data, the query will result in 0xFFFF.

10.2.10 Required module hardware ID

Module configuration: Required module hardware ID	
Address or address range	0xA008 - 0xAFC8: Module-oriented access (e.g. all of a module's parameters) 0xB800 - 0xB8FC: Parameter-oriented access (e.g. one parameter type on all modules)
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Specifies which module must be inserted in this slot (hardware ID or module product code). For more information, see "I/O module register configuration" on page 70. The module will not be started if the hardware ID of the actual module is different than the ID specified here. **Exception**: No check takes place if the hardware ID = 0.

10.2.11 Module start mode

Module configuration: Module start mode (9)	
Address or address range	0xA009 - 0xAFC9: Module-oriented access (e.g. all of a module's parameters) 0xB900 - 0xB9FC: Parameter-oriented access (e.g. one parameter type on all modules)
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Specifies the module function model to be used. Some I/O modules support other operating modes in addition to standard function model "254". For more information, see the respective module description.

10.2.12 Module configuration data index

Module configuration: Module configuration data index	
Address or address range	0xA00A - 0xAFCA: Module-oriented access (e.g. all of a module's parameters) 0xBA00 - 0xBAFC: Parameter-oriented access (e.g. one parameter type on all modules)
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	The address range 0xC000 to 0xDFFF can be used to store configuration data for I/O modules that are transferred to the respective module by the bus controller during the boot procedure (see "I/O module register configuration" on page 70). This configuration data can be taken from the description of the corresponding module or it can be created using Automation Studio. Each configuration entry takes up 4 words. The configuration data index specifies the address of the first word.

10.2.13 Module configuration data length

Module configuration: Module configuration data length	
Address or address range	0xA00B - 0xAFCB: Module-oriented access (e.g. all of a module's parameters) 0xBB00 - 0xBBFC: Parameter-oriented access (e.g. one parameter type on all modules)
Data length in words	1
Access methods	Read/Write
Permitted Modbus functions	3, 4, 6, 16, 23
Default value	-
Description	Number of configuration entries for the module. Each entry is equal to 4 words.

10.2.14 Module firmware version

Module-specific parameters: Module firmware version	
Address or address range	0xA00C - 0xAFCC: Module-oriented access (e.g. all of a module's parameters) 0xBC00 - 0xBCFC: Parameter-oriented access (e.g. one parameter type on all modules)
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Firmware version of the I/O module currently in this slot. In contrast to the firmware version of the bus controller, where the version specification is composed of a major and minor entry, I/O modules have only **one** number entry.

10.2.15 Module hardware variant

Module-specific parameters: Module hardware variant	
Address or address range	0xA00D - 0xAFCD: Module-oriented access (e.g. all of a module's parameters) 0xBD00 - 0xBDFC: Parameter-oriented access (e.g. one parameter type on all modules)
Data length in words	1
Access methods	Read
Permitted Modbus functions	3, 4, 23
Default value	-
Description	Hardware variant of the I/O module currently in this slot. In contrast to the hardware revision of the bus controller, where the specification is composed of a major and minor entry, I/O modules have only **one** number entry.

10.3 I/O module register configuration

The address range 0xC000 to 0xDFFF on the bus controller can be used to store the configuration data for up to 2048 I/O module registers. This data is then transferred to the respective modules during booting. If no explicit configuration is specified for an I/O module, then the default configuration will be used.
Both the configuration parameters and default configuration can be taken from the description of the corresponding module or easily created using Automation Studio.
Each configuration entry takes up 4 words. An I/O module can be referenced to one or more consecutive register configurations. The following reference entries in the module-specific parameters can be used for this.

Modules with identical configuration data are permitted to reference the same block in order to save space.

The reference entries are made up of the following data:

<table>
<tr><th>Modbus address
mm stands for slot index</th><th>Explanation</th></tr>
<tr><td>0xAmm8</td><td>Required module product code (hardware ID, parameter 8):
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>Hardware ID of the connected module</td><td>The module is only booted if the specified I/O module hardware ID matches the physical I/O module in this slot.
An error is reported if an I/O module is missing or the hardware ID is different (see "Miscellaneous" on page 53).
To boot subsequent I/O modules, either the configured I/O module must be physically present or the module-specific cyclic registers must have been defined for the missing module in the configuration data.
This is because the bus controller requires information about the I/O data width of each module in order to configure X2X Link. If this information is not available for a module, then none of the modules connected to it will be started.</td></tr>
<tr><td>0xFFFF</td><td>Indicates to the bus controller that the slot is empty. No mapping entries are generated for this slot regardless of whether an actual I/O module is inserted. Subsequent I/O modules are not affected by one or more empty slots.</td></tr>
<tr><td>0x0000</td><td>All I/O modules are accepted and booted with the corresponding configuration data, whether its default or configured data. To boot subsequent I/O modules, an I/O module must be physically present in this slot, or module-specific cyclic registers must have been defined in the configuration data for this slot index.
<b>This type of "wildcard" I/O module configuration is only possible if the I/O module configuration mode parameter is set to the value 0xC0.</b></td></tr>
</table></td></tr>
<tr><td>0xAmm9</td><td>Module start mode (function model)</td></tr>
<tr><td>0xAmmA</td><td>Module configuration data index. Reference to the respective starting address of the configuration block in the address range 0xC000-0xDFFF.</td></tr>
<tr><td>0xAmmB</td><td>Number of register configurations. The number 1 corresponds to <b>one entry</b> (i.e. 4 words).</td></tr>
</table>

A register configuration consists of the following 4 words:

<table>
<tr><th>Modbus address
starting at 0xC000</th><th>Description</th></tr>
<tr><td>Word 1</td><td>Register number (register address)
This word must have the hexadecimal equivalent of the module register address. The register numbers can be taken from the respective module description.</td></tr>
<tr><td>Word 2</td><td>Register type (high byte) + Register size (low byte)
This word contains the register type in the higher-value byte and the register length (in bytes) in the lower-value byte. Both values must be specified in hexadecimal.
<table>
<tr><th>Register type</th><th>Description</th></tr>
<tr><td>0</td><td>Cyclic dynamic input register (read)</td></tr>
<tr><td>1</td><td>Cyclic dynamic output register (write)</td></tr>
<tr><td>2</td><td>Cyclic fixed input register (read)</td></tr>
<tr><td>3</td><td>Cyclic fixed output register (write)</td></tr>
<tr><td>4</td><td>Reserved</td></tr>
<tr><td>5</td><td>Acyclic output registers (write)</td></tr>
</table></td></tr>
<tr><td>Word 3</td><td>Register value high word</td></tr>
<tr><td>Word 4</td><td>Register value low word</td></tr>
</table>

For more information about full configurations, see "Structure of the configuration data block" on page 36.

10.4 Example of a register configuration

In this example, an input filter and sensor type should be configured in the first slot (i.e. after the power supply module) of an X20AT2402.

10.4.1 Entering I/O module parameters

I/O module parameters reference the actual register configuration data.

Address 1 = First module after the power supply (power supply module)	**Value**	**Note**
0xA018	0x1BA8	Module product code (hardware ID) The hardware ID for a "wildcard" configuration can also be specified as 0x0000 (see "I/O module register configuration" on page 70).
0xA019	0x00FE	Module start mode according to the module documentation
0xA01A	**0xC000**	Starting address of the register configuration (configuration data index)
0xA01B	0x0002	Number of register configurations (configuration data length)

10.4.2 Entering register configuration data

Input filter:	Register 24, 1 byte
Sensor type:	Register 26, 1 byte

Address	**Value**	**Note**
0xC000	0x0018	Register number for input filter (decimal 24)
0xC001	0x0501	Acyclic register (0x05) with the size 0x01
0xC002	0x0000	Register value high word
0xC003	x	Value for the input filter configuration according to the module documentation
0xC004	0x001A	Register number for sensor type (decimal 26)
0xC005	0x0501	Acyclic register (0x05) with the size 0x01
0xC006	0x0000	Register value high word
0xC007	0x0003	Sensor type "S" according to the module documentation

11 Diagnosticsoptions

11.1 General information

The Modbus TCP bus controller offers extensive diagnostic options on the controller as well as on the connected modules. Unless otherwise stated, these diagnostic parameters can only be read. An error code is returned in response to write access.

Diagnostic data is composed of:

- Product data (e.g. module serial numbers, hardware and firmware versions)
- Operating status (e.g. watchdog expired, IP address conflict, module status)
- Statistics (e.g. Modbus TCP protocol, network, X2X Link)

11.2 Product data

The Bus controller and I/O module project data can only be read.

11.2.1 Bus controller

Product data	
Address range	**Description**
0x1080 - 0x1082	Serial number
0x1083	Product code
0x1084	Hardware major revision
0x1085	Hardware minor revision
0x1086	Active firmware major revision
0x1087	Active firmware minor revision
0x1088	FPGA hardware revision
0x1089	Active boot block
0x108A	Default firmware major revision
0x108B	Default firmware minor revision
0x108C	Update firmware major revision
0x108D	Update firmware minor revision
0x108E	Default FPGA software revision
0x108F	Update FPGA software revision

11.2.2 I/O modules

Description	Module-oriented value	Parameter-oriented value
Product code (hardware ID)	0xA**1	0xB1**
Serial number	0xA**2 - 0xA**3	0xB2** - 0xB3**
Firmware version	0xA**C	0xBC**
Hardware variant (hardware revision)	0xA**D	0xBD**

The placeholders (*) correspond to the module slot (i.e. slot index) in hexadecimal format. This parameter specifies the data for the module currently in this slot.
For additional details about module- and parameter-oriented access, see "Module-specific parameters" on page 65.

11.2.2.1 Serial number

In contrast to the bus controller serial number, which is composed of the product code and actual serial number (corresponding to the printed 11-digit barcode) and can be read via 3 word addresses, the product code and serial number for I/O modules can only be read separately (see "Composition of the module serial number" on page 67).

11.2.2.2 Firmware and hardware version

In contrast to the firmware version and hardware variant of the bus controller, where the version specification is composed of a major and minor entry, I/O modules have only one number entry.

11.2.3 Operating status

11.2.3.1 Bus controller

Value	Description
0x1184	Modified configuration flag
0x1185	Default configuration flag
0x1186	Bus controller operating status
0x1187	Bus controller error status

For details about permitted Modbus function codes and data lengths, see "Miscellaneous" on page 53.

11.2.3.2 Modified configuration flag

Value: 0x1184

This flag is automatically set to the value 0xC1 by the bus controller whenever system data is modified. This provides a way for the user to check for unintended data modifications. This flag is also stored along with the other system data in flash memory. The user can delete or set this flag by writing the constant 0xC0 or 0xC1, respectively.

Constant	Description
0xC0	Data not modified
0xC1	Data modification found

11.2.3.3 Default configuration flag

Value: 0x1185

This flag provides information about whether or not the bus controller has already been configured. If the bus controller starts up with default values, the flag receives the value 0xC1. This flag is automatically set to the value 0xC0 if system parameters are modified.
It is only possible for the user to read this flag. A restart with the constant "0xC2" is needed to reset all parameters to their default values. Write access to the Modified configuration flag also results in a change to 0xC0.

Constant	Description
0xC0	The bus controller has already been configured.
0xC1	All system parameters correspond to their default values.

11.2.3.4 Bus controller operating status

Value: 0x1186

Bit	Value	Description
0	0x0001	Bus controller no longer in its default state, i.e. settings and configurations have already been made
1	0x0002	At least one master connection exists
2	0x0004	System boot or I/O module initialization active
3	0x0008	Bus controller waiting for an IP address from the DHCP server

> ## Information:
>
> **Setting the bit to 0 corresponds to the value 0xC0 of the default configuration flag.**

11.2.3.5 Bus controller error status

Value: 0x1187

An error-free bus controller state is indicated if no bits are set.

Bit	Value	Description
0	0x0001	Watchdog timeout
1	0x0002	Flash memory read error
2	0x0004	Faulty or missing module detected during runtime
3	0x0008	Missing module detected during boot phase
4	0x0010	Incorrect module detected during boot phase
5	0x0020	Faulty I/O module configuration data
6	0x0040	IP address conflict

11.2.3.6 I/O modules

Description	Module-oriented value	Parameter-oriented value
Module status	0xA**0	0xB0**

The operating status of individual modules can be read via the 0xA**0 and 0xB0** addresses (see "Description of individual module parameters" on page 66). The placeholders (*) correspond to the module slot (i.e. slot index) in hexadecimal format.
For additional details about module- and parameter-oriented access, see "Module-specific parameters" on page 65.

Example:

Reading the module status of the first 5 modules

Value	Description
fc4	Read input register
0xB000	Starting address
0x5	Number of addresses to be read

Possible return values

Value	Description
0x00 "0"	No module connected
0x4E "N"	Bus module present but electronics module not starting. Cause: Faulty I/O power supply, or the electronics module is not connected to the bus module.
0x42 "B"	Boot procedure (OS loader test)
0x55 "U"	Boot procedure (uploading IDs)
0x70 and 0x50 "p" / "P"	Preoperational (module ready to start)
0x53 "S"	Synchronization based on the bus controller's time
0x43 "C"	Module being configured
0x52 "R"	Module active and functioning without errors
0x44 "D"	Firmware download active
0xE0	Error: Module without I/O firmware detected
0xE1	Error: Module with invalid firmware detected
0xE2	Error: Module cannot be activated, e.g. configuration error (incorrect function model, etc.)
0xE3	Error: Registers could not be registered, e.g. faulty module configuration data
0xE4	Error: Internal error, I/O module cannot be started
0xE5	Error: Module cannot be started, X2X Link frame too small
0xE6	Module not started, different module type configured for this slot

Information:

Further diagnostic information about the modules can be obtained from the X2X Link network status. The X2X Link network status is based on the bus modules or X2X Link controller, not the actual I/O module.

11.3 Statistics

11.3.1 Modbus protocol statistics

Modbus protocol statistics	
Address range	**Description**
0x10C0	Number of client connections
0x10C1 - 0x10C2	Global telegram counter
0x10C3 - 0x10C4	Local telegram counter
0x10C5 - 0x10C6	Global protocol error counter
0x10C7 - 0x10C8	Local protocol error counter
0x10C9 - 0x10CA	Global maximum command execution time in µs
0x10CB - 0x10CC	Local maximum command execution time in µs
0x10CD - 0x10CE	Global minimum command execution time in µs
0x10CF - 0x10D0	Local minimum command execution time in µs
0x10D1 - 0x10D2	Global protocol fragment counter
0x10D3 - 0x10D4	Local protocol fragment counter

11.3.2 X2X Link statistics

X2X Link statistics	
Address range	**Description**
0x11C0	X2X Link cycle counter
0x11C1	Number of X2X Link off cycles
0x11C2	Cyclic errors
0x11C3	Cyclic: Bus timing errors
0x11C4	Cyclic: Frame timing errors
0x11C5	Cyclic: Frame checksum errors
0x11C6	Cyclic: Frame pending errors
0x11C7	Cyclic: Buffer underrun
0x11C8	Cyclic: Buffer overflow
0x11C9	Acyclic errors
0x11CA	Acyclic: Bus timing errors
0x11CB	Acyclic: Frame timing errors
0x11CC	Acyclic: Frame checksum errors
0x11CD	Acyclic: Frame pending errors
0x11CE	Acyclic: Buffer underrun
0x11CF	Acyclic: Buffer overflow

11.3.3 Network statistics

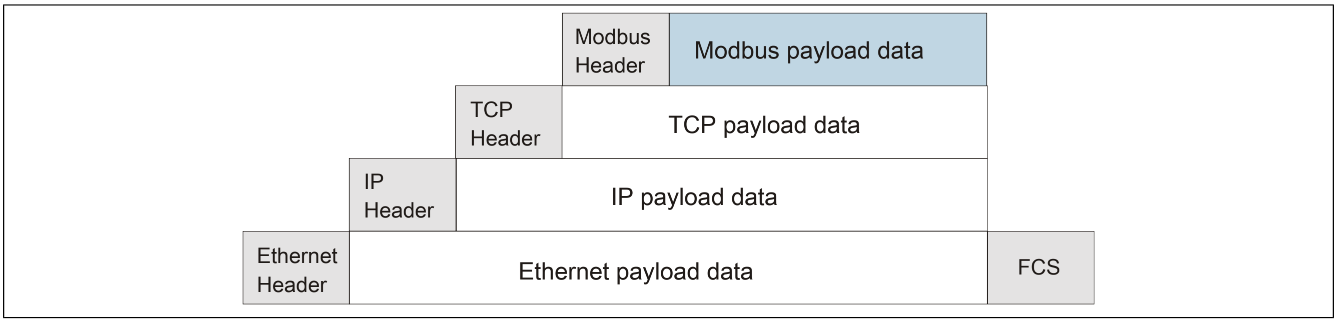
For the network statistics, the bus controllers can query the values for the following ports separately:

- **X20BC0087** and **X67BCJ321.L12**: IF1 and IF2

Network statistics	
Address range	**Description**
0x1200	IF1: Ethernet frames received
0x1201	IF1: Frames lost due to high load
0x1202	IF1: Oversized frames
0x1203	IF1: CRC error
0x1204	IF1: Frames lost
0x1205	IF1: Frames lost due to high load
0x1206	IF1: Collisions
0x1207	IF1: Frames lost due to switch overflow
0x1208	IF1: Frames lost due to switch errors
0x1210	IF2: Ethernet frames received
0x1211	IF2: Frames lost due to high load
0x1212	IF2: Oversized frames
0x1213	IF2: CRC error
0x1214	IF2: Frames lost
0x1215	IF2: Frames lost due to high load
0x1216	IF2: Collisions
0x1217	IF2: Frames lost due to switch overflow
0x1218	IF2: Frames lost due to switch errors

12 Modbus protocol basics

12.1 Communication protocol



Data is stored in "big-endian" format, i.e. the most significant byte is written to the first position in memory or the communication stream.

Example

Transferring the word 0x1234

Sequence in communication stream:	0x12	0x34		

Transferring the word 0x11223344

Sequence in communication stream:	0x11	0x22	0x33	0x44

12.2 Protocol structure

Each Modbus command stream begins with a default 7-byte header. This header depends on the command and is used to manage communication.
Each Modbus command begins with a function code of 1 byte; the protocol is at the seventh byte position (starting with byte 0).

Range	Bytes	Description		Client action	Server action
Header	0, 1	Transaction identifier	Unique command ID assigned by the client	Initialized	Copied
	2, 3	Protocol identifier	0 = Modbus protocol (constant)	Initialized	Copied
	4, 5	Length	Number of subsequent bytes	Initialized (request)	Initialized (request)
	6	Unit identifier	Remote slave ID for connecting other bus systems	Initialized	Copied
Function code	7	Always located in the first position after the header			
Modbus function-specific part The Modbus payload data can have a maximum length of 253 bytes. The function code is a part of the payload data. An entire Modbus telegram with header and payload data has a maximum length of 260 bytes.					

12.3 Error handling

If an error occurs during Modbus command execution, then a default error code is returned.

Modbus command execution is a serial process. As such, it is possible that some parts of a command can be executed without errors but that other areas inside of the same command will cause an error. One example of this is fc16 "Write to multiple registers" to an address range that is only partially writable.

In this case, the command would only be carried out up to an undefined part. In order to avoid this undefined state, make sure that no partial actions are carried out on the B&R Modbus TCP bus controller when an error occurs. This means either the command is executed completely and without errors or all partial actions already executed are discarded.

12.3.1 General structure of an error

Length in bytes	Description
7	Modbus header
1	Modbus function code + 0x80
1	Error code

12.3.2 Possible error codes

Error code	Protocol-specific name	Description
1	Illegal function	Unimplemented Modbus function
2	Illegal data address	Invalid address or address range
3	Illegal data value	Protocol parameter outside the permissible range of values
4	Slave device failure	Communication watchdog expired
6	Slave device busy	Modbus commands not possible at this time

13 Description of individual Modbus functions

13.1 Overview of Modbus function codes

Sorted by data type (bit- or word-oriented)

Access to digital data:	1, 2, 5, 15
Access to analog data:	3, 4, 6, 16, 23

Function code	Internal ID	Protocol-specific name
1	Read multiple digital outputs	Read coils
2	Read multiple digital inputs	Read discrete inputs
5	Write to one digital output	Write single coil
15	Write to multiple digital outputs	Write multiple coils
3	Read multiple analog outputs	Read holding registers
4	Read multiple analog inputs	Read input register
6	Write to one analog output	Write single register
16	Write to multiple analog outputs	Write multiple registers
23	Read and write several analog outputs	Read/Write multiple registers

Sorted by access method (read/write)

Read access:	1, 2, 3, 4, 23
Write access:	5, 6, 15, 16, 23

Function code	Internal ID	Protocol-specific name
1	Read multiple digital outputs	Read coils
2	Read multiple digital inputs	Read discrete inputs
3	Read multiple analog outputs	Read holding registers
4	Read multiple analog inputs	Read input register
5	Write to one digital output	Write single coil
6	Write to one analog output	Write single register
15	Write to multiple digital outputs	Write multiple coils
16	Write to multiple analog outputs	Write multiple registers
23	Read and write several analog outputs	Read/Write multiple registers

13.2 FC1: Read coils

This function can be used to read back multiple digital outputs.
A maximum of 2,000 bits can be read with a single request.
Digital outputs begin at address 0x0000.

Example

Read bit 1 to 4 starting at address 0x0000.

Request:

Description	Length in bytes	Example
Function code	1	**0x1**
Starting address	2	0x0000
Number of bits to be read	2	0x4

Response:

Description	Length in bytes	Example
Function code	1	**0x1**
Number of bytes	1	0x1
Bit data	1	0xF

In this example, 4 bits of data (0xF, therefore all "1") are compiled into a byte and transferred.

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
0	0	0	0	1	1	1	1
Filled with zeros				0xF			

If more than 8 bits of data are read, more bytes are sent back in response.

If the number of outstanding bits is not a multiple of 8, the remaining bits of the last byte are filled with zeros.

13.3 FC2: Read discrete inputs

This function can be used to read multiple digital inputs.
A maximum of 2,000 bits can be read with a single request.
Digital inputs begin at address 0x0000.

Example

Read bit 1 to 4 starting at address 0x0000.

Request:

Description	Length in bytes	Example
Function code	1	**0x2**
Starting address	2	0x0000
Number of bits to be read	2	0x4

Response:

Description	Length in bytes	Example
Function code	1	**0x2**
Number of bytes	1	0x1
Bit data	1	0xF

In this example, 4 bits of data (0xF, therefore all "1") are compiled into a byte and transferred.

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
0	0	0	0	1	1	1	1
Filled with zeros				0xF			

If more than 8 bits of data are read, more bytes are sent back in response.

If the number of outstanding bits is not a multiple of 8, the remaining bits of the last byte are filled with zeros.

13.4 FC3: Read holding register

This function can be used to read multiple analog inputs or outputs.
Digital inputs and outputs stored additionally in the word-oriented area can also be read with this function. A maximum of 125 registers can be read with a single request.

Example

Read 2 registers (words) starting at address 0x0800.

Request:

Description	Length in bytes	Example
Function code	1	**0x3**
Starting address	2	0x0800
Number of registers (words) to be read	2	0x2

Response:

Description	Length in bytes	Example
Function code	1	**0x3**
Number of bytes	1	0x4
Register data (word 1)	2	0xABCD
Register data (word 2)	2	0x1234

A register (word) is composed of 2 bytes, with the most significant byte always transferred as the first data unit (big-endian).

Register 1 at address 0x0800		Register 2 at address 0x0801	
High byte	Low byte	High byte	Low byte
0xAB	0xCD	0x12	0x34
0xABCD		0x1234	

13.5 FC4: Read input register

This function can be used to read multiple analog inputs or outputs.
Digital inputs and outputs stored additionally in the word-oriented area can also be read with this function. A maximum of 125 registers can be read with a single request.

Example

Read 2 registers (words) starting at address 0x0000.

Request:

Description	Length in bytes	Example
Function code	1	**0x4**
Starting address	2	0x0000
Number of registers (words) to be read	2	0x2

Response:

Description	Length in bytes	Example
Function code	1	**0x4**
Number of bytes	1	0x4
Register data (word 1)	2	0xABCD
Register data (word 2)	2	0x1234

A register (word) is composed of 2 bytes, with the most significant byte always transferred as the first data unit (big-endian).

Register 1 at address 0x0000		Register 2 at address 0x0001	
High byte	Low byte	High byte	Low byte
0xAB	0xCD	0x12	0x34
0xABCD		0x1234	

13.6 FC5: Write single coil

This function can be used to set a digital output.
Digital outputs begin at address 0x0000.

Example

Set bit 1 at address 0x0000 to high.

Request:

Description	Length in bytes	Example
Function code	1	**0x5**
Starting address	2	0x0000
Bit data	2	0xFF00

Response:

Description	Length in bytes	Example
Function code	1	**0x5**
Starting address	2	0x0000
Bit data	2	0xFF00

The controller responds with a "request echo" if no errors occur, i.e. the response is identical to (or echoes) the request.

High corresponds to the value:	0xFF00
Low corresponds to the value:	0x0000

13.7 FC6: Write single register

This function can be used to write to an analog output.
Digital outputs stored additionally in the word-oriented area can also be written to with this function.

Example

Write to a register at address 0x0800.

Request:

Description	Length in bytes	Example
Function code	1	**0x6**
Starting address	2	0x0800
Register data	2	0xABCD

Response:

Description	Length in bytes	Example
Function code	1	**0x6**
Starting address	2	0x0800
Register data	2	0xABCD

A register (word) is composed of 2 bytes, with the most significant byte always transferred as the first data unit (big-endian).

The controller responds with a "request echo" if no errors occur, i.e. the response is identical to (or echoes) the request.

13.8 FC15: Write multiple coils

This function can be used to set multiple digital outputs.
A maximum of 1968 bits can be set with a single command.
Digital outputs begin at address 0x0000.

Example

Set 12 bits (hexadecimal 0xC) to 1 starting at address 0x0000. Bits 1-8 are transferred in the fist byte (0xFF); bits 9-12 are transferred in the second byte (0xF). The remaining 4 bits of this second byte will be ignored and set to 0 by the master.

Request:

Description	Length in bytes	Example
Function code	1	**0xF**
Starting address	2	0x0000
Number of bits to be written	2	0xC
Number of bytes	1	0x2
Bit data (bit 8 to 1)	1	0xFF
Bit data (bit 16 to 9)	1	0xF

Response:

Description	Length in bytes	Example
Function code	1	**0xF**
Starting address	2	0x0000
Number of set bits	2	0xC

Byte 1							
Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
1	1	1	1	1	1	1	1
0xFF							

Byte 2							
Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9
0	0	0	0	1	1	1	1
Filled with zeros				0xF			

13.9 FC16: Write multiple registers

This function can be used to write to multiple analog outputs.
Digital outputs stored additionally in the word-oriented area can also be written to with this function.
A maximum of 123 registers can be written with a single command.

Example

Write to 2 registers starting at address 0x0800.

Request:

Description	Length in bytes	Example
Function code	1	**0x10**
Starting address	2	0x0800
Number of registers	2	0x2
Number of bytes	1	0x4
Register data (word 1)	2	0xABCD
Register data (word 2)	2	0x1234

Response:

Description	Length in bytes	Example
Function code	1	**0x10**
Starting address	2	0x0800
Number of registers	2	0x2

A register (word) is composed of 2 bytes, with the most significant byte always transferred as the first data unit (big-endian).

Register 1 at address 0x0800		Register 2 at address 0x0801	
High byte	Low byte	High byte	Low byte
0xAB	0xCD	0x12	0x34
0xABCD		0x1234	

13.10 FC23: Read/Write multiple registers

This function can be used to write to multiple analog outputs and read inputs/outputs at the same time. This function is a combination of FC3, FC4 and FC16.
Digital outputs stored additionally in the word-oriented area can also be written to with this function. A maximum of 125 registers can be read and 121 registers written.

> ### Information:
>
> **Write actions takes place before read actions.**

Example

Write to 2 registers at address 0x0800 and read 2 registers at address 0x0000.

Request:

Description	Length in bytes	Example
Function code	1	**0x17**
Starting address of registers to be read	2	0x0000
Number of registers to be read	2	0x2
Starting address of registers to be written	2	0x0800
Number of registers to be written	2	0x2
Number of bytes to be written	1	0x4
Register data (1st register to be written)	2	0xABCD
Register data (2nd register to be written)	2	0x1234

Response:

Description	Length in bytes	Example
Function code	1	**0x17**
Number of bytes read	1	0x4
Register data (1st register read)	2	0x1122
Register data (2nd register read)	2	0x3344

A register (word) is composed of 2 bytes, with the most significant byte always transferred as the first data unit (big-endian).

Registers written:

Register 1 at address 0x0800		Register 2 at address 0x0801	
High byte	Low byte	High byte	Low byte
0xAB	0xCD	0x12	0x34
0xABCD		0x1234	

Registers read:

Register 1 at address 0x0000		Register 2 at address 0x0001	
High byte	Low byte	High byte	Low byte
0x11	0x22	0x33	0x44
0x1122		0x3344	

14 Telnet interface

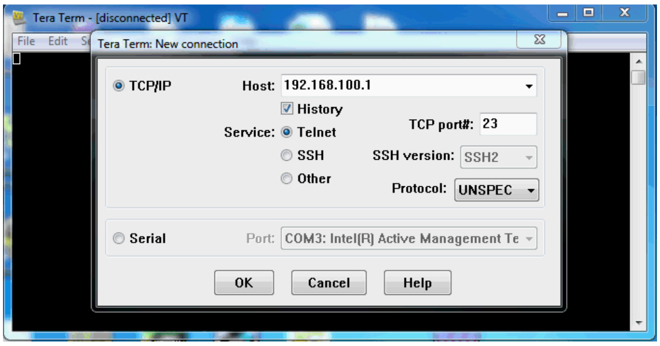
Telnet is a client/server protocol that uses TCP for data transfer (normally on port 23).
The Telnet interface for the Modbus TCP bus controller is a generic interface that can be used to execute Modbus commands 3, 4 and 6. Data length is limited to one word. Values can be specified in hexadecimal (0x) or decimal form.
In addition, the interface includes several shortcut commands, e.g. "Save data to flash memory" and "Erase flash memory".
Access via Telnet can be protected with a password. The maximum length is 14 characters and is case-sensitive (see "TelnetPasswort" on page 42).
This function is available in firmware version 1.46 and later. With firmware versions < 1.46, only the fixed password "BcModBus" can be used.

The syntax used for interface can be displayed via the "help" or "?" command. A Telnet client such as TeraTerm or PuTTY can be used to communicate via Telnet.

In Windows, Telnet can be launched by opening a command prompt (Windows Start menu / Run / "cmd") and typing "telnet" followed by the IP address of the bus controller (e.g. "telnet 192.168.100.1").

Example of settings for the Tera Term client:

1: Entering the bus controller IP address and port number

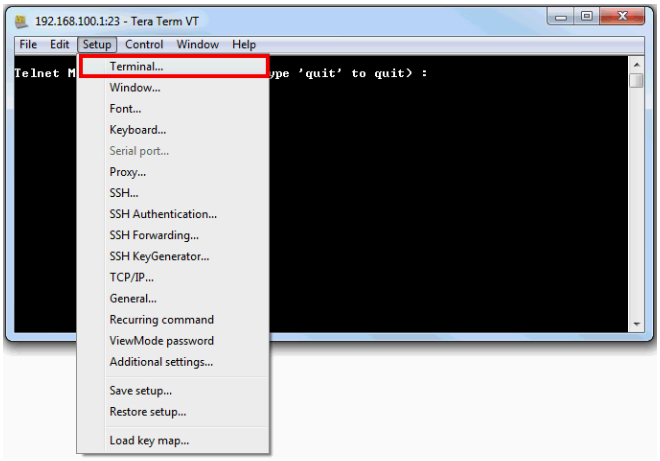


Tera Term - [disconnected] VT

Tera Term: New connection

● TCP/IP — Host: 192.168.100.1
☑ History
Service: ● Telnet — TCP port#: 23
○ SSH — SSH version: SSH2
○ Other — Protocol: UNSPEC
○ Serial — Port: COM3: Intel(R) Active Management Te

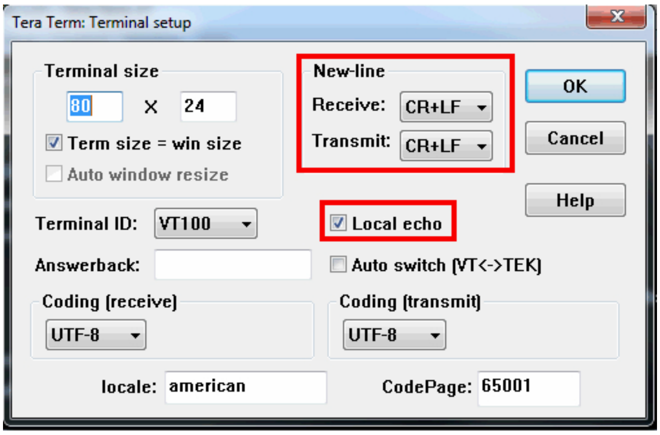
2: Selecting the terminal function



192.168.100.1:23 - Tera Term VT

```
Telnet M                                   ype 'quit' to quit> :
```

3: Important basic settings



Tera Term: Terminal setup

Terminal size: 80 × 24
☑ Term size = win size
☐ Auto window resize

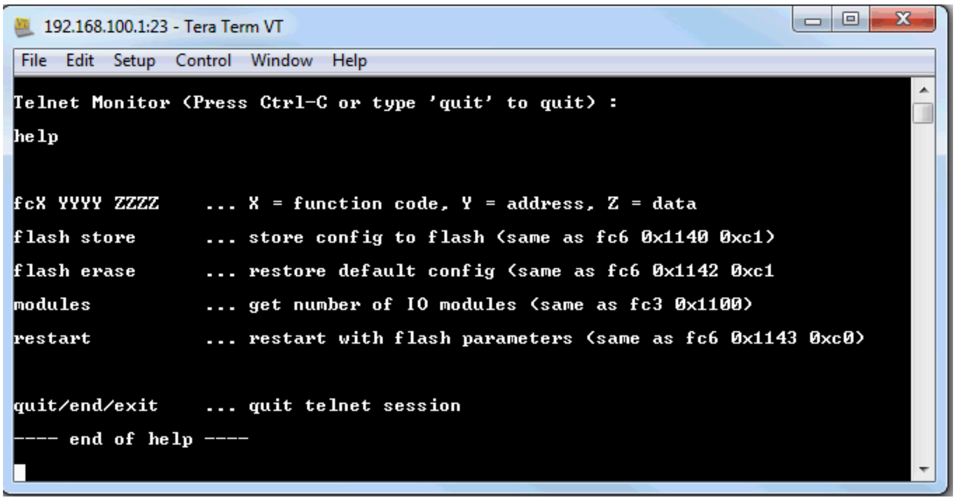
New-line
Receive: CR+LF
Transmit: CR+LF

Terminal ID: VT100
☑ Local echo
Answerback:
☐ Auto switch (VT<->TEK)

Coding (receive): UTF-8
Coding (transmit): UTF-8

locale: american
CodePage: 65001

Entering "help" or "?" displays the following information:



192.168.100.1:23 - Tera Term VT
File Edit Setup Control Window Help

```
Telnet Monitor (Press Ctrl-C or type 'quit' to quit) :
help

fcX YYYY ZZZZ    ... X = function code, Y = address, Z = data
flash store      ... store config to flash (same as fc6 0x1140 0xc1)
flash erase      ... restore default config (same as fc6 0x1142 0xc1
modules          ... get number of IO modules (same as fc3 0x1100)
restart          ... restart with flash parameters (same as fc6 0x1143 0xc0)

quit/end/exit    ... quit telnet session
---- end of help ----
```

14.1 Structure of the Telnet command line

In addition to the shortcut commands "flash store", "flash erase", "modules" and "restart", it is also possible to execute Modbus function codes 3, 4 and 6. The command line syntax for this is fc**X YYYY ZZZZ**.

X = Modbus function code

fc3 = Read holding register
fc4 = Read input register
fc6 = Write single register

YYYY = Address: Either in decimal or hexadecimal notation (4486 or 0x1186)
ZZZZ = Data: Optional specification, in decimal or hexadecimal format depending on the executed command

Information:

If the value is specified in hexadecimal format, then "0x" must precede the value.

14.2 Examples

14.2.1 Assigning an IP address

In addition to the options for assigning the bus controller an IP address, the Telnet interface provides a way to achieve simple access without having to use an additional tool, especially during commissioning. An Ethernet connection to the bus controller is required.

In this example, a new IP address (10.1.1.123) will be configured.

To do this, the following command lines must be entered in Telnet, each followed by pressing the "Enter" key.

Command:		Description:
fc6 0x1003 10	0x1003	Address area of the system parameters for the IP address (see "Communication" on page 40). These values can be entered in decimal or hexadecimal format (10 or 0xA). 1st part of the IP address
	10	10.xxx.xxx.xxx
fc6 0x1004 1	0x1004	2nd part of the IP address
	1	xxx.1.xxx.xxx
fc6 0x1005 1	0x1005	3rd part of the IP address
	1	xxx.xxx.1.xxx
fc6 0x1006 123	0x1006	4th part of the IP address
	123	xxx.xxx.xxx.123
flash store		Saves the changes from RAM to nonvolatile flash memory

> **Information:**
>
> **To enable the new IP address, the network address switches must be set to 0x00 and the bus controller must be restarted. This can be done with command "restart" in Telnet or by briefly disconnecting the power supply.**

14.2.2 AT module configuration

In this example, module X20AT4222 should be operated with a 2-wire Pt100 temperature sensor on channel 2. The module is located in the first slot after the power supply module.

The following 4 entries are used for this configuration (see "I/O module register configuration" on page 70):

1. Register number (register address)
2. Register type (high byte) + Register size (low byte)
3. Register value high word
4. Register value low word

Commands:	Value	Description:
fc6 0xA01A 0xC000	0xA**01**A	Module configuration data index. The module is located in slot **01** (i.e. it is the first X2X Link module after the power supply).
	0xC000	Starting address for the module register configuration. If a register needs to be configured, then registers 0xC000 to 0xC003 must be used. The next entry starts at 0xC004.
fc6 0xA01B 0x0001	0xA01**B**	The **B** parameter stands for the module configuration data length.
	0x0001	Only one register is needed to configure the sensor type, i.e. length = 1.
fc6 0xA019 0x0001	0xA01**9**	Parameter for the start mode (function model).
	0x0001	For 2-wire connections, the module must be configured for function model 1.
fc6 0xC000 0x0012	0xC000	**Register number**
	0x0012	The sensor type can be configured using register 18 (decimal 18 corresponds to hexadecimal 0x0012). The command is also permitted to be written as fc6 0xC000 18.
fc6 0xC001 0x0502	0xC001	**Register type**
	0x0502	This register is an output register and should be written acyclically. The register type is 5 (high byte) and has a size of 2 bytes. (see "Structure of the configuration data block" on page 36)
fc6 0xC002 0x0000	0xC002	**High word**
	0x0000	The high word is empty since the register is only 2 bytes.
fc6 0xC003 0x7727	0xC003	**Low word**
	0x7727	0x7727 is composed of the following: Bits 0 to 3 define channel 1, Bits 4 to 7, channel 2, etc. Type PT100 is set with the bit pattern 0010 (or 0x2). Since channels 1, 3 and 4 are not used, they must be configured to binary 0111 or 0x7 (channel switched off).
flash store		Saves the changes from RAM to nonvolatile flash memory