EtherNet/IP

User's manual

Version: **3.17 (November 2022)**
Order no.: **MAEIP-ENG**

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

1340333282916-3.17

1 General information

1.1 Coated modules

Coated modules are X20 modules with a protective coating for the electronics component. This coating protects X20c modules from condensation and corrosive gases.
The modules' electronics are fully compatible with the corresponding X20 modules.

For simplification purposes, only images and module IDs of uncoated modules are used in this user's manual.

The coating has been certified according to the following standards:

- Condensation: BMW GS 95011-4, 2x 1 cycle
- Corrosive gas: EN 60068-2-60, method 4, exposure 21 days

1.2 Function description

This bus controller makes it possible to connect X2X Link I/O nodes to EtherNet/IP. The bus controller can be operated via interface module X20IF10D1-1 or by 3rd-party systems with EtherNet/IP scanner functionality.

Bus controller properties:

- Auto-configuration of I/O modules
- Manual configuration of I/O modules with Automation Studio version 4.3 or later
- Can be configured by the scanner (master) using configuration assembly
- BOOTP and DHCP are supported.
- Device Level Ring (DLR) is not supported.
- Minimum fieldbus cycle time (also requested packet interval or RPI): 1 ms
- Maximum I/O data size In/Out: 511 bytes / 511 bytes

Functions:

- EtherNet/IP

EtherNet/IP

EtherNet/IP is an Ethernet-based fieldbus. The fieldbus is mainly used in automation technology.

2 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.

3 Technical description

3.1 X20 bus controller

3.1.1 Order data

Order number	Short description	Figure
	Bus controllers	
X20BC0088	X20 bus controller, 1 EtherNet/IP interface, integrated switch, web interface, 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	
	Terminal blocks	
X20TB12	X20 terminal block, 12-pin, 24 VDC keyed	

Table 1: X20BC0088 - Order data

3.1.2 Technical data

Order number	X20BC0088
Short description	
Bus controller	EtherNet/IP adapter (slave)
General information	
B&R ID code	0x26D8
Status indicators	Module status, network status, bus function
Diagnostics	
Module status	Yes, using LED status indicator and software
Bus function	Yes, using LED status indicator and software
Network status	Yes, using LED status indicator and software
Power consumption	
Bus	2 W
Additional power dissipation caused by actuators (resistive) [W]	-
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
EAC	Yes
KC	Yes
Interfaces	
Fieldbus	EtherNet/IP adapter (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

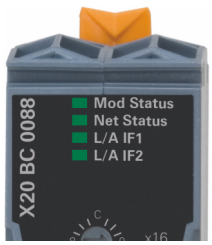
Table 2: X20BC0088 - Technical data

Order number	X20BC0088
Min. cycle time [1]	
Fieldbus	1 ms
X2X Link	500 µs
Synchronization between bus systems possible	No
Electrical properties	
Electrical isolation	EtherNet/IP 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	-
Storage	-40 to 85°C
Transport	-40 to 85°C
Relative humidity	
Operation	5 to 95%, non-condensing
Storage	5 to 95%, non-condensing
Transport	5 to 95%, non-condensing
Mechanical properties	
Note	Order 1x terminal block X20TB12 separately. Order 1x power supply module X20PS9400 or X20PS9402 separately. Order 1x bus base X20BB80 separately.
Pitch [2]	$37.5^{+0.2}$ mm

Table 2: X20BC0088 - Technical data

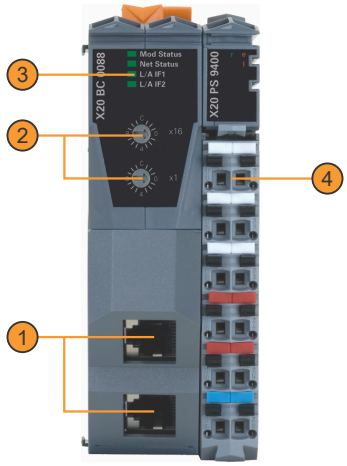
1) The minimum cycle time specifies how far the bus cycle can be reduced without communication errors occurring.
2) Pitch is based on the width of bus base X20BB80. In addition, power supply module X20PS9400 or X20PS9402 is always required for the bus controller.

3.1.3 LED status indicators

Figure	LED	Color	Status	Description
	Mod status[1]	Green	On	Indicates that there is at least one client connection
			Blinking	Bus controller not yet configured.
			Flickering	HTTP file upload (firmware or configuration file)
		Red	On	Major unrecoverable fault.
			Blinking	Major recoverable fault.
		Green/Red	Blinking	Initialization/Self-test
	Net status[1]	Green	On	Indicates at least one established active scanner (master) connection
			Blinking	Indicates no established active scanner (master) connection
			Off	Indicates no IP address has been assigned
		Rot	On	Indicates an IP address has been used more than once
			Blinking	Indicates a timeout on at least one connection
		Green/Red	Blinking	Initialization/Self-test
	L/A IFx	Green	Blinking	Ethernet activity taking place on the RJ45 interface (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 "Mod status" and "Net status" LEDs are green/red dual LEDs.

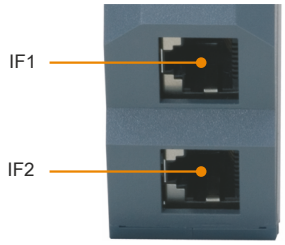
3.1.4 Operating and connection elements

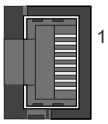


1	EtherNet/IP connection with 2x RJ45 for simple wiring	2	Network address switches
3	LED status indicators	4	Terminal block for bus controller and I/O supply

3.1.5 Ethernet interface

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



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

3.2 X67 bus controller

3.2.1 Order data

Order number	Short description	Figure
	Bus controller modules	
X67BCD321.L12	X67 bus controller, 1 EtherNet/IP 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	
X67BCD321.L12-1	X67 bus controller, 1 EtherNet/IP interface, X2X Link power supply 15 W, 16 digital channels configurable as inputs or outputs, 24 VDC, 0.5 A, pinning variant, configurable input filter, 2 event counters 50 kHz, M12 connectors, high-density module	

Table 3: X67BCD321.L12, X67BCD321.L12-1 - Order data

Required accessories
See "Required cables and connectors" on page 16. For a general overview, see section "Accessories - General overview" in the X67 system user's manual.

3.2.2 Technical data

Order number	X67BCD321.L12	X67BCD321.L12-1
Short description		
Bus controller	EtherNet/IP Adapter (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	0xACF7	0xDABF
Internal I/O module	0xB1E7	0xDACE
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	2.5 W	
Internal I/O	3.3 W	
X2X Link power supply	20.5 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	EtherNet/IP Adapter (slave)	
Variant	Internal 3x switch, male M12 circular connector, 2 female connectors on the module	
Line length	Max. 100 m between 2 stations (segment length)	

Table 4: X67BCD321.L12, X67BCD321.L12-1 - Technical data

Order number	X67BCD321.L12	X67BCD321.L12-1
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	
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 8)	
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".	

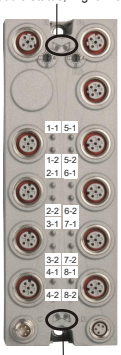
Table 4: X67BCD321.L12, X67BCD321.L12-1 - Technical data

Order number	X67BCD321.L12	X67BCD321.L12-1
Braking voltage when switching off inductive loads	50 VDC	
Electrical properties		
Electrical isolation	Channel isolated from bus EtherNet/IP 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	355 g	
Torque for connections		
M8	Max. 0.4 Nm	
M12	Max. 0.6 Nm	

Table 4: X67BCD321.L12, X67BCD321.L12-1 - Technical data

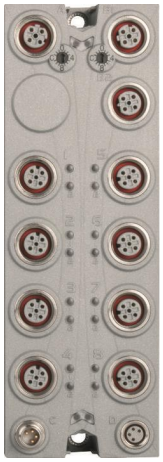
1) The minimum cycle time specifies how far the bus cycle can be reduced without communication errors occurring.
2) The power consumption of the sensors and actuators connected to the module is not permitted to exceed 12 W.

3.2.3 LED status indicators

Figure	LED	Color	Status	Description
Status indicator 1: Left: Module status, Right: Network status  Status indicator 2: Left: Green, Right: Red	**Status indicator 1:**: Status indicator for module and network functionality			
	Mod status[1]	Green	On	Indicates that there is at least one client connection.
			Blinking	Bus controller not yet configured.
		Red	On	Major unrecoverable fault.
			Blinking	Minor recoverable fault.
		Green/Red	Blinking	Initialization/Self-test
	Net status[2]	Green	On	Indicates at least one established active scanner connection.
			Blinking	Indicates no established active scanner connection.
			Off	Indicates no IP address has been assigned.
		Red	On	Indicates an IP address has been used more than once.
			Blinking	Indicates a timeout on at least one connection.
		Green/Red	Blinking	Initialization/Self-test
	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

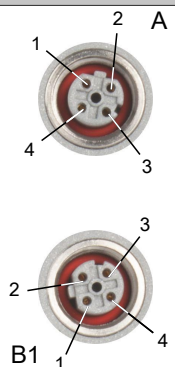
1) LED "Mod status" is a green/red dual LED. Several red blinking signals are displayed immediately after the device is switched on. However, this is a startup message, not an error (see the EtherNet/IP user's manual).
2) LED "Net status" is a green/red dual LED.

3.2.4 Operating and connection elements



3.2.5 Fieldbus interfaces

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

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

Information:

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

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

3.2.5.1 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:

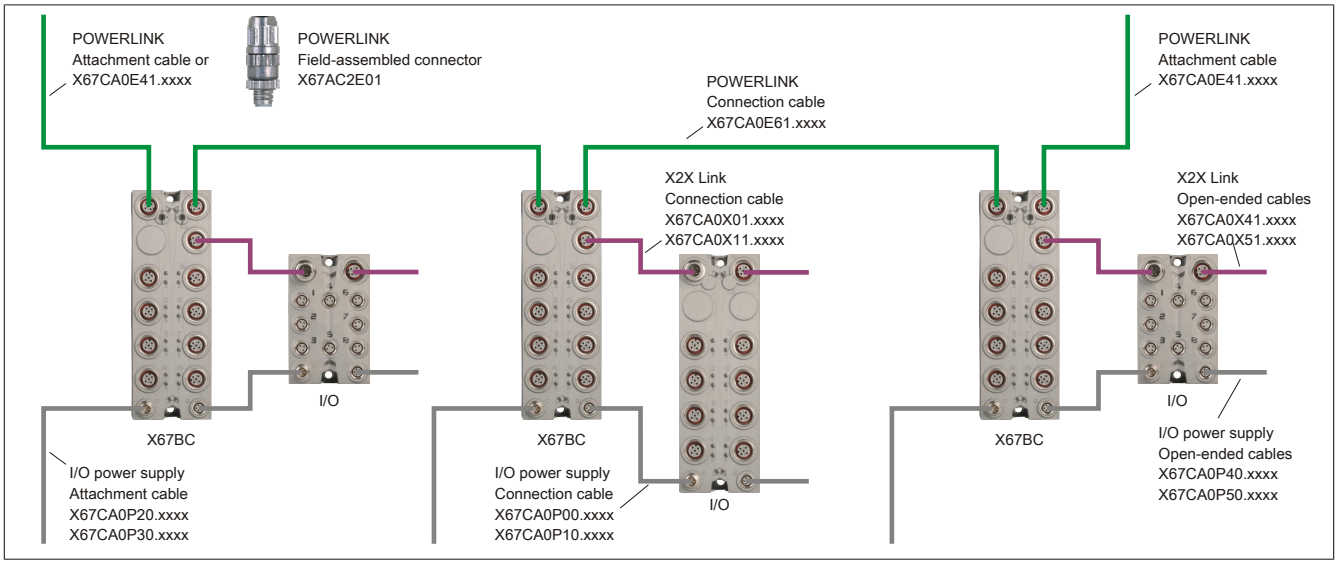
- Use Cat 5 SFTP cables.
- Observe the bend radius of the cable (see the data sheet of the cable)

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.

3.2.5.2 Required cables and connectors



4 Basic information

4.1 EtherNet/IP

EtherNet/IP (Ethernet industrial protocol) is an open Common Industrial Protocol (CIP) based fieldbus standard developed by Rockwell Automation and the Open DeviceNet Vendor Association (ODVA).

EtherNet/IP supports the provider-consumer principle for data exchange between individual network nodes.

4.2 Configuring I/O modules and data connections

After startup, the EtherNet/IP bus controller boots all I/O modules connected via B&R's X2X Link and uses them to create a local process image. For this to happen, configuration data must be created and transferred to the bus controller.

Depending on the data type, I/O data is split up into different assemblies:

- The base assemblies contain the data points of the I/O modules in order. The X2X Link station numbers are held in ascending order from left to right.
- Combined input and output assemblies group together the various base assemblies and are normally used to carry out I/O communication.

For additional information about assemblies, see "Assembly object and bus controller process image" on page 29. For additional information about configuration options, see "Configuration types of the I/O modules on the bus controller" on page 26.

4.2.1 Multifunction modules

Only standard function model "254" is supported when the bus controller is used to automatically configure X2X Link multifunction I/O modules. In order to use other function models, these must be configured accordingly with Automation Studio. For additional information about the module configuration, see "Configuration types of the I/O modules on the bus controller" on page 26.

4.3 Deleting an existing configuration

An existing configuration can be deleted in the following ways:

- Via the web server
- Via the Telnet interface
- Via fieldbus service 0x35 of bus controller class 0x64. This service does not require an attribute.

This will reset the bus controller to its factory settings.

If the configuration data in flash memory should be retained, a restart can be performed using "service 0x5 "Reset" " on page 39 of class 0x1 with attribute "1" or "2".

4.4 Automation Studio

The bus controller and all connected I/O modules can be configured using Automation Studio V4.3 or later.

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 a project is compiled, configuration files are created that can be either integrated directly in another service provider's development environment or manually transferred to the bus controller.
Automation Studio always creates a Manual configuration (full configuration).

4.5 Additional services

An integrated Web server as well as the Telnet service is available for managing and troubleshooting the bus controller and connected I/O modules.

4.6 Functionality

- UCMM Message server (not connected)
- Class 3 Message server (connected)
- Class 1 I/O server (connected)

Corresponds to the following functionality level:

- Level 1 (explicit message server)
- Level 2 (I/O message server)

4.7 Performance

Support is provided for up to 32 class 1 or class 3 connections.

4.8 Supported CIP and vendor-specific classes

Class ID	Name
0x1	Identity object
0x2	Message router object
0x4	Assembly object
0x6	Connection manager object
0x64	Bus controller object
0x65	I/O module object
0xF4	Port object
0xF5	TCP/IP interface object
0xF6	Ethernet link object

5 Bus controller properties

5.1 Speed and performance

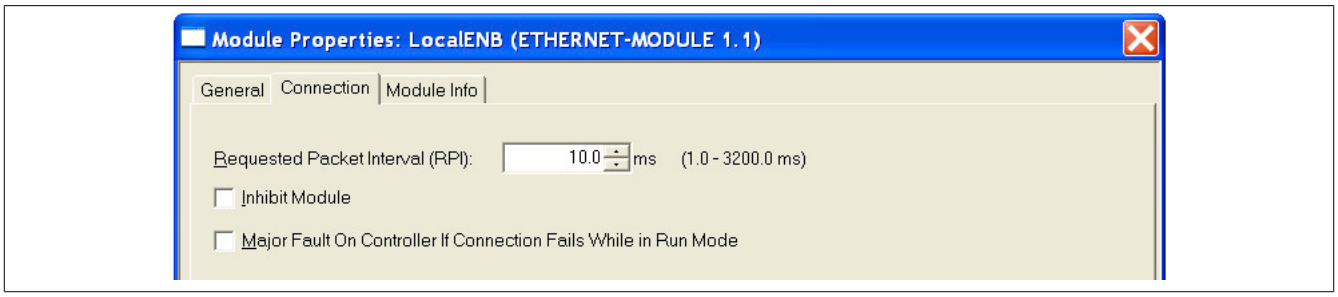
The shortest possible X2X Link cycle time is 0.5 ms, and the shortest possible RPI time is 1 ms. The number of modules behind the bus controller is limited by the X2X Link cycle time and the size of the configuration assemblies. Automation Studio issues a warning if too much I/O data has been configured for a certain cycle time. The configuration size can be reduced by grouping similar modules under a bus controller.

5.1.1 Setting the RPI and X2X Link time

RPI time

The request packet interval (RPI) controls the update rate of the I/O data for the bus controller. This value can be changed in RSLogix 5000 by right-clicking on the generic Ethernet module and selecting "Properties".

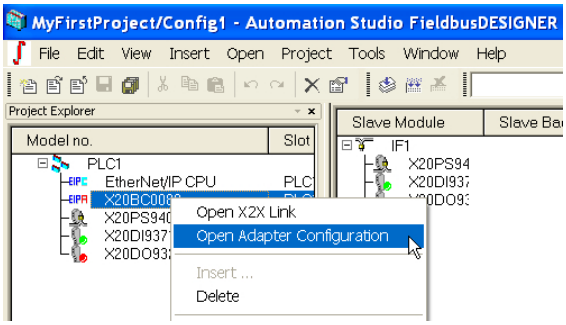
The smallest RPI value is 1 ms. The default value is 10 ms.

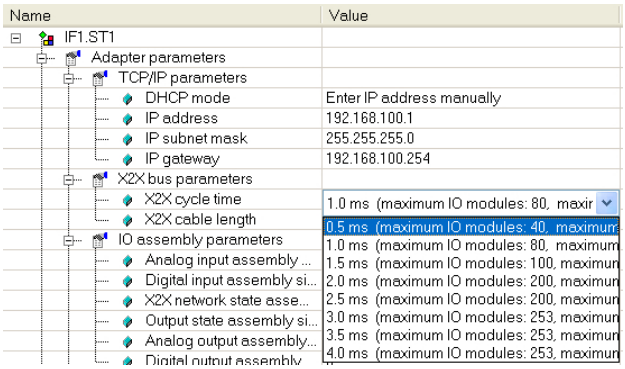


X2X Link cycle time

The X2X Link cycle time can be changed in the following ways:

- By calling the attribute 0x80 X2X Link configuration of the bus controller object (class 0x64, instance 0x1). The default value is 1 ms. In order to use the new settings, the bus controller must be restarted.
- In Automation Studio, after right-clicking on "Open adapter configuration":



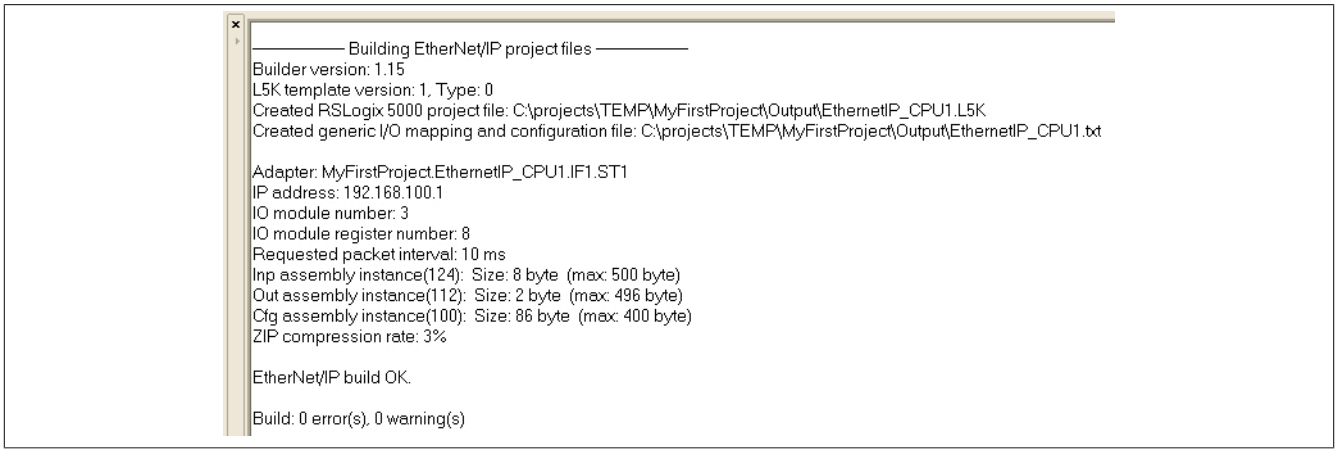


5.2 System size limits

The possible number of modules behind the bus controller is determined by the configuration buffer limit of the master system and the X2X Link cycle time. In the master system, a maximum of 400 bytes of configuration data is permitted in the buffer. If this is not sufficient, "extended configuration assemblies" must be used. See "Advanced configuration assemblies" on page 32.

The size of the configuration data depends on the number of registers used per module. If many different modules are used, this can severely limit the possible number. If many modules of the same type are used, much more can be connected by compressing the configuration data.

If the project is created by Automation Studio, the size of the configuration and input/output assemblies in the finished project are displayed.



If more modules are needed, they can be split between different bus controllers. The bus controller network can contain up to 253 analog and digital I/O modules.

Limits (from Allen-Bradley):

Configuration data:	400 bytes
Output assembly:	496 bytes
Input assembly:	500 bytes

Each byte represents 8 digital I/Os on an X20DI9371 or X20DO9322 module.

5.3 Multiple configurations

It is possible to download multiple configurations on the bus controller. This is done either via explicit messages or via the web interface. An RSLogix 5000 project created with Automation Studio is downloaded to the standard configuration (assembly instance 100 or 0x64 of class 0x04).

10 additional assembly IDs are reserved for configurations (instances 130 to 139 or 0x82 to 0x8B of the assembly object, class 0x4). After downloading to the bus controller, the active configuration can be set by service 0x37 of the bus controller object (class 0x64).

5.4 Using I/O modules

Most modules in the X20, X67 and XV series can be used behind the bus controller without limitations, including switching between function models. This includes both standard I/O modules and modules with node number switches.

However, there are limitations when using the following modules and module functions:

- Serial interface modules (e.g. X20CS10x0) can only be used behind the bus controller if they are operated in the flat stream function model.
- Stepper motor modules (X20SMxxxx) must be operated in the ramp function model.
- NetTime and functions based on it cannot be used with NetTime-capable modules.
- Module functions that require special data exchange between the module and program cannot be used (for example, transferring trace data in module X20AIx632).

6 Commissioning

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

- Static IP address
- Operation with a DHCP server

The network address switches as well as the TCP/IP object can be used to configure the 2 possibilities.

If the network address switches are 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:

- Via the web server
- Via the fieldbus
- Via the Telnet interface

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.

Operation with a DHCP server can also be configured using attribute "Configuration control" of the TCP/IP object, class 0xF5, instance 1, attribute 3, bit 1. To use this setting, the network address switches must be set to the value 0x00. In this case, the parameter from attribute 6 of the TCP/IP object, class 0xF5 is used as the hostname.

For the connection from the slave to the master, the bus controller only supports the option "Connection is pure data and is modeless"!

6.1 Connecting to the bus controller via Ethernet

The connection between the EtherNet/IP scanner (master) and the bus controller (adapter) can be established as follows:

- Direct connection via patch cable between the PC's network interface and the bus controller
- Over an Ethernet network. If necessary, multiple bus controllers can be connected to the network at the same time.

Straight-through or crossover Ethernet cables can be used. The IF1 or IF2 Ethernet interface can 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 B&R EtherNet/IP 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, the controller is assigned the IP address 192.168.100.1 after a restart.
For more details about the address switch, see "Configures network settings" on page 23.

6.2 Startup

Switching on the power triggers the initialization phase. 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 using this information.

The "Mod status" LED on the bus controller indicates any problems during startup by blinking in a certain pattern. For information about deleting a faulty configuration, see "Deleting an existing configuration" on page 17.

6.2.1 Blink codes

The boot loader indicates the following states via the module's "Mod status" status LED:

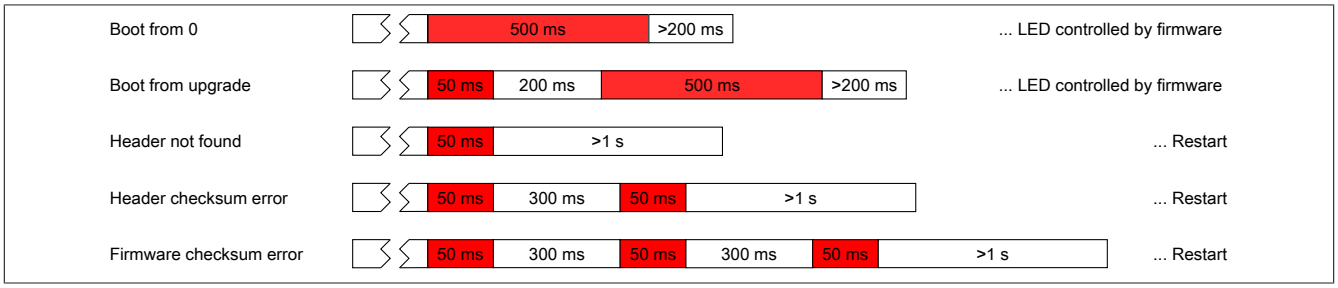


Figure 1: Blink codes during startup

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).

6.2.2 Forcing a boot from the default sector

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

To force a boot from the factory default sector, one of the network address switches must be moved continuously during booting. This is detected by the boot loader, which causes the module's "Mod status" 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.

6.3 Restarting the bus controller

The bus controller can be restarted in the following ways:

- Temporarily interrupting the power supply. This is referred to as "power cycling".
- Via the web server
- Via the Telnet interface
- Via the fieldbus

7 Configures network settings

Changes to the network settings or network address switches are only applied after a restart.

If the bus controller is restarted with the network address switch value 0xFF, it is initialized with the IP address 192.168.100.1. This address is also the factory default setting.
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.

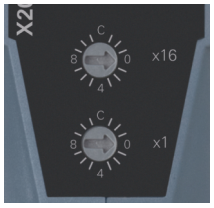
Example of the primary NetBIOS name:

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

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

7.1 Network address switches



X20

X67

The network address switches have multiple functions:

- Sets the default IP address (in the range 0x01 to 0x7F)
- Enables operation with a DHCP server (in the range 0x80 to 0xEF)
- 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 section "Overview of network address switch functionality" on page 24.

Information:

- **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, with the exception of the value 0x00.**
- **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.**
- **Changes to the network address switches are only applied after a restart.**

7.2 Overview of network address switch functionality

Switch position	Description
0x00	This switch position is the factory default setting. In this position, the network address switches have no effect on system parameters. The configured communication parameters[1] stored in the controller's flash memory such as the IP address are used. The adapter is started via DHCP if attribute 3 "Configuration control" was set to DHCP. The bus controller is started with the factory default values (as also specified in switch position 0xFF) if no valid flash data is present.
0x01 - 0x7F	The least significant byte (last byte) of the IP address stored in flash memory is replaced by the network address switch value. The IP address is 192.168.100.xxx, with xxx corresponding to the decimal value from the network address switches. Except for "Configuration control", all other bus controller parameters are read from flash memory and used without being changed.
0x80 - 0xEF	Sets the bus controller to DHCP mode for this range. A hostname is generated according to how the network address switches are set. **Example** The generated hostname is made up of 3 elements: "br" + "eip" + network address switch (3 decimal places) This means, for example, that the following hostname is generated for network address switch setting 0xD7 (dec. 215): "breip215".
0xF0 - 0xFD	Reserved. This has the same function as position 0xFF, i.e. all communication parameters are initialized with their default values.
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: "breip" + Network address switches (3 decimal places) • X2X Link configuration: 1 ms cycle time • X2X Link cable length: 0 m

1 See "TCP/IP interface object 0xF5" on page 42

7.3 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 a "ping" command together with the hostname. The bus controller registers the 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 for address switch setting 0xD7 (dec. 215): "breip215".

If the network address switches are set to 0x00, operation with a DHCP server can also be configured using the "Configuration control" attribute of the TCP/IP object (setting class 0xF5, instance 1, attribute 3, bit 1). In this case, the parameter from attribute 6, TCP/IP object, class 0xF5 is used as the hostname.

If no DNS service is available in the network, access is also possible via the two NetBIOS names of the bus controller. If the address switches are set to 0x00, it is identical to the primary NetBIOS name. Up to firmware version 3.07, the secondary NetBIOS name is identical to the hostname.
The bus controller can only be reached via its NetBIOS name if no other routers or gateways are in the way.

7.4 Setting network parameters manually

Network parameters can be modified in the following ways:

- Via the integrated web server
- Via the Telnet interface
- Via fieldbus object class 0xF5

If the IP address should be set via the TCP/IP object, then the new address will only be saved to flash if attribute 3 (configuration control) of the TCP/IP object is set to 0.

Information:

Changes to attributes in the TCP/IP object are stored immediately to flash memory and do not have to be saved explicitly. They are applied after the bus controller is restarted as long as a different setting is not made using the network address switches.

7.5 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.

7.6 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 "Configures network settings" on page 23).

The secondary NetBIOS name corresponds to the primary NetBIOS name at address switch position 0x00. This is necessary because there may be several bus controllers with the address switch 0x00 in a 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" + "eip" + Address switch value (3 decimal places).

Up to firmware version 3.07: 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).

8 Configuration types of the I/O modules on the bus controller

After it is started after the power supply has been interrupted, the EtherNet/IP bus controller detects all connected I/O - modules and creates and internal image of the input and output data.

There are 2 ways to configure the connected I/O modules:

- Automatic configuration
- Manual configuration (Manual configuration (full configuration))

If configuration data for the I/O modules is stored in flash memory on the bus controller (i.e. manual configuration), the respective modules will be configured accordingly at startup. The configuration data is stored in "Assembly object" class 0x4 in the vendor-specific instances 100 (0x64) or 130 (0x82) to 139 (0x8B). For additional details, see "Configuration assembly" on page 32.

If no configuration data is available, then the I/O modules will be started with their default settings. This operating mode is referred to as "automatic configuration".

	Automatic configuration	Manual configuration
Area of application	Use of simple input/output modules (digital/analog I/O using the default settings)	Use of simple and/or complex I/O modules
Creating the configuration data	Not required	Using Automation Studio V4.3 or later
Information via I/O data points (I/O mapping)	Module documentation or "I/O assembly mapping" in the Web server menu	From the text file generated by Automation Studio or the "I/O assembly mapping" of the web server menu
Configuration of the I/O modules	Only function model 254: "Bus controller" No possibility to intervene in input/output data Default settings of configuration registers	Any function model Possible to add or remove I/O data points from the configuration Predefined values passed to configuration registers on startup
Current boot config assembly ID attribute 0xE4 of the bus controller object	0	Instance 100: Connection-based configuration assembly Instance 130 to 139: Extended configuration assemblies

Automation Studio V4.3 or later can be used to configure the EtherNet/IP bus controller and connected I/O modules.

It creates a full configuration in the form of binary files that is downloaded to the bus controller either by a program using "Explicit messaging", assembly object, class 0x4 or manually via the integrated Web server. Downloading the configuration via the controller or program is advantageous since it is possible to replace the bus controller during service without having to re-transfer the configuration manually.

In addition, the position of the I/O data points (I/O mapping) is written to the respective I/O assemblies in a text file.

Automation Studio creates a complete project for the RSLogix programming environment from Allen-Bradley. It contains not only the configuration data for the bus controller and I/O modules, but the I/O data point mapping for the I/O data as well. Allen-Bradley controllers support the automatic transfer of configuration data while the I/O connection is being established. This uses the configuration assembly, assembly object, class 0x4, instance 100. This requires the configuration data to have a maximum size of 400 bytes and the bus controller parameter "Configuration assembly type" to be set to "Connection-based (default) configuration assembly " in Automation Studio.

For more details, see the Automation Studio help system.

8.1 Automatic configuration

If no valid data is available or enabled in the configuration assemblies during startup, then the connected I/O modules are configured automatically. In this case, the "Current boot config assembly ID" attribute of the bus controller object class 0x64 receives the value 0.

With automatic configuration, each module is operated in function model 254 ("Bus controller").

During startup, each module reports the length of the synchronous input and output registers. The bus controller uses this information to create the I/O process image. Static registers are registered by the bus controller as they are reported by the module; dynamic registers are mapped automatically by the bus controller on the X2X Link bus.

> **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 27.)**

8.1.1 Configuration of multi-function modules

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

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

8.1.2 Empty module slots

If bus modules on the X2X Link bus are empty or using node number switches (e.g. X20BM15), then the subsequent I/O modules are not started. They remain in mode PREOPERATIONAL, and attribute 0xFD of the respective I/O instance has Module status 0x50 or 0x70.

> **Information:**
>
> **With an automatic configuration, I/O modules after blank slots – i.e. those with higher X2X station numbers – are not started!**

8.2 Manual configuration (full configuration)

In a full configuration, the bus controller configures the I/O modules using the data in the Configuration assembly created by Automation Studio, for example. No information is requested from the modules regarding registers.

If the configuration data does not match the I/O modules that are actually present, then this will be indicated on both the bus controller as well as the affected I/O modules. In addition, the bus controller will also enter an error state, which may triggered certain preconfigured responses such as setting output states.

An error is reported if an I/O module is missing or the hardware ID is different. See "LED status indicators" on page 10 or "Adapter status" on page 46.

With a full configuration, I/O modules following one or more faulty I/O modules will also be started. These modules have a higher X2X node number and – as long as other errors do not occur – I/O module status 0x52 ("Operational").

8.2.1 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 27.
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).

9 Assembly object and bus controller process image

Individual I/O modules are divided up into different instances (assemblies) in the "assembly object" on page 40, class 0x4 depending on the data type.

These base assemblies contain the data points of the I/O modules in order, i.e. in increasing order of the X2X Link station number. The input/output modules are counted from left to right (with the module further to the left having the lowest number).

If additional I/O modules are added during operation, the process image is updated automatically without changing the parameters of individual assemblies, i.e. "attributes 0x40 to 0x46" on page 49 of the bus controller object. "Attributes 0x20 to 0x27" on page 47 of the bus controller object are updated in this case.

9.1 Static assemblies

The following tables provide an overview of the individual assemblies.

Output assemblies			
Instance	**Type**	**Description**	**Default size in bytes**
110, 0x6E	Base	Analog output (AO)	120
111, 0x6F	Base	Digital output (DO)	120
112, 0x70	Combination	Analog + digital output (AO + DO)	240

Input assemblies			
Instance	**Type**	**Description**	**Default size in bytes**
120, 0x78	Base	Analog input (AI)	120
121, 0x79	Base	Digital input (DI)	120
122, 0x7A	Base	Network status (NS)	120
123, 0x7B	Base	Output status (OS)	120
124, 0x7C	Combination	AI + DI + NS + OS[1]	480

1 The composition can be configured using attribute 0x46 "Composition of the combination input assembly" on page 50 of the bus controller object.

Analog assemblies

All analog or more complex I/O registers are mapped in the analog base assemblies:

- Analog output (instance 110 = 0x6E)
- Analog input (instance 120 = 0x78)

It is important to keep in mind that EtherNet/IP uses little-endian format, which means that the least significant byte is mapped first.

Example

The integer value 0x1234 is transferred as 0x34 12.

Digital assemblies

With digital assemblies, the first digital data point of each module is assigned to a new unused byte (bit offset = 0).

> **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.**

The following digital base assemblies are used:

- Digital output (instance 111 = 0x6F)
- Digital input (instance 121 = 0x79)

Input assemblies

The following input assemblies exist in addition to the analog and digital I/O data:

- X2X Link network status (instance 122 = 0x7A)
- Output status (instance 123 = 0x7B)

The network status assembly returns one byte of status information for each X2X Link station. With the X20 system, this is the bus module (e.g. X20BM11); with the X67, this is the ASIC component integrated in the I/O module. The status of the output channels that possess feedback is available as input data on the status assembly output.

Combination assemblies

In addition to the 6 base assemblies, there are 2 combination assemblies that combine the data of the respective base assemblies. Accordingly, their length results from the sum of the lengths of the individual base assemblies. Combination assemblies are normally used for I/O communication (data connections, class 1 connection):

- Combination output assembly, combined output (instance 112 = 0x70)
- Combination input assembly, combined input (instance 124 = 0x7C)

Information:

Attributes 0x40 to 0x46 of the bus controller object can be used to read or set the lengths of the individual assemblies and configure how to put together the combination input assembly.

9.2 Non-exclusive input assemblies

Input and output assemblies as well as configuration assemblies are *exclusive owner* assemblies, i.e. only one active connection at a time can be established to the corresponding assembly. Since combination assemblies only represent references to the base assembly I/O data, however, shared access to the same I/O data is theoretically possible. When using this type of combination, suitable measures must be included in the program to manage this shared I/O data access.

In contrast to *exclusive owner* assembly connections, the input or *listen only* assemblies allow read access to an existing *exclusive owner* connection.

The difference between the two access types is determined by the dependence on the corresponding *exclusive owner* assembly. All *Listen Only* connections are automatically interrupted if the corresponding *Exclusive Owner* connection has ended.

A typical use case would be when more than one controller (scanner) should access a single adapter. Only the scanner with the *Exclusive Owner* connection can set outputs. On the other scanners, either the *Input Only* instance (254) or the *Listen Only* instance (255) must be specified as a consuming assembly instead of an output assembly. The size of the consuming assembly is 0 bytes. (Source → Destination connection point)

The assembly of the respective exclusive owner connection must be configured as the input assembly. (generating assembly or Destination → Source connection point)

It is also important that the requested packet interval (RPI) of the "Source → Destination" connection is the same as that of the *exclusive owner* connection. The RPI of the "Source → Destination" connection is irrelevant.

9.2.1 Static input only assembly

Connection	Instance	Type	Description	Default size in bytes
U → Z	254, 0xFE	Base	Output to destination device (consuming assembly)	0 bytes
Z → U	Respective *Exclusive Owner* ID	Depends on the *Exclusive Owner* connection type	Input from destination device (producing assembly)	The assembly size is based on the *Exclusive Owner* connection

9.2.2 Static listen only assembly

Connection	Instance	Type	Description	Default size in bytes
U → Z	255, 0xFF	Base	Output to destination device (consuming assembly)	0 bytes
Z → U	Respective *Exclusive Owner* ID	Depends on the *Exclusive Owner* connection type	Input from destination device (producing assembly)	The assembly size is based on the *Exclusive Owner* connection

9.3 X2X Link network status assembly

The X2X Link network status provides information about the operating state of individual X2X Link stations (i.e. the bus modules on each I/O module). The operating status of the I/O modules (the electronics module, not the bus modules) can be queried using attribute 0xFD Module status of the respective instance of the I/O module object.

Each module or station on the X2X Link bus takes up one byte in the X2X network status assembly. In the case of the X20BC0088 bus controller, the first byte belongs to the power supply module for X2X station number 1.

Each X2X Link station is equipped with a hardware component (ASIC) that reports its status to the X2X Link scanner (master) during every X2X Link cycle. In this case, this is the bus controller.

Each network status byte is structured as follows:

Bit	Value	
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 (void)
4	0x10	Reserved (always 1)
5	0x20	Reserved (always 1)
6	0x40	Reserved (always 1)
7	0x80	Reserved (always 1)

This results in the following values:

Description	Value (hex)	Value (binary)
X2X Link station inactive (e.g. no X2X Link power supply)	0x00	0000 0000
No communication with the electronics module (bits 7 to 3 invalid)	0x01	0000 0001
Everything OK (I/O data valid)	0xF5	1111 0101
No communication with the electronics module (bits 7 to 3 invalid), identical to 0x01	0xF9	1111 1001
I/O data valid, 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.)	0xFD	1111 1101

9.4 Output status assembly

The status of the digital output channels that possess feedback is available as input data on the status assembly output. Examples also include digital output modules that indicate the status "Short circuit or overload" for each channel using a set bit.

Some analog modules also have input registers that can be used for status feedback. Examples include analog input modules that have registers with names like "StatusInput". In these cases, 2 bits per channel indicate whether an open circuit or measurement range violation has occurred.

Details about these special input registers for status feedback can be found in the respective module description.

When using Automation Studio to make the configuration, analog registers are mapped in the analog input assembly instead of in the analog output status assembly. Detailed mapping information is available in the text file created by Automation Studio. If the bus controller is used in automatic configuration mode, the registers are mapped to the output status assembly as described above.

9.5 Changes to I/O assemblies

If attributes 0x40 to 0x46 of the bus controller object (class 0x64) are used to make changes to the size or grouping of one or more I/O assemblies, then the bus controller service 0x36 must be subsequently called to re-initialize the assemblies.

If a class 1 I/O connection is active, these changes cannot be made; the error *Permission/Privilege check failed* will be returned instead.

Information:

The size of the assembly configured in the scan list on the scanner must match exactly the size of the settings on the bus controller. If this is not the case, an I/O connection cannot be established and *CIP forward open error* 0x315 will be reported.

9.6 Configuration assembly

Configuration data for the bus controller and its connected I/O modules can be stored in configuration assemblies.

If no configuration data is stored, then the bus controller and I/O modules use the default settings of the "Bus controller" function model for the input and output lengths, for example. These default settings can be changed at runtime using "explicit messaging". Selecting another I/O function model at runtime is not possible. For additional details about configuration changes at runtime, see "Automatic configuration" on page 27.

The configuration data for the EtherNet/IP bus controller is created with Automation Studio.

9.6.1 Connection-based (default) configuration assembly

Setting "Connection-based configuration assembly" for parameter "Configuration assembly type" in Automation Studio only generates one binary file (e.g. "EthernetIP_CPU1_IF1_ST1_std.bin") with a fixed size of 400 byte.

A configuration assembly with this size can be transferred automatically from Allen-Bradley systems to the adapter or bus controller when the connection is being established. It is also possible to download this file to the controller manually, from the application or via the Web server integrated in the bus controller by selecting "Configuration download" from the menu.

Instance	Type	Description	Size in bytes
100, 0x64	Base	Connection-based (default) configuration assembly. Used to configure the bus controller and connected I/O modules	400

9.6.2 Advanced configuration assemblies

Setting "Extended configuration assemblies" for parameter "Configuration assembly type" in Automation Studio is used to generate "extended" configuration data for large configurations (data > 400 bytes). The binary file generated by Automation Studio, e.g. "EthernetIP_CPU1_IF1_ST1_ext.bin" with a size of 4500 bytes contains the entire configuration and can be downloaded to the device via menu option "Configuration download" of the Web server integrated in the bus controller.

Instance	Type	Description	Size in bytes
130, 0x82	Base	1. Extended configuration assembly	450
131, 0x83	Base	2. Extended configuration assembly	450
...	...	...	...
138, 0x8A	Base	9. Extended configuration assembly	450
139, 0x8B	Base	10. Extended configuration assembly	450

It is also possible to download individual configuration files to the controller from the application using "explicit messages" on the controller. The files with 450 bytes are used for this, e.g. EthernetIP_CPU1_IF1_ST1_ext_0.bin, EthernetIP_CPU1_IF1_ST1_ext_1.bin, etc.

The binary file with 4500 bytes fills all 10 extended configuration assemblies (130 to 139). All existing configuration data in the extended assemblies is overwritten when transferring to the bus controller.

If more than one configuration should be stored in the B&R bus controller, this can be done using the "Advanced configuration" menu of the web server or "explicit messaging" in the application on the controller. In this case, the individual files, e.g. EthernetIP_CPU1_IF1_ST1_ext_0.bin, EthernetIP_CPU1_IF1_ST1_ext_1.bin, can be stored in the 10 extended configuration assemblies (instances 130 to 139 or 0x82 to 0x8B).

If a configuration is larger than 450 bytes, then the configuration block consists of the necessary amount of 450-byte files. These files must be stored consecutively.

Example

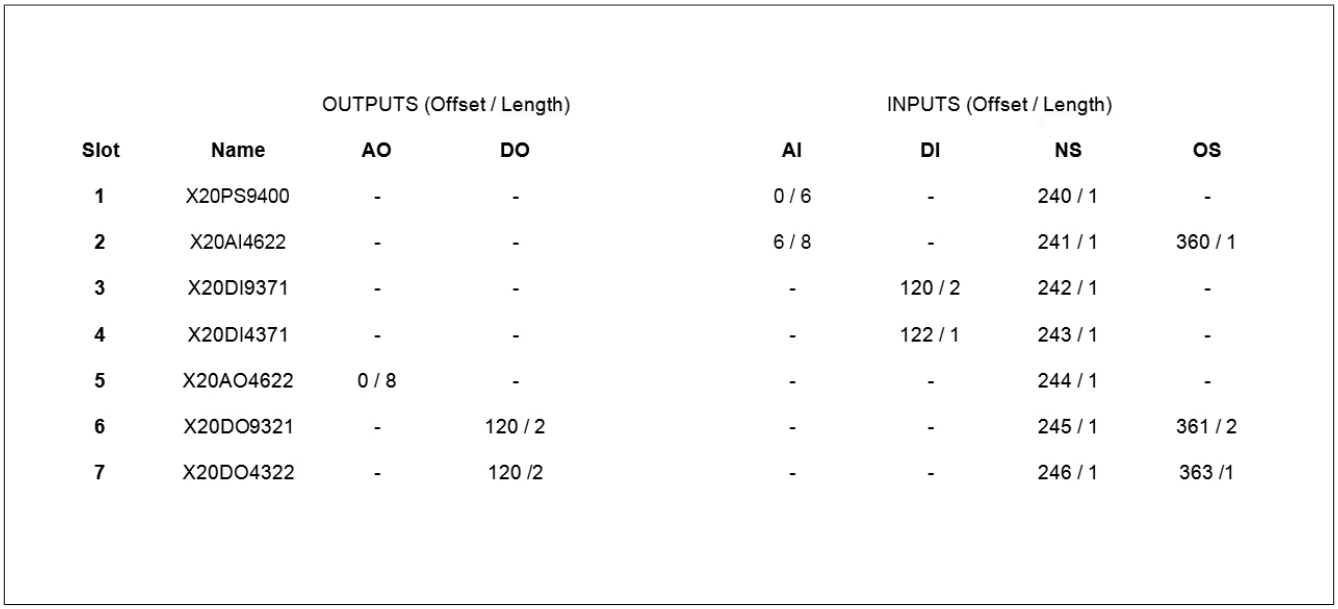
A configuration with a size of 600 bytes consists of the files EthernetIP_CPU1_IF1_ST1_ext_0.bin and EthernetIP_CPU1_IF1_ST1_ext_1.bin. The first file is stored in the third extended configuration assembly (instance 132); the second is stored in the fourth (instance 133) since the first two extended configuration assemblies are already occupied by other data.

The desired configuration can be started by executing service 0x37 class 0x64. The start assembly ID must be passed to the service as a UINT parameter. 0x0084 must be used as the parameter value in the example above.

9.7 Example of a process image

Module name	Module type	Output	Input
X20PS9400	Power supply module		3 analog channels (6 bytes AI)
X20AI4622	Analog input module		4 analog channels (8 bytes AI) 1-byte status register (OS)
X20DI9371	Digital inputs		12 digital channels (2 bytes DI)
X20DI4371	Digital inputs		4 digital channels (1 byte DI)
X20AO4622	Analog outputs	4 analog channels (8 bytes AO)	
X20DO9321	Digital outputs	12 digital channels (2 byte DO)	2-byte status register (OS)
X20DO4322	Digital outputs	4 digital channels (1 byte DO)	1-byte status register (OS)

The integrated Web server in the bus controller can be used to view the lengths and offsets of individual I/O module data types in both combination assemblies by selecting "I/O assembly mapping" from the menu:



		OUTPUTS (Offset / Length)		INPUTS (Offset / Length)			
Slot	**Name**	**AO**	**DO**	**AI**	**DI**	**NS**	**OS**
1	X20PS9400	-	-	0 / 6	-	240 / 1	-
2	X20AI4622	-	-	6 / 8	-	241 / 1	360 / 1
3	X20DI9371	-	-	-	120 / 2	242 / 1	-
4	X20DI4371	-	-	-	122 / 1	243 / 1	-
5	X20AO4622	0 / 8	-	-	-	244 / 1	-
6	X20DO9321	-	120 / 2	-	-	245 / 1	361 / 2
7	X20DO4322	-	120 /2	-	-	246 / 1	363 /1

Figure 2: Web server I/O assembly mapping

In this image, the value before the slash specifies the byte offset in the output or input combination assemblies; the value after the slash specifies the length of the respective data in bytes.

In this example, the default settings for the length of the base assemblies has been left at 120 bytes in each case. 8 bytes of analog output data (AO) for the X20AO4622 module is located at offset 0 in the output combination assembly, i.e. at the start of the data. Byte 0 is the LSB, byte 1 the MSB of the first channel. Byte 2 is the LSB of channel 2, etc. since EtherNet/IP uses the little-endian format.

112 bytes of unused data then follows in accordance with the standard length of the AO base assembly.

Byte 120 contains the digital output data (DO) for channels 1 to 8 of the X20DO9321 module; byte 121 contains channels 9 to 12 on bits 0 to 3.

Byte 122 contains the digital output data for channels 1 to 4 of the X20DO4322 module (bits 0 to 3).

The 120-byte input combination assembly first contains 6 bytes of analog input data (AI) from power supply unit X20PS9400. These are the 3 WORD channels for status, current and voltage. 8 bytes for the 4 analog inputs of the X20AI4622 follow with an offset of 6 bytes. The remaining 108 bytes are unused and return zero. The data for the digital input base assembly (DI) begins at byte 120 with a total of 3 bytes of data from both DI modules.

In the default setting, one byte with the network status for each X2X station is located at offset 240. For details, see "X2X Link network status assembly" on page 31.

The data for the output status assembly (OS) is located at offset 360. The X20AI4622 module returns 1 byte of data in this case, each with 2 bits per channel for the short circuit and overflow status. Both DO modules return 1 bit of "short circuit or overload" status information for each output channel. Bit 0 of byte 361 contains the status of channel 1 of the X20DO9321 module; bit 7 contains the status of channel 8. Bits 0 to 3 of the next byte (byte 362) contain the status of channels 9 to 12; the other bits are unused and return zero. Bits 0 to 3 of byte 363 contain the status of channels 1 to 4 of the X20DO4322 module. The remaining bits are unused.

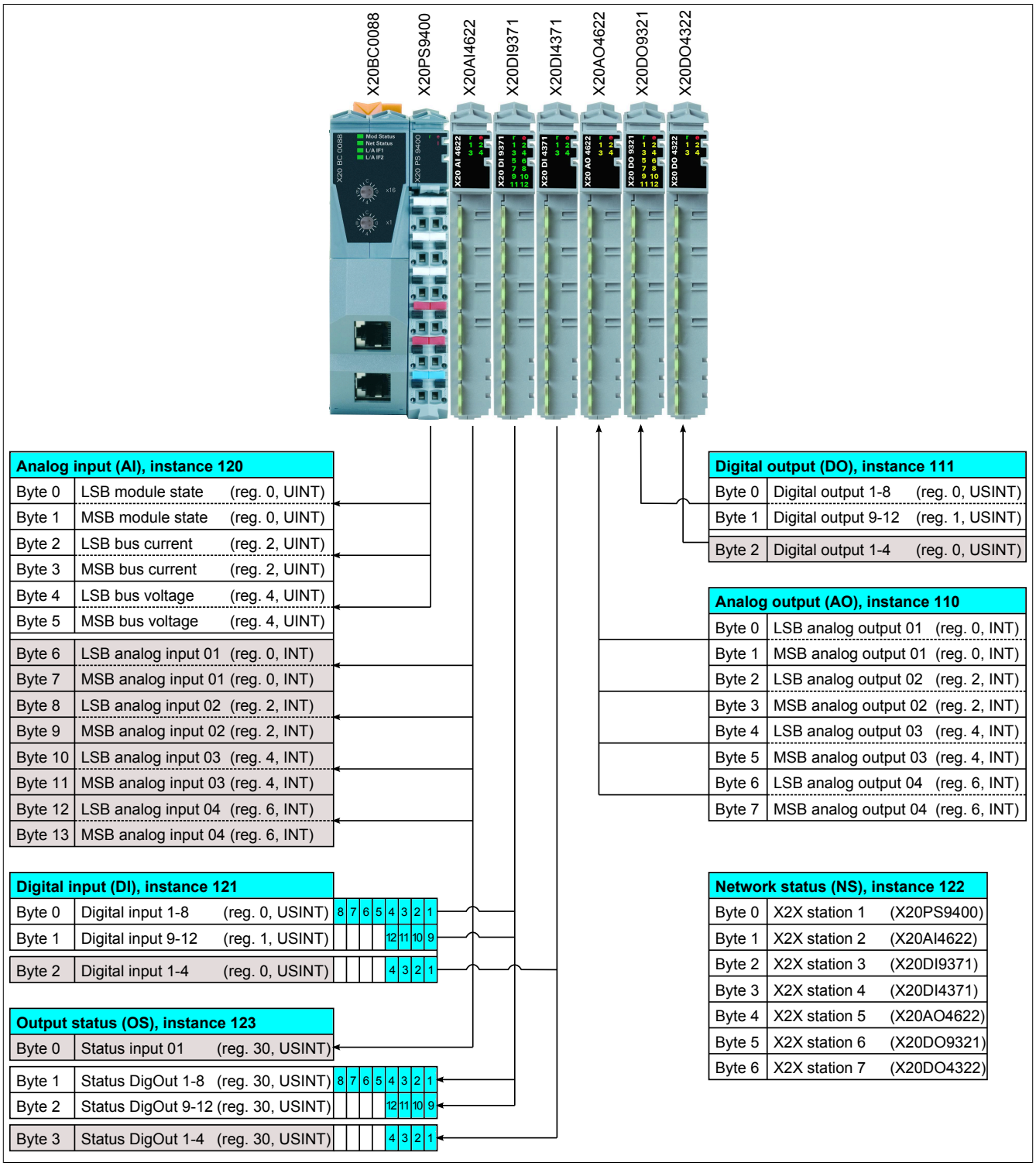


Figure 3: Overview of process image registers

10 Adapter actions

Adapter states (events) can be connected to certain actions.

The linking of the actions with the corresponding states can be programmed via the bus controller object CIP class 0x64, Actions or simply configured in the Automation Studio. Every state can be assigned an Action. The status of the bus controller can be read by Attribute Adapter status.

10.1 Available adapter states

The following states can be configured with actions:

States	Description
Communication loss	The class 1 exclusive owner connection to the scanner is lost (timeout). The network status LED "Net Status" blinks red.
Program mode	The scanner is in programming mode; the adapter idles. Nothing indicated by LED.
Module failed	I/O module removed during operation or module defective. The module status LED "Mod Status" blinks red (Major Recoverable Fault)
Module missing at power-up	Module not detected during the boot phase. The module status LED "Mod Status" blinks red (Major Recoverable Fault)
Module mismatch at power-up	Incorrect module detected during the boot phase. The module status LED "Mod Status" blinks red (Major Recoverable Fault)

10.1.1 Communication loss

This state occurs if the adapter does not detect any scanner activity within a certain amount of time (class 1 connection timeout).

This timeout state is reset by either reestablishing the connection that caused the timeout or explicitly executing the "Reset timeout" service 0x32 of the bus controller object CIP class 0x64. The desired behavior can be configured using the 0x63 attribute "Communication loss (timeout) reset mode" on page 50 of the bus controller object.

All timeout-specific parameters are defined with the *forward open* service and therefore inseparably linked to a class 1 or class 3 connection. Various timeouts can occur depending on the connection type.

The following timeouts are possible:

Type of timeout	Actions
Class 1 exclusive owner connection timeout	Displayed via network LED status indicator "Net status". Only this timeout triggers an action.
Class 1 input only connection timeout	Displayed via network LED status indicator "Net status". No timeout action is triggered.
Class 1 listen only connection timeout	Displayed via network LED status indicator "Net Status". No timeout action is triggered.
Class 3 connection timeout	Displayed via network LED status indicator "Net Status". No timeout action is triggered.

10.1.2 Program mode

The adapter goes into idle mode when a scanner with an active connection is switched to idle mode. This is usually the case if the associated controller is switched to program mode.

Idle mode can only be reset by the scanner. This state is not indicated by any LEDs.

10.1.3 Module failed

This event occurs if an I/O module is removed or identified as defective during operation.

This state can be reset by restoring the original I/O module configuration.

This state is indicated as *Major Fault Recoverable* with a red blinking module status LED "Mod Status".

10.1.4 Module missing at power-up

This state occurs if a module is not detected during the boot phase.

If no module configuration data is available, it is only possible to detect that modules are missing if there are empty slots between connected modules. If the user has linked this event with an action, then this state is indicated as *Major Fault Recoverable* with a red blinking module status LED "Mod Status" and can only be reset by rebooting.

If the event **has not** been linked to an action, e.g. with empty action *No action*, there will be no LED indication and the event will only be reflected in the Adapter status.

10.1.5 Module mismatch at power-up

This state occurs if an incorrect module is detected during the boot phase.

Incorrect module types can only be detected if module configuration data is available. This state is indicated as *Major Fault Recoverable* with a red blinking module status LED "Mod Status" and can only be reset by rebooting.

10.2 Available actions

Action type	Description	Parameter value
No action	No action is executed.	0
Set outputs to zero	All outputs are set to zero.	1
Set default	All outputs are set to their predefined values. **This feature is not currently implemented.**	2
Freeze outputs	The status of the outputs remains frozen and cannot be changed.	3
Disable new class 1 connections	The scanner can no longer establish a class 1 connection, i.e. *Forward open* is no longer permitted on the adapter. ► If an adapter state occurs that has been linked with this action (*Disable forward open*), then an automatic adapter configuration is no longer possible via *Config Assembly*. The source of the adapter error must be cleared and a restart must be performed before this state can be exited. If the cause of error cannot be eliminated, there is an option to boot the adapter with its default settings. With the default setting, all actions are set to *Set outputs to zero*. The scanner can then establish a class 1 connection and download all configuration data from the adapters. This data is stored automatically to flash memory, which causes the adapter to revert to its original state.	4

10.3 Action hierarchy

If several adapter states occur at the same time, a predefined hierarchy determines which action is carried out first. The order of actions is defined as follows:

Priority	Action
1	Module mismatch
2	Module missing
3	Communication loss (timeout)
4	Module failed
5	Program mode (idle)

10.3.1 Practical example: Hierarchy of actions

First, to clear up terminology: An event refers to something that causes a change in the adapter's state.

During operation, the *Module failed* event occurs. The *Freeze Outputs* action, which in this example is connected with the *Module Failed* event, is executed (i.e. the output data of all I/O modules is "frozen" in the present state).

While state *Module failed*is maintained, another event occurs: *Communication loss* (timeout). Since the *Communication loss* event has a higher priority than *Module failed*, the action associated with the *Communication loss* event is executed. In our example, this is the *Set outputs to zero* action. All output data is set to 0.

Both adapter states are retained, i.e. the adapter is currently in the *Communication loss* + *Module failed* state.

After the connection lost due to timeout is reestablished, the *Communication loss* state is reset. If the scanner now attempts to set outputs using the re-established connection, this action cannot be executed because the state *Module Failed* is still active and all outputs are blocked with the action *Freeze Outputs*.

The scanner can only update the I/O modules' output data after the state *Module Failed* has been corrected by the user.

10.4 Action scope

In general, all actions apply "globally to the adapter", i.e. actions affect the output data of all I/O modules. Nevertheless, the actions linked to the *Communication loss* (timeout) and *Program mode (idle)* events can be configured for a local scope as well. This would cause actions to only affect the output data that is connected with the assemblies on which the events *Communication Loss* or *Program Mode* have occurred.

Example

2 exclusive I/O connections have been configured. The first only uses analog I/O assemblies; the other uses digital I/O assemblies. If a timeout or idle event occurs in the I/O connection managing the digital I/O points, then only the digital outputs will be affected if the action has been configured for a local scope. With "Adapter global", both the analog as well as the digital outputs would be affected.

The two attributes 0x62 "Communication loss (timeout) scope" and 0x65 "Program mode (idle) scope" of bus controller object class 0x64, instance 1 or the corresponding parameters in Automation Studio under "Adapter configuration" are used to configure the action scope.

11 Supported CIP objects

11.1 Class attributes

A class attribute refers to the class as a whole, not to a particular instance.

Each of the CIP objects described uses one or more of the following class attributes. Please refer to the overview of the respective CIP object.

Attribute ID [hex]	Access	Data type	Description
0x1	Get	UINT	Object revision
0x2	Get	UINT	Largest instance number of the object generated in this class level
0x3	Get	UINT	Number of generated instances
0x4	Get	STRUCT of	
		UINT	Number of optional attributes
		ARRAY of UINT	List of optional attribute numbers
0x5	Get	STRUCT of	
		UINT	Number of optional services
		ARRAY of UINT	List of optional service codes
0x6	Get	UDINT	Largest possible ID number of class attributes
0x7	Get	UDINT	Largest possible ID number of instance attributes

11.2 Identity object

(CIP class 0x1)

This object identifies and provides general information about the bus controller. There is only one instance of the *identity object*.

Class attributes	1, 2, 3, 6, 7
Class services	0x1, 0xE
Instance attributes	1, 2, 3, 4, 5, 6, 7, 8
Instance services	0x1, 0x5, 0xE

11.2.1 Instance attributes

Attribute ID [hex]	Access	Data type	Description	Function
0x1	Get	UINT	Vendor ID	**B&R code:** 0x0377 (887)
0x2	Get	UINT	Device type	**Default value:** 0x000C (12) = Adapter device
0x3	Get	UINT	Product code	Hardware ID of the bus controller. This is identical to the first 4 characters printed on the module's housing. **Bus controller ID:** 0x26D8 (9944)
0x4	Get	STRUCT of:		0xZZYY
		USINT	Major revision	ZZ = Firmware major revision
		USINT	Minor revision	YY = Firmware minor revision
0x5	Get	WORD	Status	See "Table for "Status", attribute 5" on page 39.
0x6	Get	UDINT	Serial number	Identical to the seven characters on the module's housing after the hardware ID
0x7	Get	SHORTSTRING	Product name	0x1C (length = 28) and hexadecimal equivalent of "B&R I/O controller X20BC0088"
0x8	Get	USINT	Status	0 = Does not exist 1 = Device self-test 2 = Standby 3 = Operational 4 = Major recoverable fault 5 = Major unrecoverable fault 6 to 254 = Reserved 0x03

Table for "Status", attribute 5

Bit	Name	Function
0	Owned	TRUE indicates that there is an owner for the device or object within the device. In the master/slave paradigm, setting this bit means that a master has been assigned to a predefined master/slave connection set.
1		Reserved, should be 0
2	Configured	TRUE indicates that the firmware is no longer set to the preconfigured default behavior. This does not include how communication is configured.
3		Reserved, should be 0
4 - 7	Extended device status (EDS)	The extended device status indicates whether the device is following the general definition for using this bit through the use of the DeviceStatusAssembly keyword in the [Device] section of the EDS file. For the meaning of individual bits, see "Extended status description" on page 39.
8	Minor recoverable fault	TRUE indicates that the device has self-diagnosed a problem that appears to be recoverable. This problem does not cause the device to enter into an error state.
9	Minor unrecoverable fault	TRUE indicates that the device has self-diagnosed a problem that does not appear to be recoverable. This problem does not cause the device to enter into an error state.
10	Major recoverable fault	TRUE indicates that the device has self-diagnosed a problem that forces it to enter the "Major recoverable fault" error state.
11	Major unrecoverable fault	TRUE indicates that the device has self-diagnosed a problem that forces it to enter the "Major unrecoverable fault" error state.
12 - 15		Reserved, should be 0

Extended status description

Bits 4 - 7	Extended status description of the device
0000	Self-test or unknown
0001	Firmware update in progress
0010	At least one faulty I/O connection
0011	No I/O connection established
0100	Faulty nonvolatile configuration
0101	Significant error: Either bit 10 or 11 set to TRUE
0110	At least one I/O connection in RUN mode
0111	At least one I/O connection established, all in idle mode
1000-1001	Reserved, should be 0
1010-1111	Vendor- or product-specific

11.2.2 Identity service object

Service code (hex)	Supported by	Service name	Description
0x1	Class/Instance	Get_Attribute_All	Returns a predefined list of object attributes
0x5	Instance	Reset	Executes the reset service for the device. This service has a USINT parameter called "Type". 0 = Restart (default) 1 = Resets to factory default configuration with subsequent restart 2 = Resets to factory default configuration except for communication connection parameters, subsequent restart 3 to 255 = Reserved **Resetting to the factory default configuration is only valid until the next restart! Parameters in flash memory are not overwritten by the service. In order to permanently delete stored parameters, see "B&R-specific service 0x35 of class 0x64" on page 55**
0xE	Class/Instance	Get_Attribute_Single	Supplies the contents of the corresponding attributes

11.3 Message router object

(CIP class 0x2)

The message router object provides a message connection that can be used by a client to execute any class or instance services within a physical device.

Class attributes	1, 2, 3, 4, 5, 6, 7
Class services	0x1, 0xE
Instance attributes	1, 2, 3 (firmware version 3.07 and later)
Instance services	0x1, 0xE

11.3.1 Instance attributes

Attribute ID [hex]	Access	Data type	Description
0x1	Get	STRUCT of	List of all supported objects
		UINT	Number of supported classes in the class array
		ARRAY of UINT	List of all supported class codes
0x2	Get	UINT	Maximum number of supported connections

11.3.2 Service objects

Service code (hex)	Supported by	Service name	Description
0x1	Class/Instance	Get_Attributes_All	Returns a list of object attributes
0xE	Class/Instance	Get_Attribute_Single	Supplies the contents of the corresponding attribute

11.4 Assembly object

(CIP class 0x4)

The assembly object unifies the attributes of various objects. This allows data for each object to be transmitted or received using a single connection. The assembly object can be used for input and output data. The direction is considered from the network's point of view. Input data refers to sending data to the network while output data refers to data received from the network.

Class attributes	1, 2, 3, 6, 7
Class services	0x1, 0xE
Instance attributes	3, 4 (firmware version 3.07 and later)
Instance services	0xE, 0x10

11.4.1 Instance attributes

Attribute ID [hex]	Access	Data type	Description
0x3	Set	ARRAY of Byte	Data

11.4.2 Service objects

Service code (hex)	Supported by	Service name	Description
0xE	Class/Instance	Get_Attribute_Single	Supplies the contents of the corresponding attribute
0x10	Instance	Set_Attribute_ Single	Modifies the value of an attribute

11.5 Connection manager object

(CIP class 0x6)

This object is used for connected and connectionless communication. Communication can then also take place via various subnetworks.

Class attributes	1, 2, 3, 4, 6, 7
Class services	0x1, 0xE
Instance attributes	1, 2, 3, 4, 5, 6, 7, 8, 11 (firmware version 3.07 and later)
Instance services	0x1, 0xE, 0x4E, 0x52, 0x54

11.5.1 Instance attributes

Attribute ID [hex]	Access	Data type	Description
0x1	Set[1)]	UINT	Number of received forward open service requests.
0x2	Set[1)]	UINT	Number of forward open service requests that were rejected due to an invalid format.
0x3	Set[1)]	UINT	Number of forward open service requests that were rejected due to insufficient resources.
0x4	Set[1)]	UINT	Number of forward open service requests that were not rejected due to an invalid format or insufficient resources.
0x5	Set[1)]	UINT	Number of received forward close service requests.
0x6	Set[1)]	UINT	Number of forward close service requests that were rejected due to an invalid format.
0x7	Set[1)]	UINT	Number of forward open service requests that were not rejected due to an invalid format.
0x8	Set[1)]	UINT	Total number of all connection timeouts that occurred in connections controlled in this connection manager.

1) A device can refuse the Set request to this attribute if the transmitted attribute value is not zero. In this case, it transmits the general status code 0x09 (invalid attribute value).

11.5.2 Service objects

Service code (hex)	Supported by	Service name	Description
0x1	Class/Instance	Get_Attributes_All	Returns a predefined list of object attributes
0xE	Class/Instance	Get_Attribute_Single	Returns the contents of the corresponding attributes
0x4E	Instance	Forward_Close	Closes a connection
0x52	Instance	Unconnected_Send	Unconnected transmit service
0x54	Instance	Forward_Open	Opens a connection. Maximum data size: 511 bytes

11.6 Port object

(CIP class 0xF4)

The port object defines the CIP connections present on the device.

Class attributes	1, 2, 3, 6, 7
Extended class attributes	8, 9
Class services	0x1, 0xE
Instance attributes	1, 2, 3, 4, 7, 10, 11 (firmware version 3.07 and later)
Instance services	0x1, 0xE

11.6.1 Extended class attributes

Attribute ID [hex]	Access	Data type	Description	Default value
0x8	Get	UINT	Instance of the connection object that describes the connection object and was used to send this request to the device	
0x9	Get	ARRAY of STRUCT of	Array of structures that contain the instance attributes 1 and 2 for each instance	[1)]
		UINT	Enumerates the connection types	See "instance attribute 1" on page 42
		UINT	CIP connection number that is linked to this connection.	See "instance attribute 2" on page 42

1) The index of the array is determined by the instance number, i.e. from 1 to the maximum number of instances. The value at index 1 (offset 0) and non-instanced instances should be 0.

11.6.2 Instance attributes

Attribute ID [hex]	Access	Data type	Description	Function
0x1	Get	UINT	Enumerates the possible connection types. All connection types with the exception of 0 indicate routing-capable connections of the corresponding type.	Values for the connection types: 0 Connection does not support CIP routing. Attribute 2 is ignored. 1 Reserved to preserve compatibility with existing protocols 2 ControlNet 3 ControlNet redundant 4 EtherNet/IP 5 DeviceNet 6 - 99 Reserved 100 - 199 Manufacturer-specific 200 CompoNet 201 Modbus TCP 202 Modbus SL 203 SERCOS III 204 - 65534 Reserved 65535 Not configured
0x2	Get	UINT	CIP connection number that is linked to this connection. This attribute is ignored if the connection type = 0.	The manufacturer assigns a unique number for each communication port. Value 1 is reserved for internal use (e.g. backplane). Value 0 is reserved and is not permitted to be used.
0x3	Get	STRUCT of		
		UINT	Number of 16-bit words in the following path	Range = 2 to 6
		Padded EPATH	Logical path segment that identifies the object for this port	The path should consist of one logical class segment and one logical instance segment. The maximum size is 12 bytes.
0x4	Get	SHORT_STRING	Name of the physical network port	"Port A", for example. The maximum number of characters is 64. This name must be different for each physical port. If several CIP ports use the same physical port, the same physical name must be used.
0x7	Get	Padded EPATH	Node number switch value of the device	The switch value should be identical to attribute 2.

11.6.3 Service objects

Service code (hex)	Supported by	Service name	Description
0x1	Class/Instance	Get_Attributes_All	Returns a predefined list of object attributes
0xE	Class/Instance	Get_Attribute_Single	Supplies the contents of the corresponding attribute

11.7 TCP/IP interface object

(CIP class 0xF5)

The TCP/IP interface object provides a mechanism for configuring the bus controller's TCP/IP network interface. Examples include configurable components such as the IP address, network mask, gateway address and hostname of the device.

The B&R EtherNet/IP bus controller supports only one instance of the TCP/IP interface object since there is only a single IEEE 802.3 communication interface. The Ethernet link object class 0xF6 provides attributes for accessing the built-in 3-port switch.

Class attributes	1, 2, 3, 6, 7
Class services	0x1, 0xE
Instance attributes	1, 2, 3, 4, 5, 6, 8, 9, 13, 16, 17 (firmware version 3.07 and later)
Instance services	0x1, 0x2, 0xE, 0x10

11.7.1 Instance attributes

Attribute ID (hex)	Access	Data type	Description	Function
0x1	Get	DWORD	Interface status	See "Interface status, attribute 1" on page 43. **Default value:** 0x00000002
0x2	Get	DWORD	Interface configuration properties	**Bit** 0: BOOTP client (False) 1: DNS client (False) 2: DHCP client (True) 3: DHCP DNS update (True = Device can transmit its hostname on DHCP request) 4: Configuration settable (True = Interface control flags are settable, see attribute 3) 5 to 31: Reserved **Default value:** 0x0000001C
0x3	Get / Set	DWORD	Interface control flags	Startup configuration **Bit** 0: As saved in flash memory (default) 1: By BOOTP 2: By DHCP 3 to 15: Reserved
0x4	Get	STRUCT of	Path to the physically linked object	Identifies the object that is connected to the underlying physical communication object
		UINT	Length of the path	Number of 16-bit words in the path **Default value:** 0x0002
		Padded EPATH	Logical segments that identify the physical link	Path addresses to the internal port of the built-in 3-port switch **Default value:** Class = 0xF6, instance = 3
0x5	Get / Set	STRUCT of	TCP/IP network interface configuration	Contains TCP/IP configuration parameters In order to avoid incomplete or incompatible configurations, parameters cannot be set individually. The user should first retrieve the attribute, change the desired values and then write back the attribute.
		UDINT	IP address	This value is 0 if an IP address has not been configured. Otherwise, a valid class A, B or C address should be written. **Default value:** 0x0164A8C0 (corresponds to 192.168.100.1)
		UDINT	Network mask	This value is 0 if a network mask has not been configured. **Default value:** 0x00FFFFFF (corresponds to 255.255.255.0)
		UDINT	Gateway address	This value is 0 if a gateway address has not been configured. Otherwise, a valid class A, B or C address should be written. **Default value:** 0xFE64A8C0 (corresponds to 192.168.100.254)
		UDINT	Primary name server	The value 0 indicates that a primary name server has not been configured. Otherwise, a valid class A, B or C address should be written. **Default value:** 0x00000000
		UDINT	Secondary name server	The value 0 indicates that a secondary name server has not been configured. Otherwise, a valid class A, B or C address should be written. **Default value:** 0x00000000
		STRING	Domain name	ASCII characters. The maximum length is 48 characters. They should be filled to an even number (filler byte not included in the length). The length is 0 if a domain name has not been configured. **Default value:** 0x000 (length = 0, empty string)
0x6	Get / Set	STRING	Hostname	ASCII characters. The maximum length is 64 characters. It should be filled to an even number (filler byte not included in the length). The length is 0 if a hostname has not been configured. **Default value:** 0x000E (length = 14) + hex equivalence of "br +MAC (2+12 characters)

Interface status, attribute 1

Bit	Name	Description
0 - 3	Interface configuration status	Indicates the status of the interface configuration attribute 0: The interface configuration attribute has not yet been configured. 1: The interface configuration attribute has received a valid configuration of BOOTP, DHCP or nonvolatile memory. 2: The interface configuration attribute has received a valid configuration of hardware settings such as a thumbwheel switch, handwheel, etc. 3 to 15: Reserved for future use
4	Mcast pending	Indicates a pending configuration change in the TTL value and/or Mcast configuration attribute This bit should be set if either the TTL value or Mcast configuration attribute is set and should be deleted the next time the device is started.
5 - 31	Reserved	Reserved for future use and should be set to 0

11.7.2 Services of the TCP/IP object

Service code (hex)	Supported by	Service name	Description
0x1	Class/Instance	Get_Attribute_All	Returns a predefined list for this object attribute.
0x2	Instance	Set_Attribute_All	Modifies all modifiable attributes.
0xE	Class/Instance	Get_Attribute_Single	Returns the contents of the corresponding attributes.
0x10	Instance	Set_Attribute_Single	Modifies a single attribute value.

11.8 Ethernet link object

(CIP class 0xF6)

The Ethernet link object manages connection-specific counters and status information for the IEEE 802.3 interface. 3 instances (IF1, IF2 and internal) are provided for the bus controller.

Class attributes	1, 2, 3, 6, 7
Class services	0x1, 0xE
Instance attributes	1, 2, 3 4, 5, 6, 7, 8, 10, 11 (firmware version 3.07 and later)
Instance services	0x1, 0xE

11.8.1 Instance attributes

Attribute ID [hex]	Access	Data type	Description	Function
0x1	Get	UDINT	Transfer rate currently in use	Speed in Mbps (e.g. 0, 10, 100, 1000, etc.)
0x2	Get	DWORD	Interface status flag	See "Interface status flag" on page 44.
0x3	Get	ARRAY [0..5] of USINT	Physical address	Device-specific MAC address

Interface status flag

The interface status flag contains information about the status and configuration of the device.

Bit	Name	Function
0	Connection status	Indicates whether the interface is connected to an active network 0 No connection to a network exists 1 Connection to a network exists
1	Full-/Half-duplex	Indicates the duplex mode currently in use 0 Half-duplex mode 1 Full-duplex mode **This flag is undefined if the connection status flag is 0.**
2 - 4	Autonegotiation	Indicates the current status of autonegotiation 0 Autonegotiation in progress 1 to 2 Not supported by the bus controller. The built-in switch continues attempting to establish a connection by trying out the different connection modes. 3 Duplex mode and speed successfully determined 4 No autonegotiation. The duplex mode and speed are set manually.
5		0 The interface can apply changes to link parameters automatically (autonegotiation, duplex mode, transfer rate). 1 Changes are applied only after the "Reset" service is executed on the identity object.
6		2 The interface did not find any local hardware errors. 1 The interface found a local hardware error (e.g. no transceiver).
7 - 31		Reserved

11.8.2 Service objects

Service code (hex)	Supported by	Service name	Description
0x1	Class/Instance	Get_Attributes_All	Returns a predefined list of object attributes
0xE	Class/Instance	Get_Attribute_Single	Supplies the contents of the corresponding attribute

12 B&R-specific objects

12.1 Bus controller object

(CIP class 0x64)

The bus controller object can be used to configure all global bus controller parameters. All I/O module parameters are managed individually using the "I/O module object " on page 56.

12.1.1 Class attributes

Attribute ID (hex)	Access	Data type	Description	Default value
0x1	Get	UINT	Revision	0x0001
0x2	Get	UINT	Maximum number of instances	0x0001

12.1.2 Instance attributes

Attribute ID (hex)	Access	Data type	Description	Group
0x1	Get	UDINT	Adapter status	Product and bus controller status
0x2	Get	UINT	Hardware major revision	
0x3	Get	UINT	Hardware minor revision	
0x4	Get	UINT	FPGA hardware revision	
0x5	Get	UINT	Active boot block	
0x6	Get	UINT	Default firmware major revision	
0x7	Get	UINT	Default firmware minor revision	
0x8	Get	UINT	Update firmware major revision	
0x9	Get	UINT	Update firmware minor revision	
0xA	Get	UINT	Default FPGA software revision	
0xB	Get	UINT	Update FPGA software revision	
0x20	Get	UINT	Number of modules	Input and output data
0x21	Get	UINT	Length of the analog input data in bytes	
0x22	Get	UINT	Length of the analog output data in bytes	
0x23	Get	UINT	Length of the digital input data in bytes	
0x24	Get	UINT	Length of the digital output data in bytes	
0x25	Get	UINT	Length of the X2X network status information in bytes	
0x26	Get	UINT	Length of the output status information in bytes	
0x27	Get	UINT	Highest X2X station number currently in use	
0x40	Set/Get	UINT	Size of the analog input assemblies in bytes (AI)	Assembly sizes
0x41	Set/Get	UINT	Size of the analog output assemblies in bytes (AO)	
0x42	Set/Get	UINT	Size of the digital input assemblies in bytes (DI)	
0x43	Set/Get	UINT	Size of the digital output assemblies in bytes (DO)	
0x44	Set/Get	UINT	Size of the X2X network status assemblies in bytes (NS)	
0x45	Set/Get	UINT	Size of the output status assemblies in bytes (OS)	
0x46	Set/Get	UINT	Composition of the combination input assembly	
0x60	Set/Get	UINT	Global action delay time [ms]	Actions
0x61	Set/Get	UINT	Communication loss (timeout) action	
0x62	Set/Get	UINT	Communication loss (timeout) scope	
0x63	Set/Get	UINT	Communication loss (timeout) reset mode	
0x64	Set/Get	UINT	Program mode (idle) action	
0x65	Set/Get	UINT	Program mode (idle) scope	
0x66	Set/Get	UINT	Action for faulty or missing module in state "Operational"	
0x67	Set/Get	UINT	Action for missing module(s) during the boot phase	
0x68	Set/Get	UINT	Action for incorrect module type(s) during the boot phase	
0x80	Set/Get	UINT	X2X Link configuration	X2X Link configuration
0x81	Set/Get	UINT	X2X Link cable length [m]	
0xE0	Get	UINT	Reading network address switches	Miscellaneous
0xE1	Set/Get	UINT	Module initialization delay [ms]	
0xE2	Set/Get	UINT	Enable/disable the Telnet password	
0xE3	Set/Get	UINT	IP maximum transmission unit [bytes]	
0xE4	Get	UINT	Current boot config assembly ID	
0xE5	Get	UINT	Read the number of configured I/O modules	
0xE9	Set/Get	Array of Byte	Controller for the interfaces	

12.1.2.1 Product and bus controller status

Adapter status

Attribute ID (hex)	0x1
Data type	UDINT
Access	Get
Default value	-
Description	Reads the adapter status. 32 bits of information are available. Bits 0 to 10 indicate error-free states; bits 11 to 31 indicate error states. Individual states are also indicated by the two LED status indicators on the bus controller.

Bit	Value	Description
0	0x00000001	The adapter was configured using configuration assemblies.
1	0x00000002	At least one class 1 or class 3 connection is active.
2	0x00000004	System start or I/O module initialization active
3	0x00000008	Adapter in program mode (idle)
4	0x00000010	Firmware upload via web interface active
5	0x00000020	Configuration update via web interface active
6	0x00000040	I/O module firmware upload via web interface active
7	0x00000080	A device description CFG phase is active.[1)]
8 - 10	0x00000100 - 0x00000400	Reserved
11	0x00000800	A class 1 exclusive owner timeout occurred.
12	0x00001000	An input only, listen only or class 3 timeout occurred.
13	0x00002000	Faulty or missing module detected during runtime
14	0x00004000	Missing module detected during boot phase
15	0x00008000	Incorrect module detected during boot phase
16	0x00010000	The adapter has not yet received an IP address assignment from DHCP.
17	0x00020000	2 or more identical IP addresses exist in the network.
18	0x00040000	General EIP stack error
19	0x00080000	Communication resources: Limit reached
20	0x00100000	Stack socket error occurred
21	0x00200000	Insufficient memory
22	0x00400000	Error detected while reading the primary flash memory page
23	0x00800000	Faulty configuration assembly data
24	0x01000000	"Scanner auto-connect" error
25	0x02000000	Faulty firmware update detected. The bus controller is booting with the default firmware.
26	0x04000000	Error when generating the configuration
27	0x08000000	Configuration resource is currently locked.
28	0x10000000	I/O mapping error[1)]
29	0x20000000	Reserved
30	0x40000000	Invalid DHCP settings[1)]
31	0x80000000	Fatal error (hardware or software problem)[1)]

1) Firmware version 3.07 or later.

Hardware major revision

Attribute ID (hex)	0x2
Data type	UINT
Access	Get
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.

Hardware minor revision

Attribute ID (hex)	0x3
Data type	UINT
Access	Get
Default value	-
Description	Hardware minor revision (number after the decimal point, e.g. V1.02 → 2).

1 See also "Hardware major revision" on page 46.

FPGA hardware revision

Attribute ID (hex)	0x4
Data type	UINT
Access	Get
Default value	-
Description	FPGA hardware revision Specifies the hardware revision of the installed FPGA chip.

Active boot block

Attribute ID (hex)	0x5
Data type	UINT
Access	Get
Default value	1
Description	This attribute can be used to determine the flash memory block from which the firmware or FPGA software was loaded. Flash block: Description 0: Default firmware (factory firmware) 1: Update firmware

1 See also "Startup" on page 22.

Default firmware major revision

Attribute ID (hex)	0x6
Data type	UINT
Access	Get
Default value	-
Description	Default firmware major revision (factory firmware)

Default firmware minor revision

Attribute ID (hex)	0x7
Data type	UINT
Access	Get
Default value	-
Description	Default firmware minor revision (factory firmware)

Update firmware major revision

Attribute ID (hex)	0x8
Data type	UINT
Access	Get
Default value	-
Description	Update firmware major revision

Update firmware minor revision

Attribute ID (hex)	0x9
Data type	UINT
Access	Get
Default value	-
Description	Update firmware minor revision

Default FPGA software revision

Attribute ID (hex)	0xA
Data type	UINT
Access	Get
Default value	
Description	Factory FPGA software revision (default block)[1]

1 See "Active boot block" on page 47.

Update FPGA software revision

Attribute ID (hex)	0xB
Data type	UINT
Access	Get
Default value	
Description	FPGA software revision of the update block[1]

1 See "Active boot block" on page 47.

12.1.2.2 Input and output data

Number of modules

Attribute ID (hex)	0x20
Data type	UINT
Access	Get
Default value	-
Description	Number of successfully started I/O modules. Configured dummy modules are not counted. If additional I/O modules are started at runtime, then this attribute is updated (incremented). If modules fail during runtime, this is assessed as an error. The number of modules attribute does not change in this case.

1 Bit 13 in the adapter status, see "Product and bus controller status" on page 46

Length of the analog input data in bytes

Attribute ID (hex)	0x21
Data type	UINT
Access	Get
Default value	-
Description	Length of the analog input data (AI) in bytes. The sum of all configured or, in the event of automatic configuration, all successfully started I/O modules is output.

Length of the analog output data in bytes

Attribute ID (hex)	0x22
Data type	UINT
Access	Get
Default value	-
Description	Length of the analog output data in bytes. The sum of all configured or, in the event of automatic configuration, all successfully started I/O modules is output.

Length of the digital input data in bytes

Attribute ID (hex)	0x23
Data type	UINT
Access	Get
Default value	-
Description	Length of the digital input data in bytes. The sum of all configured or, in the event of automatic configuration, all successfully started I/O modules is output.

Length of the digital output data in bytes

Attribute ID (hex)	0x24
Data type	UINT
Access	Get
Default value	-
Description	Length of the digital output data (DO) in bytes The sum of all configured or, in the event of automatic configuration, all successfully started I/O modules is output.

Length of the X2X network status information in bytes

Attribute ID (hex)	0x25
Data type	UINT
Access	Get
Default value	-
Description	Length of the X2X network status information (NS) in bytes. The sum of all configured or, in the event of automatic configuration, all successfully started I/O modules is output. The X2X network status provides information about the operating state of individual X2X Link stations[1]. The operating state of the I/O modules[2] themselves can be queried using module-specific parameters[3].

1 These are the bus modules for the respective I/O modules.
2 The so-called electronics modules in contrast to the bus modules.
3 Attribute 0xFD of the I/O module object, see "Instance attributes" on page 56

Length of the output status information in bytes

Attribute ID (hex)	0x26
Data type	UINT
Access	Get
Default value	-
Description	Length of the output status information (OS) in bytes. The sum of all configured or, in the event of automatic configuration, all successfully started I/O modules is output. The output status contains input registers of I/O modules that serve to provide feedback from status information[1].

1 See "Output status assembly" on page 31.

Highest X2X station number currently in use

Attribute ID (hex)	0x27
Data type	UINT
Access	Get
Default value	-
Description	Returns the X2X Link station number of the I/O module with the highest station number currently in use that is currently in state "Operational". This is the last I/O module that is operated or configured on the X2X Link bus. If the number of configured I/O modules is greater than this station number, then the number of configured I/O modules is returned. Index 0 is returned if the system is still in the boot phase. **Range of values:** [0, 1 to n with n < 254]

12.1.2.3 Assembly sizes

The B&R EtherNet/IP bus controller provides various static assembly instances that can be used for I/O communication. See "Assembly object and bus controller process image" on page 29.

Combination assemblies consist of several base assemblies. Accordingly, their total size results from the sum of the lengths of the individual base assemblies.

If the total size of a combination assembly is configured larger than the permissible maximum value of 502 bytes, then this assembly will not be created during a (subsequent) assembly reinitialization and can therefore not be used for I/O communication.

Information:

Changes to assembly sizes only take effect after bus controller service 0x36 has been executed.

The size of the base assembly is configured using the following attributes. The range of values of assembly sizes is 0 to 502 bytes.

Size of the analog input assemblies in bytes (AI)

Attribute ID (hex)	0x40
Data type	UINT
Access	Set/Get
Default value	120 bytes
Description	Allows the reading and writing of the size of the analog input assembly (AI, instance 120). The size is specified in bytes. Attribute 0x46 "Composition of the combination input assembly" on page 50 can be used to control whether this base assembly is part of the combination input assembly (instance 124).

Size of the analog output assemblies in bytes (AO)

Attribute ID (hex)	0x41
Data type	UINT
Access	Set/Get
Default value	120 bytes
Description	Allows the reading and writing of the size of the analog output assembly (AO, instance 110). The size is specified in bytes. This base assembly is a permanent part of the combination output assembly (instance 112).

Size of the digital input assemblies in bytes (DI)

Attribute ID (hex)	0x42
Data type	UINT
Access	Set/Get
Default value	120 bytes
Description	Allows the reading and writing of the size of the digital input assembly (DI, instance 121). The size is specified in bytes. Attribute 0x46 "Composition of the combination input assembly" on page 50 can be used to control whether this base assembly is part of the combination input assembly (instance 124).

Size of the digital output assemblies in bytes (DO)

Attribute ID (hex)	0x43
Data type	UINT
Access	Set/Get
Default value	120 bytes
Description	Allows the reading and writing of the size of the digital output assembly (DO, instance 111). The size is specified in bytes. This base assembly is a permanent part of the combination output assembly (instance 112).

Size of the X2X network status assemblies in bytes (NS)

Attribute ID (hex)	0x44
Data type	UINT
Access	Set/Get
Default value	120 bytes
Description	Allows the reading and writing of the size of the X2X Link network status assembly (NS, instance 122). The size is specified in bytes. Attribute 0x46 "Composition of the combination input assembly" on page 50 can be used to control whether this base assembly is part of the combination input assembly (instance 124).

Size of the output status assemblies in bytes (OS)

Attribute ID (hex)	0x45
Data type	UINT
Access	Set/Get
Default value	120 bytes
Description	Allows the reading and writing of the size of the output status assembly (OS, instance 123). The size is specified in bytes. Attribute 0x46 "Composition of the combination input assembly" on page 50 can be used to control whether this base assembly is part of the combination input assembly (instance 124).

Composition of the combination input assembly

Attribute ID (hex)	0x46
Data type	UINT
Access	Set/Get
Default value	15 (all 4 base input assemblies enabled)
Description	The combination input assembly basically consists of the 4 base assemblies: Analog input + Digital input + X2X network status + Output status (AI + DI + NS + OS) This attribute can be used to adjust the composition of the combination input assembly. Individual base assemblies can be turned on and off. The total size of the combination assembly is calculated from the sum of the active base assemblies. The order of the base assemblies cannot be changed, however. Bit 0: Analog input Bit 1: Digital input Bit 2: Network status Bit 3: Output status Bit set: Assembly active Bit not set: Assembly inactive **Example** A combination input assembly with the following configuration should be created: Digital input + Output status 1) Configuration of the size of the corresponding base assemblies 2) Configuration of the composition of the combination input assembly with 0xA (binary 1010) 3) Configuration of the scanner: The size of the combination input assembly consists of the sum of the sizes of the corresponding base assemblies. 4) Reinitialization of the adapter assemblies by executing bus controller service 0x36

12.1.2.4 Actions

Global action delay time [ms]

Attribute ID (hex)	0x60
Data type	UINT
Access	Set/Get
Default value	0 ms
Description	Delays the execution of an action. This delay only takes effect if the adapter (bus controller) switches from state "Operational" to a state that is connected to an action. If the adapter is already in a state other than "Operational", then all further actions will be performed without a delay. This parameter is specified in milliseconds [ms]; the default setting is 0 ms (no delay).

Communication loss (timeout) action

Attribute ID (hex)	0x61
Data type	UINT
Access	Set/Get
Default value	1 (set outputs to zero)
Description	Configures the action for state "Communication loss" on page 35 (timeout) For possible actions, see "Available actions" on page 36.

Communication loss (timeout) scope

Attribute ID (hex)	0x62
Data type	UINT
Access	Set/Get
Default value	1 (adapter-global)
Description	Defines whether the *Communication loss (timeout)* action is "adapter-global" or "assembly-local". Adapter-global applies to all outputs of I/O modules connected to the bus controller. Assembly-local applies only to those outputs that are connected to assemblies for which the *Communication loss* event has occurred. Value 0: Local Value 1: Global See also "Action scope" on page 37.

Communication loss (timeout) reset mode

Attribute ID (hex)	0x63
Data type	UINT
Access	Set/Get
Default value	1 (connection-oriented)
Description	Defines whether an existing *Communication loss (timeout)* state is reset when reestablishing the corresponding connection or explicitly via bus controller service 0x35. Value 0: Explicitly via bus controller service 0x35 Value 1: By the successful re-establishment of the I/O connection

1 See "B&R-specific services" on page 55.

Program mode (idle) action

Attribute ID (hex)	0x64
Data type	UINT
Access	Set/Get
Default value	1 (set outputs to zero)
Description	Configures the action for state "Program mode" on page 35 (idle) For possible actions, see "Available actions" on page 36.

Program mode (idle) scope

Attribute ID (hex)	0x65
Data type	UINT
Access	Set/Get
Default value	1 (adapter-global)
Description	Defines whether the *program mode (idle)* action is "adapter-global" or "assembly-local". Adapter-global applies to all outputs of I/O modules connected to the bus controller. Assembly-local applies only to those outputs that are connected to assemblies for which the *Program mode* event has occurred. Value Scope 0 Local 1 Global See also "Action scope" on page 37.

Action for faulty or missing module in state "Operational"

Attribute ID (hex)	0x66
Data type	UINT
Access	Set/Get
Default value	1 (set outputs to zero)
Description	Configures the action for state "Module failed" on page 36 For possible actions, see "Available actions" on page 36.

Action for missing module(s) during the boot phase

Attribute ID (hex)	0x67
Data type	UINT
Access	Set/Get
Default value	1 (set outputs to zero)
Description	Configures the action for state "Module missing at power-up" on page 36. For possible actions, see "Available actions" on page 36.

Action for incorrect module type(s) during the boot phase

Attribute ID (hex)	0x68
Data type	UINT
Access	Set/Get
Default value	1 (set outputs to zero)
Description	Configures the action for state "Module mismatch at power-up" on page 36 For possible actions, see "Available actions" on page 36.

12.1.2.5 X2X Link configuration

X2X Link configuration

Attribute ID (hex)	0x80
Data type	UINT
Access	Set/Get
Default value	6 (=1 ms)
Description	The X2X Link cycle time and the data width achieved are 2 system parameters that cannot be configured separately from each other. This value can be optimized depending on the required cycle time and the number of connected I/O modules.

Value	Cycle time	Description
0	4 ms	Max. 253 I/O modules, max. 1400 bytes sync data
1	3.5 ms	Max. 253 I/O modules, max. 1150 bytes sync data
2	3 ms	Max. 253 I/O modules, max. 900 bytes sync data
3	2.5 ms	Max. 200 I/O modules, max. 800 bytes sync data
4	2 ms	Max. 200 I/O modules, max. 500 bytes sync data
5	1.5 ms	Max. 100 I/O modules, max. 450 bytes sync data
6	1 ms	Max. 80 I/O modules, max. 300 bytes sync data
7	0.5 ms	Max. 40 I/O modules, max. 120 bytes sync data

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

X2X Link cable length [m]

Attribute ID (hex)	0x81
Data type	UINT
Access	Set/Get
Default value	0 [m]
Description	Used to optimize X2X Link timing with respect to low ESD emissions. 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 two X2X Link stations (100 m) and the maximum number of stations (253 modules), which equals in total 25.3 km. The software/hardware must be restarted in order to apply changes that have been made.

12.1.2.6 Miscellaneous

Reading network address switches

Attribute ID (hex)	0xE0
Data type	UINT
Access	Get
Default value	-
Description	Reads the two network address switches located on the front of the bus controller. The switch with the ID "x16" provides the upper digit of the hexadecimal value "x1" provides the lower digit. The higher-valued byte in the UINT data type remains empty.

Module initialization delay [ms]

Attribute ID (hex)	0xE1
Data type	UINT
Access	Set/Get
Default value	3000 ms
Description	This attribute can be used to configure or read the module initialization delay. This delay is specified in [ms]. The value is only saved permanently after executing service 0x33 "Storing the system data to flash memory" on page 55 of bus controller object class 0x64, instance 1. After a restart, the system goes into a module initialization phase. 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. It is not possible to establish a class 1 or class 3 connection during the module initialization phase. The adapter is in initialization or self-test mode, which is indicated by the "Mod status" LED.

Enable/disable the Telnet password

Attribute ID (hex)	0xE2
Data type	UINT
Access	Set/Get
Default value	0 (password disabled)
Description	Value Description 0 Password disabled 1 Password enabled The default password is "BcEip". This function can be used, but it is not possible to change the password. Attackers who know they are dealing with a EtherNet/IP node would not be stopped even by a complicated password since it is also possible to communicate with the bus controller via the unprotected CIP protocol.

IP maximum transmission unit [bytes]

Attribute ID (hex)	0xE3
Data type	UINT
Access	Set/Get
Default value	1500
Description	The *maximum transmission unit* (MTU) specifies the maximum size of the complete TCP/IP packet. **The function is not currently in use.**

Current boot config assembly ID

Attribute ID (hex)	0xE4
Data type	UINT
Access	Get
Default value	0
Description	Reads the current boot config assembly ID Value: Description 0: The bus controller is being operated in automatic configuration mode. 100: Manual configuration via configuration assembly instance 100 130 to 139: Manual configuration via extended configuration assembly instances 130 to 139 The boot config assembly ID parameter is updated under the following conditions: 1: If a scanner opens a class 1 connection and transmits in the process a (connection-based) configuration assembly whose data does not correspond to the current configuration (value: 100). 2: If an extended configuration assembly is enabled explicitly via bus controller service 0x37 and a configuration change results (value: 130 to 139). **Example 1** The bus controller is connected to a scanner that establishes a class 1 connection and in the process transmits the configuration data. Since this is new data, the boot config assembly ID parameter is set to the ID of the transferred configuration assembly (usually 100). If the same configuration data is now saved to the extended configuration assembly by the application with the ID 130 and then ID 130 is enabled via service 0x37, then the bus controller is not restarted and the boot config assembly ID parameter preserves its last value. In this case, the configuration data remains unchanged. **Example 2** New configuration that does not match the configuration currently in use is written to one or more extended configuration assemblies and then enabled via service 0x37. The bus controller restarts automatically and the boot config assembly ID parameter is reset (value: 130 to 139). A scanner then establishes a class 1 connection and transmits configuration data that is identical to the current configuration to the bus controller. In this case, the boot config assembly ID parameter is not changed since the configuration data has not changed.

Read the number of configured I/O modules

Attribute ID (hex)	0xE5
Data type	UINT
Access	Get
Default value	-
Description	Reads the number of configured I/O modules

Controller for the interfaces

Attribute ID (hex)	0xE9
Data type	Array[1..6] of Byte
Access	Set/Get
Default value	**Parameter** / **Default value** / **Description** PIN / 0, 0, 0, 0 / The PIN is not active. The interface controller can be written to with any PIN. cmd / 0x00 / No command active. state / 0xFF / All interfaces are enabled or open.
Description	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, HTTP and Telnet interfaces, which are not absolutely necessary for basic EtherNet/IP functionality. Changes are effective immediately but not automatically saved in flash memory. **Data is saved remanently only after calling service 0x33 - Save system data to flash memory.** This function is available in firmware version 3.07 or later. **Structure of the interface controller** Writing is only possible with service Set_Attribute_Single. The length of the data must be 6 bytes. **Interface controller (6-byte array)**: PIN (Byte 1, Byte 2, Byte 3, Byte 4); ICP (cmd (byte), state (byte)) **Explanation of parameters** **Parameter** / **Values** / **Description** PIN / x, x, x, x / Protection for the interface settings. After successful initialization, a change is only possible with a valid pin. cmd / 0 / No command active. cmd / 1 / Resets the interface controller to its default values. state / 0xFF / State of the interfaces. The following interfaces can be switched off: **Interface** / **State** / **Description** UDP service channel / Bit 0 / **Value** / **Description**: 1 / Interface is available.; 0 / Interface is blocked. Telnet / Bit 1 / **Value** / **Description**: 1 / Interface is available.; 0 / Interface is blocked. HTTP web server / Bit 2 / **Value** / **Description**: 0 / Interface is available.; 1 / Interface is blocked. Possible errors **Name** / **Code** / **Description** INVALID_PARAMETER_VALUE / 0x03 / Invalid parameter PERMISSION_DENIED / 0x0F / Invalid PIN NOT_ENOUGH_DATA / 0x13 / Invalid data length: Too little data TOO_MUCH_DATA / 0x15 / Invalid data length: Too much data **Using the PIN** • 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. **Network address switch function** 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.

12.1.3 Common services

Service code [hex]	Supported by	Service name	Description
0xE	Class/Instance	Get_Attribute_Single	Supplies the contents of the corresponding attribute
0x10	Instance	Set_Attribute_Single	Modifies an attribute value

12.1.4 B&R-specific services

Service code [hex]	Data type, parameter	Description
0x32	Parameters: -	Resets all pending timeouts, see "Communication loss" on page 35
0x33	Parameters: -	Saves all system data to flash memory. This permanently stores all changes to volatile flash memory that have not been stored there automatically by other services. An example of a change that is automatically stored in permanent memory is the set command to the "Configuration Control" attribute 0x3 of the TP/IP object (class 0xF5, see "TCP/IP interface object " on page 42).
0x34	Parameters: -	Reads all system data from flash memory. This overwrites all changes in RAM with the permanently saved settings.
0x35	Parameters: -	Deletes the entire flash memory. This restores the factory default settings in flash memory.
0x36	Parameters: -	Reinitializes all assemblies, see "Changes to I/O assemblies" on page 31 and "Assembly sizes" on page 49.
0x37	Parameters: UINT Start Assembly ID (Value range:130 to 139)	Enables one or more extended configuration assemblies The specified parameter makes it possible to generate several independent configurations in the 10 assemblies (see "Advanced configuration assemblies" on page 32). Service 0x37 can be used to enable the desired configuration with the corresponding start assembly ID. All 10 assemblies for a single configuration can also be used (use of extended combination configuration assembly file, *_ext.bin). **Example** Configuration A is saved to assemblies 130, 131 and 132 (e.g. the files *_ext_0.bin, *_ext_1.bin and *_ext2.bin from Automation Studio). Configuration B is saved to assembly 135. Configuration C is saved to assembly 136. The following services must be executed to enable the individual configurations. For A configurations: Service: 0x37, data 0x82 00 (parameter: ID 130) For B configurations: Service: 0x37, data 0x87 00 (parameter: ID 135) For C configurations: Service: 0x37, data 0x88 00 (parameter: ID 136)
0x38	Parameters: -	Clears all I/O module configuration data from RAM. Flash memory will not be cleared. After executing flash memory command 0x33, the configuration data is also cleared from flash memory.
0x40	Parameters: UINT	Generates a new configuration based on an existing configuration or the connected I/O modules. This can include both configured and unconfigured I/O modules. Function of the bits in the parameter: **Bit** / **Function** 0 – The configuration is compressed in ZIP format. 1 – The EtherNet/IP stack configuration assemblies are written. 2 – All data that has been generated is stored in flash memory. 3 – The RAW I/O module register configuration is written.
0x41	Parameters: -	Clears the Parameter list for all I/O modules
0x42	Parameters: UINT	Applies the Parameter list to existing RAW configurations. This changes all existing configurations. The result is stored in the RAW configuration (Flash Shadow RAM). Optionally, the generated RAW configuration can be stored in flash memory. Possible parameter value. **Value** / **Function** 1 – All data that has been generated is stored in flash memory.

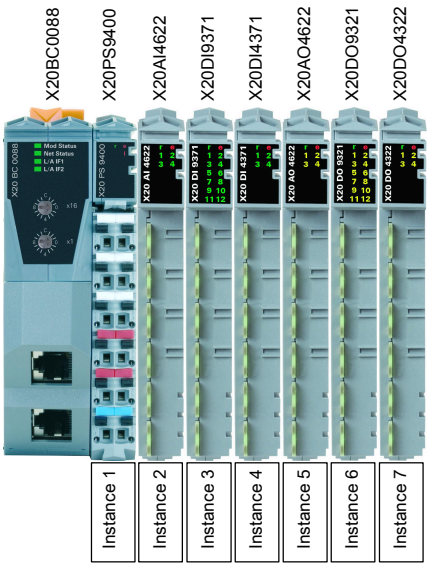
12.2 I/O module object

(CIP class 0x65)

The I/O module object can be used to manage all parameters relevant to I/O modules.

> **Information:**
>
> **Each instance of this I/O module object corresponds to the electronics module being operated on the X2X Link station. Instance 0x1 stands for the first I/O module in the X2X Link network, for example.**

Example:



For the B&R X20BC0088 bus controller, instance 0x1 is always the power supply module, e.g. X20PS9400 or X20PS9402.

In addition, service functions make it possible to use asynchronous I/O module communication. This allows asynchronous I/O registers to be accessed during operation in order to change settings, for example. For details about these registers, see the respective module documentation.

12.2.1 Class attributes

Attribute ID (hex)	Access	Data type	Description	Default value
0x1	Get	UINT	Revision	0x0001
0x2	Get	UINT	Maximum number of instances, i.e. maximum number of supported I/O modules	0x00FD

12.2.2 Instance attributes

Attribute ID (hex)	Access	Data type	Data type
0x01	Get	ARRAY [0..5] of BYTE	Configured module hardware ID
0x02	Get	ARRAY [0..5] of BYTE	Current module hardware ID
0x03	Get	USINT	Total length of input data
0x05	Get	USINT	Total length of output data
0xA0	Get	UINT	Read the number of I/O module registers
0xA1	Get	ARRAY [0..n] of UDINT	List of register addresses for an I/O module
0xA2	Get	ARRAY [0..n] of UDINT	List of register values for an I/O module
0xE0	Get	UINT	Analog input data length in bytes (AI)
0xE1	Get	UINT	Analog output data length in bytes (AO)
0xE2	Get	UINT	Digital input data length in bytes (DI)
0xE3	Get	UINT	Digital output data length in bytes (DO)
0xE4	Get	UINT	Network status data length in bytes (NS)
0xE5	Get	UINT	Output status data length in bytes (OS)
0xFA	Get	UINT	Module firmware version
0xFB	Get	UINT	Module hardware variant
0xFC	Get	UDINT	Module serial number
0xFD	Get	UINT	Module status

Configured module hardware ID

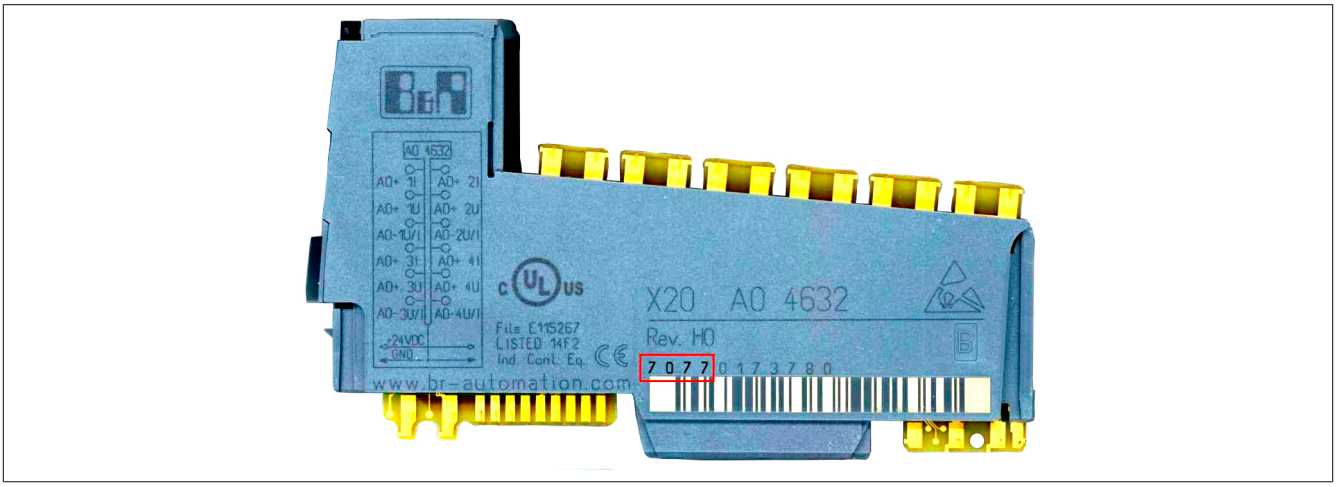
Attribute ID (hex)	0x01
Data type	ARRAY [0..5] of BYTE
Access	Get
Default value	0x0000, *module hardware ID*, 0x0000
Description	Allows the configured *module hardware ID* to be read The value 0x0000 is returned in the event of an automatic configuration or unconfigured slots[1]. Configured yet unoccupied slots return the value 0xFFFF. Byte[0,1]: Vendor ID (currently always 0x0000) Byte[2,3]: *Module hardware ID* Byte[4,5]: Reserved (always 0x0000)

1 See "Auto mode" on page 28.

Current module hardware ID

Attribute ID (hex)	0x02
Data type	ARRAY [0..5] of BYTE
Access	Get
Default value	0x0000, *module hardware ID*, 0x0000
Description	Allows the current *module hardware ID* to be read If an I/O module is not connected to the addressed module slot, then the value 0x0000 is returned. Byte[0,1]: Vendor ID (currently always 0x0000) Byte[2,3]: *Module hardware ID* Byte[4,5]: Reserved (always 0x0000)

For the *module hardware ID*, see the respective module documentation. In addition, a serial number is printed on each electronics module; the *module hardware ID* corresponds to the first 4 positions of the serial number. (See figure: Hardware ID is also colored black.)



Information:

The IDs up to 9999 are printed as decimal numbers and must be converted to hex values for comparison!

Total length of input data

Attribute ID (hex)	0x03
Data type	UINT
Access	Get
Default value	-
Description	Specifies the total length of the input data for the respective I/O module in bytes The sum of the following frames is formed: AI, DI, NS, OS. The result 0 is output if slots are empty.

Total length of output data

Attribute ID (hex)	0x05
Data type	UINT
Access	Get
Default value	-
Description	Specifies the total length of the output data for the respective I/O module in bytes The sum of the following frames is formed: AO, DO. The result 0 is output if slots are empty.

Read the number of I/O module registers

Attribute ID (hex)	0xA0
Data type	UINT
Access	Get
Default value	-
Description	Returns the number of configuration registers for the current I/O module taking into account the Parameter list.

List of register addresses for an I/O module

Attribute ID (hex)	0xA1
Data type	ARRAY [0..n] of UDINT (a maximum of 124 values)
Access	Get
Default value	-
Description	Returns a list of configuration register addresses for the current I/O module. Together with attribute 0xA2, this attribute forms the Parameter list. The 32-bit register address that is read consists of a physical address and a virtual sub-address • HIGH word: Physical address (little-endian format) • LO word: Virtual sub-address (little-endian format) The sub-address is sequential number generated by the bus controller that specifies the number of entries in the parameter list. **Information:** **Errors occur in the configuration data if there are more than 124 entries in the parameter list.**

List of register values for an I/O module

Attribute ID (hex)	0xA2
Data type	ARRAY [0..n] of UDINT (maximum 124 values)
Access	Get
Default value	-
Description	Returns a list of configuration register values for the current I/O module. This attribute, in conjunction with attribute 0xA1, constitutes the Parameter list. **Information:** **Errors occur in the configuration data if there are more than 124 entries in the parameter list.**

Analog input data length in bytes (AI)

Attribute ID (hex)	0xE0
Data type	UINT
Access	Get
Default value	-
Description	Analog input (AI) data length of the respective I/O module in bytes. The result 0 is output if slots are empty.

Analog output data length in bytes (AO)

Attribute ID (hex)	0xE1
Data type	UINT
Access	Get
Default value	-
Description	Analog output (AO) data length of the respective I/O module in bytes. The result 0 is output if slots are empty.

Digital input data length in bytes (DI)

Attribute ID (hex)	0xE2
Data type	UINT
Access	Get
Default value	-
Description	Digital input (DI) data length of the respective I/O module in bytes. The result 0 is output if slots are empty.

Digital output data length in bytes (DO)

Attribute ID (hex)	0xE3
Data type	UINT
Access	Get
Default value	-
Description	Digital output (DO) data length of the respective I/O module in bytes. The result 0 is output if slots are empty.

Network status data length in bytes (NS)

Attribute ID (hex)	0xE4
Data type	UINT
Access	Get
Default value	-
Description	Reads the X2X Link network status (NS) data length of the respective I/O module in bytes. The result 0 is output if X2X Link stations are not present.

Output status data length in bytes (OS)

Attribute ID (hex)	0xE5
Data type	UINT
Access	Get
Default value	-
Description	Reads the output status (OS) data length of the respective I/O module in bytes. The result 0 is output if slots are empty.

Module firmware version

Attribute ID (hex)	0xFA
Data type	UINT
Access	Get
Default value	-
Description	Firmware version of the I/O module currently found on 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. The result 0 is output if slots are empty.

Module hardware variant

Attribute ID (hex)	0xFB
Data type	UINT
Access	Get
Default value	-
Description	Hardware variants of the I/O module currently found on 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. The result 0 is output if slots are empty.

Module serial number

Attribute ID (hex)	0xFC
Data type	UDINT
Access	Get
Default value	-
Description	Every B&R module has a unique serial number. The complete serial number is made up of the module hardware ID and module serial number as follows: Serial number = Module hardware ID * 1E+7 + Module serial number The module's serial number is printed in decimal form on the module's housing.

Module status

Attribute ID (hex)	0xFD
Data type	UINT
Access	Get
Default value	-
Description	Reads module status of a connected I/O module.

Value	Function
0x00 "0"	No X2X Link station found for this instance (slot)
0x42 "B"	Boot procedure (OS loader test)
0x43 "C"	Module being configured
0x44 "D"	Firmware download active
0x4E "N"	X2X Link station found, but the I/O module cannot be started for the following reasons: No I/O power supply, or the module is not connected to the X2X Link ASIC component.
0x50 "P"	Mode PREOPERATIONAL (module ready for mode RUN)
0x52 "R"	Mode RUN (operational)
0x55 "U"	Boot procedure (uploading IDs)
0x70 "p"	Mode PREOPERATIONAL (module ready for mode RUN)
0xE0	Error: Module without firmware detected
0xE1	Error: Module with invalid firmware detected
0xE2	Error: Module cannot be activated, possible configuration error (incorrect function model, etc.)
0xE3	Error: Register cannot be mapped, possible configuration error, etc.
0xE4	Internal error, I/O module cannot be started
0xE5	Module cannot be started, X2X Link frame too small → If necessary, increase the X2X Link cycle time.
0xE6	Module not started, different module type configured for this slot

12.2.3 Service objects

Service code (hex)	Supported by	Service name	Description
0xE	Class/Instance	Get_Attribute_Single	Supplies the contents of the corresponding attribute
0x10	Instance	Set_Attribute_Single	Modifies an attribute value

12.2.4 B&R-specific services

Service ID	Data type, parameter	Description
0x32	Parameters: UINT Register address Results: DINT Register value	Reads I/O module registers (in little-endian format)
0x33	Parameters: UINT Register address Results: DINT Register value	Writes asynchronous I/O module registers (in little-endian format)
0x40	Parameters: UINT Hardware register address UINT Sub-address UDINT Register value USINT Action Results: -	Adds an entry to I/O module Parameter list. Possible values for parameter "Action": Value / Function: 1 Add register; 2 Deleting a register; 3 Change register value Sub-address 0xFFFF should always be used for the "Add register" action. For the other actions, the value from attribute 0xA1 "List of register addresses for an I/O module" on page 58 is used.
0x41	Parameters: - Results: -	Deletes the entire I/O module Parameter list

12.2.4.1 Reading I/O module registers

The 0x32 service of the I/O module object makes it possible to access synchronous and asynchronous I/O module registers during operation (read-only). Each instance of the I/O module object corresponds to the electronics module being operated on that particular X2X Link station. Instance 0x1 stands for the first I/O module in the X2X Link network, for example.

For details about I/O registers such as addresses, values, etc., see the respective module documentation. The register address is passed to the service as a 16-bit parameter value; the resulting register value is available as a 32-bit DINT value.

Example

This example reads the input filter of a digital input module, for example the X20DI9371 in slot 2. The input filter is mapped in register 18 (0x12).

Execution looks like this:

Service=0x32, class=0x65 (101), instance=2, attribute=<empty> or any, data=0x1200 (decimal 12 in little-endian format)

The input filter is set to 1 ms by default. Since the time is specified as a multiple of 100 µs for this module, the corresponding value is 10. The result is therefore output as 0x0A000000 (dec. 10, little-endian format).

12.2.4.2 Writing asynchronous I/O module registers

Service 0x33 allows the writing of asynchronous module registers. As parameters, the register address must be passed to the service as a 16-bit UINT value and the register value to be written as a 32-bit DINT value. These 2 + 4 = 6 bytes must all be transferred in little-endian format. This service does not return a result, i.e. as long as protocol error (CIP error) is not output, then the write procedure was successful.

Information:

If synchronous registers are written using this service while a class 1 I/O connection exists at the same time, then these registers will usually be overwritten cyclically by I/O data.

Example

This example converts channel 1 of an analog input module such as the X20AI4622 from the default voltage measurement setting to current measurement. The necessary configuration value is 0x01 for this module according to the register documentation. The module is located in slot 11. Input channels 2 to 4 should continue to be used for voltage measurement. The channel type is mapped in register 18 (0x12) for this module.

Service=0x33, Class=0x65 (101), Instance=0xB (11), Attribute=<empty> or any, Data=0x1200 **01**00 0000 (little-endian format)

13 Diagnostic tools

The 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. IP address conflicts, incorrect or missing I/O modules, module status)
- Statistics (e.g. EtherNet/IP protocol, network, X2X Link)

13.1 Product data

It is only possible to read product data from bus controllers and I/O modules.

13.1.1 Bus controller

	Class	Instance	Attribute	Data type
Product code (hardware ID)	0x1	0x1	0x3	UINT
Serial number	0x1	0x1	0x6	UDINT
Hardware major revision	0x64	0x1	0x2	UINT
Hardware minor revision	0x64	0x1	0x3	UINT
FPGA hardware revision	0x64	0x1	0x4	UINT
Active boot block	0x64	0x1	0x5	UINT
Default firmware major revision	0x64	0x1	0x6	UINT
Default firmware minor revision	0x64	0x1	0x7	UINT
Update firmware major revision	0x64	0x1	0x8	UINT
Update firmware minor revision	0x64	0x1	0x9	UINT
Default FPGA software revision	0x64	0x1	0xA	UINT
Update FPGA software revision	0x64	0x1	0xB	UINT

13.1.2 I/O modules

	Class	Instance	Attribute	Data type
Configured module hardware ID	0x65	Respective slot	0x1	ARRAY[0..5] of BYTE
Current module hardware ID	0x65	Respective slot	0x2	ARRAY[0..5] of BYTE
Module firmware version	0x65	Respective slot	0xFA	UINT
Module hardware variant	0x65	Respective slot	0xFB	UINT
Module serial number	0x65	Respective slot	0xFC	UDINT

These parameters specify the data for the module currently in this slot with the exception of the "configured module hardware ID".

13.2 Operating status

13.2.1 Bus controller

	Class	Instance	Attribute	Data type
Adapter status	0x64	0x1	0x1	UDINT
Number of modules	0x64	0x1	0x20	UINT
Length of the analog input data in bytes	0x64	0x1	0x21	UINT
Length of the analog output data in bytes	0x64	0x1	0x22	UINT
Length of the analog input data in bytes	0x64	0x1	0x23	UINT
Length of the digital output data in bytes	0x64	0x1	0x24	UINT
Length of the X2X network status information in bytes	0x64	0x1	0x25	UINT
Length of the output status information in bytes	0x64	0x1	0x26	UINT
Highest X2X station number currently in use	0x64	0x1	0x27	UINT

13.2.2 I/O modules

The operating state of individual I/O modules themselves can be read using the 0xFD attribute.

	Class	Instance	Attribute	Data type
Module status	0x65	Respective slot	0xFD	UINT

Value	Function
0x00 "0"	No X2X Link station found for this instance (slot)
0x42 "B"	Boot procedure (OS loader test)
0x43 "C"	Module being configured
0x44 "D"	Firmware download active
0x4E "N"	X2X Link station found, but the I/O module cannot be started for the following reasons: No I/O power supply, or the module is not connected to the X2X Link ASIC component.
0x50 "P"	Mode PREOPERATIONAL (module ready for mode RUN)
0x52 "R"	Mode RUN (operational)
0x55 "U"	Boot procedure (uploading IDs)
0x70 "p"	Mode PREOPERATIONAL (module ready for mode RUN)
0xE0	Error: Module without firmware detected
0xE1	Error: Module with invalid firmware detected
0xE2	Error: Module cannot be activated, possible configuration error (incorrect function model, etc.)
0xE3	Error: Register cannot be mapped, possible configuration error, etc.
0xE4	Internal error, I/O module cannot be started
0xE5	Module cannot be started, X2X Link frame too small → If necessary, increase the X2X Link cycle time.
0xE6	Module not started, different module type configured for this slot

Information:

Additional diagnostic information with regard to modules can be found the X2X Link network status assembly. The X2X Link network status is based on the bus modules or X2X Link controller, however, not the actual I/O module.

14 Web server

The web server integrated in the bus controller starting with Rev. B4, firmware V.1.06 offers the following:

- Easy configuration of the most common settings
- Possibility of transferring configuration files
- Extensive diagnostic options and an overview of the composition of both combination assemblies
- The ability to update the firmware

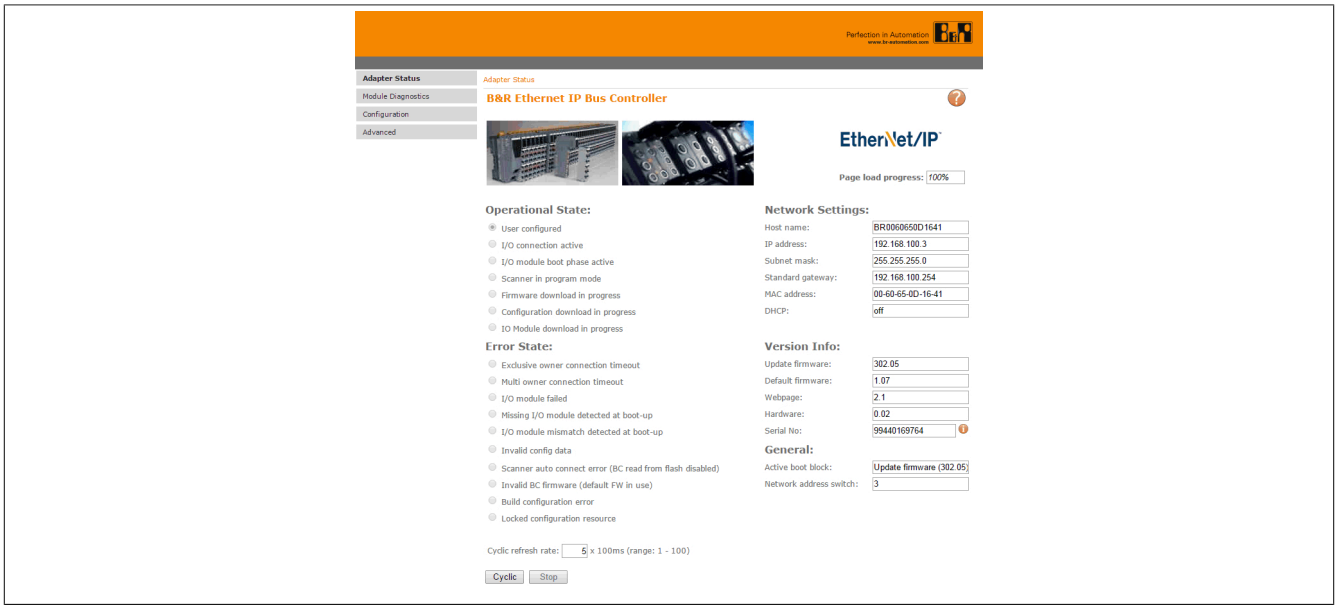
It also includes a help system that can be opened by clicking the "?" icon on any of the pages.

The web server can be opened from any browser by specifying the IP address or hostname in the address bar.

Examples

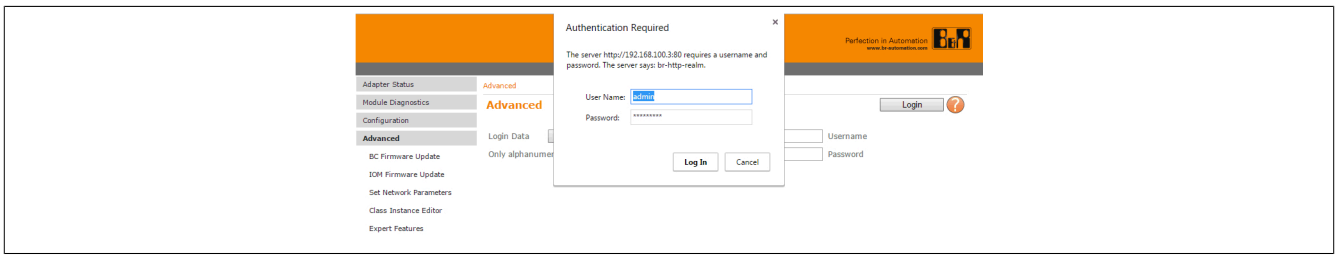
http://192.168.100.1

http://breip128 if the address switch value is 0xFF or 0x80



14.1 "Advanced" menu

The pages under menu option "Advanced" require write access or the execution of services. For this reason, they will request authentication before they allow these functions to be accessed.



Default settings for login data:

> **Information:**
>
> **Username = admin**
>
> **Password = Corresponds to the module name (X20BC0088 *, X67BCD321.L12 or X67BCD321.L12-1)**
>
> **Please note that both of these parameters are case-sensitive!**

* also applies to the coated module.

14.1.1 Download firmware

This menu option allows new firmware to be downloaded to the bus controller.

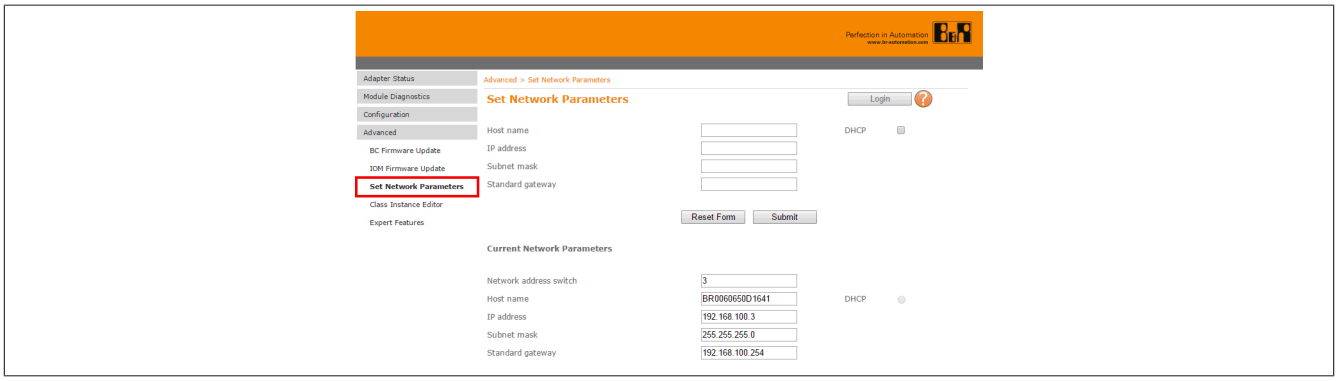
14.1.2 Download the I/O module firmware

This menu option allows you to download new firmware to I/O modules. The update is performed on all modules whose hardware variant and module ID match the firmware.

14.1.3 Network configuration

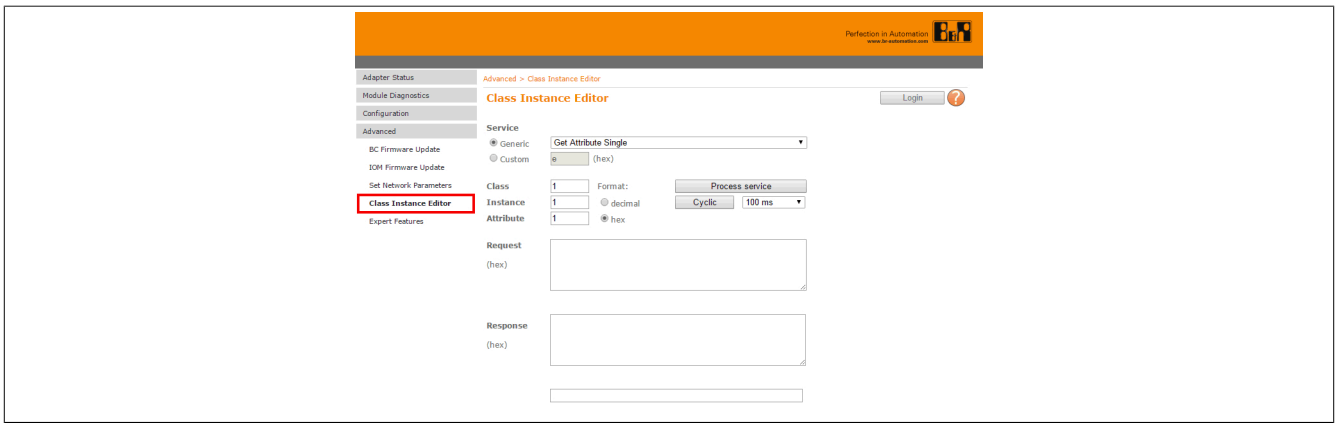
This menu provides an easy way to change all network parameters.

> **Information:**
>
> **Starting with firmware version 3.xx, the network parameters can only be changed with network address switch setting 0x00.**



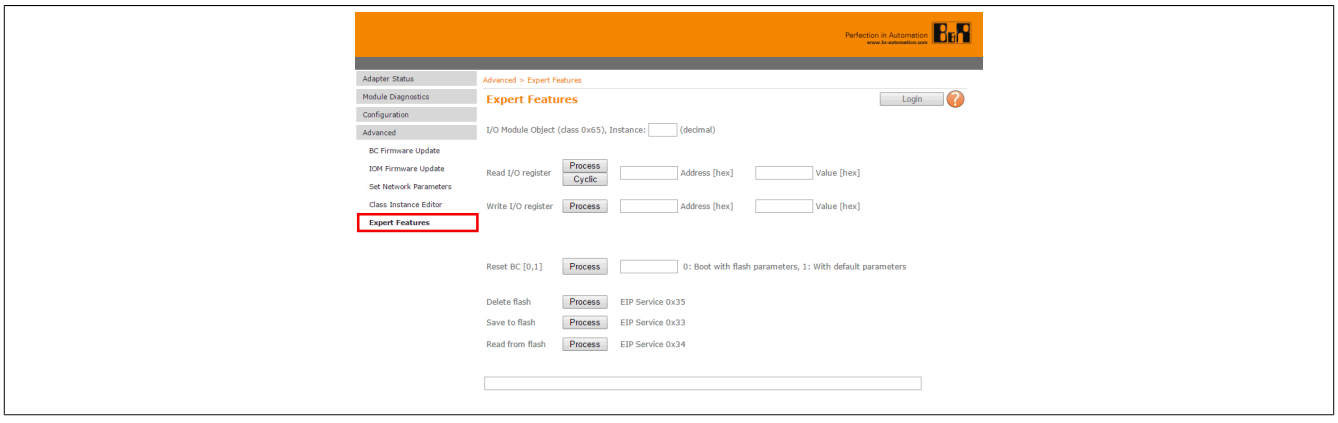
14.1.4 Class instance editor

The class instance editor makes it possible to directly query and change the attributes of the CIP object dictionary.



14.1.5 Expert features

This menu can be used to read or write X2X registers. In also makes it possible to load, save and delete the bus controller configuration.



15 Configuration management

Starting with version 2.x of the bus controller, "Configuration" has been added to in the integrated web server in order to make it easy to edit I/O module configurations.

15.1 Parameter list

The parameter list is located in the bus controller and is where the register parameters for a module are stored. Each module has its own list. It is therefore possible to edit module configurations directly on the bus controller without having to rely on external tools.

This parameter list can be generated and updated as follows:

- Downloading a configuration file generated by Automation Studio
- Generating an automatic configuration
- Calling command "Build configuration" on page 70 on the bus controller web page

All configuration changes made on the integrated bus controller web page or using CIP class 0x65 0x40 service calls are stored in this list first.

After calling command "Apply and build configuration" on page 67 or service CIP class 0x64 0x40, a new configuration is created from this list.

The current configuration of a specific module can be queried using the two I/O module object attributes "List of register addresses for an I/O module" on page 58 and "List of register values for an I/O module" on page 58.

Information:

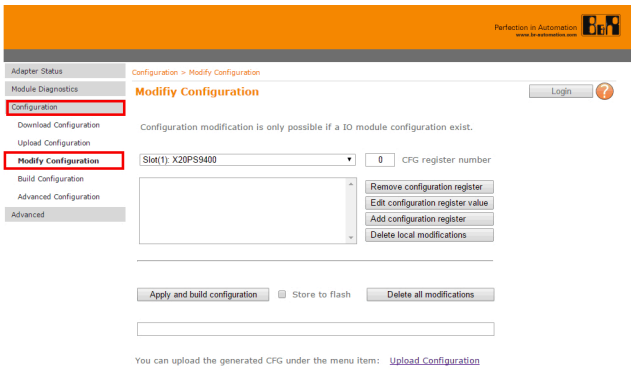
The number of entries in the parameter list for a module must not exceed 124.
The total number of all entries in the parameter list on the bus controller is not permitted to exceed 1024.

15.2 Editing the Configuration

The "Modify Configuration" sub-item on the integrated web page allows modification of the acyclic configuration registers that are used to configure I/O modules on startup.

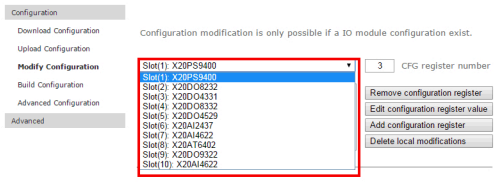
Acyclic configuration registers can be added, edited and deleted for each individual I/O module. First, a Parameter list is generated on the bus controller, where all the changes will be entered.

These changes will only be applied and implemented when the command "Apply and build configuration" or service CIP class 0x64 0x40 is carried out.



Selecting I/O modules

The I/O module that should be reconfigured is selected from a menu.



After selection, all registers in the current configuration list for this I/O module will appear in the corresponding list.

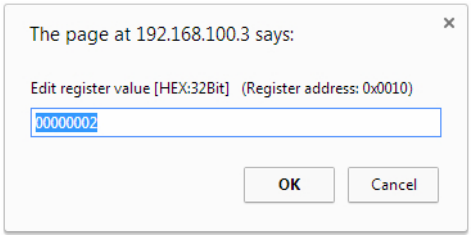
Deleting a register

The "Remove configuration register" button can be used to deleted a configuration register that has been selected in the list.

Editing existing registers

The "Edit configuration register value" button can be used to edit a configuration register that has been selected in the list.

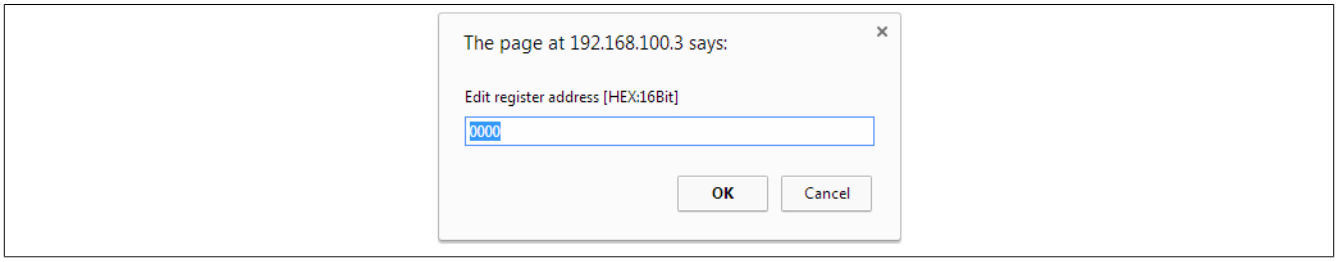
Dialog box 1: Enter the new register value.



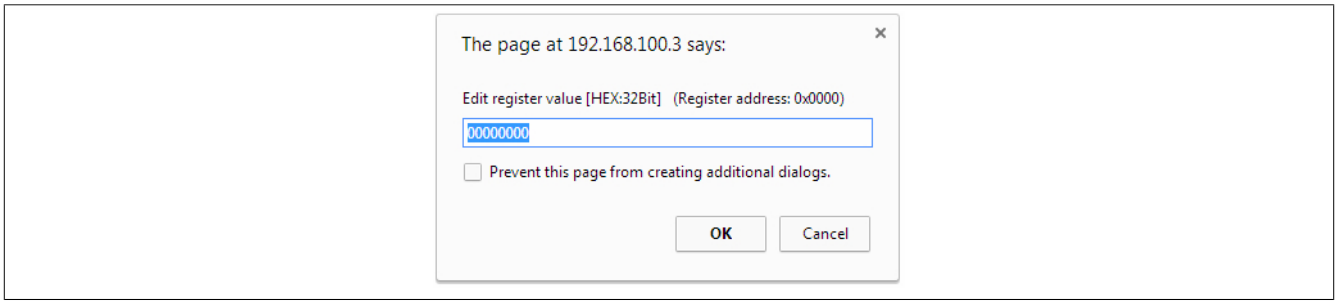
Adding a new register

The "Add configuration register" button can be used to add a new configuration register. This is done in 2 steps:

1. Dialog box 1: Enter the new register number.



2. Dialog box 2: Enter the value for the new register.

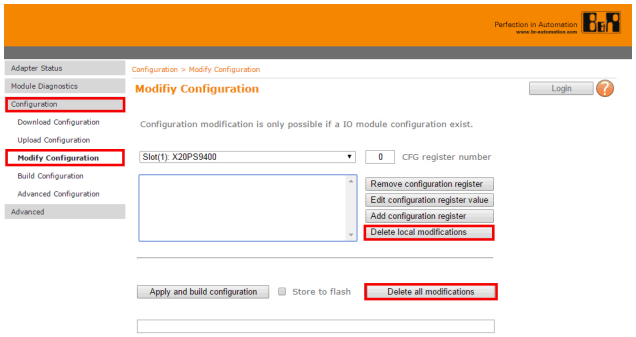


Information:

In all dialog boxes:

- **Entries that begin with 0x are interpreted as hex values.**
- **Entries without an 0x are interpreted as decimal values.**

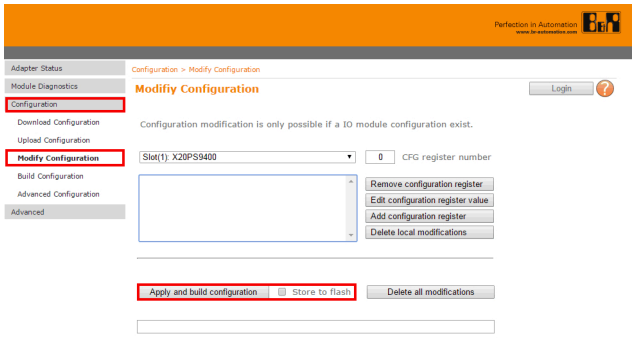
15.3 Deleting configuration modifications



The "Delete local modifications" button deletes all configuration changes that have been made on the currently selected module ("Slot x") but not yet saved.

The "Delete all modifications" button deletes all configuration changes that have been made on all modules but not yet saved.

15.4 Applying the configuration



The "Apply and build configuration" button can be used to make all configuration changes included in the Parameter list and store them in RAM on the bus controller.

This does not affect the configuration currently running on the bus controller. This function can be used, for example, to create configurations for similar systems, which can then be uploaded using Function "Uploading the configuration" on page 70.

The "Store to Flash" checkbox can be used to permanently store the configuration that has been generated to Flash memory.

> ### Information:
>
> **Modified configurations are not applied until the bus controller is restarted.**

15.5 Generating configurations

On the integrated web page for the bus controller, the "Build Configuration" sub-item can be found under Configuration.

This function makes it possible to generate a configuration file on the bus controller:

- When the bus controller is operated in automatic configuration mode, all modules that are physically present on the bus and in "RUN" mode are added to the configuration file.
- If the bus controller has been configured manually, then a Manual configuration (full configuration) is already saved on the bus controller. In this case, the configuration in the flash memory is applied to the configuration file, as well as any modules not already included in this configuration that are physically present on the bus and in "RUN" mode.

The "Build configuration" button can be used to generate the configuration and store it in RAM on the bus controller. The configuration that has been generated is displayed in the status bar under "Build configuration".

This does not affect the configuration currently running on the bus controller. This function can be used, for example, to create configurations for similar systems, which can then be uploaded using Function "Uploading the configuration" on page 70.

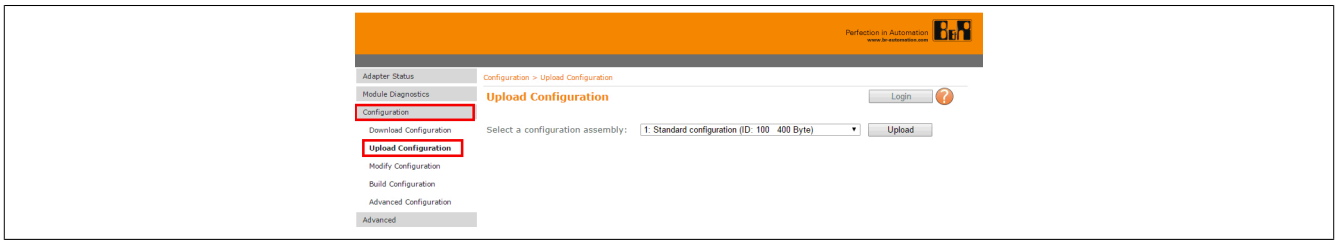
The "Store to Flash" checkbox can be used to permanently store the configuration that has been generated to Flash memory.

Information:

Modified configurations are not applied until the bus controller is restarted.

15.6 Uploading the configuration

On the integrated web page for the bus controller, it is possible to navigate to the "Upload Configuration" sub-item under Configuration. This allows the configuration that is currently stored in flash memory on the bus controller to be uploaded.



The "Upload" button uploads the configuration selected in "Select a configuration assembly" from the bus controller.

16 The Telnet interface

Telnet is a client/server protocol that uses TCP for data transfer (normally on port 23).
The Telnet interface on the EtherNet/IP bus controller provides a generic interface that can be used for the following:

- Changing network settings such as IP address, subnet mask, gateway, DHCP configuration, etc.
- Reading and writing any CIP attributes
- Executing services

Parameters can be specified in hexadecimal (0x) or decimal form.

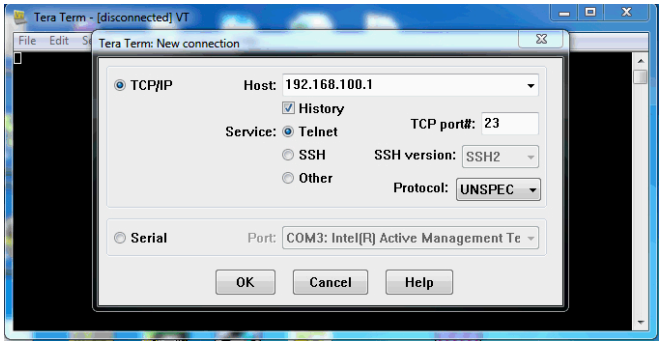
In addition, the interface includes several shortcut commands, e.g. "Save data to flash" and "Flash erase".

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.

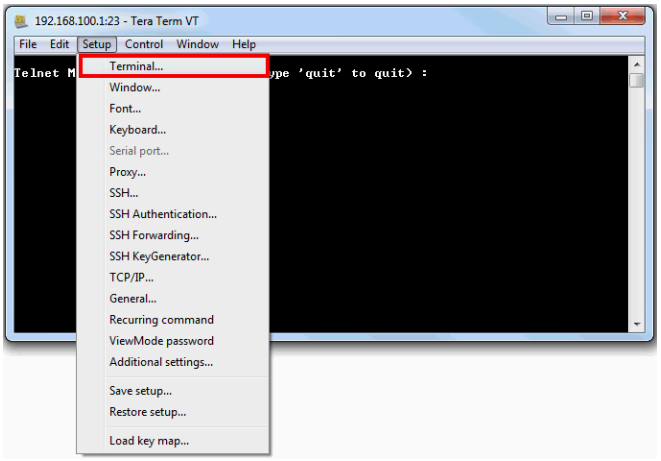
In Windows, Telnet can be launched by opening a command prompt (Windows Start menu / Run / "cmd") and typing "telnet" followed by the IP address of the bus controller (e.g. "telnet 192.168.100.1").

Example of settings for the TeraTerm client:

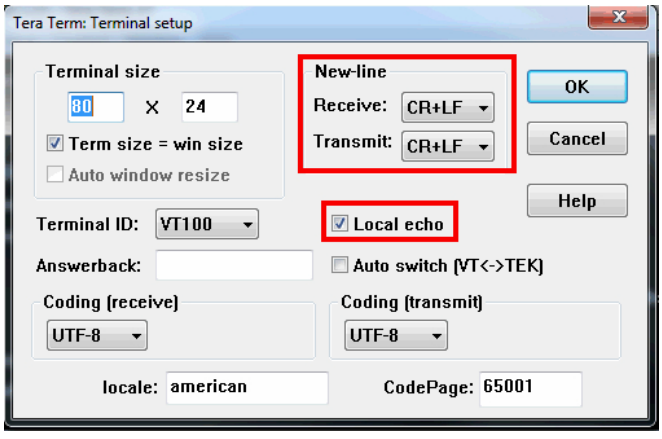
1: Entering the bus controller IP address and port number



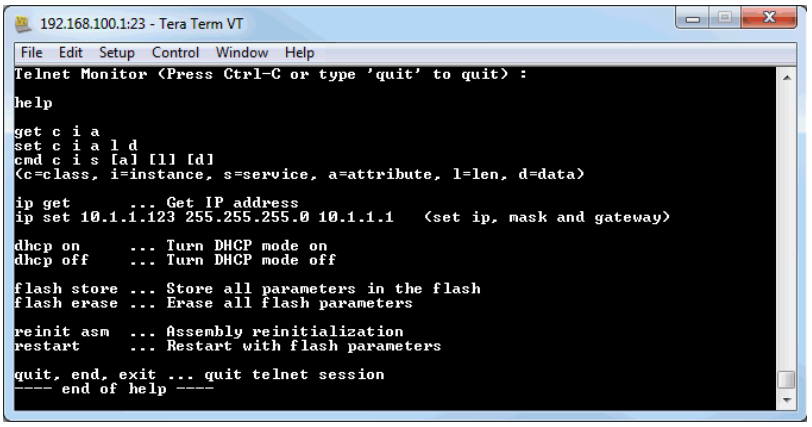
2: Selecting the terminal function



3: Important basic settings



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



16.1 Overview of Telnet commands

Usage	Commands	Acronyms
Shortcuts	"ip get" "ip set" "dhcp on/off" "flash store" "flash erase" "reinit asm" "restart"	
Reading and writing CIP attributes	"get c i a" "set c i a"	"c": For the class "i": For the instance "a": For the CIP attribute The following must be specified for write access: "d": For the data "l": Length of the data in bytes
Executing services	"cmd c i s"	"s": For the CIP service code "a": For the attribute. This is optional. If parameter data is not used but still necessary, "0" must be specified. "l2: For the length of the service parameters in bytes "d": For the parameter data

Examples

The example in section "Reading I/O module registers" on page 60 looks like this as a Telnet command:

cmd 0x65 2 0x32 0 2 0x12 → Returns "0x0000000A (10)" in normal cases.

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

16.2 Usage examples

16.2.1 Assigning an IP address

In addition to its other features, the Telnet interface makes it extremely easy to assign the bus controller an IP address. This is especially beneficial during initial startup since no additional tools are needed.

The only requirement is an Ethernet connection to the bus controller.

By setting the network address switch value 0xFF, the bus controller assigns itself IP address 192.168.100.1 after a restart. A Telnet connection can then be established by opening a command prompt window and entering "telnet 192.168.100.1".

Example

The IP address should be set to 192.168.1.123. Since the IP address is stored with the subnet mask and default gateway in a single attribute (class 0xF5, instance 1, attribute 5), execution looks like this:

→ ip set 192.168.1.123 255.255.255.0 192.168.1.254

This will disconnect the Telnet connection. If attribute 3 "Configuration control" of the TCP/IP object is set to the default setting of 0 during this, then the new network settings will be saved directly to flash memory.

The new settings will only take effect after the application is restarted. It is important to note that most network address switch values will change these settings. Settings from flash memory will only be applied if the switches are set to the value 0x00.

> **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 the "restart" command in Telnet or by briefly disconnecting the power supply.**

> **Information:**
>
> **Starting with firmware version 3.xx, the network parameters can only be changed with network address switch setting 0x00.**

16.2.2 Resetting to factory settings (clearing flash memory)

The easiest way to revert back to the factory settings is to delete flash memory.

This will delete all stored configurations, such as configuration assemblies 100 and 130 to 139. The bus controller then operates the connected I/O modules with the default settings of the automatic configuration.

The Telnet command is this:

flash erase

The Telnet console will display "OK" as confirmation. Restarting the bus controller will then load the factory settings from flash memory to RAM.

16.2.3 Querying I/O assembly lengths

The bus controller supports the reading of attribute 4 of the (CIP class 0x4) assembly object. This allows the lengths of the respective I/O assemblies to be queried.

Usually, the combination input and output assemblies are used for I/O communication. These are:

Combination output assembly: Instance 112 (0x70)

Combination input assembly: Instance 124 (0x7C)

With the queries

get 4 112 4 and

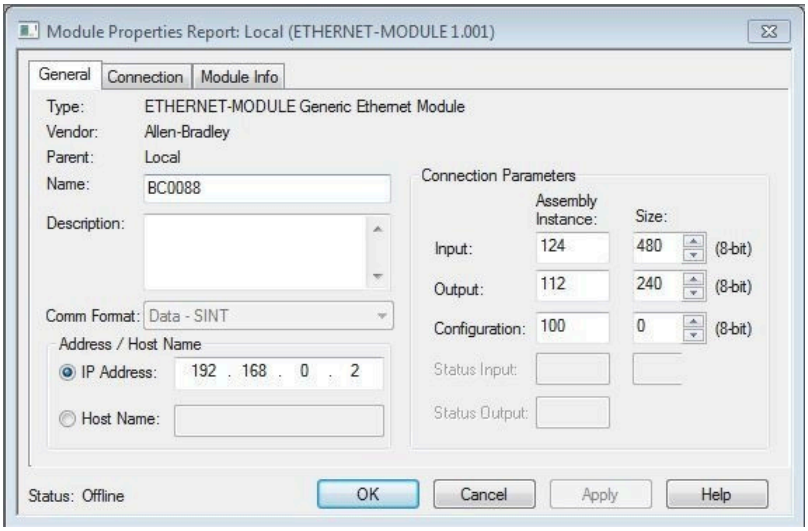
get 4 124 4, the respective assembly lengths are retrieved in bytes.

Default settings:

get 4 112 4 → 0x00F0 (240)

get 4 124 4 → 0x01E0 (480)

These specifications can be used to integrate the bus controller as a "generic Ethernet module" in the corresponding development environment. The data format "Comm format" must be set to bytes for this (i.e. data type "SINT").



17 Configuration examples for Rockwell RSLogix and B&R Automation Studio

The bus controller can be configured in 2 different ways:

- The default or automatic configuration permits the programmer to stay within the RSLogix 5000 development environment. No additional software is necessary. Each module connected to the bus controller is configured with default settings and the I/O data points are addressed by their corresponding byte offsets. Settings that go beyond the default settings must be made using either "explicit messages" in the RSLogix 5000 or a direct connection to the modules, e.g. via the Telnet interface of the bus controller.
- The manual configuration in Automation Studios provides more configuration options for the programmer. Modules can be set beyond the default configuration and I/O data points can be assigned a descriptive name.
  Automation Studio creates a .L5K file that can be imported directly into RSLogix.

17.1 Automatic configuration in Rockwell RSLogix

The default or automatic configuration is carried out with RSLogix 5000 using the default settings stored within the individual I/O modules. If required, these settings can be changed later using explicit messages or a direct connection to the bus controller.

This mode is recommended for systems that either have a limited number of digital and analog I/O data points, or where the default setting of the modules is sufficient. Additional modules can be added at any time with no or very little changes to the system.

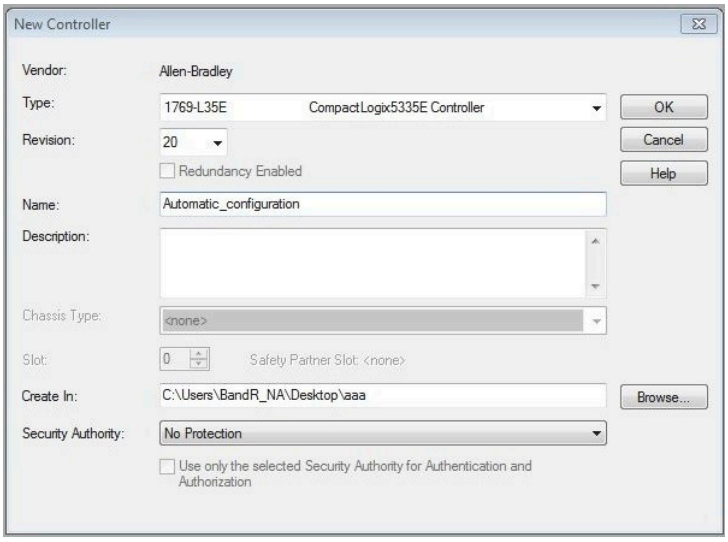
Software and hardware used for this example:

- X20BC0088 B&R EtherNet/IP adapter bus controller
- Rockwell CompactLogix_1769_L35E controller as EtherNet/IP scanner
- Rockwell RSLogix 5000 V20

17.1.1 Creating a new project

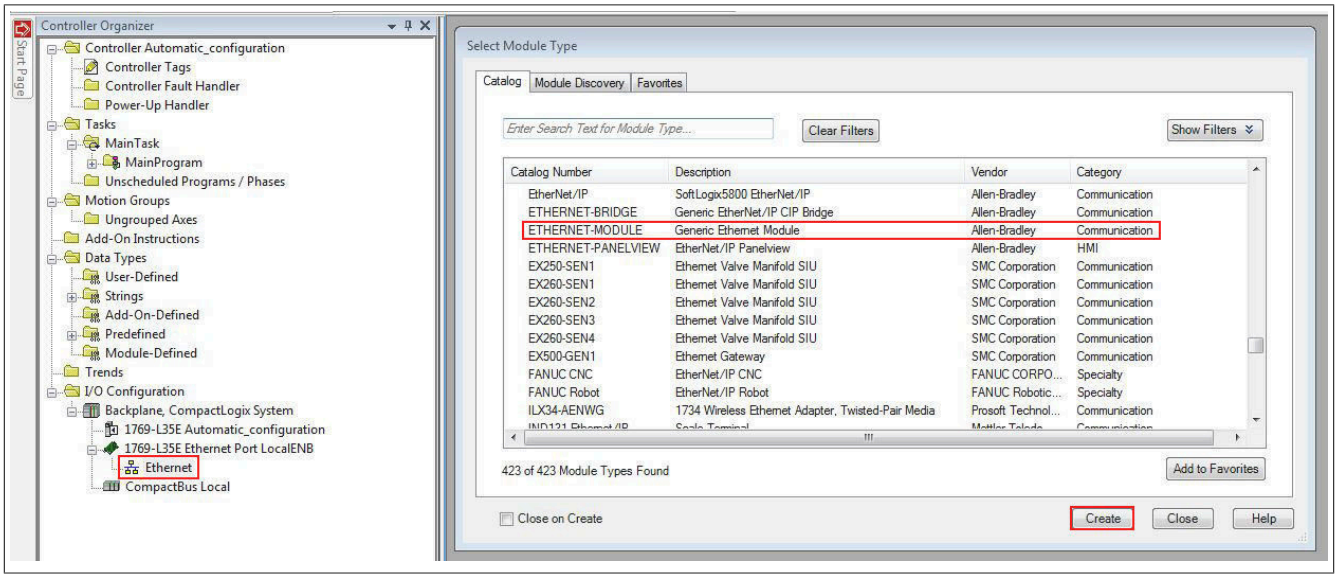
● If necessary, a new project can be created after opening the RSLogix 5000 development environment.

To do this, *File → New* is selected, and the controller type, controller revision, name and path of the new project are specified.



17.1.2 Adding and configuring the EtherNet/IP adapter

● A new module can be added by right-clicking on the Ethernet section of the controller and selecting "New module". The generic Ethernet module "ETHERNET-MODULE", for example, is added here.



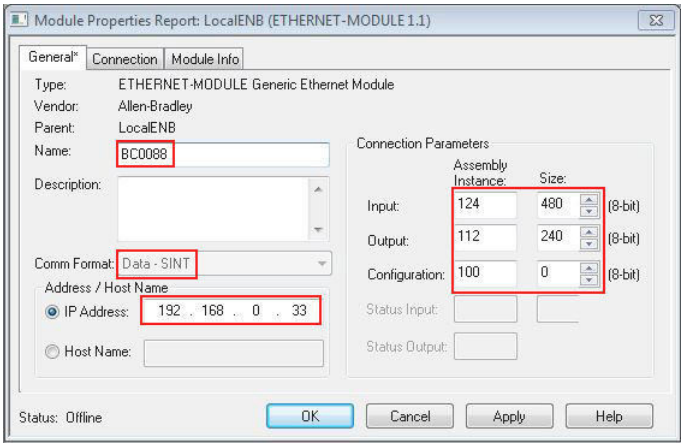
● Define module properties:

- Specify the name of the bus controller module

- Set *Comm format* to "Data - SINT". (Other settings require an adjustment of the assembly sizes.)

- Set the IP address. (For possible options, see "Overview of network address switch functionality" on page 24 and "Configures network settings" on page 23.)

- Set assembly instance and size. (See "Explanation of I/O assemblies" on page 77.)



17.1.3 Explanation of I/O assemblies

Configuration assembly

Instance ID	Type	Description	Size in bytes
100, 0x64	Base	Configuration of the I/O modules	400[1]

1 400 bytes is the maximum configuration size. With a default configuration, this size must be set to 0, since no configuration data is sent to the modules.

Output assemblies

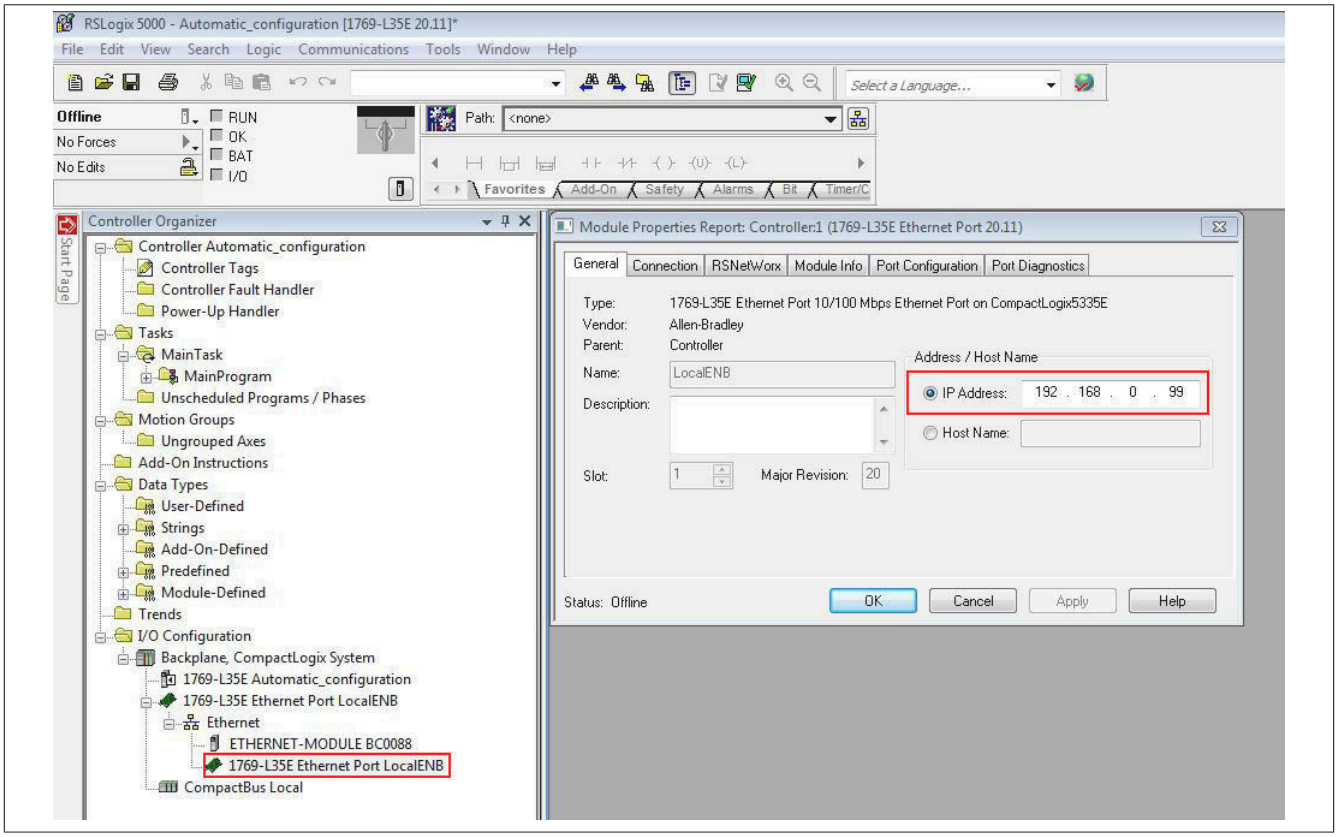
Instance ID	Type	Description	Default size in bytes
110, 0x6E	Base	Analog output	120
111, 0x6F	Base	Digital output	120
112, 0x70	Combined	Analog + digital output	240

Input assemblies

Instance ID	Type	Description	Default size in bytes
120, 0x78	Base	Analog input	120
121, 0x79	Base	Digital input	120
122, 0x7A	Base	Network status	120
123 0x7B	Base	Output status	120
124, 0x7C	Combined	Analog input + Digital input + Network status + Output status	480

17.1.4 Assign IP address

● Right-clicking on the local Ethernet port of the controller opens the Properties window. The IP address of the EtherNet/IP scanner is assigned here. This IP address must be identical to the local IP address of the controller.



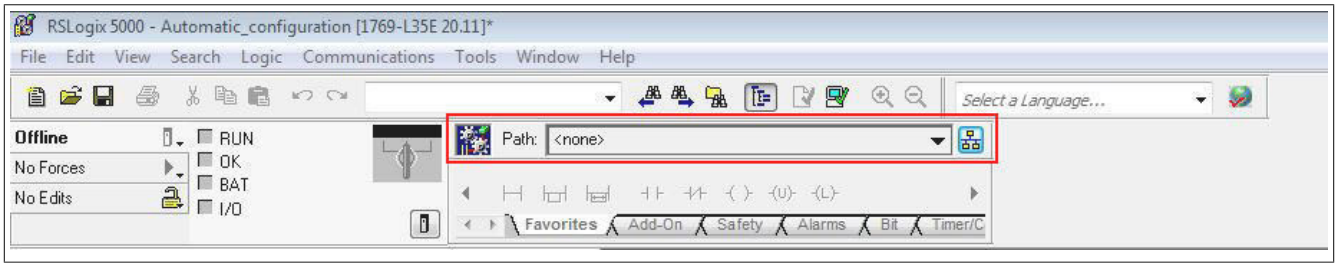
17.1.5 Establishing a connection

Information:

To connect the controller to RSLogix, the controller must already have a valid IP address. Setting the IP address of the controller is done differently depending on the controller used and must be looked up in the respective controller documentation.

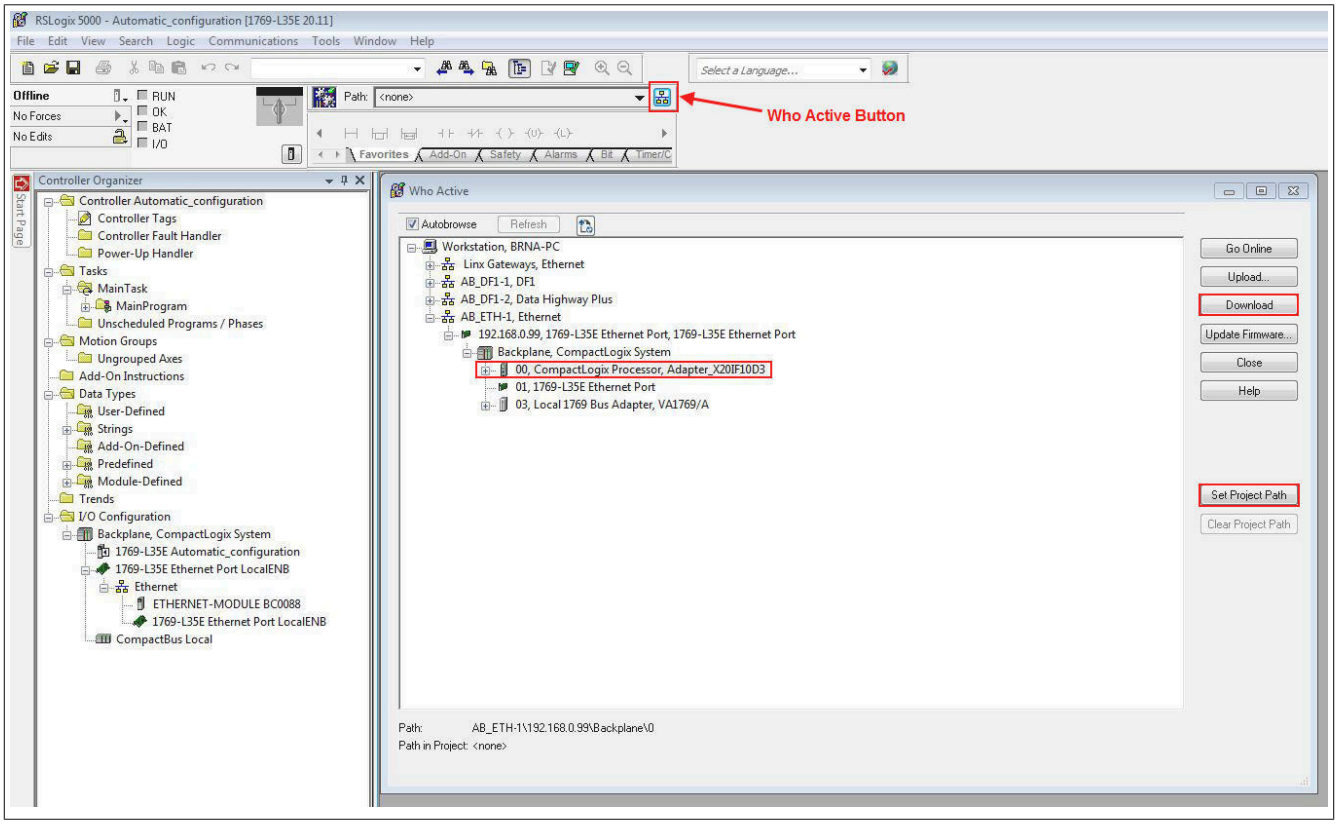
● To connect RSLogix to the controller, the path to the controller must be defined in RSLogix.

If the path to the controller has not yet been created, it must be created using RSLinx. For details, see "Creating a path with Rockwell RSLinx" on page 90.



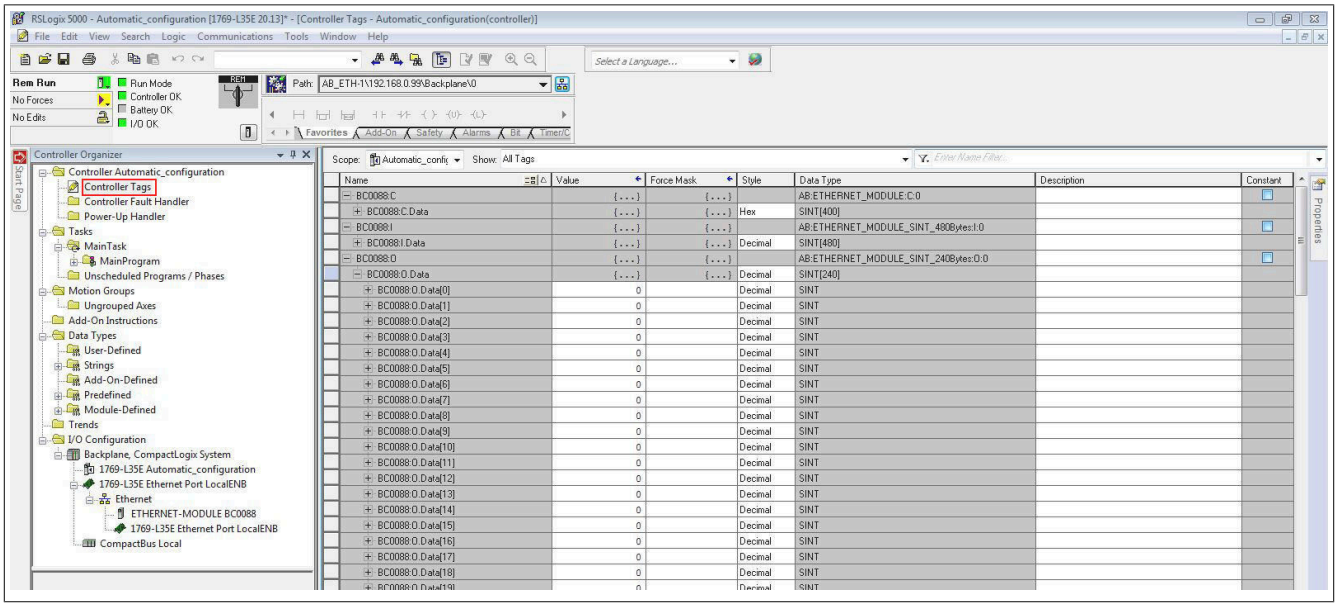
● If the path is valid, the addresses of all available EtherNet/IP scanners (controller) and adapters become visible after clicking on button **Who active**.

Open the desired EtherNet/IP scanner wide enough so that the corresponding processor is visible; mark it, click on **Set project path** and download the project.



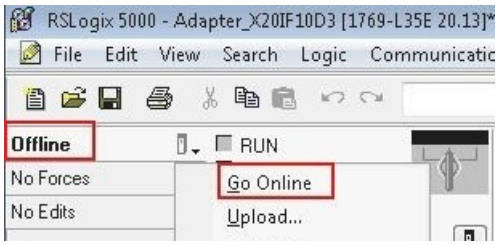
17.1.6 Reading and setting inputs/outputs of the EtherNet/IP adapter

- The configured outputs of the EtherNet/IP adapter can be set and the inputs can be read under *Controller tags*.

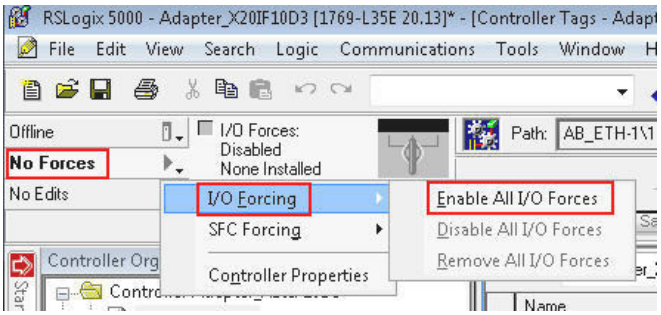


Online mode must be enabled for this. In order to force outputs, **Enable all I/O forces** must also be enabled.

Enable online mode.



Enable "Force".



Digital output

The default offset for digital outputs is *BC0088:O.Data[120]*.

The first digital output is addressed with *BC0088:O.Data[120].0*, the second with *BC0088:O.Data[120].1* and so on.

Name	Value	Force Mask	Style	Data Type
BC0088:O.Data[118]	0		Decimal	SINT
BC0088:O.Data[119]	0		Decimal	SINT
BC0088:O.Data[120]	2#0101_0101		Binary	SINT
BC0088:O.Data[120].0	2#1		Binary	BOOL
BC0088:O.Data[120].1	2#0		Binary	BOOL
BC0088:O.Data[120].2	2#1		Binary	BOOL
BC0088:O.Data[120].3	2#0		Binary	BOOL
BC0088:O.Data[120].4	2#1		Binary	BOOL
BC0088:O.Data[120].5	2#0		Binary	BOOL
BC0088:O.Data[120].6	2#1		Binary	BOOL
BC0088:O.Data[120].7	2#0		Binary	BOOL
BC0088:O.Data[121]	0		Decimal	SINT

Digital input

The default offset for digital inputs is *BC0088:I.Data[120]*.

The first digital input is addressed with *BC0088:I.Data[120].0*, the second with *BC0088:I.Data[120].1* and so on.

Name	Value	Force Mask	Style	Data Type
BC0088:I.Data[118]	0		Decimal	SINT
BC0088:I.Data[119]	0		Decimal	SINT
BC0088:I.Data[120]	2#0101_0101		Binary	SINT
BC0088:I.Data[120].0	2#1		Binary	BOOL
BC0088:I.Data[120].1	2#0		Binary	BOOL
BC0088:I.Data[120].2	2#1		Binary	BOOL
BC0088:I.Data[120].3	2#0		Binary	BOOL
BC0088:I.Data[120].4	2#1		Binary	BOOL
BC0088:I.Data[120].5	2#0		Binary	BOOL
BC0088:I.Data[120].6	2#1		Binary	BOOL
BC0088:I.Data[120].7	2#0		Binary	BOOL
BC0088:I.Data[121]	0		Decimal	SINT

Analog output

The default offset for analog outputs is *BC0088:O.Data[0]*. Each analog output is represented by an integer or 2 SINT values.

The first analog output is addressed by SINTs *BC0088:O.Data[0]* and *BC0088:O.Data[1]*, the second by SINTs *BC0088:O.Data[2]* and *BC0088:O.Data[3]* and so on.

Name	Value	Force Mask	Style	Data Type
BC0088:C	{...}	{...}		AB:ETHERNET_MODULE:C:0
BC0088:I	{...}	{...}		AB:ETHERNET_MODULE_SINT_480Bytes:I:0
BC0088:O	{...}	{...}		AB:ETHERNET_MODULE_SINT_240Bytes:O:0
BC0088:O.Data	{...}	{...}	Decimal	SINT[240]
BC0088:O.Data[0]	0		Decimal	SINT
BC0088:O.Data[1]	0		Decimal	SINT

Analog input

The default offset for analog inputs is *BC0088:I.Data[0]*. Each analog input is represented by an integer or 2 SINT values.

The first analog input is addressed by SINTs BC0088:I.Data[0] and BC0088:I.Data[1], the second by SINTs BC0088:I.Data[2] and BC0088:I.Data[3] and so on.

Name	Value	Force Mask	Style	Data Type
BC0088:C	{...}	{...}		AB:ETHERNET_MODULE:C:0
BC0088:I	{...}	{...}		AB:ETHERNET_MODULE_SINT_480Bytes:I:0
BC0088:I.Data	{...}	{...}	Decimal	SINT[480]
BC0088:I.Data[0]	0		Decimal	SINT
BC0088:I.Data[1]	0		Decimal	SINT

17.2 Manual configuration in B&R Automation Studio

Automation Studio provides an interface with which all B&R I/Os can be individually configured and named.

When the configuration is complete, an .L5K file is generated. This file can be imported into the Rockwell engineering environment (RSLogix 5000).

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.

17.2.1 Creating a project

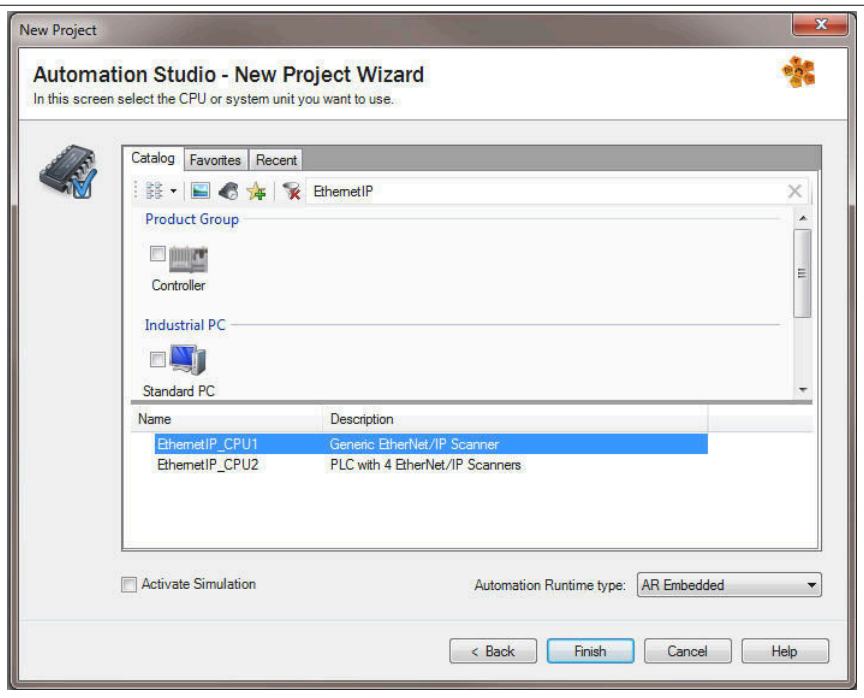
Once Automation Studio is started, a new project can be created by selecting *File → New project* or the corresponding icon. The wizard for creating new projects opens. The project name is specified here and the path to the project directory is displayed.

● In the next step, the name of the configuration can be selected and a new configuration created with **Continue**. The setting for the hardware configuration must be left at "Define a new hardware configuration manually".

● In the next dialog box, the target system is selected. This target system corresponds to the EtherNet/IP scanner and acts as a connection point for the B&R bus controller.
After entering the keyword "EthernetIP" in the search window, 2 types of scanners appear:

- **EthernetIP_CPU1:** Controllers with 1 EtherNet/IP scanner
- **EthernetIP_CPU2:** Controllers with multiple EtherNet/IP scanners (EtherNet/IP communication modules). Up to 4 modules are supported.

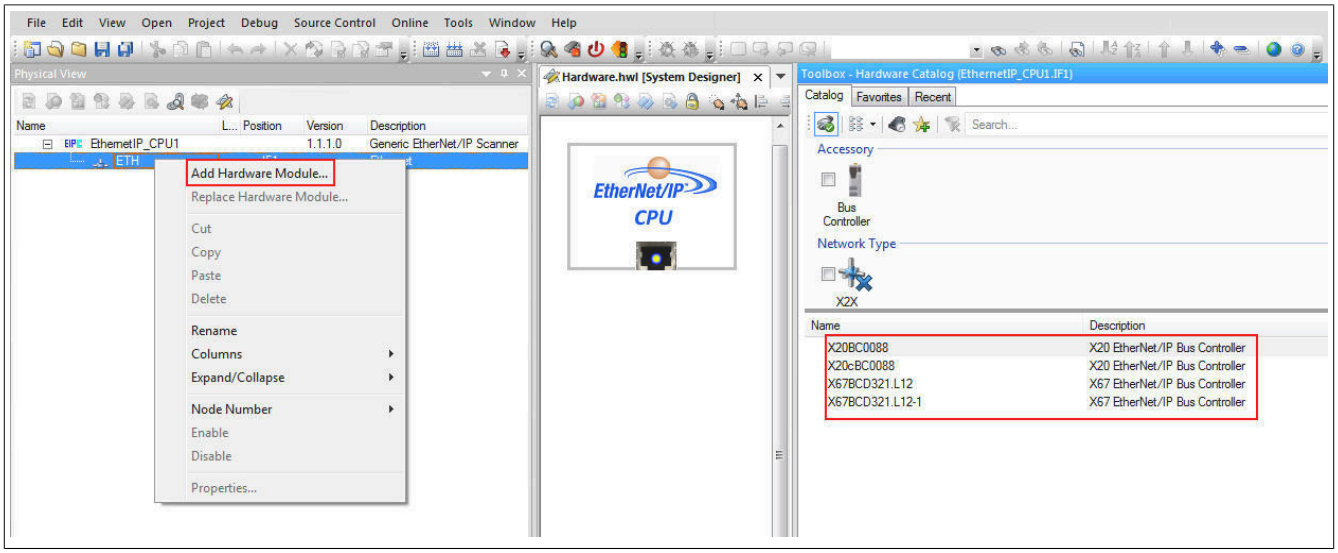
The second controller type (EthernetIP_CPU2) is used for specific Allen-Bradley controllers where multiple independent EtherNet/IP scanners can be used in one controller. In all other cases, the first controller type (EthernetIP_CPU1) must be used.



● Clicking on **Finish** closes the wizard and creates the new project.

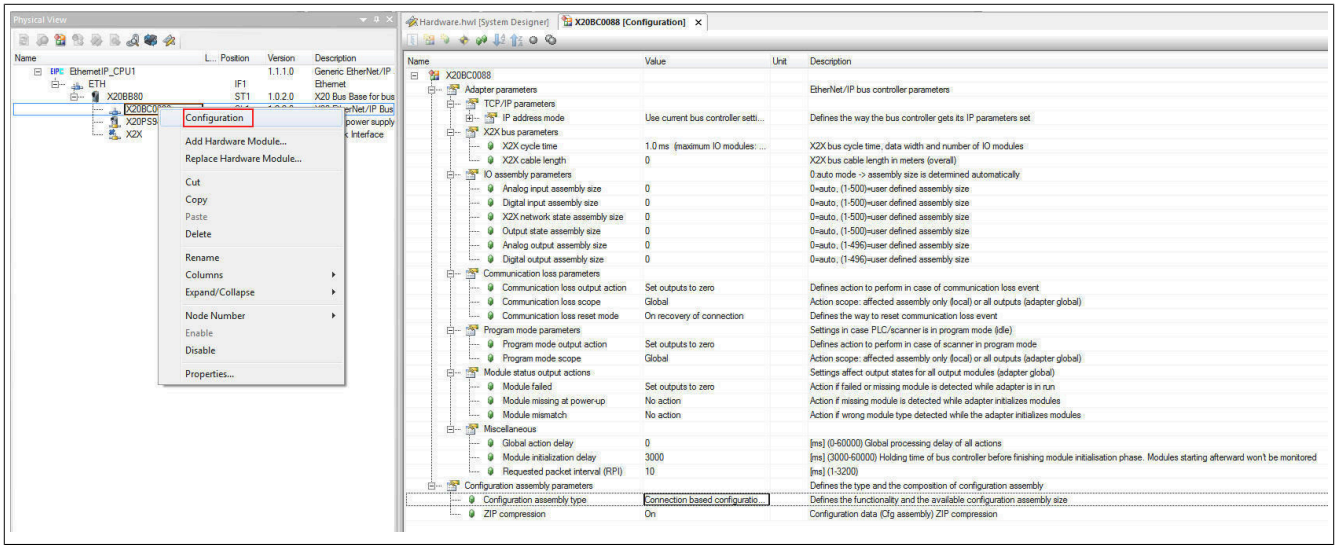
17.2.2 Adding and configuring EtherNet/IP bus controllers

● The adapter (bus controller) is added to the interface of the EtherNet/IP CPU. The desired bus controller can be selected by right clicking and selecting **Add hardware modules**.



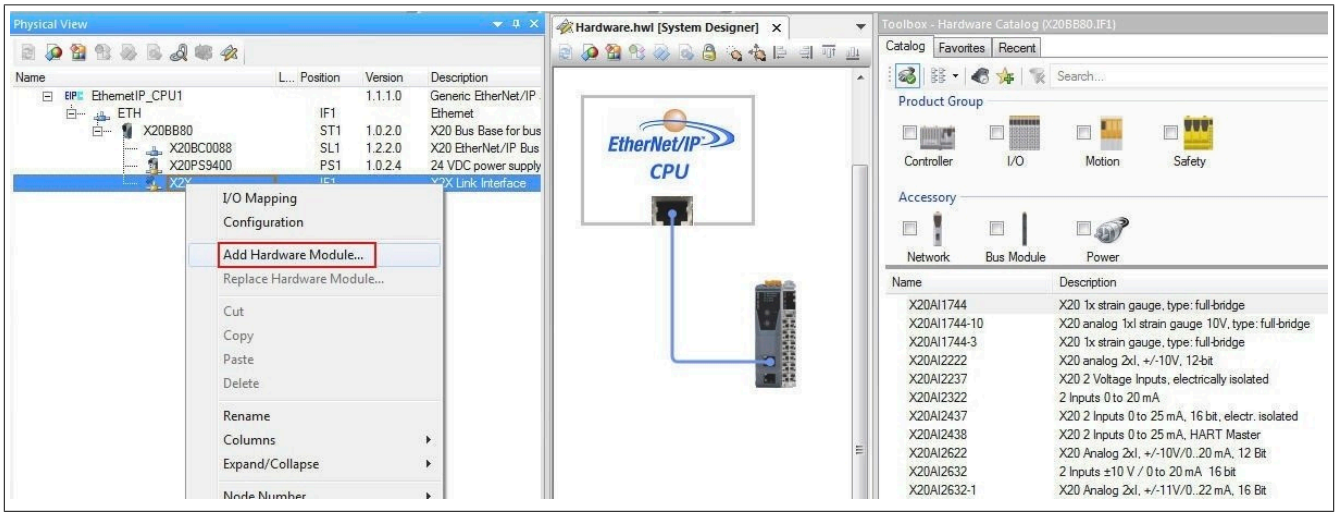
● The EtherNet/IP adapter is configured by selecting **Configuration** from the shortcut menu of the bus controller in the Physical View.

For a description of configuration parameters, see Automation Help.



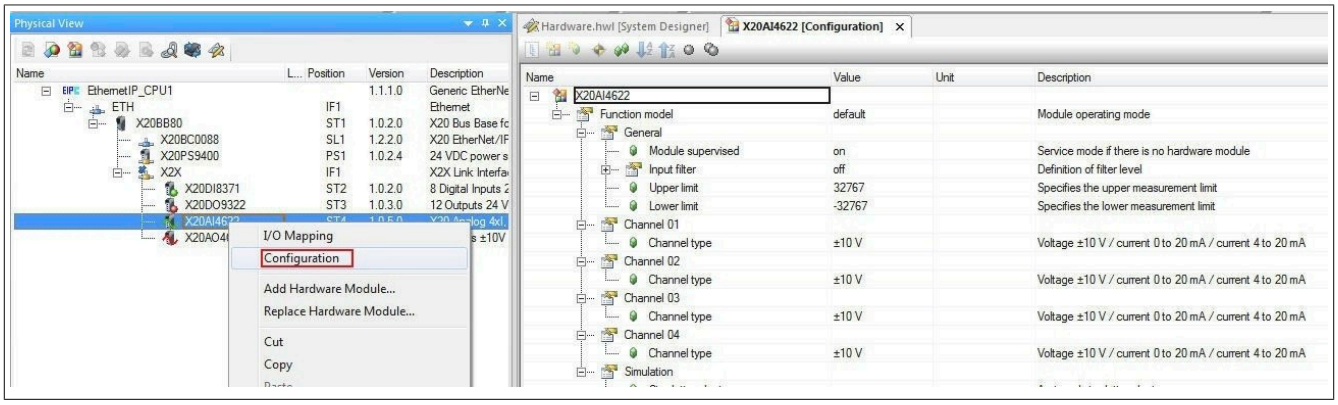
17.2.3 Adding and configuring I/O modules

● The I/O modules are added to the bus controller on the X2X interface. The desired I/O modules can be selected by right-clicking and selecting **Add hardware module**.

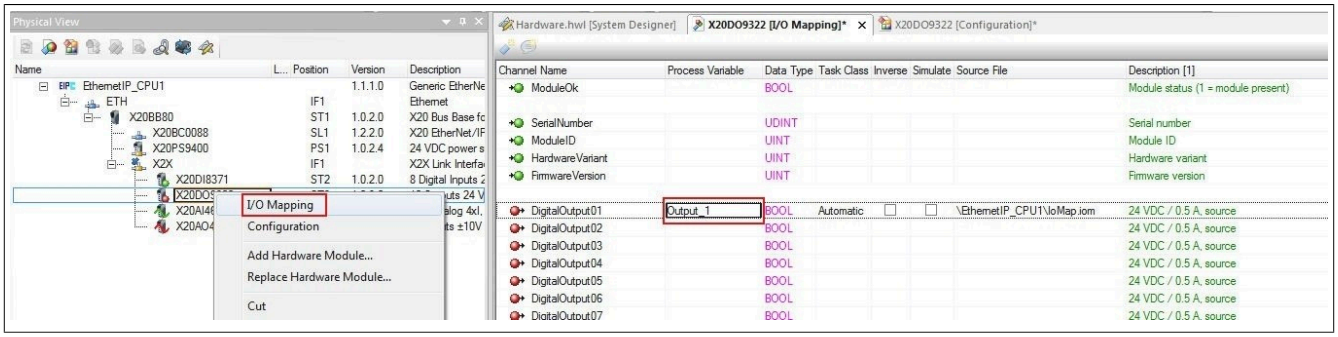


● Now the module-specific configuration can be carried out for each I/O module. The configuration is done by selecting **Configuration** in the shortcut menu of the respective I/O module in the Physical View.

For the description of the configuration parameters, see the data sheet of the respective I/O module.

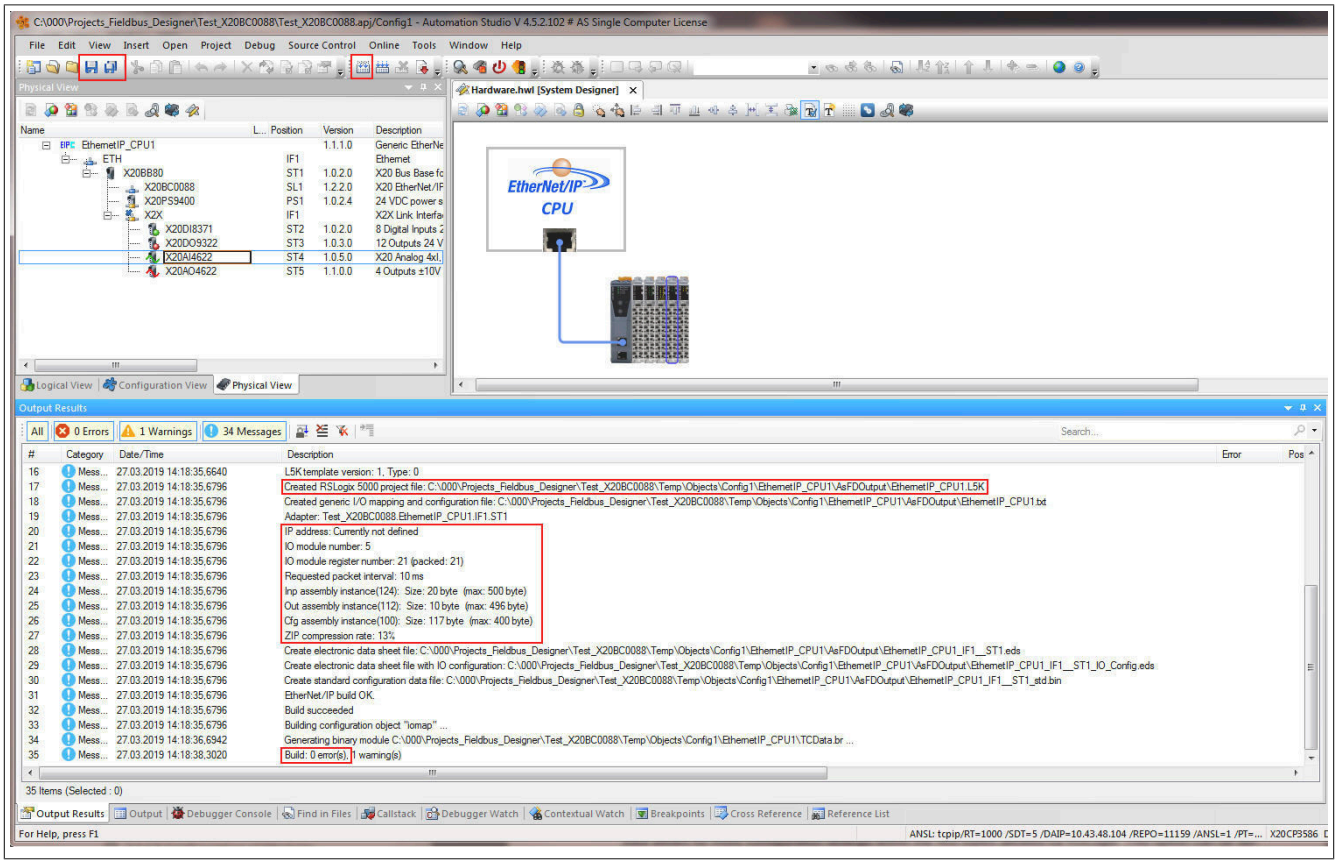


● The cyclic data points are displayed via shortcut menu **I/O mapping**. The default name of each data point is displayed under "Channel name" (e.g. DigitalOutput01). If another name is added to column "Process variable", this name is then used instead of the default name (e.g. Output_1).



17.2.4 Creating the L5K configuration file

● When all necessary I/O modules have been added to the bus controller and configured, the generation process for the configuration file can be started. The project is saved at this point, and the generation process is started.



● The "Output results" window displays information about the generated file:

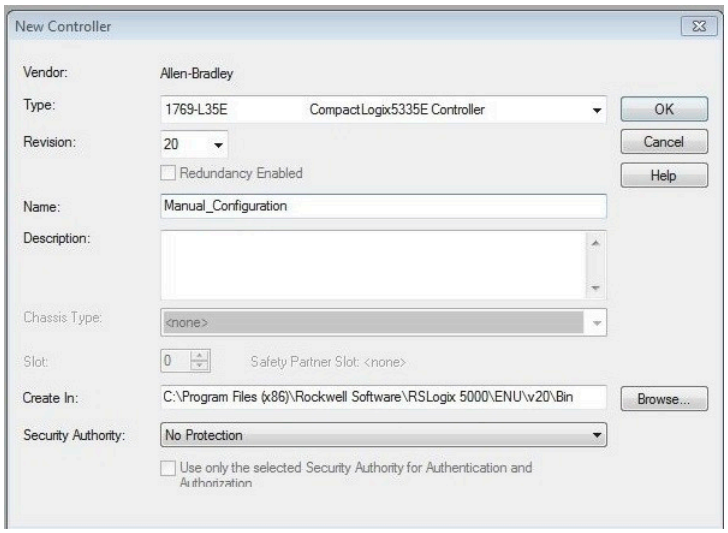
- Storage location of the generated configuration file
- IP address
- Number of I/O modules
- Requested packet interval (RPI) time
- Assembly sizes: If an assembly size exceeds the maximum size, an error is displayed.
- ZIP compression rate: The ZIP compression rate refers to the compression rate of the configuration assembly. Packed configuration data permits more configuration settings within the 400 byte limit permitted by RSLogix. This option can be disabled in the bus controller configuration dialog box.
- Number of errors occurred

17.3 Importing the L5K configuration file into Rockwell RSLogix

17.3.1 Creating a new project

● If necessary, a new project can be created after opening the RSLogix 5000 development environment.

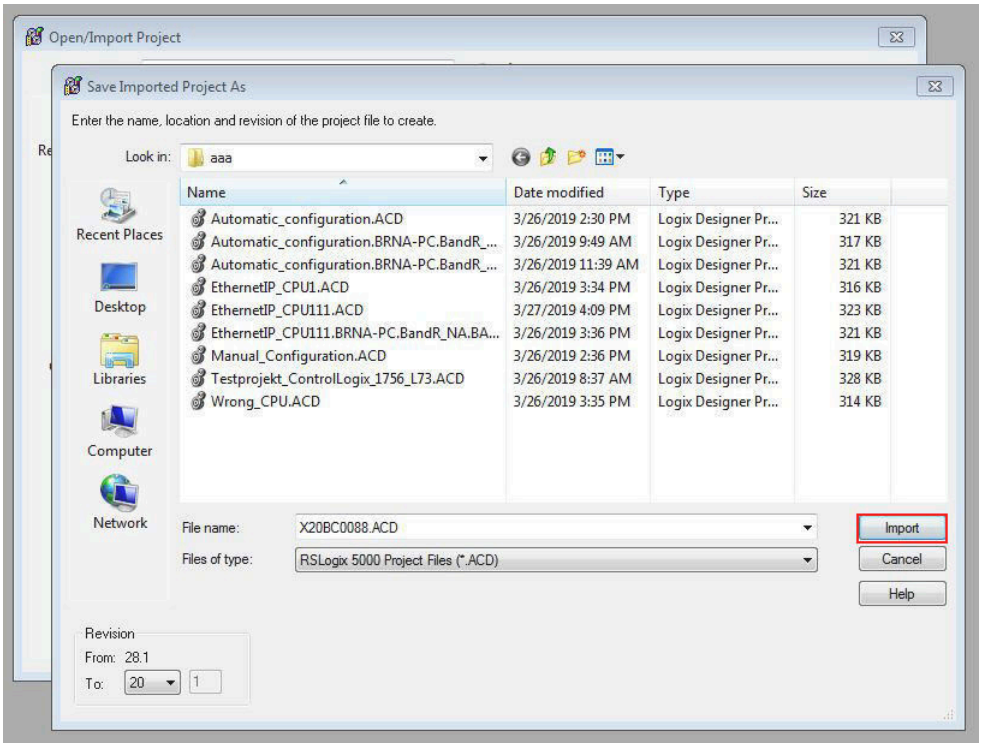
To do this, *File* → *New* is selected, and the controller type, controller revision, name and path of the new project are specified.



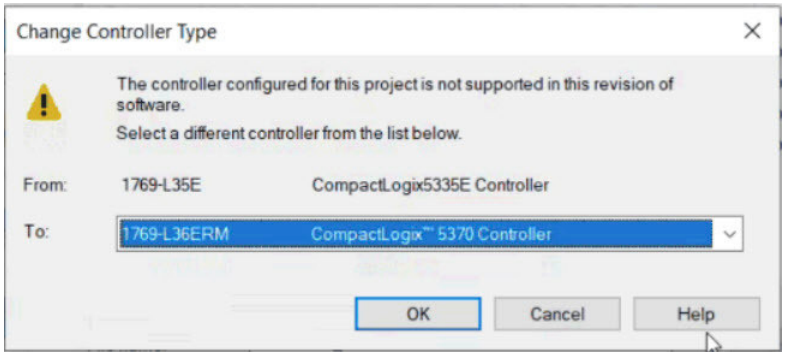
17.3.2 Importing the L5K configuration file

● The L5K configuration file generated in Automation Studio must then be imported into the development environment.

● The desired .L5K file can be selected via *File* → *Open*. Confirm with **Import**.

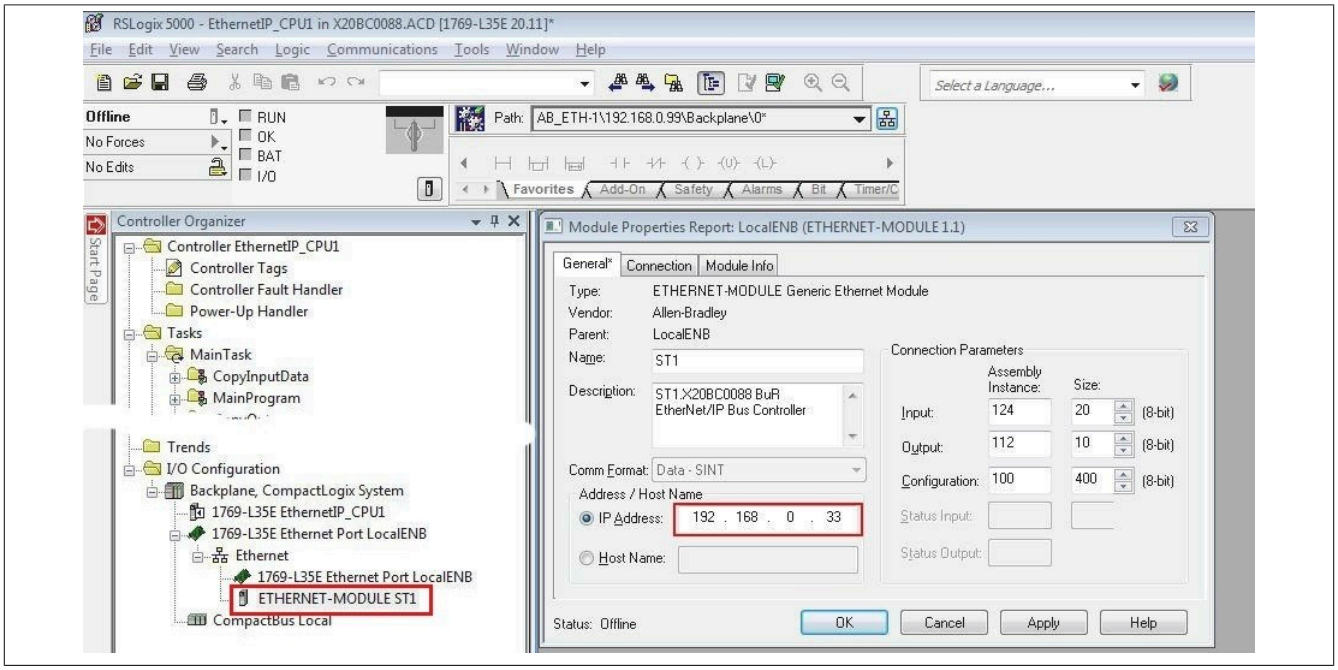


● If the controller type entered in the .L5K file is not supported by the development environment, it is recommended to replace the controller type with an offered alternative controller type.

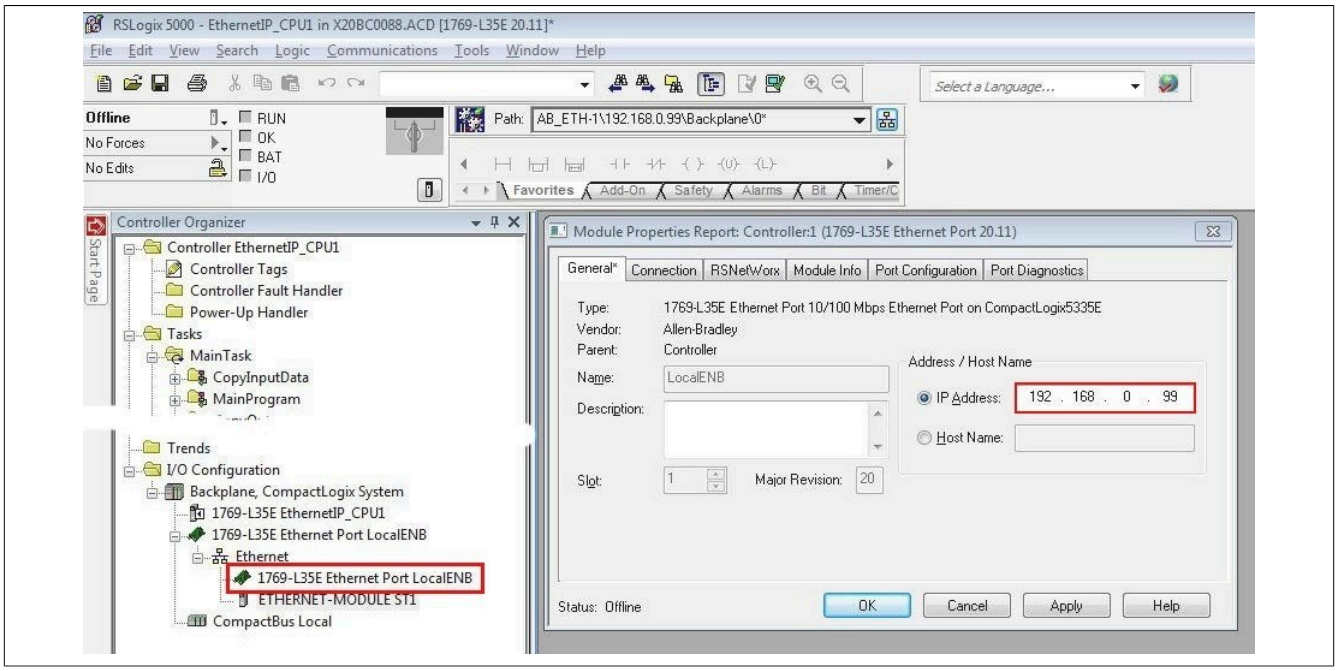


17.3.3 Assigning IP addresses

● Right-clicking on the added EtherNet/IP adapter opens the Properties window. The IP address of the EtherNet/IP adapter is adjusted here (if necessary).



● Right-clicking on the local Ethernet port of the controller opens the Properties window. The IP address of the EtherNet/IP scanner is assigned here. This IP address must be identical to the local IP address of the controller.



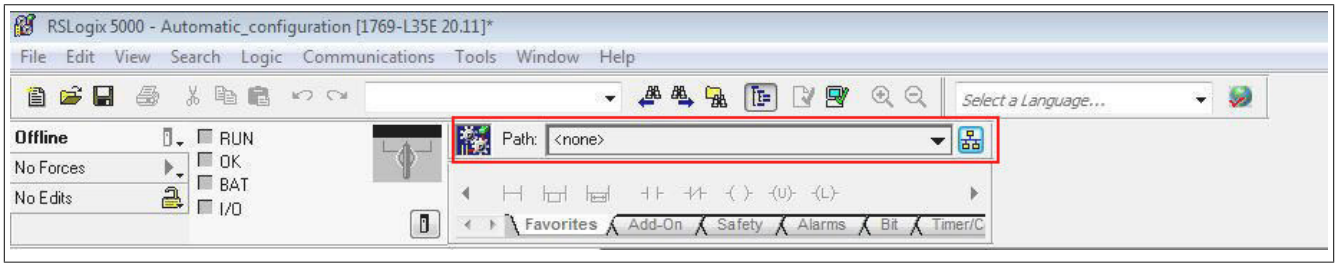
17.3.4 Establishing a connection to the controller and downloading the configuration

Information:

To connect the controller to RSLogix, the controller must already have a valid IP address. Setting the IP address of the controller is done differently depending on the controller used and must be looked up in the respective controller documentation.

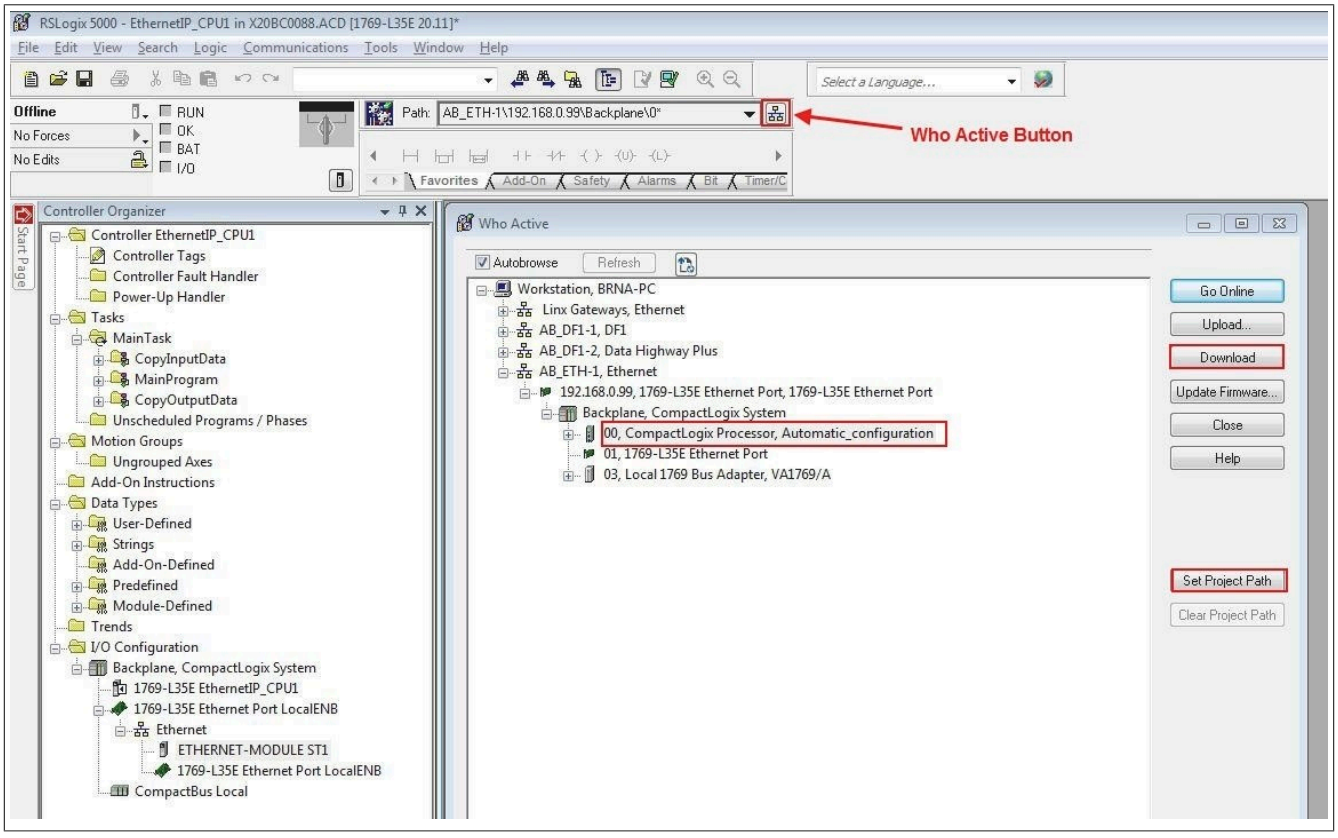
● To connect RSLogix to the controller, the path to the controller must be defined in RSLogix.

If the path to the controller has not yet been created, it must be created using RSLinx. For details, see "Creating a path with Rockwell RSLinx" on page 90.



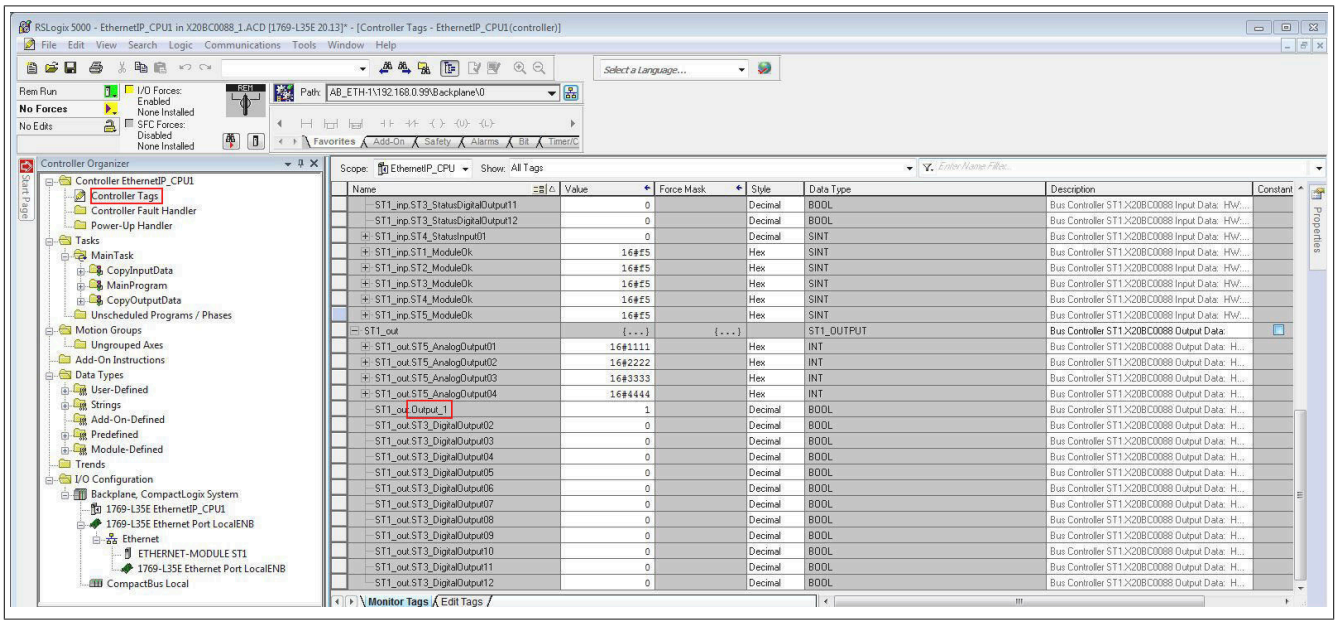
● If the path is valid, the addresses of all available EtherNet/IP scanners (controller) and adapters become visible after clicking on button **Who active**.

Open the desired EtherNet/IP scanner wide enough so that the corresponding processor is visible; mark it, click on **Set project path** and download the project.



17.3.5 Reading and setting inputs/outputs of the EtherNet/IP adapter

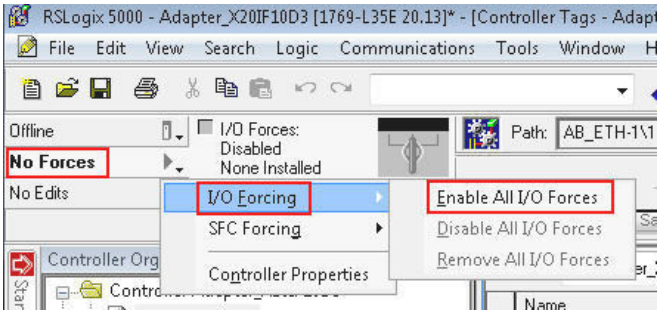
● The configured outputs of the EtherNet/IP adapter can be set and the inputs can be read under *Controller tags*. The symbolic names of the process variables that were previously changed in the I/O assignment of the individual modules in Automation Studio are also displayed here.



Online mode must be enabled for this. In order to force outputs, **Enable all I/O forces** must also be enabled.

Enable online mode.

Enable "Force".



Overview of monitor tags

Name	Information
copyInpCounter copyOutCounter mainCounter	Lifecycle counter
ST1:C	Configuration assembly
ST1:I	Input assembly
ST1:O	Output assembly Copied to ST1_out in every cycle of task CopyOutputData of the main program.
ST1_inp	Input data Copied to ST1:I in every cycle of task CopyInputData of the main program.
ST1_out	Output data

Information:

The I/O data should not be worked with directly since it is overwritten cyclically by the copy tasks.

17.4 Creating a path with Rockwell RSLinx

The RSLinx application is installed with the RSLogix 5000 installation.

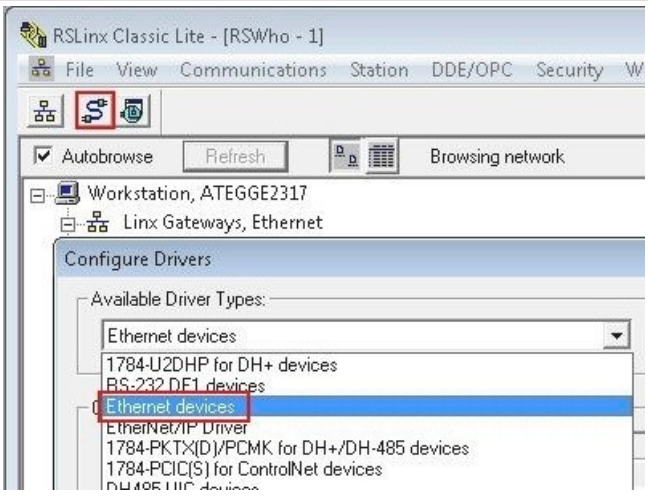
Starting the graphical user interface

> **Information:**
>
> **The RSLinx graphical user interface might not start in certain operating systems such as Windows Vista, Windows 7 and Server 2008.**

If RSLinx Classic is running as a server, it is not possible to start the graphical user interface. This is only available if RSLinx Classic is in application mode.
To toggle between execution as a service and application mode, use the "RSLinx Classic Launch" control panel. This is available here:
Start → Programs → Rockwell software → RSLinx → RsLinx Classic Launch control panel

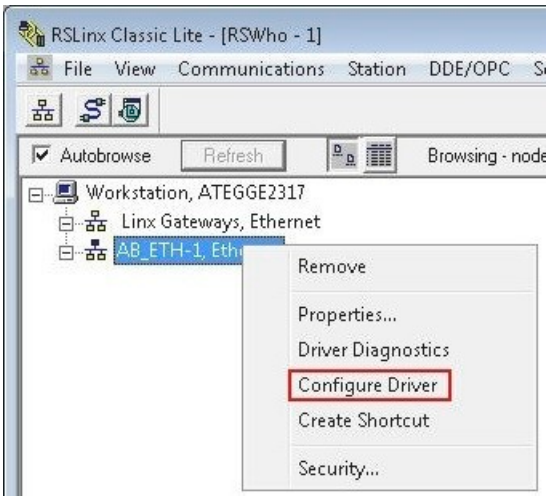
In order to disable **Always run as service**, you must first click on button **Stop**. Other Rockwell software may need to be closed before the service is ended.
Finally, **Always run as service** can be disabled and RSLinx can be started as an application by clicking on **Start**.

● To create the path, open **Configure drivers** and select "Ethernet devices" as the driver type.

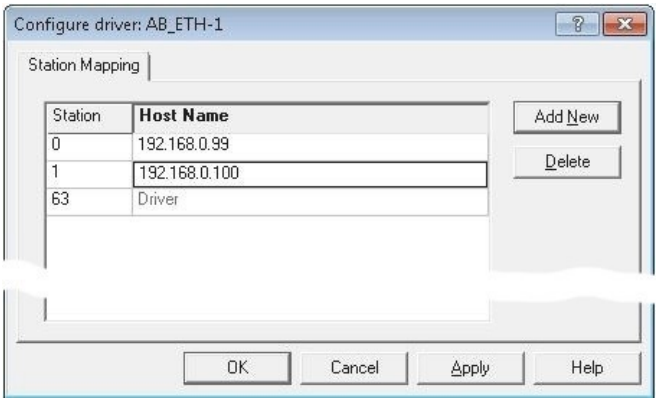


● Right-click on the newly added driver type (AB_ETH1, Ethernet) and select **Configure driver**. In the configuration dialog box, enter the IP addresses of the EtherNet/IP scanner (controller) and the adapter.

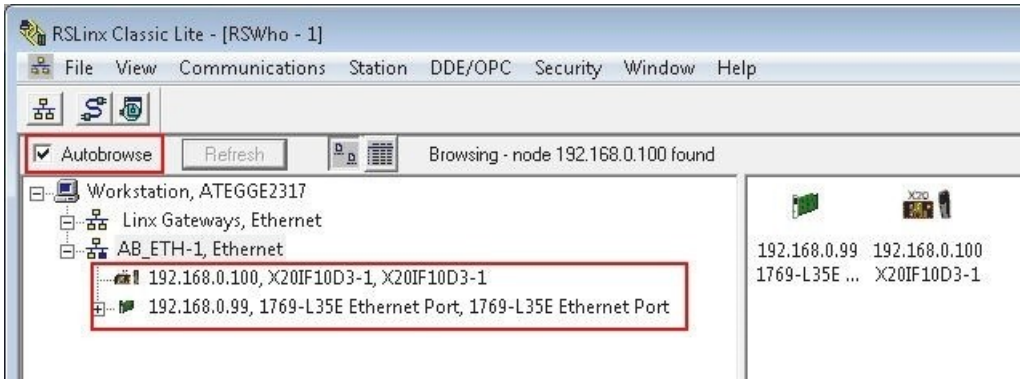
Select driver



Enter IP address



● If "Autobrowse" is enabled, both devices should be found shortly thereafter and displayed in RSLinx.



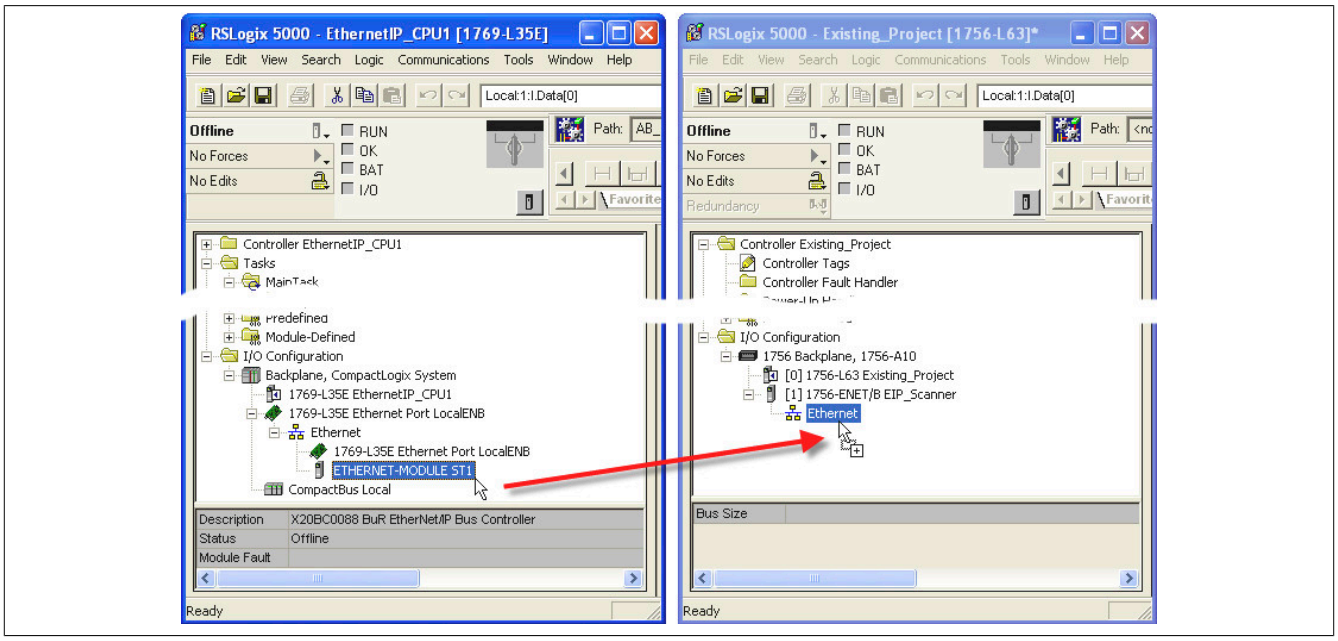
● Close RSLinx again and continue working with RSLogix 5000.

17.5 Transferring a configuration between Rockwell RSLogix projects

The following steps describe how bus controller information can be transferred between different RSLogix 5000 projects. This makes it easier, for example, to integrate the bus controller into new projects or to change the controller.

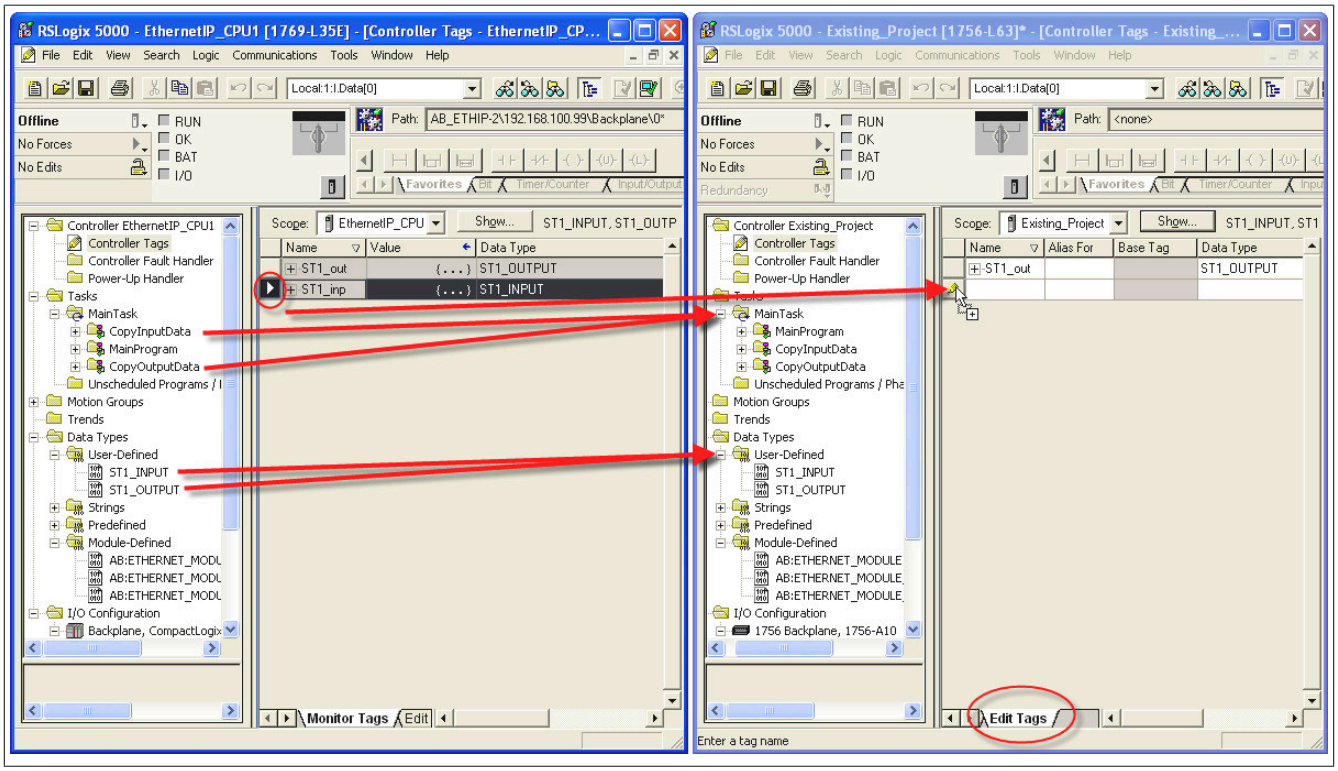
● Open a second instance of RSLogix 5000. The first instance contains the information to be copied. The second instance contains the new or existing project that the information will be copied into.

● Click in the project on the ETHERNET-MODULE ST1 whose configuration (L5K file) was created with Automation Studio and drag it into the new project. This copies the bus controller and its assemblies to the new project.



● If the names of the I/O channels are to be retained, the next step is to transfer the user-defined data types. Like the bus controller, the data types are transferred between the open instances using drag-and-drop.

● Define controller tags for the input and output assemblies. Controller tags can be defined or dragged and dropped between instances. The life cycle counters (copyInpCounter, copyOutCounter and mainCounter) are also transferred.

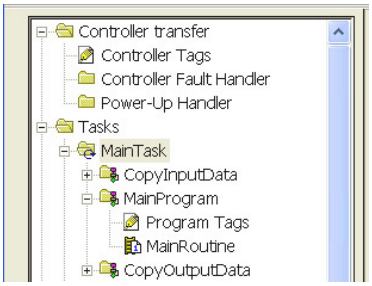


● Transfer the programs between the projects. Like bus controllers and data types, they can be transferred using drag-and-drop.

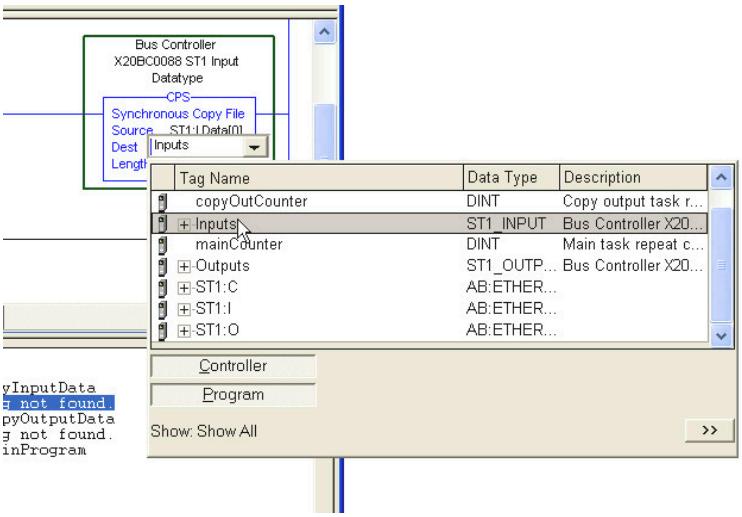
Information:

Changing the order is not permitted.

- **Copy inputs**
- **Execute main program**
- **Copy outputs**



● Make sure that the I/O structures are correctly bound to the copy function blocks.



● The project transfer has been completed.