BUCHER automation

User Manual



60880627_02

JetControl 365MC

Controller

This document has been compiled by Bucher Automation AG with due diligence based on the state of the art as known to them. Any revisions and technical advancements of our products are not automatically made available in a revised document.
Bucher Automation AG shall not be liable for any errors either in form or content, or for any missing updates, as well as for any damage or detriment resulting from such failure.

Bucher Automation AG
Thomas-Alva-Edison-Ring 10
71672 Marbach/Neckar, Germany
T +49 7141 2550-0
info@bucherautomation.com

Technical support
T +49 7141 2550-444
support@bucherautomation.com

Sales
T +49 7141 2550-663
sales@bucherautomation.com

www.bucherautomation.com

Translation of the german original User Manual

Revision	1.32.2
Date of issue	6/10/2024

Table of Contents

1 Introduction ... 7
1.1 Information on this document ... 7
1.2 Typographical conventions ... 7
2 Safety ... 8
2.1 General Information ... 8
2.2 Purpose ... 8
2.2.1 Intended use ... 8
2.2.2 Usage other than intended ... 8
2.3 Warnings used in this document ... 9
3 Product Description ... 10
3.1 System overview ... 10
3.2 Design ... 11
3.3 Product features ... 12
3.4 Status indication ... 12
3.4.1 Diagnostics capabilities by means of status indication ... 13
3.4.2 LED states during the boot process ... 14
3.4.3 LED states after detection of an IP address conflict ... 15
3.5 Nameplate ... 15
3.6 Scope of Delivery ... 16
3.7 Order reference/options ... 16
4 Technical Specifications ... 18
4.1 Dimensions ... 18
4.2 Mechanical Specifications ... 19
4.3 Electrical properties ... 19
4.3.1 System power supply ... 21
4.4 Real-time clock ... 22
4.5 Environmental Conditions ... 22
4.6 EMC Values ... 23
4.6.1 Enclosure ... 23
4.6.2 Shielded data and I/O lines ... 23
4.6.3 DC power supply inputs and outputs ... 24
5 Mechanical Installation ... 25
5.1 Installing Device on DIN Rail ... 26
5.2 Removing Device from DIN Rail ... 27
5.3 Dismounting Enclosure from Backplane Module ... 28
6 Electrical Connection ... 29

6.1 Improving EMC Noise Immunity ... 29
6.2 Interfaces ... 30
6.2.1 Selector switch S11 - Mode selector ... 30
6.2.2 Terminal X10 - Power supply ... 31
6.2.3 Port X11 – Serial interface ... 32
6.2.4 Ports X14, X15 - Ethernet ... 34
6.2.5 Port X19 - CAN Bus ... 35
6.2.6 Slot X61 - SD Memory Card ... 39
6.3 Commissioning ... 41
6.3.1 Configuring error states ... 43
6.3.2 Configuration in JetSym ... 43
7 Identification and Configuration ... 46
7.1 Identification ... 46
7.1.1 Electronic Data Sheet (EDS) ... 46
7.1.2 Version registers ... 50
7.2 IP configuration ... 51
7.2.1 Factory settings ... 51
7.2.2 Determining the IP address of the controller using JetIPScan ... 52
7.2.3 Configuration memory ... 53
7.2.4 Configuration file "config.ini" ... 54
7.2.5 Configuration registers ... 58
7.2.6 Setting the IP address ... 59
7.2.7 Using names for IP addresses ... 70
7.3 Operating system ... 71
7.3.1 Operating system update of the controller ... 71
7.3.2 Operating system update of a JX2 or JX3 module ... 74
7.4 File System ... 77
7.4.1 Properties ... 77
8 Programming ... 79
8.1 Abbreviations, module register properties and formats ... 79
8.2 Storage options - Overview ... 80
8.2.1 Operating system memory ... 80
8.2.2 File system memory ... 80
8.2.3 Application Program Memory ... 80
8.2.4 Memory for volatile application program variables ... 82
8.2.5 Memory for non-volatile application program registers ... 83
8.2.6 Memory for non-volatile application program variables ... 84
8.2.7 Registers on I/O modules ... 85

8.2.8 Memory for non-volatile registers on the backplane module ... 86
8.2.9 Special registers ... 87
8.2.10 Inputs and outputs ... 88
8.2.11 Flag ... 90
8.3 Saving and loading an application program ... 92
8.4 Addressing expansion modules ... 92
8.4.1 Numbering registers and I/Os ... 93
8.5 System commands ... 98
8.6 Real-time clock ... 99
8.7 Startup delay time ... 99
8.8 System runtime registers ... 99
8.9 Monitoring interface activities ... 100
8.10 Ethernet system bus ... 100
8.11 Control of alphanumeric displays and HMIs ... 101
8.12 Controlling printer and serial interfaces ... 102
8.13 Project Engineering of a JX2 Station ... 102
8.14 Project Engineering of a JX3 Station ... 103
8.15 JCF-SV1 "simple servo function" ... 103
8.16 E-mail feature ... 104
8.17 Modbus/TCP ... 105
8.17.1 Modbus/TCP server ... 106
8.17.2 Modbus/TCP client with STX functions ... 108
8.18 User-programmable serial interface ... 110
8.19 User-programmable IP interface ... 110
8.20 User-programmable CAN interface ... 110
8.21 CANopen STX API ... 111
8.22 Motion control ... 114
8.23 FTP server ... 116
8.24 HTTP server ... 117
8.25 FTP client ... 118
8.26 Automatic copying of controller data (AutoCopy) ... 119
9 Registers - Overview ... 120
10 Maintenance ... 148
10.1 Repairs ... 148
10.2 Storage and Shipment ... 148
10.3 Return and Disposal ... 149
11 Service ... 150
11.1 Technical support ... 150

12 Spare parts and accessories 151
12.1 Spare Parts 151
12.2 Accessories 151
12.2.1 CAN bus cable 152
12.2.2 Ethernet Cable 152
12.2.3 Cable for the serial interface 152

1 Introduction

1.1 Information on this document

This document forms an integral part of the product and must be read and understood prior to using it. It contains important and safety-related information for the proper use of the product as intended.

Target Groups

This document is intended for specialists with appropriate qualifications.
Only competent and trained personnel are allowed to put this device into operation.
During the whole product life cycle, safe handling and operation of the device must be ensured. In the case of missing or inadequate technical knowledge or knowledge of this document any liability is excluded.

Availability of Information

Make sure this document is kept at the ready in the vicinity of the product throughout its service life.
For information on new revisions of this document, visit the download area on our website. This document is not subject to any updating service.

Start | Bucher Automation - We automate your success.

For further information refer to the following information products:

– Online help for the JetSym software
  Functions of software products with application examples
– Application-oriented manuals
  Product-independent documentation
– Version updates
  Information about changes to the software products and operating system of your device

1.2 Typographical conventions

This manual uses different typographical effects to support you in finding and classifying information. Below, there is an example of a step-by-step instruction:

✓ This symbol indicates requirements which have to be met before executing the following action.

► This sign or a numbering at the beginning of a paragraph marks an action instruction that must be executed by the user. Execute the instructions one after the other.

⇨ The target after a list of instructions indicates reactions to, or results of these actions.

ⓘ Info

Further information and practical tips

In the info box you will find helpful information and practical tips about your product.

2 Safety

2.1 General Information

The product corresponds to the current state of science and technology when placed on the market.
In addition to this user manual, the laws, regulations and directives of the country of operation or the EU apply to the operation of the product. The operator is responsible for compliance with the relevant accident prevention regulations and generally recognized safety rules.

2.2 Purpose

2.2.1 Intended use

This device has been designed to control machinery, such as conveyors, production machines, and handling machines.

Operate the device only in accordance with the intended conditions of use and within the limits set forth in the technical specifications.
Intended use of the product includes its operation in accordance with this manual.

SELV

The operating voltage of this device is classified as Safety Extra Low Voltage and is, therefore, not subject to the EU Low Voltage Directive. The device may only be operated from a SELV source.

2.2.2 Usage other than intended

This device must not be used in technical systems which to a high degree have to be fail-safe.

Machinery Directive

This device is no safety-related part as per Machinery Directive 2006/42/EC, and must, therefore, not be used for safety-relevant applications. This device is NOT intended for the purpose of personal safety, and must, therefore, not be used to protect persons.

2.3 Warnings used in this document

⚠ DANGER



High risk

Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.

⚠ WARNING



Medium risk

Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.

⚠ CAUTION



Low risk

Indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

NOTICE



Material damage

Indicates a situation which, if not avoided, could result in malfunctions or material damage.

3 Product Description

The modular JC-365MC control system is designed for the medium performance range. It covers all areas of industrial automation.

Besides traditional control functions, this controller also offers the possibility to program servo drives as point-to-point axes, axis groups and complex path controls thanks to its motion control functionalities.

3.1 System overview

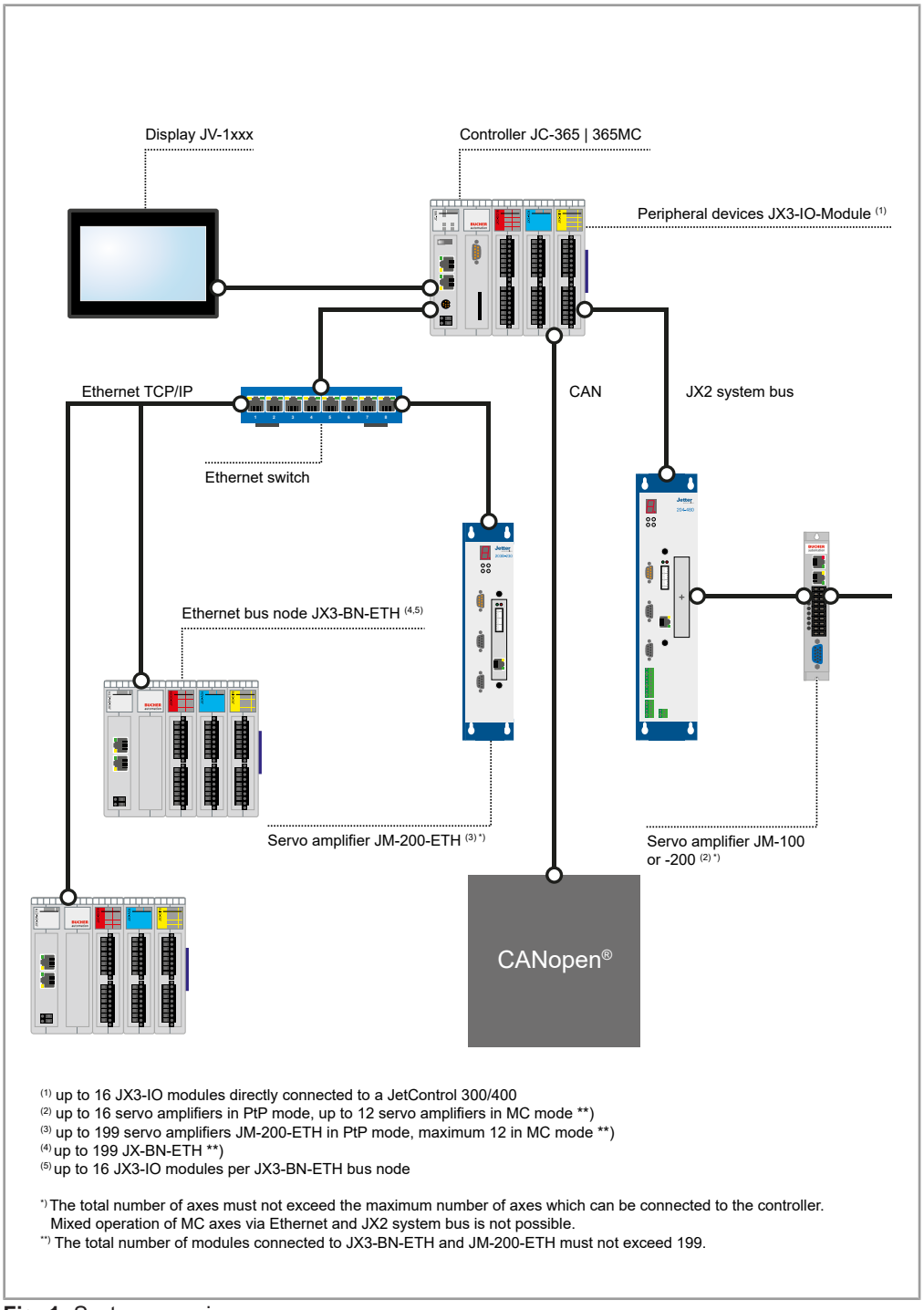


Fig. 1: System overview

3.2 Design

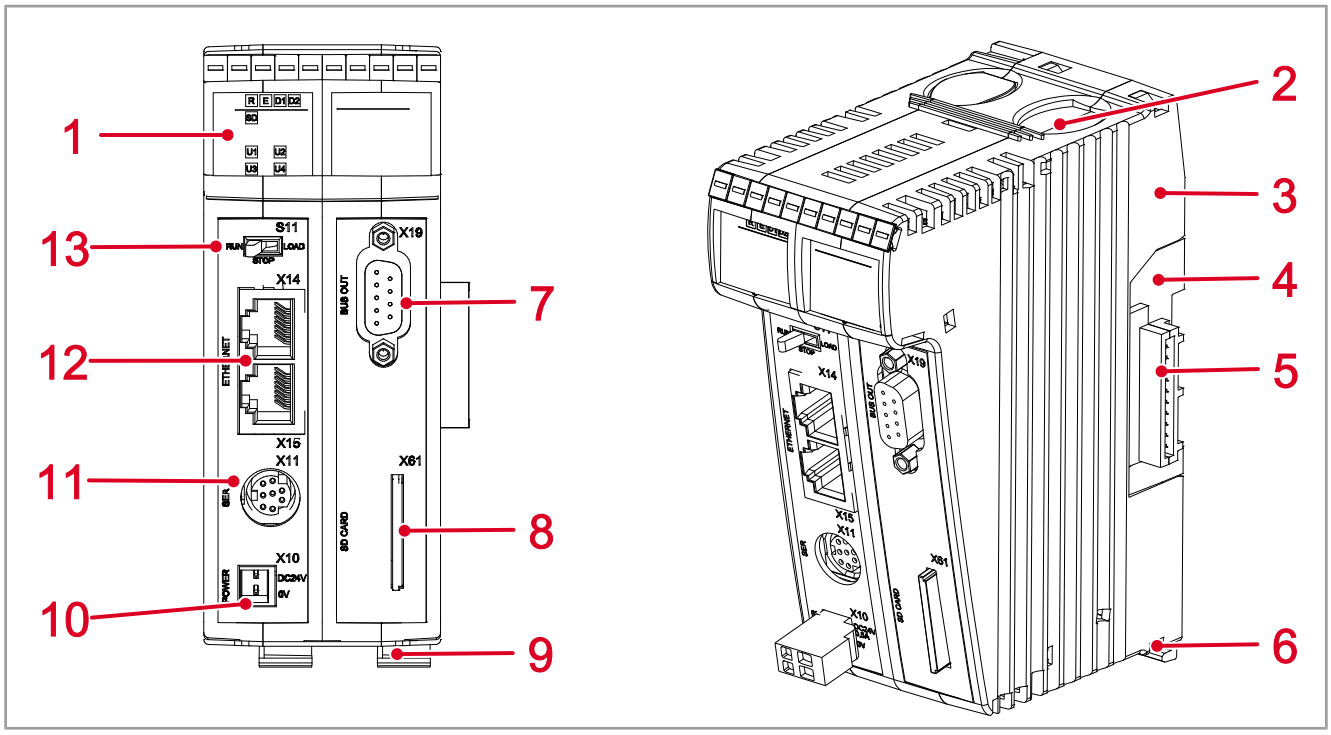


Fig. 2: Design

1	Status indication
2	Upper DIN rail latch
3	Module enclosure
4	Backplane modules with DIP switches
5	X119 plug: Interface to JX3-IO modules
6	DIN rail release latch
7	Socket X19: JX2 system bus and CAN bus interface
8	Socket X61: SD card slot
9	Lower latch
10	Terminal X10: Power supply
11	Socket X11: Serial port
12	Socket X14, X15: Ethernet port
13	Selector S11: Mode selector

3.3 Product features

- Up to a maximum of 16 PtP axes via JX2 system bus
- Up to 64 PtP axes via Ethernet
- Up to 12 axes as path control (a maximum of 6 axes on the JX2 system bus)
- 2 Ethernet ports with integrated switch
- Programming in high-level language STX according to IEC 61131-3
- Non-volatile variable memory:
  240,000 bytes (with option -R: 480,000)
  60,000 registers (with -R option: 120,000)
- STX program/data memory: 24 MB
- 1 serial port (RS-232/422/485)
- Either 1 JX2 system bus port or 1 CAN port (CANopen)
- 1 additional CAN port (CANopen)
- A maximum of 16 JX3 modules can be directly added to the unit
- Real-time clock
- Modbus/TCP client and server
- SD memory card

Additional options

Your JC-365MC may be equipped with additional options (see Order reference/options [▶ 16]). You must already specify the additional options for your controller when placing the order. The controller cannot be equipped with additional features afterwards.

3.4 Status indication

Status indication LEDs indicate the communication status of the device as well as the status of the power supply.

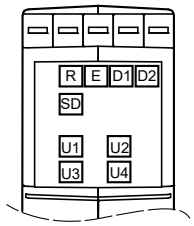


Fig. 3: Status indication

LED	Description
R	State of operating system
E	General error
D1	Special states
D2	State of boot loader
SD	Access to SD card

LED	Description
U1 … U4	Application-specific programming is possible

3.4.1 Diagnostics capabilities by means of status indication

Color and status of the LEDs are an excellent source of information for analyzing various states. In the JetSym programming tool, diagnostics can be performed in the Hardware Manager or via the Setup pane by entering the corresponding register number.

LED	Status	Color	Description
R	OFF	---	No power supply or defective.
	Blinking at 1 Hz	Green	Either the controller is still booting up, or it does not execute the application program.
	Blinking at 4 Hz	Green	Reset or fatal error.
	ON	Green	Application program is being executed.
E	OFF	---	No error occurred.
	Blinking at 1 Hz	Red	There is no valid OS.
	Blinking at 4 Hz	Red	Reset, fatal error, or checking the network consistency.
	ON	Red	Error; refer to error register.
D1	OFF	---	Normal operating condition.
	Blinking at 1 Hz	Red	Automatic IP configuration; AutoCopy function is completed, or first half of start delay is being executed.
	Blinking at 4 Hz	Red	Reset; fatal error, or second half of start delay is being executed.
	ON	Red	**autocopy.ini** file is being executed.
D2	OFF	---	Boot loader is not running.
	Blinking at 1 Hz	Red	Boot loader: Automatic IP configuration.
	Blinking at 4 Hz	Red	Reset or fatal error.
	ON	Red	Boot loader is being executed.

3.4.2 LED states during the boot process

If the following requirements are met, the bus nodes goes through the normal boot process without errors:

– Mode selector S11 is in *RUN* position.
– There is a valid OS.
– There must be a valid application program.

The LED flashing patterns indicate the different stages of the boot process.

LEDs	R	E	D1	D2	State
Phase 1					**Reset**
Color	Green	Red	Red	Red	Reset
Blinking pattern	4 Hz	4 Hz	4 Hz	4 Hz	
Phase 2					**Operating system**
Color	Green	-	-	Red	The device is initializing the operating system.
Blinking pattern	1 Hz	OFF	OFF	ON	
Phase 3					**Parser**
Color	Green	-	-	-	The OS reads the settings of the DIP switch on the backplane module and checks whether an Ethernet switch exists.
Blinking pattern	1 Hz	OFF	OFF	OFF	
Phase 4					**Interfaces and programs**
Color	Green	Red	-	-	The OS initializes realtime clock, Ethernet interface and file system.
Blinking pattern	1 Hz	ON	OFF	OFF	
Phase 5a					**Start delay**
The controller only executes phases **5a** and **5b** only if a time value is entered in R 202971 *Start delay*.					
Color	Green	Red	Red	-	The first half of the start delay is in progress.
Blinking pattern	1 Hz	ON	1 Hz	OFF	
Phase 5b					**Start delay**
Color	Green	Red	Red	-	The second half of the start delay is in progress.
Blinking pattern	1 Hz	ON	4 Hz	OFF	
Phase 6					**Application program**
Color	Green	Red	Red	-	The OS initializes the modules on the system bus, as well as software features (Web, Modbus/TCP, etc.). Then it loads the application program.
Blinking pattern	1 Hz	ON	ON	OFF	

LEDs	R	E	D1	D2	State
Phase 7					Operating state
Color	Green	-	-	-	The operating state is displayed
Blinking pattern	1 Hz	OFF	OFF	OFF	Logic voltage supply is OK; application program has been stopped
	ON	OFF	OFF	OFF	Logic voltage supply is OK; application program is running

Tab. 1: LED states during the boot process

3.4.3 LED states after detection of an IP address conflict

During boot-up or when the IP address was changed at runtime, the controller checks its own set IP address. If the controller detects that its IP address is already used on the network, it takes the following actions:

– The controller sets bit 24 in R 200009.

– The controller sets bit 7 in R 200008 (collective error bit of R 200009).

– Then, the LEDs on the controller indicate the following:

LEDs	E	D1	D2	State
Color	Red	Red	Red	The controller has detected an IP address conflict.
Blinking pattern	ON	1 Hz	1 Hz	

To exit this state, the following two steps must be taken:

1. Resolve the IP address conflict.
2. Reboot the controller.

3.5 Nameplate

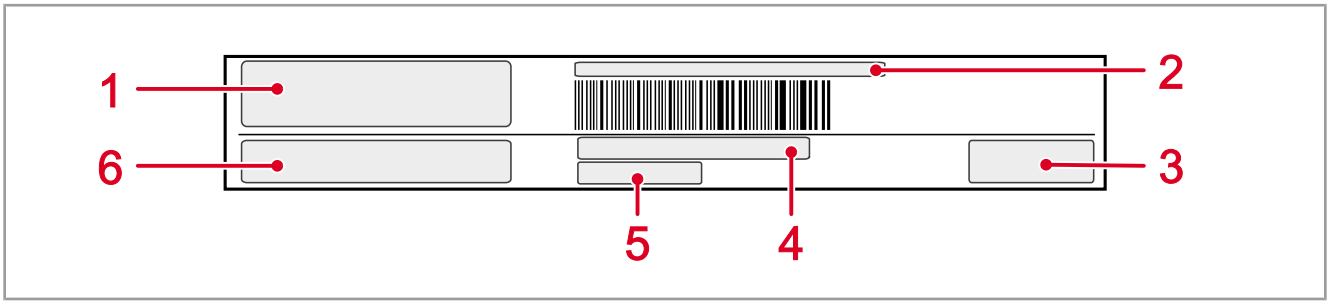


Fig. 4: Sample nameplate

1	Company logo
2	Serial number
3	Certification mark
4	Item number
5	Hardware revision
6	Item name

3.6 Scope of Delivery

Scope of delivery	Item number	Quantity
JC-365MC	Depending on options	1
Male connector in spring-cage technology, 2-pin	60870409	1
Terminal labels	60870411	10
Keying pins	60870410	1
Installation manual	60873051	1

3.7 Order reference/options

The order reference consists of the name of the controller and the desired options. Each of the additional options listed below supplements the controller. The order reference only reflects existing options.

JC-365MC - **A** - **R**

Element	Description
JC-365MC	Controller with Motion Control feature
A	Number of axes: 4, 8, or unlimited; if unlimited, A is not applicable, e.g. JC-365MC-R Number of axes: 16 max. on the JX2 system bus Path control: 4, 8 or 12 servo axes
R	Additional option: Memory expansion to 120,000 non-volatile registers

ⓘ Info

Ordering additional options

Specify your desired options in the order. The controller cannot be equipped with additional features afterwards.

Number of axes

4, 8 or an unlimited number of axes can be connected to the JC-365MC controller.

Item no.	Order reference
10001337	JC-365MC-4
10001338	JC-365MC-8
10001336	JC-365MC

Integrated Web server and e-mail feature

With its integrated web server and e-mail feature the controller supports the following functions:

– **FTP server/FTP client:** The FTP server lets you handle directories and files using an FTP client.
– **HTTP server:** The user can store HTML pages on the controller and display register contents at run time using SSI (Server side includes).
– **SMTP client:** The SMTP client lets you send e-mails.

Modbus/TCP

The controller supports the Modbus/TCP protocol. The controller can act as both server and client.

4 Technical Specifications

This chapter contains information on both electrical and mechanical data as well as on operating data of the JC-365MC device.

4.1 Dimensions

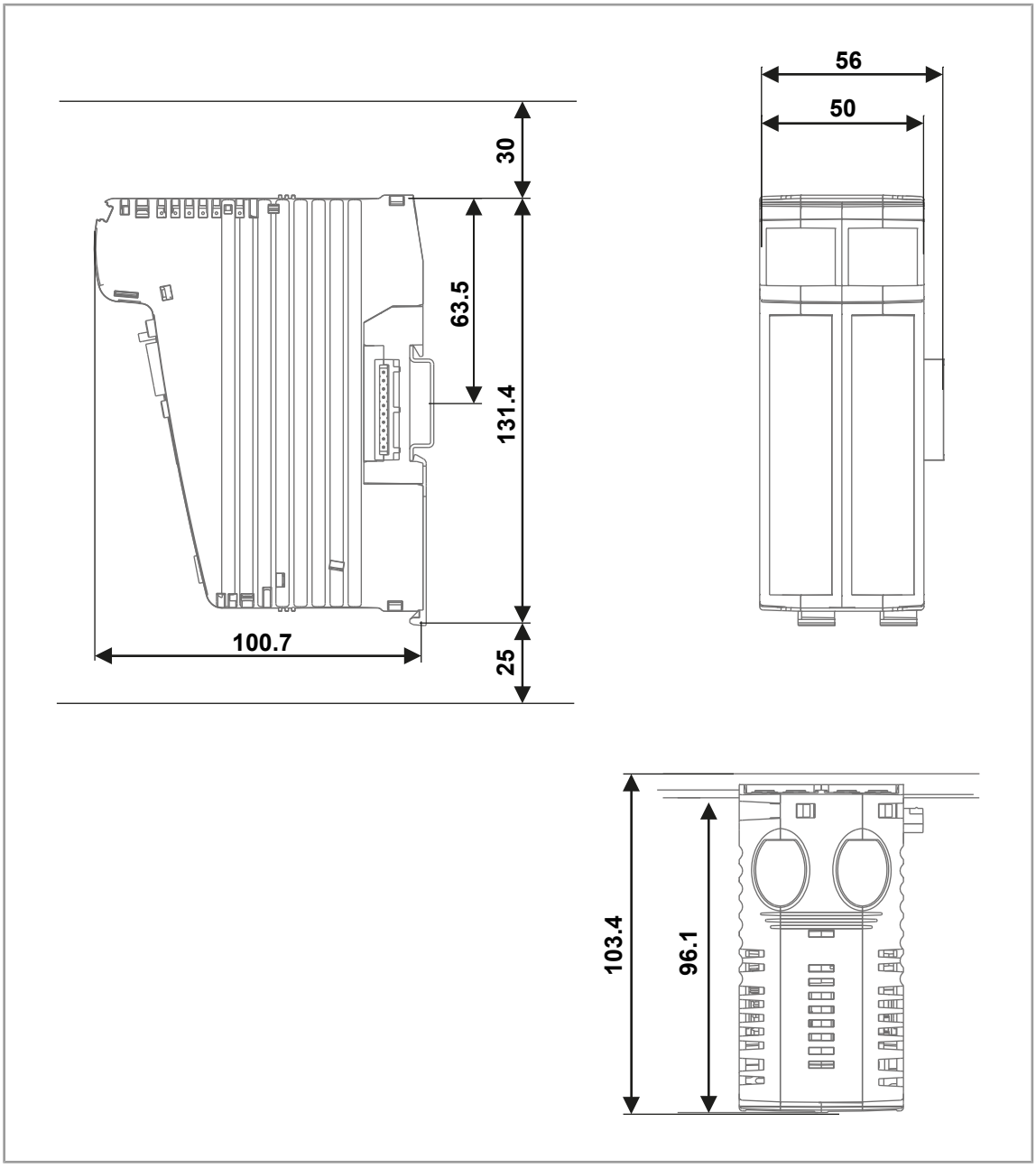


Fig. 5: Dimension in mm

Info **CAD data**

CAD data of the device can be found in the download area of our homepage.

4.2 Mechanical Specifications

Parameter	Description	Standards
Mounting orientation	Vertically mounted on DIN rail	
Weight	275 g	
Enclosure specifications		
Material	Plastic	
Maximum height of fall		
with shipping packaging	1 m	DIN EN 61131-2 DIN EN 60068-2-31
with product packaging	0.3 m	
Vibration resistance		
Frequency sweeps	1 octave/minute, sinusoidal	DIN EN 61131-2 DIN EN 60068-2-6
Constant amplitude	3.5 mm	5 Hz ≤ f ≤ 9 Hz
Constant acceleration	1 g	9 Hz ≤ f ≤ 150 Hz
Number and direction	10 sweeps for all 3 spatial axes	
Shock resistance		
Type of shock	Half-sine wave	DIN EN 61131-2 DIN EN 60068-2-27
Intensity and duration	15 g for 11 ms	
Number and direction	3 shocks in the directions of all 3 spatial axes	
Degree of protection		
Degree of protection	IP20	DIN EN 60529

Tab. 2: Mechanical specifications

4.3 Electrical properties

Power supply (terminal X10)

Parameter	Description
Supply voltage	DC 24 V
Permissible voltage range	-15 % … +20 %
Input current	
Without HMI	1.0 A max
With HMI	1.5 A max
Power consumption	
Without HMI	24 W max
With HMI	36 W max

Tab. 3: Power supply (terminal X10)

Serial interface (socket X11)

Parameter	Description
Terminal type	MiniDIN, shielded
Total of pins	8
Electrical isolation	None
Number of interfaces	1
Interface standards	RS-232/RS-422/RS-485-2
Baud rates	1,200 ... 115,200 baud
Bits per character	5, 6, 7, 8
Number of stop bits	1, 2
Parity	Even, odd, none, 1, 0
Max. line length	< 30 m

Tab. 4: Serial interface (socket X11)

Ethernet ports (sockets X14, X15)

Parameter	Description
Terminal type	RJ45 socket
Port count	2
	Interconnected via internal switch
Bit rate	10 Mbit/s, 100 Mbit/s (Cat 5e)
Auto-crossover	Yes

Tab. 5: Ethernet ports (sockets X14, X15)

CAN bus interface (socket X19)

Parameter	Description
Terminal type	Sub-D socket
Total of pins	9
Electrical isolation	None

Tab. 6: CAN bus interface (socket X19)

SD card (slot X61)

Parameter	Description
Plug-in card type	Standard SD card
Mechanical operation	push/push
Maximum memory size	32 MB ... 32 GB
Drive format	FAT
Protection against inserting the SD card in the wrong direction	Yes

Tab. 7: SD card (slot X61)

Memory configurations

Parameter	Description
Non-volatile memory	240,000 bytes with –R option: 480000
	60,000 application registers (32 bit) with –R option: 120000
STX program memory/data memory	24 MB
Flash drive	24 MB

Tab. 8: Memory configurations

Electrical safety

Parameter	Description	Standards
Class of protection	III	DIN EN 61131-2
Dielectric test voltage	Functional ground is internally connected to the device ground	
Protective connection	0	
Overvoltage category	II	

Tab. 9: Electrical safety

4.3.1 System power supply

The controller supplies the local system bus with logic and power supply voltage.

These two types of voltage are for supplying the connected expansion modules.

System bus

Parameter	Description
Bus type	JX3 system bus
Logic voltage	
Supply voltage	DC +5 V
Permissible voltage range	-15 % ... +10 %
Additional voltage	
Supply voltage	DC +24 V
Permissible voltage range	-15 % ... +20 %

Tab. 10: System bus

Connected JX3 IO modules

Parameter	Description
Logic voltage	
Current consumption	I_{5V} = max. 1,200 mA
Power consumption	6 W max.
Additional voltage	
Current consumption	$I_{24\ V}$ = max. 750 mA
Power consumption	18 W max.

Tab. 11: JX3 IO modules on the system bus

4.4 Real-time clock

Parameter	Description	
Power reserve (if the controller was previously switched on for at least 1 hour)	Minimum	1 week
	Typically	2 weeks
Deviation	maximum	1 minute per month

Tab. 12: Real-time clock

4.5 Environmental Conditions

Parameter	Description	Standards
Operating temperature	0 °C … +50 °C	DIN EN 61131-2
Storage temperature	-40 °C … +70 °C	DIN EN 60068-2-1
Humidity	10 % … 95 %, non-condensing	DIN EN 60068-2-2
Max. operating altitude	2,000 m above sea level	DIN EN 61131-2
Corrosion immunity and chemical resistance	No special protection against corrosion. Ambient air must be free from higher concentrations of acids, alkaline solutions, corrosive agents, salts, metal vapors and other corrosive or electroconductive contaminants.	
Degree of pollution – Electronics	Level 2	DIN EN 61131-2
	Usually, the pollution is non-conductive. However, temporary conductivity due to condensation may occur.	

Tab. 13: Environmental conditions

4.6 EMC Values

4.6.1 Enclosure

Emitted Interference

Parameter	Values	Standards
Frequency band	**30 MHz … 230 MHz**	DIN EN 61000-6-3 DIN EN 61131-2 DIN EN 55011
Limit value	30 dB (µV/m) at 10 m distance	
Frequency band	**230 MHz … 1,000 MHz**	
Limit value	37 dB (µV/m) at 10 m distance	
	Class B	

Tab. 14: Emitted interference

Immunity to Interference

Parameter	Values	Standards
Magnetic field with mains frequency		
Frequency	50 Hz	DIN EN 61131-2 DIN EN 61000-6-2 DIN EN 61000-4-8
Magnetic field	30 A/m	
RF field, amplitude-modulated		
Frequency band	80 MHz … 2 GHz	DIN EN 61131-2 DIN EN 61000-6-2 DIN EN 61000-4-3
Test field strength	10 V/m	
	AM 80 % with 1 kHz	
	Criterion A	
ESD		
Discharge through air Test peak voltage	8 kV	DIN EN 61131-2 DIN EN 61000-6-2 DIN EN 61000-4-2
Discharge through air Test peak voltage	4 kV	
	Criterion A	

Tab. 15: Immunity to interference

4.6.2 Shielded data and I/O lines

Immunity to interference

Parameter	Values	Standards
RF field, asymmetric, amplitude modulated		
Frequency band	0.15 MHz … 80 MHz	DIN EN 61131-2 DIN EN 61000-6-2 DIN EN 61000-4-6
Test voltage	10 V	
	AM 80 % at 1 kHz	
Source impedance	150 Ω	
	Criterion A	
Bursts		

Parameter	Values	Standards
Test voltage	1 kV	DIN EN 61000-6-2 DIN EN 61000-6-2 DIN EN 61000-4-4
	tr/tn 5/50 ns	
Repetition frequency	5 kHz	
	Criterion A	
Surge voltages, asymmetric, line to earth		
Common-mode interference	tr/th 1.2/50 µs	DIN EN 61131-2 DIN EN 61000-6-2 DIN EN 61000-4-5
	1 kV	

Tab. 16: Immunity of shielded data and I/O lines

4.6.3 DC power supply inputs and outputs

Immunity to interference

Parameter	Values	Standards
Asymmetric RF, amplitude-modulated		
Frequency band	0.15 MHz … 80 MHz	DIN EN 61131-2 DIN EN 61000-6-2 DIN EN 61000-4-6
Test voltage	10 V	
	AM 80% at 1 kHz	
Source impedance	150 Ω	
	Criterion A	
Bursts		
Test voltage	2 kV	DIN EN 61131-2 DIN EN 61000-6-2 DIN EN 61000-4-4
	tr/tn 5/50 ns	
Repetition frequency	5 kHz	
	Criterion A	
Surge voltages, symmetric, line to cable		
Series-mode interference	tr/th 1.2/50 µs	DIN EN 61131-2 DIN EN 61000-6-2 DIN EN 61000-4-5
	0.5 kV	
Surge voltages, asymmetric, line to earth		
Common-mode interference	tr/th 1.2/50 µs	DIN EN 61131-2 DIN EN 61000-6-2 DIN EN 61000-4-5
	1 kV	

Tab. 17: DC power supply inputs and outputs

5 Mechanical Installation

This chapter describes how to install and remove the JC-365MC.

NOTICE



Damaged devices

Damaged devices may cause considerable physical damage.

- ► Check the device for external damage and faulty connections.
- ► Ensure to install only fully functional devices.

5.1 Installing Device on DIN Rail

NOTICE

Functional impairment caused by unfavorable installation

- ► Install the device only in vertical position on the DIN rail (DIN EN 60715).
- ► Keep a minimum distance of 30 mm to surrounding parts.

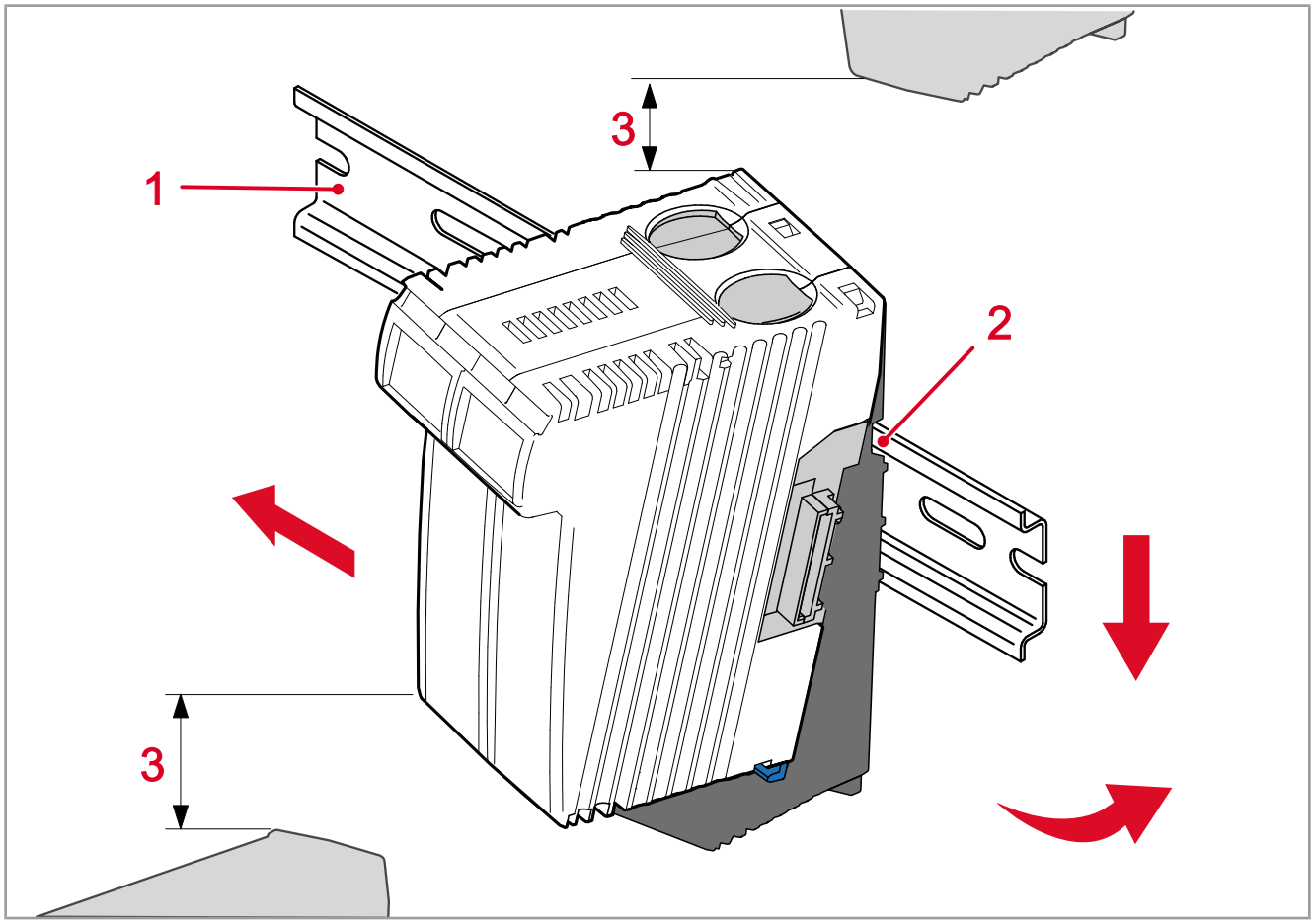


Fig. 6: Installing the device on the DIN rail

1	DIN rail
2	Upper latch
3	Minimum distance to surrounding parts (30 mm min.)

1. Disconnect the system from the power supply.
2. Place the upper latch (2) in an angled position on the DIN rail (1).
3. Snap the lower latch of the device onto the DIN rail.
4. Slide the device on the DIN rail to its intended position.

5.2 Removing Device from DIN Rail

Applying force to the release latch disengages the device from the DIN rail.

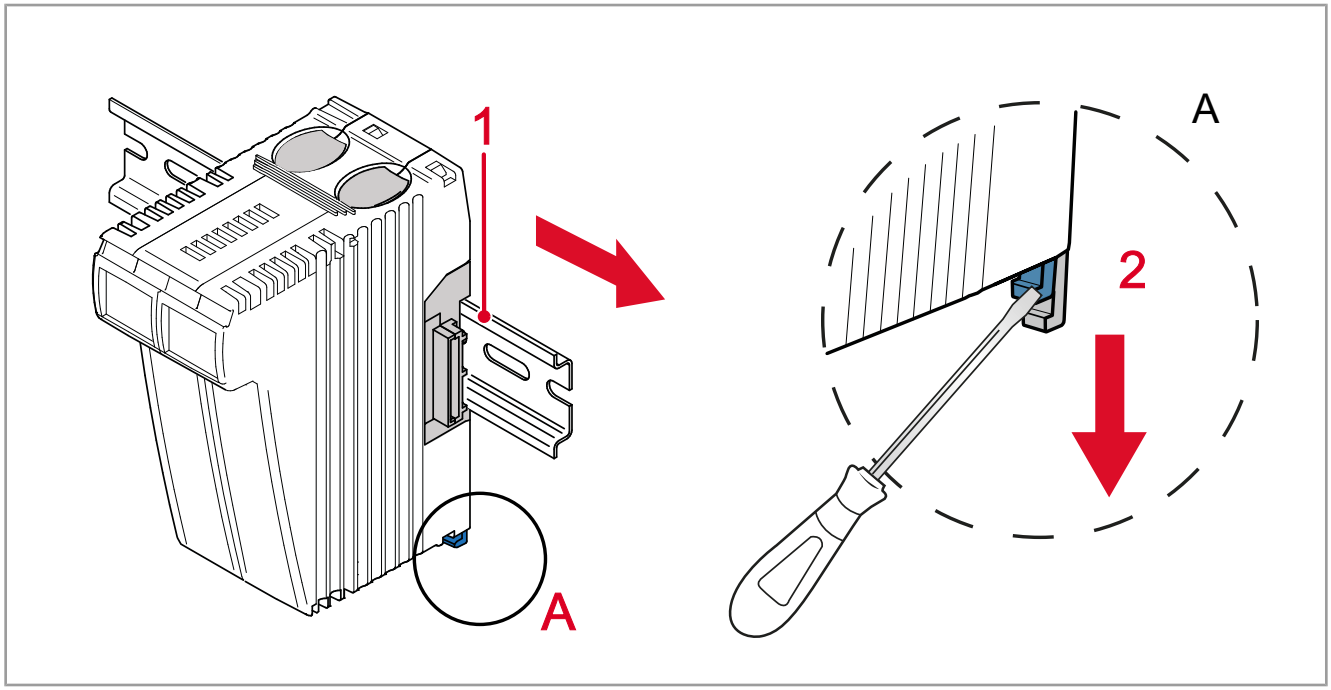


Fig. 7: Removing the device from the DIN rail

1	DIN rail
2	DIN rail release latch
A	Detail view

1. Disconnect the system from the power supply.
2. Pull the device off the overall assembly.
3. Pry the release latch (2) downwards and pull the device off the DIN rail (1).

5.3 Dismounting Enclosure from Backplane Module

Pressing down on the upper and the lower backplane latches of the device allows you to remove the enclosure from the backplane module.

NOTICE

Mechanical damage and reduced EMC noise immunity

If devices are replaced, degree of protection IP20 is not ensured. Touching the EMC clip may lead to mechanical damage to the device and reduced EMC noise immunity.

► Do not touch any electronic components once the enclosure has been removed from the backplane module.

The following information is retained on the backplane module:

- IP address
- Subnet mask
- Gateway
- DNS server
- Controller name
- File name for the AutoCopy function
- IP port numbers for the debug server and the IP server

Switching Between Controller Models

Any controller of the JetControl 300 series can be replaced by another controller model of this series. To increase performance, for example, the controller model JC-340-0 may be replaced by controller model JC-350-3.

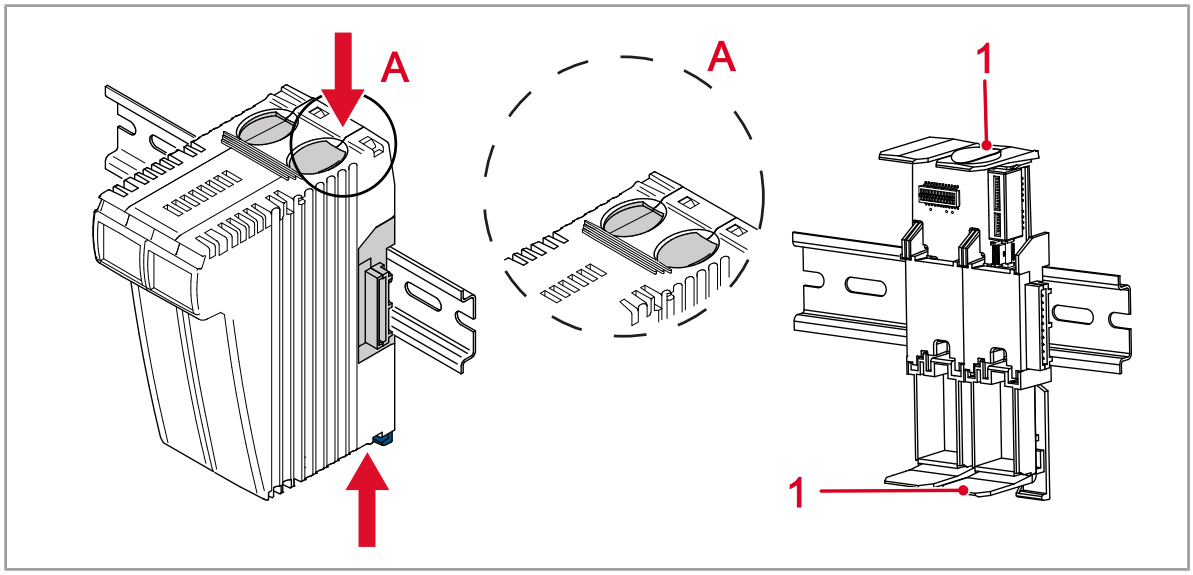


Fig. 8: Removing the enclosure from the backplane module

1	Latch
A	Detail view

1. Disconnect the system from the power supply.
2. Simultaneously, press and hold the upper and lower latches (1) of the device.
3. With the latches pressed down pull the enclosure away from DIN rail.

6 Electrical Connection

NOTICE



Damages to material or functional impairment

Improper implementation of the wiring harness may cause mechanical stress.

- ► Protect the cables from bending, twisting or chafing.
- ► Install strain reliefs for the connecting cables.

6.1 Improving EMC Noise Immunity

A system's immunity to noise is determined by its weakest component. Key factors are correct connections, lines and proper shielding. Take note of the measures described in this section.

ⓘ Info

Further Information

For further information on the noise immunity of a system, please refer to the Application Note 016 *EMC-Compatible Installation of the Electric Cabinet* available for download on our website.

DIN rail

- – Attach the JC-365MC device to a DIN rail to EN 60715 with the dimensions 35 x 7.5 mm.
- – The DIN rail must be electrically conducting and grounded by either of the following two ways:
  - – Directly
  - – Via rear panel of the electric cabinet

Application Note 016

Follow the instructions given in Application Note 016 *EMC-Compatible Installation of the Electric Cabinet*.

The following instructions are excerpts from Application Note 016:

- – Keep signal and power lines **separated from each other** at all times. Bucher Automation AG recommends that there be a minimum distance of 20 cm. Cables and lines should cross each other at an angle of 90°.
- – Shield the following cables and lines:
  - – Analog lines
  - – Data lines
  - – Motor cables coming from inverter drives (servo output stage, frequency converter)
  - – Lines between components and interference suppressor filter, where the suppressor filter is located not immediately adjacent to the component.
- – Place the shield **on both sides**.
- – Keep unshielded wire ends of shielded cables as short as possible.
- – Span the shield **entirely** across the isolation. For **wide-area grounding**, clamp it down tightly with an extensively earthed strain relief.

Use of Connectors

– Clamp the shield down **entirely** using the shielding clamp of the metalized connector enclosure (impedance shielding), or the EMC-compliant gland bushing. For **wide-area** grounding, clamp it down tightly using a strain relief.

– Only use metalized connectors, e.g. Sub-D with metalized enclosure. Make sure that the strain relief is directly connected with the enclosure.

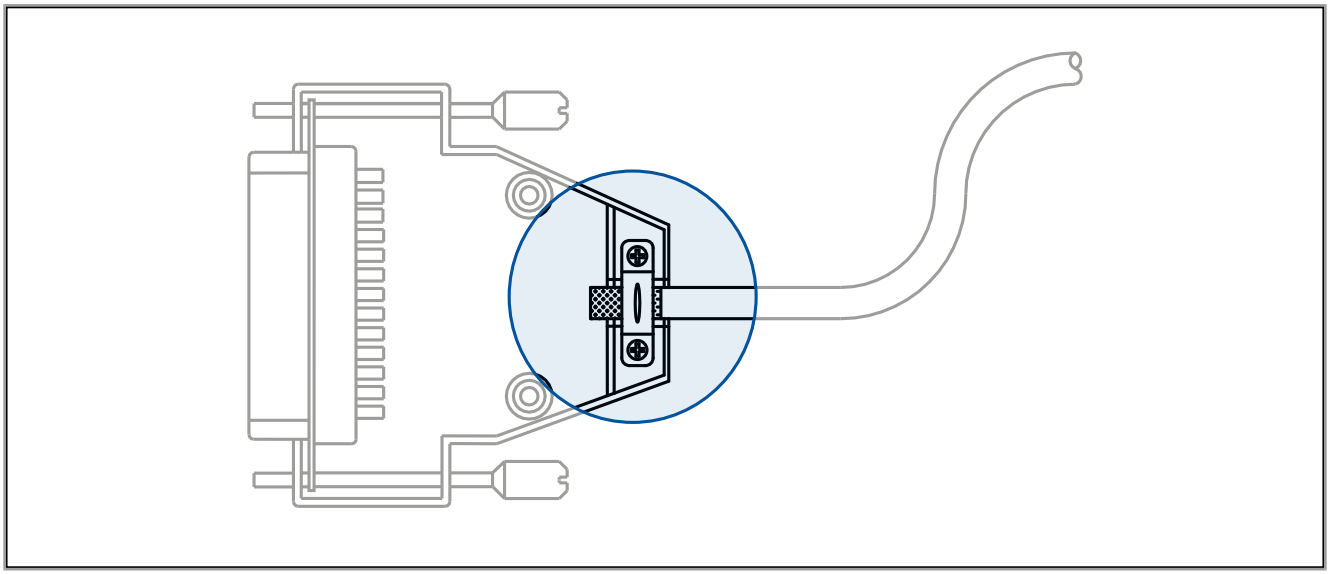


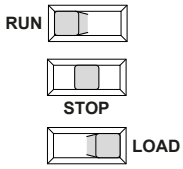
Fig. 9: Strain relief on the Sub-D enclosure

6.2 Interfaces

6.2.1 Selector switch S11 - Mode selector

The position of mode selector S11 is identified while the controller is booting up. The post-boot behavior of the controller depends on the mode selector position. Any changes made to the mode selector while the controller is running will have no effect on the operating mode.

Switch position



Operating mode	Description
RUN	The controller launches the application program.
STOP	The controller does not launch the application program.
LOAD	The controller does not launch the application program. The controller executes the AutoCopy function. Upon completion of the AutoCopy function, the controller requires rebooting.

Tab. 18: S11 mode switch positions

Functions of the mode selector

In checking the position of mode selector S11 the JC-365MC proceeds in the following way:

Step	Description	
1	Power supply of the controller is at terminal X10.	
2	The boot loader of the controller checks the position of selector S11.	
	If ...	**... then ...**
	... mode selector S11 = *RUN* or *STOP*,	... the OS is launched; --> proceed with step 4
	... selector S11 = *LOAD* position, and an SD card has been inserted,	... the OS is launched; --> proceed with step 3
	... selector S11 = *LOAD* position, and an SD card has not been inserted,	... the boot loader waits until an OS update is carried out.
3	The controller loads the **autocopy.ini** file.	
	If ...	**... then ...**
	... the file could be loaded,	... the instructions contained in it are executed.
	... the file could not be loaded,	... the instructions cannot be executed.
4	The controller checks the position of selector S11.	
	If ...	**... then ...**
	... mode selector S11 = *RUN*,	... the application program is launched.
	... mode selector S11 = *STOP*,	... the application program does not start.
5	**If ...**	**... then ...**
	... the position of mode selector S11 is changed once the controller has been turned on,	... this has no effect on the functioning of the controller.

6.2.2 Terminal X10 - Power supply

Ports and interfaces

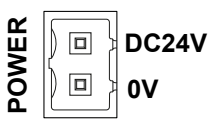
X10 lets you connect the following devices:

– Voltage supply for the JC-365MC controller

– Voltage supply of the connected JX3 IO modules provided they are not supplied by a separate power supply module.

– Voltage supply of an HMI (LCD xxx) of the Bucher Automation AG connected to X11

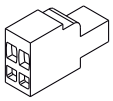
Pinning



Pin	Description
DC24V	Supply voltage
0 V	Reference potential (GND)

2-pin connector, spring-cage technology

The controller comes with a 2-pin spring-clamp connector.



Category	Description	Standards
Connector		
Designation	BU_02_E_BLZF_GE_RM3.5	
Connector technology	Spring-cage connection	
Type	2-pin, 3.5 mm pitch	
Connectible conductors		
Outer diameter of the isolation	2.90 mm max	
AWG	16 … 28	
Clamping range	0.13 … 1.5 mm²	
Stripping length	10 mm	
Specification without wire end ferrules		
Solid conductor	H05(07) V-U	
	0.2 … 1.5 mm²	
Flexible conductor	H05(07) V-K	
	0.2 … 1.5 mm²	
Specification with wire end ferrules		
Wire end ferrule without sleeve	0.2 … 1.5 mm²	DIN 46228/1
Wire end ferrule with sleeve	0.2 … 1.5 mm²	DIN 46228/4
Crimping tool	PZ 4, PZ 6 ROTO, PZ 6/5	DIN 46228

Tab. 19: Connector specification, 2-pin connector, spring-cage technology

6.2.3 Port X11 – Serial interface

Ports and interfaces

Port X11 lets you connect the following devices:

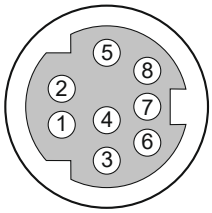
– PC
– Bucher Automation AG HMIs
– Any device with R-232/422/485 interface

ⓘ Info

Limitations

– While the device supports several different hardware drivers, only one hardware interface is available. This means: If communication takes place for example via RS-422, the system cannot communicate simultaneous and independently via RS-232.
– The maximum cable length is < 30 m.

Pinning



Pin	Signal	Description
1	RDA	RS-422; receive data inverted
2	GND	Reference potential
3	RDB	RS-422; receive data not inverted
4	RxD	RS-232; receive data
5	SDB	RS-422; transmit data not inverted RS-485; Transmit/receive data not inverted
6	DC24V	HMI supply voltage
7	SDA	RS-422; transmit data inverted RS-485; Transmit/receive data inverted
8	TxD	RS-232; transmit data

Cable for connector X11

To connect devices to port X11, you can order cables separately as accessories [▶ 152].

Schematic diagram of port X11

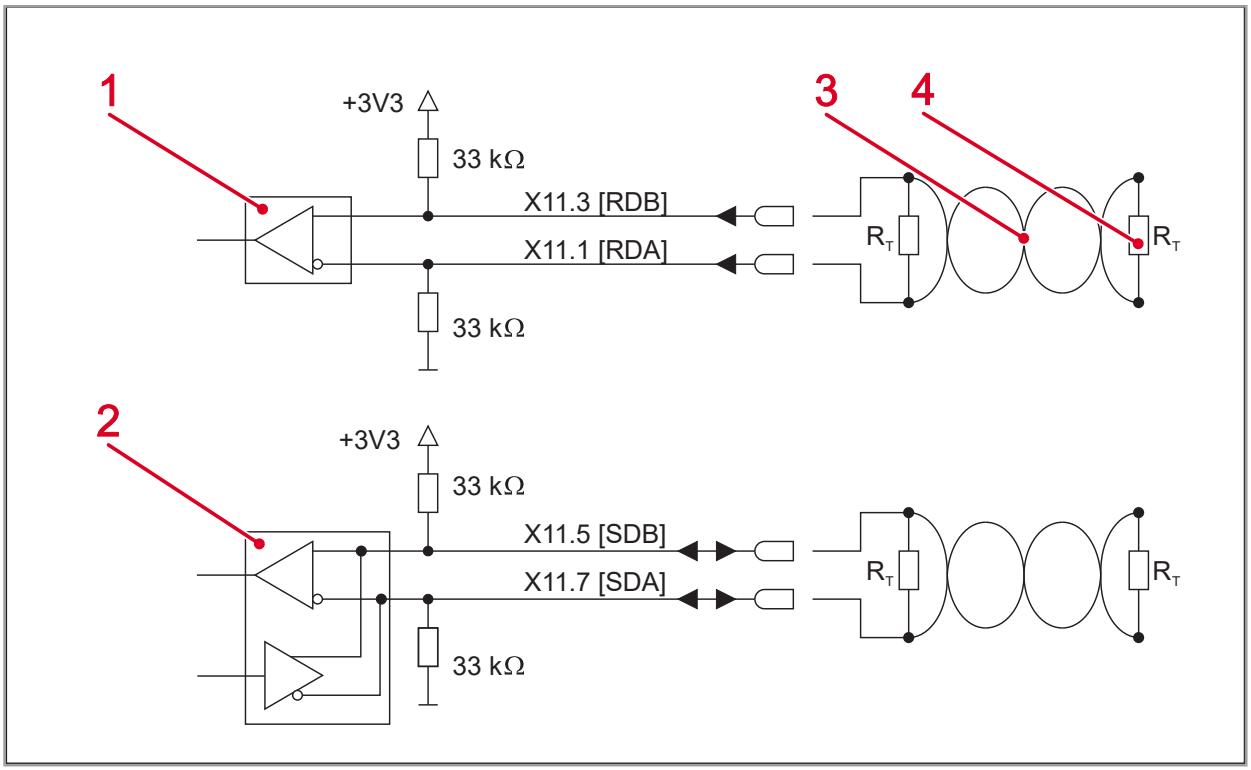


Fig. 10: X11 port block diagram

Position	Part	Function with RS-422	Function with RS-485
1	Receiver	Receives data	Unused
2	Receiver/transmitter	Transmits data	Receives and transmits data
3	Serial line	Twisted line of the serial interface	
4	R_T	Terminating resistor	

Terminating resistor

Connect a terminating resistor to both serial lines in the following cases:

– Long lines
– High baud rates

Select a terminating resistor which corresponds to the impedance of the line used.

Connecting alphanumeric displays and HMIs

The serial interface (port X11) of the JC-365MC lets you connect HMIs by Bucher Automation.

Info

Further information

For more information on this topic refer to the application-oriented manual *Controlling alphanumeric HMI devices (LCD) and printers* available in the download area of our homepage.

6.2.4 Ports X14, X15 - Ethernet

Ports and interfaces

Ports X14 and X15 let you connect the following devices:

– PC
– HMI by Bucher Automation AG, e. g. JV-10xx
– Any device with Ethernet TCP/IP interface (10/100 Mbits)

Pinning

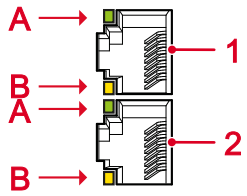


Fig. 11: Ports X14, X15

Position	Socket	LED	Color	Description
1	X14	A	Green	LINK: There is a connection to the network
		B	Yellow	ACT: Data transmission
2	X15	A	Green	LINK: There is a connection to the network
		B	Yellow	ACT: Data transmission

Cables for ports X14, X15

To connect devices to ports X14 and X15, you can order cables separately as accessories.

6.2.5 Port X19 - CAN Bus

Ports and interfaces

Port X19 lets you connect the following devices:

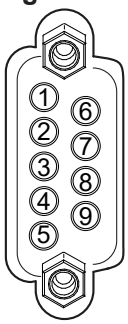
- CAN bus 1 or 2 (CANopen STX API):
  - CANopen modules
- CAN bus 1 (JX2 system bus):
  - JX2 I/O module
  - JX2 slave modules
  - Drives JetMove 1xx, JetMove 2xx, and JetMove 6xx
  - Third-party CANopen modules, e.g. valve terminals

Info

Further information

For more information on this subject refer to the application-oriented manual *CANopen STX API* available for download from our homepage.

Pin assignment



Pin	Signal	Description
1	CMODE0	Commissioning
2	CAN-L	Data signal for CAN bus 1
3	GND	Reference potential
4	CMODE1	Commissioning
5	Unused	
6	CAN-L_2	Data signal for CAN bus 2
7	CAN-H	Data signal for CAN bus 1
8	CAN-H_2	Data signal for CAN bus 2
9	Unused	

CAN bus cable

To connect devices to the CAN bus, you can order cables separately as accessories [▶ 152].

Y-cable (splitter cable)

Info

Y-cable (splitter cable)

To use both CAN buses, connect a Y-cable (splitter cable) to port X19.

One end of the Y-cable (CAN 1) lets you use either the JX2 system bus protocol, or the CANopen protocol (configuration via R 200002077). The other end of the Y-cable (CAN 2) lets you use the CANopen protocol. The functions of the programming interface CANopen STX API implement the CANopen protocol. For more information, see the application-oriented manual CANopen STX API.

You can order the Y-cable separately as an accessory [▶ 152] .

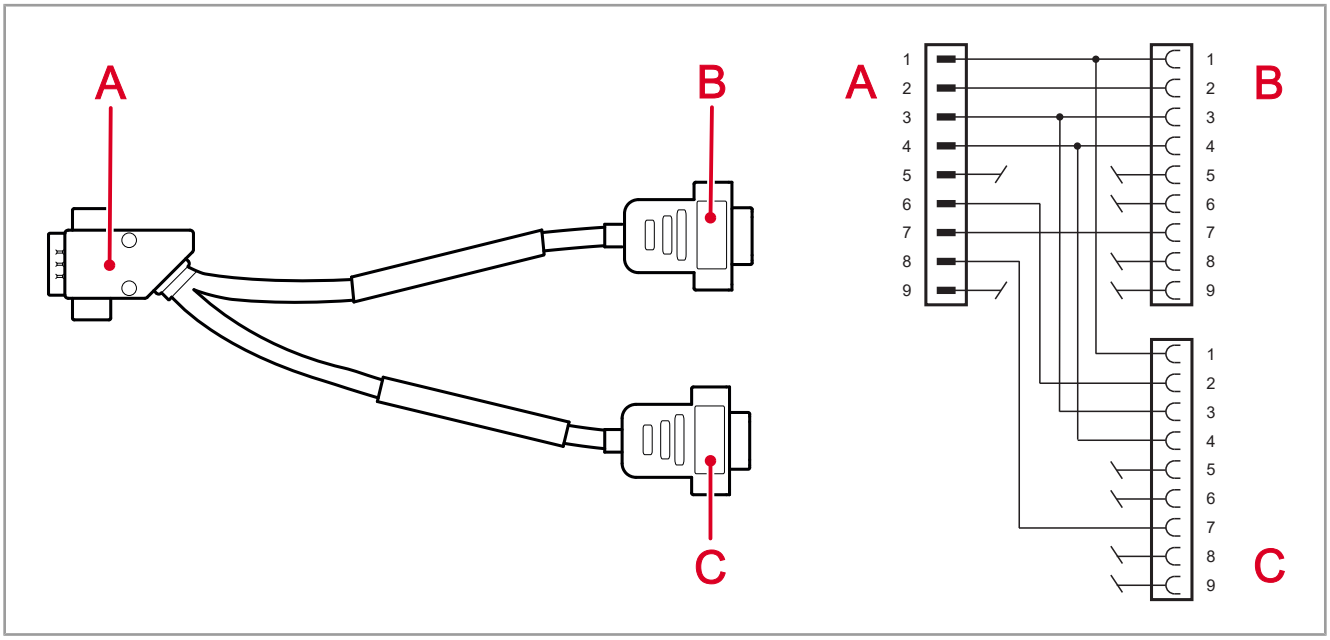


Fig. 12: Y-cable (splitter cable) for the CAN bus interface

Position	Part	Description
A	Sub-D plug, 9-pin	For connection to BUS OUT
B	Sub-D socket, 9-pin	For connection to BUS IN (CAN 1)
C	Sub-D socket, 9-pin	For connection to BUS IN (CAN 2)

Sub-D plug (A)

Pin assignment of the 9-pin sub-D plug on the Y-cable:

Pin	Signal	Description
1	CMODE0	Commissioning
2	CAN-L	Data signal CAN 1
3	GND	Reference potential
4	CMODE1	Commissioning
6	CAN-L_2	Data signal CAN 2
7	CAN-H	Data signal CAN 1
8	CAN-H_2	Data signal CAN 2

Sub-D socket (B)

Pin assignment of the 9-pin Sub-D socket (CAN 1) on the Y-cable:

Pin	Signal	Description
1	CMODE0	Commissioning
2	CAN-L	Data signal CAN 1
3	GND	Reference potential
4	CMODE1	Commissioning
7	CAN-H	Data signal CAN 1

Sub-D socket (C)

Pin assignment of the 9-pin Sub-D socket (CAN 2) on the Y-cable:

Pin	Signal	Description
1	CMODE0	Commissioning
2	CAN-L_2	Data signal CAN 2
3	GND	Reference potential
4	CMODE1	Commissioning
7	CAN-H_2	Data signal CAN 2

JX2 System Bus Cable

JX2 system bus cable specifications

Parameter	Description
Wire cross-section	1,000 kBd: 0.25 mm2 ... 0.34 mm^2 500 kBd: 0.34 mm2 ... 0.50 mm^2 250 kBd: 0.34 mm2 ... 0.60 mm^2 125 kBd: 0.50 mm2 ... 0.60 mm^2
Cable capacitance	60 pF/m max
Resistivity	1,000 kBd: 70 Ω/km max 500 kBd: 60 Ω/km max 250 kBd: 60 Ω/km max 125 kBd: 60 Ω/km max
Number of cores	5
Shield	Complete, not paired
Twisting	CAN_L and CAN_H cables are twisted pairwise

Tab. 20: JX2 system bus cable specifications

Wiring diagram

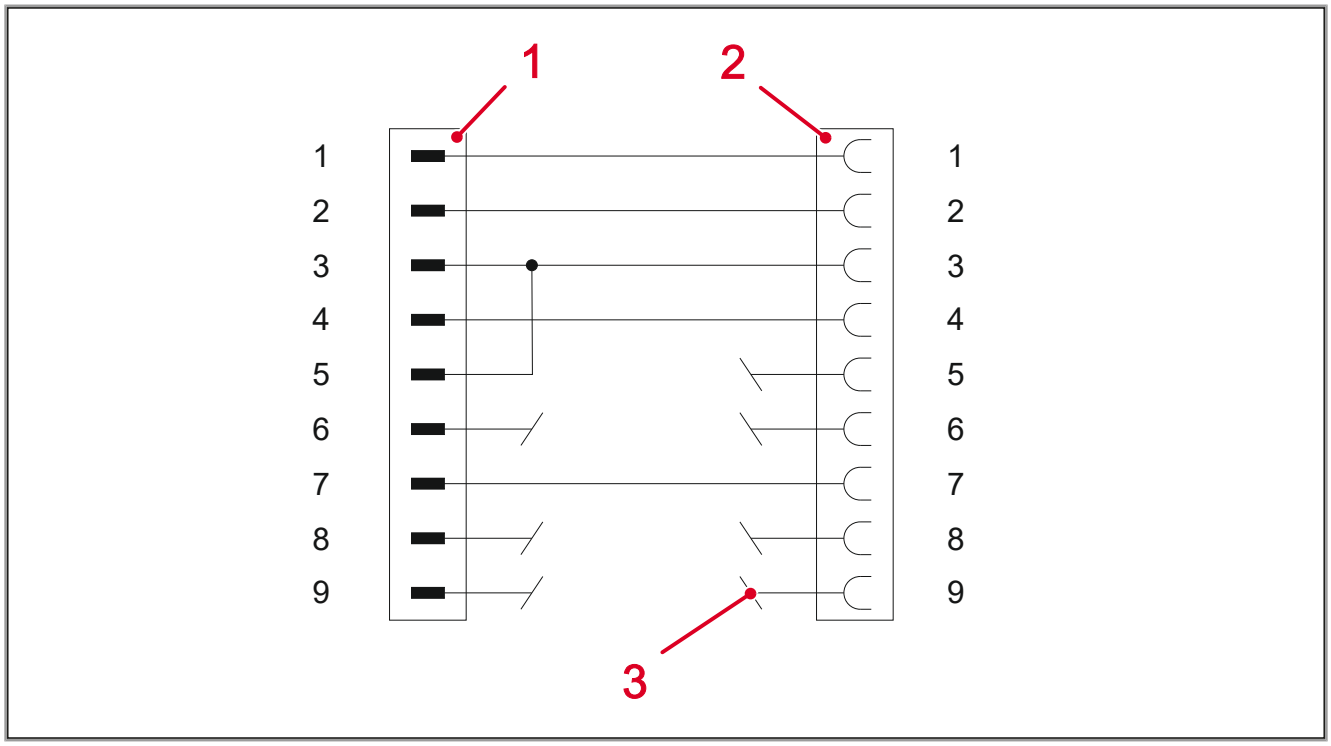


Fig. 13: Wiring diagram of the system bus cable

Number	Part	Description
1	Sub-D plug, 9-pin	For connection to BUS OUT
2	Sub-D socket, 9-pin	For connection to BUS IN
3	Not connected	Do not connect these pins

Sub-D plug

Pinout of the 9-pin Sub-D plug on the JX2 system bus cable:

Pin	Signal name	Description
1	CMODE0	Commissioning
2	CAN-L	Data signal
3	GND	Reference potential
4	CMODE1	Commissioning
5	TERM	Short-circuited with pin 3
7	CAN-H	Data signal

Sub-D socket

Pinout of the 9-pin Sub-D socket on the JX2 system bus cable:

Pin	Signal name	Description
1	CMODE0	Commissioning
2	CAN-L	Data signal
3	GND	Reference potential
4	CMODE1	Commissioning
7	CAN-H	Data signal

JX2 system bus - Line lengths and baud rates

Line lengths

The maximum cable length depends on the baud rate used and the number of expansion modules connected to the bus.

Baud rate	Cable length	Stub length	Total cable length
1,000 kBaud	25 m max.	0.3 m max.	3 m
500 kBaud	100 m max.	1.0 m max.	39 m
250 kBaud	200 m max.	3.0 m max.	78 m
125 kBaud	200 m max.	-	-

Calculation

When engineering the line length, follow the rules listed below:

– Each connected device reduces the maximum line length by 1.0 m.

Baud rates

The baud rate setting depends on the number of modules connected to the JX2 system bus:

JX2-I/O modules JX2 slave modules JetMove JX3-BN-CAN	JX-SIO CANopen modules	1000 kBaud	500 kBaud	250 kBaud	125 kBaud
x		x	x	x	x
	x	x	x	x	x
x	x	x			x

6.2.6 Slot X61 - SD Memory Card

The SD card slot is for accommodating standard SD memory cards. The controller accesses data stored on the SD card which is used as file system extension.

Inserting the SD card

► Insert the SD card into the SD slot as illustrated below.

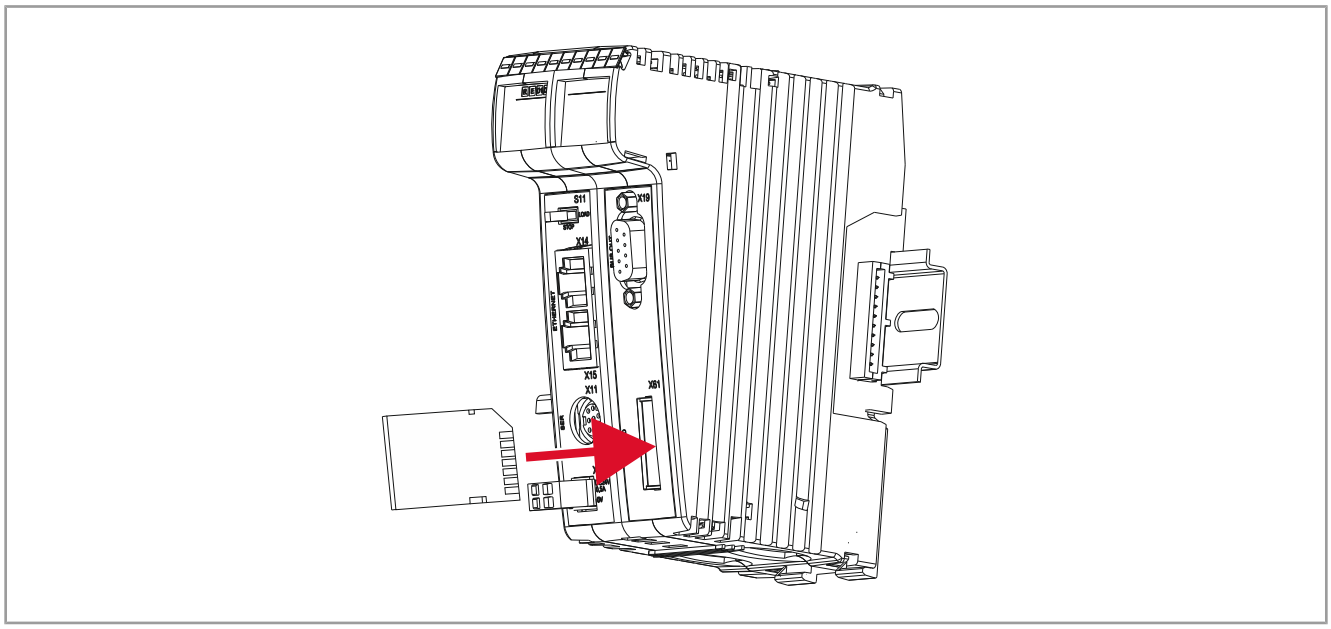


Fig. 14: Inserting the SD card

⇨ If the SD card has been inserted correctly, the **SD** status LED lights up for 300 ms.

Removing the SD card

NOTICE

Data loss

► Make sure the SD card is not accessed, while you are removing it.

► First, close all files which are stored to the SD card.

1. Push the SD card further into the slot and let go of it.

   ⇨ The slot enables the SD card.

2. Pull the SD card out straight.

⇨ After the SD card was removed, the **SD** status LED lights up twice for 100 ms each.

6.3 Commissioning

Commissioning the device is based on the following configuration:

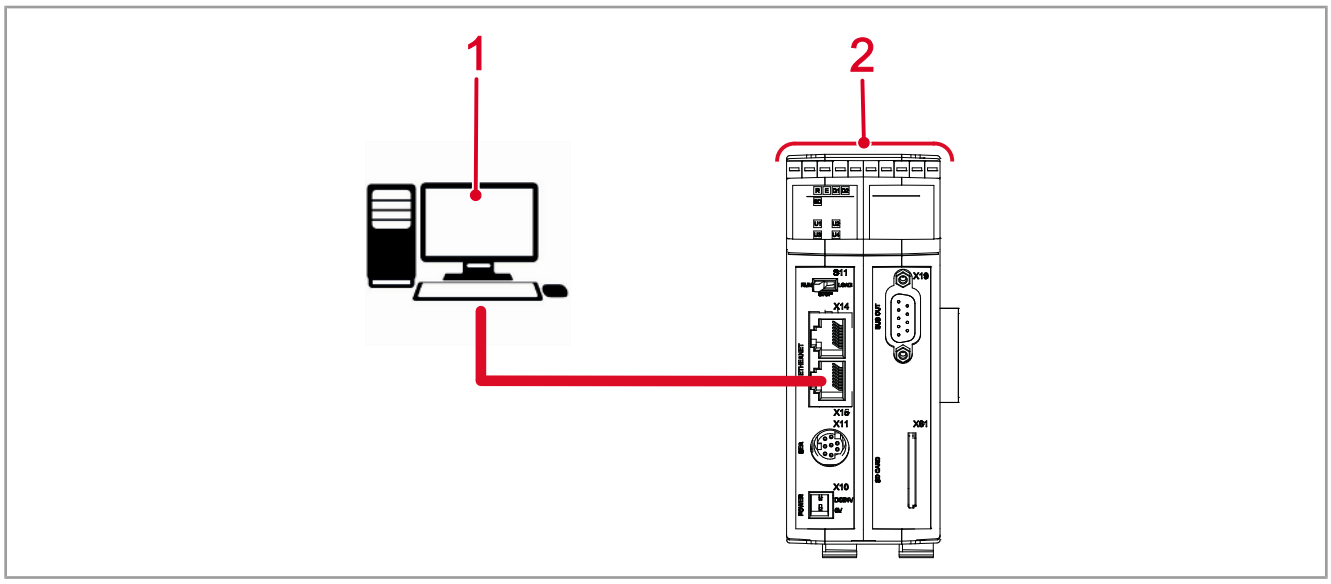


Fig. 15: Configuration

Number	Part	Description
1	PC	Programming system
2	JC-365MC	Controller

Ethernet connection with the controller

The IP address factory setting of the JC-365MC controller is 192.168.1.1. Configure the Ethernet interface of your PC to communicate with the controller via this IP address. In this example, a valid IP address of the PC is 192.168.1.20. It is import for both devices to share the same subnet.

Info

Behavior after power-up

The position of switch S11 (mode selector) on the controller must be chosen depending on the application. If the mode selector is in STOP position when the controller is powered up, the application program will not launch.

State of the LEDs

Following a correct commissioning, the LEDs are lit as follows:

LED	Status	Color	Description
R	Blinking at 1 Hz	Green	Logic voltage supply is OK; application program has been stopped
E	OFF	---	No error
D1	OFF	---	Normal operating condition
D2	OFF	---	Boot loader is not running
SD	OFF	---	The controller has no access to the SD card
U1 … U4	OFF	---	LEDs which are programmed depending on the application

Creating an application program

To create and check the program, proceed as follows:

1. Launch the JetSym programming environment.
2. Create a new project.
3. In JetSym, start the Hardware Manager by clicking the **Hardware** tab or hitting the keyboard command **[Alt] + [5]**.
   Open the window for controller configuration by double-clicking the **CPU** folder in Hardware Manager.
   Select the JC-365MC controller type.
4. Enter the following information:
   - the version of the installed OS system
   - **Ethernet** as interface type
   - the IP address
5. Open the program editor.
6. Enter the program specifications.
7. To compile the program, in the **Build** menu click the **Build** menu item (keyboard shortcut **[F7]**).
8. To upload the project to the controller, in the **Build** menu click the **Download** menu item (keyboard shortcut [Ctrl] + [F5]).
9. Open a setup pane.
10. Enter the variable name (Count).
11. Activate the setup.

ⓘ Info

Further information

For further information on this subject refer to the JetSym Online Help, chapter *Commissioning/Initial Commissioning/Controllers/JetControl controllers*.

JetSym STX program

The following program increments the content of a variable by one every 2 seconds:

```
Var
    Count:   Int;
End_Var;

Task increment Autorun
    Loop
        Inc(Count);
        Delay(T#2s);
    End_Loop;
End_Task;
```

Setup pane

The JetSym setup pane shows the content of the variable:

	Name	Nummer	Inhalt	Typ
1	Count		39 1	
2				
3				
4				

Number	Description	Function
1	Present content of the variable	The content of the variable is incremented by one every 2 s

6.3.1 Configuring error states

JetSym lets you configure the JC-365MC controller.

Commissioning

- ✓ JetSym has been installed on the PC.
- ✓ JetSym has been licensed (see JetSym Online Help).
- ✓ Limitations to be taken into account when engineering a JX3 station have been observed.

1. Make sure that the power supply is disconnected!
2. Assemble any JX3 modules as required to form JX3 station. Take into account the restrictions applying to the design of a JX3 station.
3. Use an Ethernet cable to connect the JC-365MC controller and the PC.
4. Set an IP address on the JC-365MC controller (see IP configuration [▶ 51]).
5. On the PC, set an IP address which differs from the IP address of the controller. Example: The controller has got IP address 192.168.1.1. Thus, for the PC, IP address 192.168.1.20 can be used. Make sure that the first 3 elements of the IP addresses are identical.
6. Switch on the power supply for the JX3 station.
7. Launch JetSym. Configure the JX3 station following the sample program.
8. Configure the JX3 station using the Hardware Manager.
9. Enter the sample program.
10. Upload the program to the controller.

ⓘ Info

Number of connectible expansion modules

You can calculate the exact amount of connectible expansion modules by means of the system bus configurator, JX3-sysbus_configurator_xxx_e, which is available for download on our website.

ⓘ Info

Further information

For more information on this subject, refer to the JetSym Online Help.

6.3.2 Configuration in JetSym

A simple example illustrates how to use JetSym for configuration. Connect the expansion module JX3-DIO16 as a second module to a JC-365MC controller.

In a minimum program, a flashing light has been programmed. Output 9 of the JX3-DIO16 module is set and then reset again.

Configuration

This example is based on the following configuration:

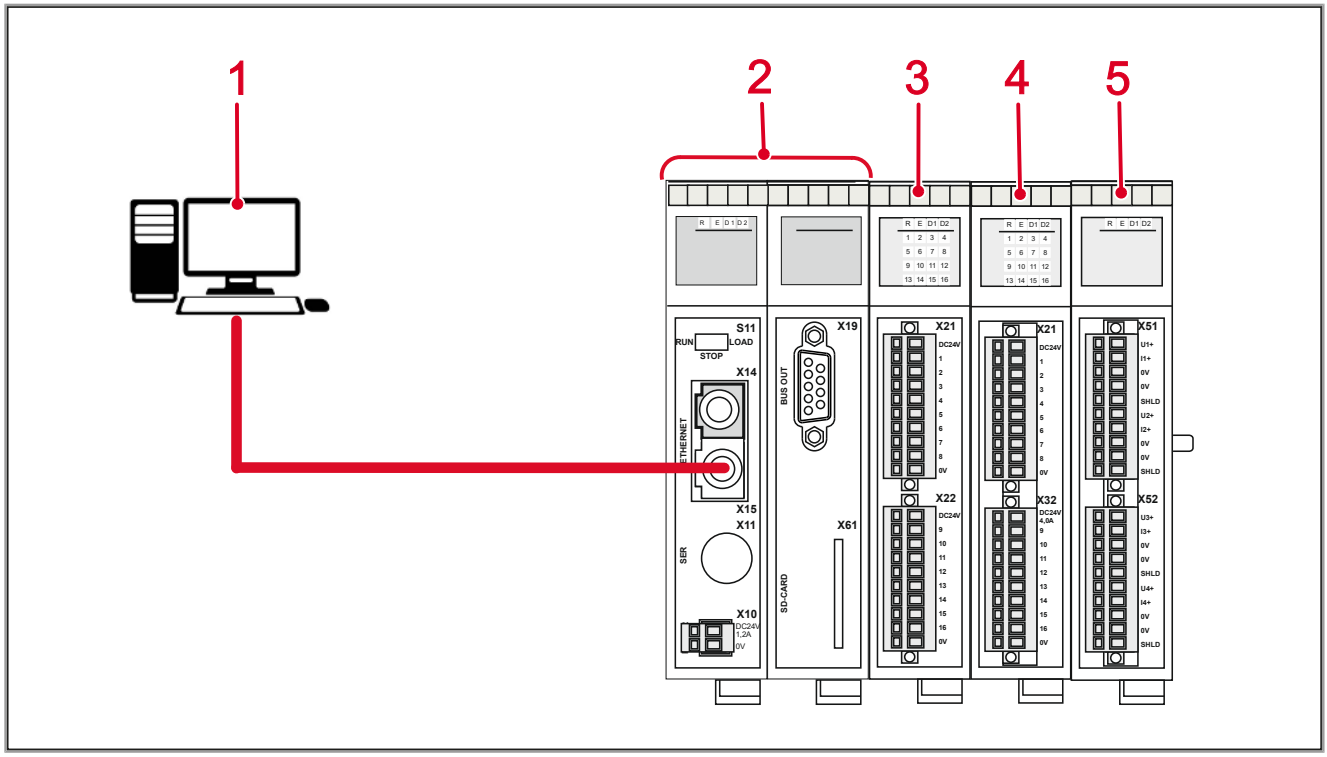


Fig. 16:

Number	Part	Description
1	PC	Programming system
2	JC-365MC	Controller
3	JX3-DI16	Expansion module
4	JX3-DIO16	Expansion module
5	JX3-AO4	Expansion module

1. Connect the power supply to the terminals X21.DC24V/X21.0V and X32.DC24V/X32.0V of the module JX3-DIO16.

   ⇨ Now, you can activate the digital outputs X32.9...16.
2. Launch the JetSym programming environment.
3. Create a new project.
4. Connect the Bucher Automation AG controller and expansion modules with the PC via Ethernet system bus.
5. Switch on the power supply.
6. In JetSym, start the Hardware Manager by clicking the **Hardware** tab or hitting the keyboard command **[Alt] + [5]**.
7. Configure the JX3 station by means of the Hardware Manager.

8. Activate the programming environment using the **[Alt]+ [0]** keyboard command. Alternatively, click the **File** tab.
9. Enter the program shown below.
10. Compile the program.
11. Upload the program to the controller.

⇨ LED **9** of the JX3-DIO16 module is blinking. The status indication updates every 5 seconds.

JetSym STX program

Output 9 of the module JX3-DIO16 is set and then reset again.

```
Task Flashing_light Autorun
    Loop
        OUTPUTS[100000309] := True;
        Delay(T#5s);
        OUTPUTS[100000309] := False;
        Delay(T#5s);
    End_Loop;
End_Task;
```

7 Identification and Configuration

7.1 Identification

This chapter describes how to identify the JC-365MC device:

– Determining the hardware revision
– Retrieving Electronic Data Sheet (EDS) information. The EDS holds numerous non-volatile production-relevant data.
– Determining the OS version of the device and its software components

7.1.1 Electronic Data Sheet (EDS)

Each JC-365MC features an Electronic Data Sheet (EDS). Numerous production-relevant data are permanently stored in the EDS. The EDS data can be read out via files in the file system of the controller or via special registers.

EDS file "eds.ini"

EDS data can be retrieved via the **eds.ini** file.

Properties

– You can access this file through the file system of the controller.
– For an FTP connection, the user needs administrator rights (*admin* user) or system rights (*system* user).
– The EDS file of the controller is located in the **System** folder.
– The EDS file of JX3 modules is located in the directory of the corresponding module */System/*JX3-*ModuleXX*.
– This file is read-only.
– Formatting the flash drive or SD card has no effect on the file.

File structure

The EDS file is a text file with its entries grouped into several sections.

Example

This is an example of an EDS file for a JC-365MC:

```
;Bucher Automation AG Electronic Data Sheet

[IDENTIFICATION]
Version = 2
Code = 878
Name = JC-365MC
PcbRev = 00
PcbOpt = 00
OSVersionMin = 0.0.0.0
BLVersionMin = 0.0.0.0

[PRODUCTION]
Version = 0
SerNum = 20150120000000
Day = 2
Month = 6
Year = 2015
TestNum = -1
TestRev = 255.255.255.255

[FEATURES]
Version = 1
MAC-Addr = 00:50:CB:00:00:00
Serial = 1
Switch = 1
STX = 1
NVRegs = 60000
JX3-Bus = 1
CAN = 1
SD-Card = 1
MotionControl = 1
Axes = -1
Web = 1
ModbusTCP = 1
SDLed = 1
UserLeds = 1
RTC = 1
```

[IDENTIFICATION] section

The general hardware configuration can be retrieved from the [IDENTIFICATION] section.

Name	Example	Function
Version	2	Version of this section
Code	878	Module code for JC-365MC
Name	JC-365MC	Corresponds to the information on the nameplate
PcbRev	00	PCB revision
PcbOpt	00	PCB option
OSVersionMin	0.0.0.0	The product is available as of this OS version

Name	Example	Function
BLVersionMin	0.0.0.0	The product is available as of this bootloader version

Tab. 21: [IDENTIFICATION] section

[PRODUCTION] section

The serial number and production date can be retrieved from the [PRODUCTION] section.

Name	Example	Function
Version	0	Version of this section
SerNum	20150602000000	Corresponds to the information on the nameplate
Day	02	Production date: Day
Month	06	Production date: Month
Year	2015	Production date: Year
TestNum	-1	Internal usage
TestRev	255255255255	Internal usage

Tab. 22: [PRODUCTION] section

[FEATURES] section

In the [FEATURES] section, special properties of the controller are specified. The OS of the controller interprets any properties that have not been entered in the file as non-existing.

Name	Example	Function
Version	1	Version of this section
MAC-Addr	00:50:CB:00:00:00	MAC address of the Ethernet interface
Serial	1	The serial interface is available
Switch	1	RUN/STOP/LOAD mode selector is available
STX	1	Runtime environment for the application program is available
NVRegs	60000	Total of non-volatile registers
JX3 bus	1	Bus interface for JX3 modules is available
CAN	1	Bus interface for JX2 modules is available
SD card	1	Slot for the SD memory card is available
SD LED	1	The LED for the SD memory card is available
UserLEDs	1	LEDs U1 through U4 are supported
RTC	1	A real-time clock is available
MotionControl	1	Motion Control feature is available
Axes	-1	Number of supported JX2 axis modules
Web	1	Web server and e-mail client are available
ModbusTCP	1	Modbus/TCP client and server are available

Tab. 23: [FEATURES] section

EDS registers

EDS registers let you retrieve entries made in the Electronic Data Sheet (EDS). These registers contain the exactly same information as the EDS file. They are read only (ro).

Register numbers

The basic register number is dependent on the controller. The register number is calculated by adding the number of the module register (MR) to the number of the basic register.

Device	Basic register number	Register numbers
JC-365MC	100000	100500 ... 100817

Tab. 24: Register numbers of the EDS

Registers - Overview

The following table lists the EDS registers of a controller, as well as their connection to the entries in the EDS file **/System/eds.ini**. You can use this register array to display the EDS of the controller or any JX3 module. To do this, select the controller or the desired JX3 module via the module registers 500 and 501. Data of the selected EDS can be retrieved from MR 600 and the following.

Registers	Section in eds.ini	Name in eds.ini	Description
MR 500	-	-	Functional group 0: CPU 1: JX3 module
MR 501	-	-	Module number (if MR 500 > 0)
MR 600	IDENTIFICATION	Version	Version of this section
MR 601		Code	Module code
MR 602 … MR 612		Name	Module name or controller name
MR 613		PcbRev	PCB revision
MR 614		PcbOpt	PCB option
MR 700	PRODUCTION	Version	Version of this section
MR 701 … MR 707		SerNum	Serial number
MR 708		Day	Production date: Day
MR 709		Month	Production date: Month
MR 710		Year	Production date: Year
MR 711		TestNum	Internal usage
MR 712		TestRev	Internal usage

Registers	Section in eds.ini	Name in eds.ini	Description
MR 800	FEATURES	Version	Version of this section
MR 801		MAC-Addr	MAC address (Bucher Automation)
MR 802		MAC-Addr	MAC address (device)
MR 803		Serial	Serial port
MR 804		Switch	Mode selector RUN/STOP/LOAD
MR 805		STX	Runtime environment for the application program
MR 806		NVRegs	Number of non-volatile registers
MR 807		JX3 bus	Bus interface for JX3 modules
MR 808		CAN	CAN bus for JX2 modules
MR 809		SD card	SD card slot
MR 810		MotionControl	MC software
MR 811		Axes	Number of supported JX2 axis modules
MR 812		Web	Web server and e-mail client
MR 813		ModbusTCP	Modbus/TCP client and server
MR 815		SD LED	LED of the SD card slot
MR 816		UserLEDs	User-defined LEDs
MR 817		RTC	Real-time clock

Tab. 25: Overview of EDS registers

7.1.2 Version registers

The operating system provides several registers which can be used to read out the hardware revision or OS version of the device and its components.
You will need this information when contacting the Bucher Automation AG support hotline in case of a problem.

Hardware revisions

The device has special registers, the content of which lets you identify the hardware.

Registers - Overview

The registers listed below let you retrieve the hardware revisions:

Registers	Description
108020	Hardware revision of the backplane module
108021	Hardware revision - CPU board
200170	Controller model

Tab. 26: Register overview - Hardware revisions

Operating system version

The device features special registers containing unique OS version numbers.

Software version numbers

The software version number of the device is a 4-digit value.

1 . 2 . 3 . 4

Digits	Description
1	Major or main version number
2	Minor or secondary version number
3	Branch or intermediate version number
4	Build version number

Tab. 27: Software version number format

Released version

A released version is identified by both the branch and build values being 0.

Register overview

The operating system versions can be read from the registers listed below:

Registers	Description
200168	Boot loader version
200169	Operating system version
210001	Version of the execution unit for the STX application program
100002000	Version of the JX3 system bus driver
200002000	Version of the JX2 system bus driver

Tab. 28: Registers - Overview

7.2 IP configuration

This chapter describes the IP configuration for the controller. The following parameters can be set:

– IP address of the controller
– Subnet mask
– IP address of the default gateway
– IP address of the DNS server
– Controller name
– IP port number for the JetSym debugger
– Basic port number for communication via JetIP
– Name of the AutoCopy command file

7.2.1 Factory settings

Prior to shipment of the JC-365MC controller, various parameters are set to a default value. These parameters can be changed by the user.

Factory settings

Parameter	Value
IP address of the controller	192.168.1.1
Subnet mask	255.255.255.0
IP address of the default gateway	0.0.0.0
IP address of DNS server	0.0.0.0
Controller name	JC-365MC
IP port number for debugger	52000
IP port number for JetIP	50000
Name for AutoCopy command file	/SD/autocopy.ini
DIP switch sliders	DIP switch slider 1 = ON All other DIP switch sliders = OFF
User's password *admin*	admin
User's password *system*	system

Tab. 29: Factory settings

7.2.2 Determining the IP address of the controller using JetIPScan

Introduction

The JetIPScan tool lets you determine the IP address, subnet mask and the IP address of the default gateway of the JC-365MC.

ⓘ Info

Downloading JetIPScan

Bucher Automation AG provides the JetIPScan program on its homepage. You will find the file **jetipscan_1-11-00.zip** for download under *Downloads - Software - Other Software Tools - JetIpScan*.

Contents of the ZIP file

The zip file **jetipscan_1-11-00.zip** contains the following files:

– The program JetIPScan_V_1-11-00.exe
– The help jetipscan_01_help_en.png
– The batch file read_IP_via_JETIPSCAN.bat to determine the IP address
– The batch file write_IP_via_JETIPSCAN_10_150.bat to set IP address 192.168.10.150 for the controller

The batch files launch the program JetIPScan.

The files are unzipped to the folder **jetipscan_1-11-00**.

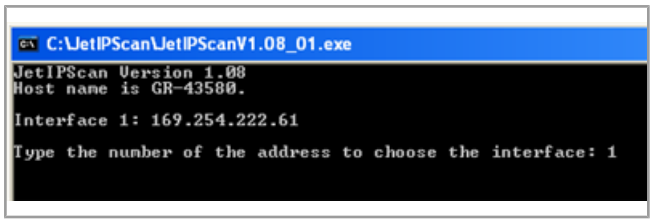
Determining the IP address

To determine the IP address of the JC-365MC, proceed as follows:

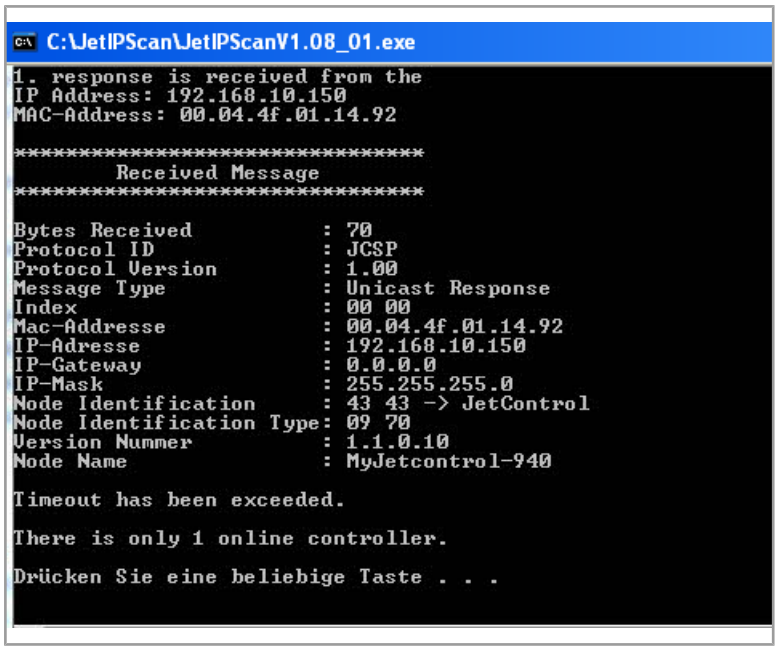
✓ An Ethernet connection exists between the PC and the JC-365MC.

1. Launch the JetIPScan program on your PC, for example via batch file **read_IP_via_JETIPSCAN.bat**.
2. Write a batch file. The content of the batch file is **JetIPScan_V_1-11-00.exe -R**.
3. Execute the batch file.

   ⇨ JetIPScan is launched and shows all IP addresses, which are presently active on your PC.
4. Select the interface (IP address) of the device whose IP address you want to determine.



⇨ Among others, JetIPScan shows the IP address of the JC-365MC.



7.2.3 Configuration memory

During the boot phase, the JC-365MC reads the parameters for initializing the IP interface from the **config.ini** configuration file which is physically part of the configuration memory. The configuration memory is located in a remanent memory (EEPROM) on the backplane module.

According to the contents of the configuration file, the controller writes values into configuration registers and then initializes the IP interface.

The user can access the data stored in the configuration memory in the following ways:

– Read and change data via the **config.ini** file in the system directory using FTP

– The user can read out or change data via configuration registers. When saving, a new configuration file is created from the data.

ⓘ Info

When does this take effect?

The controller reads out data located in the configuration memory only during the boot process. If you make changes to the configuration memory, reboot the controller. Only this way, these changes will take effect.

Default values

Prior to processing data from the configuration memory, JC-365MC checks them for plausibility. If the entries are not valid or do not exist, the controller uses the following default values:

Parameter	Default value
IP address of the controller	192.168.10.15
Subnet mask	255.255.255.0
IP address of the default gateway	0.0.0.0
IP address of DNS server	0.0.0.0
Controller name	JC-365MC
Suffix type of the name	0
IP port number for debugger	52000
IP port number for JetIP	50000
Name for AutoCopy command file	/SD/autocopy.ini

Tab. 30: Configuration memory - Default values

Storage location/controller replacement

The configuration memory is located on the backplane module. Owing to this approach, configuration data will be preserved when the function module is replaced.

7.2.4 Configuration file "config.ini"

If the data in the configuration memory is illegal or not valid, the JC-365MC uses the settings in the **config.ini** file to initialize the IP interface. This file has been stored to the EEPROM in the backplane module.

Properties

– You can access this file through the file system of the JC-365MC controller.
– For an FTP connection, the user must have administrator or system rights.
– This file is located in the folder **System**.
– You cannot delete the file, but only overwrite it.
– Formatting the Flash disk drive or the SD card leaves the file unchanged.

File structure

The configuration file is a text file the entries of which are grouped into several sections. The JC-365MC controller will replace missing IP configuration parameters by their factory-set fallback values.

Example

This is an example of a configuration file **config.ini**:

```
;JC-365MC System Configuration
;Copyright (c) 2008 by Bucher Automation AG

[IP]
Address = 192.168. 50. 1
SubnetMask = 255.255.255. 0
DefGateway = 192.168. 50. 11
DNSServer = 192.168. 1. 44

[HOSTNAME]
SuffixType = 0
Name = JC-365MC

[PORTS]
JetIPBase = 50000
JVMDebug = 52000

[FILES]
AutoCopyIni = /SD/autocopy.ini
```

Section [IP]

In the [IP] section the required IP addresses and the subnet mask are specified.

Address

Property	Description	
In the given example	192.168.50.1	
Function	IP address The settings of the DIP switch on the backplane module can overwrite the least significant byte.	
Allowed values	> 1.0.0.0	< 223.255.255.255
Illegal values	Network address, broadcast address	
In the event of an illegal value	All 4 values will be reset to default.	

Tab. 31: Address

SubnetMask

Property	Description
In the given example	255.255.255.0
Feature	Specifies the subnet mask
Allowed values	≥ 128.0.0.0
Illegal values	1 and 0 mixed
In the event of an illegal value	All 4 values will be reset to default.

Tab. 32: SubnetMask

DefGateWay

Property	Description	
In the given example	192.168.50.1	
Feature	IP address of the gateway to other subnets; equals 0.0.0.0, if no other nodes are available via the address/subnet mask.	
Allowed values	≥ 0.0.0.0	< 223.255.255.255
Illegal values	– Network address – Broadcast address – If no other nodes are available via the address/subnet mask – The Address value	
In the event of an illegal value	Will be set to 0.0.0.0	

Tab. 33: DefGateWay

DNSServer

Property	Description	
In the given example	192.168.1.44	
Feature	IP address of the server for the Domain Name System	
Allowed values	≥ 0.0.0.0	< 223.255.255.255
In the event of an illegal value	Will be set to 0.0.0.0	

Tab. 34: DNSServer

Section [HOSTNAME]

The [HOSTNAME] section contains the name of the JC-365MC. If desired, the controller automatically generates an individual name. Hostname is not currently used.

SuffixType

Property	Description	
In the given example	0	
Function	The type of the automatically generated suffix that is attached to the controller name	
Allowed values	0	No suffix
	1	Low-order byte of the IP address in decimal notation
	2	Low-order byte of the IP address in hexadecimal notation
In the event of an illegal value	0	

Tab. 35: SuffixType

Name

Property	Description	
In the given example	JC-365MC	
Function	Specifies the controller name	
Allowed values	First character	'A' ... 'Z', 'a' ... 'z'
	Next characters	'A' ... 'Z', 'a' ... 'z', '0' ... '9', '-'
In the event of an illegal value	JC-365MC	

Tab. 36: Name

Section [PORTS]

In section [PORTS] the IP port numbers of data and debug servers are specified. The IP port numbers must be consistent with, for example, the port numbers set in JetSym.

JetIPBase

Property	Description
In the given example	50000
Function	IP port for OS update and communication between devices
Allowed values	1024 ... 65535
In the event of an illegal value	50000

Tab. 37: JetIPBase

JVMDebug

Property	Description
In the given example	52000
Function	IP port for debugger/setup in JetSym
Allowed values	1024 ... 65535
In the event of an illegal value	52000

Tab. 38: JVMDebug

Section [FILES]

In section [FILES] the name of the command file for the AutoCopy function is specified.

AutoCopyIni

Property	Description
In the given example	/SD/autocopy.ini
Function	Command file for the AutoCopy function
Allowed values	Allowed path and file name
In the event of an illegal value	/SD/autocopy.ini

Tab. 39: AutoCopyIni

Changing the IP configuration via the configuration file

You can change the IP configuration directly in the **config.ini** configuration file.
To do this, carry out the following steps:

1. Create on your PC a configuration file named **config.ini** using a text editor and make the corresponding entries.
2. Open an FTP connection between the PC and the JC-365MC.
3. Choose a user login with administrator or system rights.
   Default login information:
   User: admin, Password: admin
   User: system; Password: system
4. Open the *System* folder of the JC-365MC.
5. Copy the **config.ini** configuration file you created into the *system* folder.
6. Clear the FTP connection.
7. Reboot.

⇨ The new configuration is active.

Configuration registers let you also make changes to the IP configuration.

7.2.5 Configuration registers

Configuration registers grant you read or write access to the parameters of the IP configuration. The configuration registers have 2 register areas that may differ in content.

Register numbers

The basic register numbers of both ranges are dependent on the device. The register number is calculated by adding the number of the module register (MR) to the number of the basic register.

Device	Data range	Basic register number	Register numbers
JC-365MC	Configuration memory	101100	101100 ... 101165
	Parameters used	101200	101200 ... 101265

Tab. 40: Register numbers of the configuration registers

Register areas

– Registers R 101100 ... R 101165 contain the data which the controller has determined from the contents of the configuration file during the boot phase. The user has read-only access to this data.
– Registers R 101200 ... R 101265 contain the data which are actually used to initialize the IP interface. The user has read and write access to this data.

The data of the two register areas may differ for the following reasons:

– Parameters of the IP interface can be changed at runtime. This affects the value of the registers R 101200 ... R 101202.
– The position of the DIP switch affects the actual IP address and thus the value of R 101200.

Registers - Overview

Registers	Section in config.ini	Name in config.ini	Description
MR 0	IP	Address	IP address of the controller
MR 1		SubnetMask	Specifies the subnet mask
MR 2		DefGateWay	IP address of the gateway to other subnets
MR 3		DNSServer	IP address of the server for the Domain Name System
MR 32	HOSTNAME	SuffixType	The type of the automatically generated suffix that is attached to the controller name
MR 33 … MR 51		Name	Specifies the controller name
MR 64	PORTS	JetIPBase	IP port number for OS updates and communication between controllers
MR 65		JVMDebug	IP port number for debugger/setup in JetSym
MR 80	FILES	AutoCopyIni	Name for AutoCopy command file

Tab. 41: Overview of configuration registers

7.2.6 Setting the IP address

To communicate via Ethernet, you must set a unique IP address on the controller.

Replacing the controller

The IP address is stored in the configuration memory on the backplane module. If you plug another controller of the same controller series onto the backplane module, the following configuration data are retained:

– IP address of the controller
– Subnet mask
– IP address of the default gateway
– IP address of DNS server
– Controller name
– Suffix type of the name
– IP port number for debugger
– IP port number for JetIP
– Name for AutoCopy command file

Configuration options

You can configure the IP address in the following ways:

– Setting the default IP address
– Configuration via file **config.ini**
– Configuration via **config.ini** file and DIP switch

– Configuration via the configuration registers (remanent)
– Configuration during runtime via special registers (not remanent)
– Configuration via JetIPScan program (remanent)

Changing the IP address

1. Switch off the power supply to the device.
2. Remove the controller enclosure from the backplane module.
3. Make the corresponding DIP switch settings.
4. Reattach the controller enclosure to the backplane module.

⇨ After the restart, the device can be reached at the new IP address.

Setting the default IP address

The default IP address is 192.168.10.15 You can reset the IP address to its default value at any time.

DIP switch slider settings

To set the default IP address 192.168.10.15, position the DIP switch sliders (1 … 12 = OFF) as shown below:

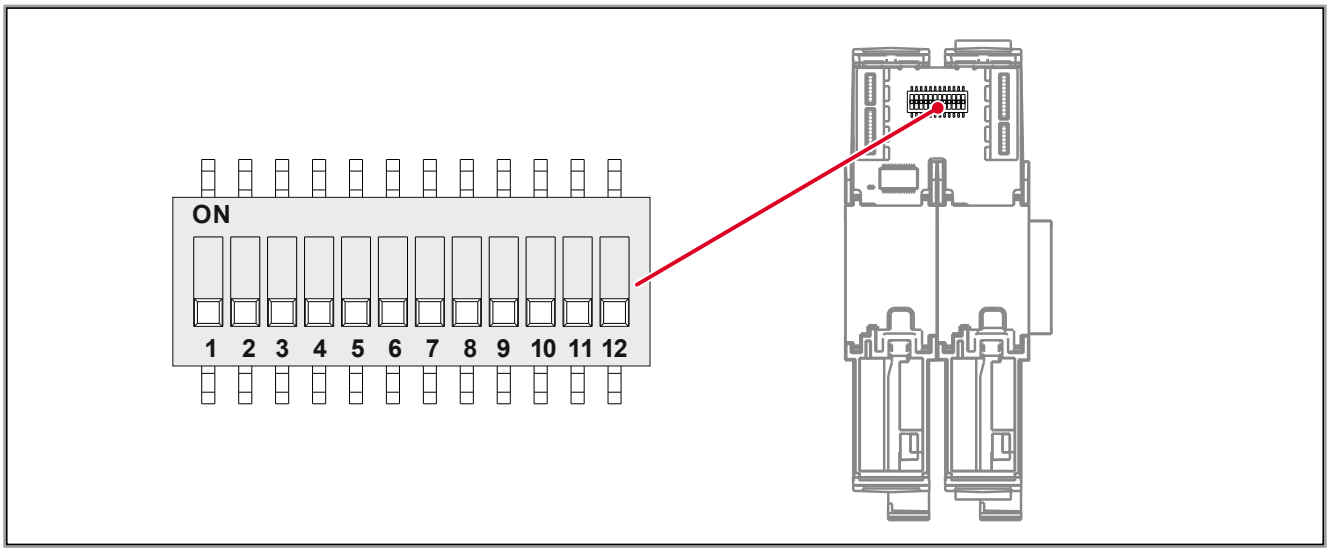


Fig. 17: DIP switch sliders 1 ... 12 OFF

Changing the IP address of the controller

Introduction

The JetIPScan tool lets you change the IP address, subnet mask and the IP address of the default gateway of the JC-365MC.

ⓘ Info

Downloading JetIPScan

Bucher Automation AG provides the JetIPScan program on its homepage. You will find the file **jetipscan_1-11-00.zip** for download under *Downloads - Software - Other Software Tools - JetIpScan*.

Contents of the ZIP file

The zip file **jetipscan_1-11-00.zip** contains the following files:

- The program JetIPScan_V_1-11-00.exe
- The help jetipscan_01_help_en.png
- The batch file read_IP_via_JETIPSCAN.bat to determine the IP address
- The batch file write_IP_via_JETIPSCAN_10_150.bat to set IP address 192.168.10.150 for the controller

The batch files launch the program JetIPScan.

The files are unzipped to the folder **jetipscan_1-11-00**.

Possible commands of the software JetIPScan

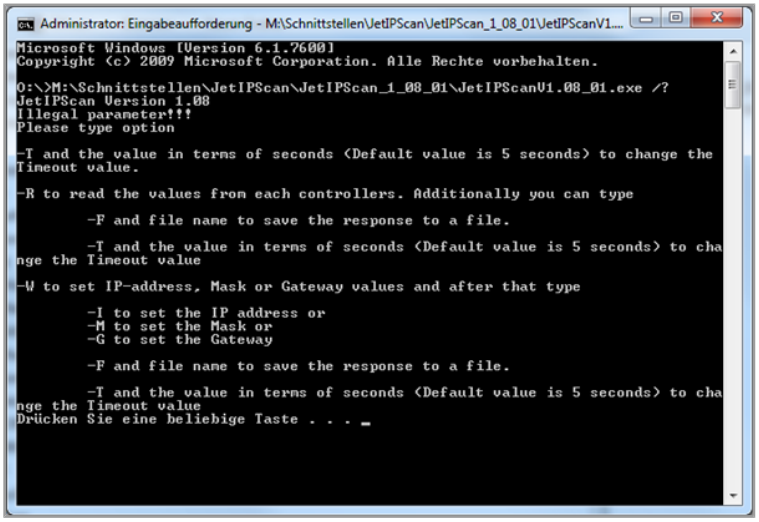


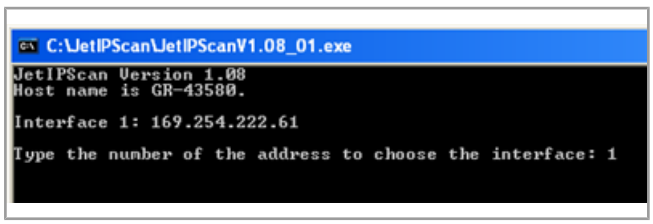
Fig. 18: JetIPScan - command options

Changing the IP address

✓ An Ethernet connection exists between the PC and the JC-365MC.

1. Then start the JetIPScan tool on your PC, e. g. by means of the batch file **write_IP_via_JETIP-SCAN_10_150.bat**.
2. Write a batch file. The content of the batch file is **JetIPScan_V_1-11-00.exe -W -I 192.168.10.150**.
3. Execute the batch file.

   ⇨ JetIPScan is launched and shows all IP addresses, which are presently active on your PC.
4. Select the interface (IP address) of the device whose IP address you want to change.



   ⇨ JetIPScan shows all the devices found.
5. To make changes to the IP address of a device, select the respective device from the list.

   ⇨ JetIPScan changes the IP address of the JC-365MC to 192.168.10.150.

Changing the subnet mask

1. Write a batch file. The content of the batch file is, for example, **JetIPScan_V_1-11-00.exe -W -M 250.255.248.0**.
2. Execute the batch file.
3. JetIPScan is launched and shows all interfaces, which are presently active on your PC.
4. For all further steps, please refer to the instruction **Changing the IP address**.

Changing the Default Gateway

1. Write a batch file. The content of the batch file is, for example, **JetIPScan_V_1-11-00.exe -W -G 192.168.4.1**.
2. Execute the batch file.

   ⇨ JetIPScan is launched and shows all interfaces, which are presently active on your PC.
3. For all further steps, please refer to the instruction **Changing the IP address**.

Setting the IP address via configuration file

Setting the IP address

The IP address can be set via the **config.ini** configuration file.

```
[IP]
Address = aaa.bbb.ccc.ddd
...
```

Element	Description
`Address`	Enter the IP address into this line.
`aaa`	1. byte of IP address
`bbb`	2. byte of IP address
`ccc`	3. byte of IP address
`ddd`	4. byte of IP address

Tab. 42: Elements of the IP address

DIP switch slider settings

To enable the JC-365MC controller to read the IP address from the **config.ini** file, position the DIP switch sliders (1 … 8 = ON) as shown below:

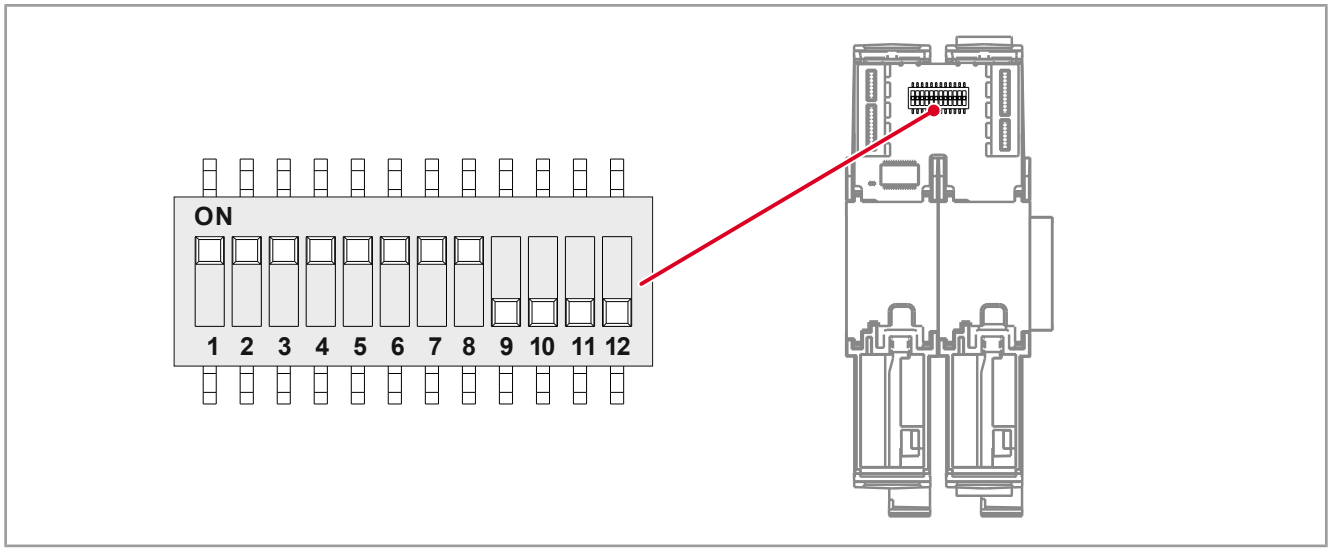


Fig. 19: DIP switch sliders 1 ... 8 ON

Transferring the configuration file

1. Establish an FTP connection between PC and JC-365MC.
2. Choose a user login with administrator or system rights.
   Default login information:
   User: *admin*; Password: *admin*
   User: *system*; Password: *system*
3. Open the **System** folder of the JC-365MC.
4. Copy the **config.ini** file to the **System** folder.
5. Clear the FTP connection.
6. Reboot.

Setting the IP address via configuration file and DIP switch

You can set the IP address by combining the settings in the **config.ini** configuration file and the DIP switch sliders on the backplane module. This method uses contents of the **config.ini** file to populate the upper 3 bytes of the IP address, and the setting of the DIP switch slider 1 … 8 for the lower byte.

Setting the IP address

Set the upper 3 bytes of the IP address via the **config.ini** file.

```
[IP]
Address = aaa.bbb.ccc.1
...
```

Element	Description
`Address`	Line where to enter the upper 3 bytes of the IP address
`aaa`	First byte of IP address
`bbb`	Second byte of IP address
`ccc`	Third byte of IP address
`ddd`	Dummy entry; must be **1**

Tab. 43: IP address elements

DIP switch slider settings

To enable the JC-365MC controller to read the IP address from the **config.ini** file and the respective switch elements, position the DIP switch sliders as shown below:

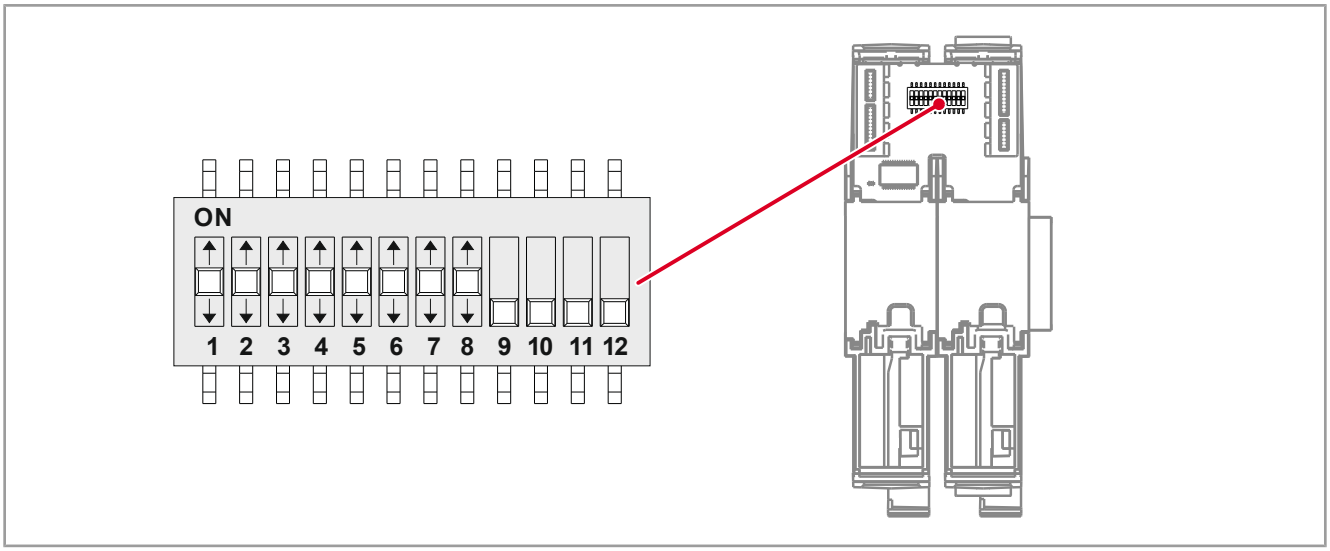


Fig. 20: DIP switch sliders 1 ... 8 arbitrarily

DIP switch									IP address
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9 … 12**	
OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	Default IP address
ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	aaa.bbb.ccc.1
OFF	**ON**	OFF	OFF	OFF	OFF	OFF	OFF	OFF	aaa.bbb.ccc.2
ON	**ON**	OFF	OFF	OFF	OFF	OFF	OFF	OFF	aaa.bbb.ccc.3
...	…	…	…	…	…	…	…	…	…
OFF	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**	OFF	aaa.bbb.ccc.254
ON	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**	OFF	from config.ini

Transferring the configuration file

1. Establish an FTP connection between PC and JC-365MC.
2. Choose a user login with administrator or system rights.
   Default login information:
   User: *admin*; Password: *admin*
   User: *system*; Password: *system*
3. Open the **System** folder of the JC-365MC.
4. Copy the **config.ini** file to the **System** folder.
5. Clear the FTP connection.
6. Reboot.

Setting the IP address via non-volatile registers

The IP interface is initialized by the settings in the configuration memory during the boot phase. You can change the following settings via registers. The values are then stored to the remanent memory:

– IP address of the controller
– Subnet mask
– IP address of the default gateway
– IP address of DNS server
– Host name and suffix type
– Port numbers for JetIP and JetSym debugger
– Name for AutoCopy command file

Registers - Overview

Register (range)	Description
101200	IP address
101201	Subnet mask
101202	IP address of the default gateway
101203	IP address of DNS server
101232	Host name suffix type
101233 … 101251	Host name
101264	Port number for JetIP
101265	Port number for STX debugger
101280 … 101298	Name for AutoCopy command file
101299	Saving the settings (0x77566152)

Tab. 44: Overview over the configuration memory registers

NOTICE



Limitation of the write cycles

The EEPROM data on the backplane module allow for 100,000 write cycles.

► Read a value first.
► Compare the value.
► Write only if you actually want to change the value.

Setting the configuration values to be non-volatile

To change the configuration values to become non-volatile, proceed as follows:

1. Enter the desired configuration data into one or several registers in the range 101200 … 101298.
2. To have the controller apply the values, you must enter a password.
   For this, enter value 2002149714 (0x77566152) into register 101299.
3. Wait for the controller to write value 0 into MR 101299.

   ⇨ The save process is now completed.
4. Boot the controller.

⇨ The settings are completed. Communication is possible again.

ⓘ Info

Influence of the DIP switches on the IP address

The position of the DIP switch sliders has an influence on the actual IP address (see Configuration registers [▶ 58]).

Effect

By writing to register 101299, the controller executes the following:

– The controller creates a configuration file out of the values.
– It saves the configuration file to the backplane module as **/System/config.ini**.
– If you have entered comments and formatting details into this file, the comments and formatting details will be lost during this process.

Setting the IP address during runtime

The IP interface is initialized by the settings in the configuration memory during the boot phase. Registers let you make changes to the settings listed below. These changes will then be saved to a volatile memory:

– IP address of the controller
– Subnet mask
– IP address of the default gateway

ⓘ Info

Important note

The settings made during runtime do not overwrite the parameters in the configuration memory. When de-energizing the controller, your settings will be lost.

To set the IP address and the subnet mask, proceed as follows:

- ✓ While settings are being made, no communication via IP interface is allowed. Otherwise, this would lead to a loss of data.
- ✓ The values entered must be valid. This can be ensured, e.g. by including a validity check in the application program.
  If you set the parameters at runtime, no check takes place.

1. Enter the value 0.0.0.0 into R 104533.
2. Enter the value 0.0.0.0 into R 104532.
3. Enter the desired IP address into R 104531.
4. Enter the desired subnet mask into R 104532.
5. Enter the desired IP address of the default gateway into R 104533.

⇨ The settings are completed. Communication is possible again.

IP address in the GNN operating mode

In GNN (Global Network Number) mode , the JC-365MC controller functions as a network node within a greater control network. The controller acquires its IP address during boot-up from the **config.ini** configuration file. Next, the main controller, which is the NetConsistency master, commands the network node to log in with its GNN. After accepting the network node, the main controller compares the set IP configuration with the actual IP configuration of the network node. If a difference results, the main controller makes the corresponding changes in the set IP configuration in the network node.

Setting the IP address

The IP address can be set via the **config.ini** configuration file.

```
[IP]
Address = aaa.bbb.ccc.ddd
...
```

Element	Description
`Address`	Enter the IP address into this line.
`aaa`	1. byte of IP address
`bbb`	2. byte of IP address
`ccc`	3. byte of IP address
`ddd`	4. byte of IP address

Tab. 45: Elements of the IP address

DIP switch slider settings

To enable GNN mode, move DIP switch slider 9 to **ON**.
This enables the JC-365MC controller to read the IP address from the **config.ini** file, and the GNN from the lower 8 positions of the DIP switches sliders. The DIP switch sliders are binary-coded.

– Valid values for the GNN: 1 ... 199.

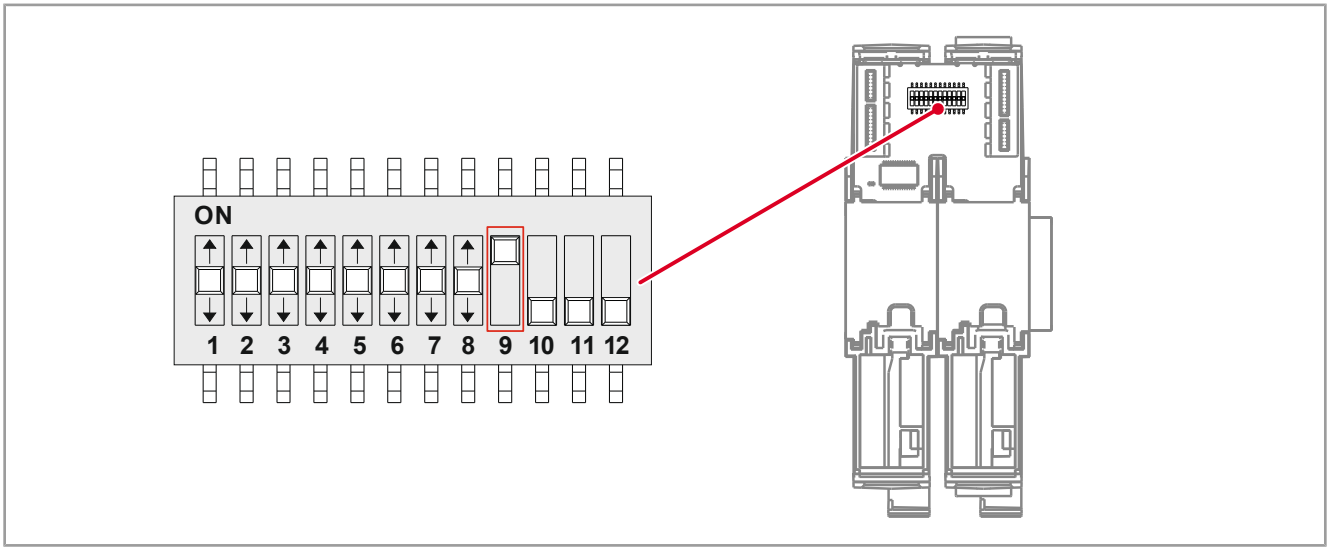


Fig. 21: DIP switch GNN

DIP switch										GNN
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10 … 12**	
OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	**ON**	OFF	Invalid
ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	**ON**	OFF	1
OFF	**ON**	OFF	OFF	OFF	OFF	OFF	OFF	**ON**	OFF	2
ON	**ON**	OFF	OFF	OFF	OFF	OFF	OFF	**ON**	OFF	3
...	…	…	…	…	…	…	…	…	…	
ON	**ON**	**ON**	OFF	OFF	OFF	**ON**	**ON**	**ON**	OFF	199
...	…	…	…	…	…	…	…	…	…	
ON	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**	**ON**	OFF	Invalid

7.2.7 Using names for IP addresses

You can also use names when specifying IP addresses of external communication partners, for example, when configuring the e-mail client. The controller resolves the names into IP addresses. A configuration file or the Domain Name System is used to resolve a name into its corresponding IP address. Names can be resolved to IP addresses during the boot process, or during runtime of the controller.

Name resolution

Names are resolved to IP addresses in the following way:

<table>
<tr><th>Step</th><th colspan="2">Description</th></tr>
<tr><td>1</td><td colspan="2">In the boot phase, the IP address of the DNS server is read from the configuration memory.</td></tr>
<tr><td>2</td><td colspan="2">In the boot phase, the /etc/hosts file is read. Then, the controller creates a translation table with the names and IP addresses found in this file.</td></tr>
<tr><td>3</td><td colspan="2">After the boot process the controller detects a name instead of an IP address.</td></tr>
<tr><td rowspan="3">4</td><td colspan="2">Based on this translation table, the controller tries to resolve the name into a related IP address.</td></tr>
<tr><td>If …</td><td>… then …</td></tr>
<tr><td>... the controller could resolve the name, ...</td><td>... proceed with step 6.</td></tr>
<tr><td></td><td>... the controller could not resolve the name, ...</td><td>... proceed with step 5.</td></tr>
<tr><td rowspan="3">5</td><td colspan="2">The controller tries to resolve the name into a related IP address by sending a request to the DNS server.</td></tr>
<tr><td>If …</td><td>… then …</td></tr>
<tr><td>... the controller has resolved the name,</td><td>... it enters the name and IP address into the translation table;
→ proceed with step 6.</td></tr>
<tr><td></td><td>... the controller has not resolved the name,</td><td>... the controller aborts the function, e.g. the system function for sending an e-mail, and issues an error message.</td></tr>
<tr><td>6</td><td colspan="2">The controller uses the found IP address for further communication.</td></tr>
</table>

Tab. 46: Name resolution

Configuration File "hosts"

The **hosts** configuration file holds the static assignment between names and IP addresses. During the boot phase, the JC-365MC reads the file once.

– File format: Text
– Location: */etc*
– File name: **hosts**

Domain Name System (DNS)

If a name cannot be found in the **/etc/hosts** file, the JC-365MC tries to resolve the IP address by DNS server request. During the boot process the JC-365MC reads the IP address of the DNS server from the configuration memory.

7.3 Operating system

We are continuously striving to enhance the operating systems of our products. Enhancing means adding new features, and upgrading existing functions. Current OS files are available for download on our homepage in the downloads area of the respective product.

ⓘ Info

Further information

More information on this subject is available on our website.

Start | Bucher Automation - We automate your success.

Operating system update

It is possible to update the operating system of the following devices if connected to a JC-365MC:

– This controller
– Bus node JX3-BN-ETH on the Bucher Automation Ethernet system bus
– JX3 IO modules on the JX3 system bus
– Slave modules on a JX2 system bus

7.3.1 Operating system update of the controller

This chapter describes how to carry out an operating system update on the JC-365MC controller. You have got several options for transferring the OS file to the controller:

– Using the programming tool JetSym
– Via an FTP connection
– From an SD card
– From within the application program

Operating system update using the programming tool

The JetSym programming tool offers an convenient way of transferring an OS file to the JC-365MC controller.

Performing the update

✓ The mode selector of the controller is in **RUN** or **STOP** position before controller boot-up.
✓ An OS file for the JC-365MC controller is available.
✓ A UDP/IP and a TCP/IP connection between programming tool and JC-365MC controller is possible.
✓ The number of the IP port is set in the configuration memory as IP basic port number for JetIP communication.
✓ The OS is running.
✓ The controller keeps running throughout the update.

1. In the JetSym **Build** menu, select **Update OS...**

   ⇨ The file selection dialog opens.

2. Select the new OS file here.

   ⇨ JetSym opens a confirmation dialog.

3. Launch the OS upload by clicking the button **Yes**.
4. Wait until the update process is completed.

5. Set the mode selector to **STOP** or **RUN** position.
6. To activate the transferred OS, re-boot the controller.

Updating the operating system by means of FTP

Using an FTP client an OS file can be transferred to the JC-365MC controller.

Performing the update

- ✓ Before the controller is switched on, the mode selector of the controller is in the **RUN** or **STOP** position.
- ✓ An OS file for the JC-365MC controller is available.
- ✓ An FTP connection to the controller is possible.
- ✓ The login parameters for a user with administrator or system rights are at hand.
- ✓ The OS is running.
- ✓ Make sure the controller remains energized during the operating system update.

1. Open an FTP connection to the JC-365MC.
2. Log in with administrator or system rights.
3. Navigate to the directory */System/OS*.
4. Transfer the OS file.
5. Wait until the update process is completed.
6. Clear the FTP connection.
7. Set the mode selector to **STOP** or **RUN**.
8. To activate the transferred OS, re-boot the controller.

Operating system update from an SD card

An automatic OS update of the controller from an SD card can be carried out using the AutoCopy function.

ⓘ Info

Further information

For more information on this subject refer to the application-oriented manual *File System* available for download from our homepage.

Updating the OS from within the application program

The file functions included in the STX language allow for a program-controlled OS update of a JC-365MC from within an OS file.

Performing the update

- ✓ An OS file is available in the JC-365MC file system.
- ✓ The operating system and the application program must be running.
- ✓ Make sure the controller remains energized during the operating system update.

1. Copy the OS file to a file of any name and of the extension ***.os** in the directory */System/OS*.
2. To activate the updated OS, for example by writing to the system command register, re-boot the controller.

STX program

```
Var
    SourceName:          String;
    DestinationName:      String;
    UpdateIt:              Bool;
End_Var;

//*****************************************************
// Name: OSupdate
// 1. Enable tracing in JetSym
// 2. Sett the name of the source file in 'SourceName'
// 3. Sett the flag 'UpdateIt'
//*****************************************************

Task OSupdate Autorun
    Var
        ResCopy: Int;
    End_Var;

    DestinationName := '/System/OS/OperatingSystem.os';
    Loop
        UpdateIt := False;
        When UpdateIt Continue;
        ResCopy := FileCopy(SourceName,
                            DestinationName);
        Trace('Result : ' + IntToStr(ResCopy) + '$n');
    End_Loop;
End_Task;
```

7.3.2 Operating system update of a JX2 or JX3 module

This chapter describes how to execute an OS update via the controller for a JX2 module connected to the JX2 system bus, or for a JX3 module connected to the JX3 system bus.

You have got several options to transfer the OS file to the module:

– From within the JetSym programming tool
– Via FTP connection
– From SD card
– From within the application program

Operating system update using the programming tool

The JetSym programming tool lets you transfer an OS file to a JX2 module on the JX2 system bus or to a JX3 module on the JX3 system bus via the controller in a convenient way.

Performing the update

✓ The mode selector of the controller is in the **RUN** or **STOP** position before the controller is switched on.
✓ An OS file for the module is available.
✓ A UDP/IP and a TCP/IP connection between programming tool and controller is possible.
✓ The number of the IP port is set in the configuration memory as IP basic port number for the JetIP communication.
✓ The operating system is running.
✓ The controller has initialized the system bus including the connected modules without errors.
✓ Make sure the controller remains energized during the operating system update.
✓ The modules are and remain switched on.

1. In JetSym, from the **Build** menu select menu item **Update OS...** .

   ⇨ The file selection dialog opens.
2. Select the new OS file here.

   ⇨ JetSym prompts a confirmation dialog.
3. Confirm by clicking **Yes.**

   ⇨ JetSym prompts an input box for entering the interface type and module number.
4. Enter the interface type (2 for the JX2 system bus or 1 for the JX3 system bus) and the module number (2 ... 23).
5. Launch the OS upload by clicking the button **Update**.
6. Wait until the update process is completed. If you want to update further modules, repeat steps 1 ... 6.
7. Set the mode selector to **STOP** or **RUN**.
8. To activate the transferred OS, re-boot the controller.

Updating the operating system by means of FTP

An FTP client lets you transfer an OS file to a JX2 module on the JX2 system bus or to a JX3 module on the JX3 system bus of the controller.

Performing the update

- ✓ The mode selector of the controller is in the RUN or STOP position before the controller is switched on.
- ✓ An OS file for the module is available.
- ✓ An FTP connection to the controller is possible.
- ✓ The login parameters for a user with administrot or system rights are at hand.
- ✓ The operating system is running.
- ✓ The controller has initialized the system bus including the connected modules without errors.
- ✓ Make sure the controller remains energized during the operating system update.
- ✓ The modules are and remain switched on.

1. Establish an FTP connection to the controller.
2. Log in with administrator or system rights.
3. Navigate to the directory */System/JX2-Slave/OS* or *System/JX3-Module/OS*.
4. Transfer the OS file.
5. Wait until the update process is completed.
6. Clear the FTP connection.
7. Set the mode selector to STOP or RUN.
8. To activate the transferred OS, re-boot the controller.

Operating system update from an SD card

An automatic OS update of a JX2 or JX3 module from an SD card can be executed via the AutoCopy function.

ⓘ Info

Further information

For more information on this subject refer to the application-oriented manual *File System* available for download from our homepage.

Updating the OS from within the application program

The file functions included in the STX language scope let you carry out a program-controlled OS update from an OS file for a JX2 module connected to the JX2 system bus, or for a JX3 module connected to the JX3 system bus.

Performing the update

- ✓ An OS file for the module is available in the file system of the controller.
- ✓ The operating system of the controller and the application program are running.
- ✓ The controller has initialized the system bus including the connected modules without errors.
- ✓ Make sure the controller remains energized during the operating system update.
- ✓ The modules are and remain switched on.

1. Copy the operating system file into a file with any name and the file extension ***.os** in the directory */System/*JX2-*Slave/OS/*, or */System/*JX3-*Module/OS/*
2. If you want to update other modules, repeat step 1.
3. To activate the updated OS, for example by writing to the system command register, re-boot the controller.

STX program

```
Var
    SourceName:          String[100];
    DestinationName:      String[100];
    UpdateIt:              Bool;
End_Var;

//*****************************************************
// Name: OSupdate
// 1. Enable tracing in JetSym
// 2. Sett the name of the source file in 'SourceName'
// 3. Setting the name of the target file in 'DestinationName'
// 4. Sett the flag 'UpdateIt'
//*****************************************************

Task OSupdate Autorun
    Var
        ResCopy: Int;
    End_Var;

    Loop
        UpdateIt := False;
        When UpdateIt Continue;
        ResCopy := FileCopy(SourceName,
                            DestinationName);
        Trace('Result : ' + IntToStr(ResCopy) + '$n');
    End_Loop;
End_Task;
```

7.4 File System

The file system lets you access files located on the internal flash disk and the SD card. Some files may be protected against read/write access or deletion. Some of these files are virtual files, such as firmware images, or protected files, such as EDS files.

NOTICE



Malfunctions caused by missing or damaged system files

Careless working with system files can result in malfunctions of the device.

► Do not delete or move any system files.

File categories

The files of the file system are categorized as follows:

– System directories or system files used by the operating system
– Files accessible to the user

System directories

The user is not allowed to delete system directories. System directories even survive formatting.

Directory	Description
/System	– System configuration – System information
/SD	– Root directory of the SD memory card

Tab. 47: System directories

ⓘ Info

Further information

For more information on this subject refer to the application-oriented manual *File System* available for download from our homepage.

7.4.1 Properties

The following properties apply to the internal flash disk and the SD card:

– 8 files max. to be opened simultaneously.
– Separate directory names by a slash "/", not by a backslash "\".
– When the controller creates a file, the file contains date and time assigned by the controller.
– Date, time, and file size are not available for all system files.

Flash disk - Properties

Size

The following disk space is available to the user:

– 24 MB

Features

The internal flash disk drive has got the following properties:

– Up to 7 directory levels and 1 file level are allowed.
– Differentiation between upper and lower case.
– Directory and file names with a length of up to 63 characters are possible.
– All characters except "/" and ".." are permitted for directory and file names.
– User/access administration for a maximum number of 31 locks and 33 users.

SD memory card - Properties

Size

The available capacity depends on the SD card used:

– Tested size: up to 32 GB (SDHC)

Properties

– The SD memory card must be compatible with FAT 16 or FAT 32.
– No case sensitivity.
– Directory and file name length must not exceed 63 characters.
– The following characters are not permitted in directory and file names: "/", "\", ":", "*", "?", "<", ">" and "|"
– The number of subdirectory levels depends on the formatting.
– There is no user/access management.

Bucher Automation AG shall be liable only for those SD cards marketed as an option.

8 Programming

Programming of the JC-365MC is performed using the JetSym programming tool.

8.1 Abbreviations, module register properties and formats

Abbreviations

The abbreviations used in this document are listed in the table below:

Abbreviation	Description
R 100	Register 100
MR 150	Module register 150

Tab. 48: Abbreviations

Module register properties

Each module register is characterized by certain properties. Most properties are identical for many module registers. In the following description, module register properties are mentioned only if a property deviates from the default properties listed below.

Property	Standard design
Access	Read/write
Value after reset	0 or undefined (e.g. revision/version number)
Takes effect	Immediately
Write access	Always
Data type	Integer

Tab. 49: Module register properties

Numerical formats

The numerical formats used in this document are listed in the table below:

Notation	Format of numerical values
100	Decimal
0x100	Hexadecimal
0b100	Binary

Tab. 50: Numerical formats

8.2 Storage options - Overview

The controller JC-365MC features several types of program and data memories. This memory is located directly in the CPU or in separate memory or I/O modules.

There is volatile and non-volatile memory:

– Volatile memory loses its content at switching off.
– Non-volatile memory keeps its content even when the power supply is off.

8.2.1 Operating system memory

The operating system is located in the system area of the controller's flash memory.

Properties

– Internal non-volatile flash memory for the operating system
– After switching on the device with mode selector S11 in RUN or STOP position, the operating system is loaded from the flash memory into the RAM of the controller and started.

Types of access

– The user is not allowed to directly access the OS memory.
– The operating system can be changed via an update.

8.2.2 File system memory

The file system memory is for storing data and program files.

Properties

– Non-volatile
– Slow access: Milliseconds up to seconds
– Limited number of write/erase cycles: Approx. 1 million
– Size of internal flash disk: 24 MB
– Size of the external SD memory card: 32 MB ... 32 GB

Types of access

– By the operating system
– By JetSym
– Via FTP connection
– By the email client
– Via a browser (via the HTTP server)
– Via file commands from the application program and through the AutoCopy function

8.2.3 Application Program Memory

By default, the application program is uploaded from JetSym to the controller and is remanently stored to its flash memory.

When the program is started, the application program is loaded by the operating system from the flash memory into the volatile memory (SDRAM) and executed there.

Properties

– Stored as file within the file system; file name: <Programmname.es3>
– Default directory */app*
– Files may also be stored to other directories (or on SD card)
– The size limited by the available flash memory and the SDRAM capacity
– Several programs can be stored in subdirectories in the */app* directory
– The **start.ini** file lets you select the program to be started (see Saving and loading an application program [▶ 92]).

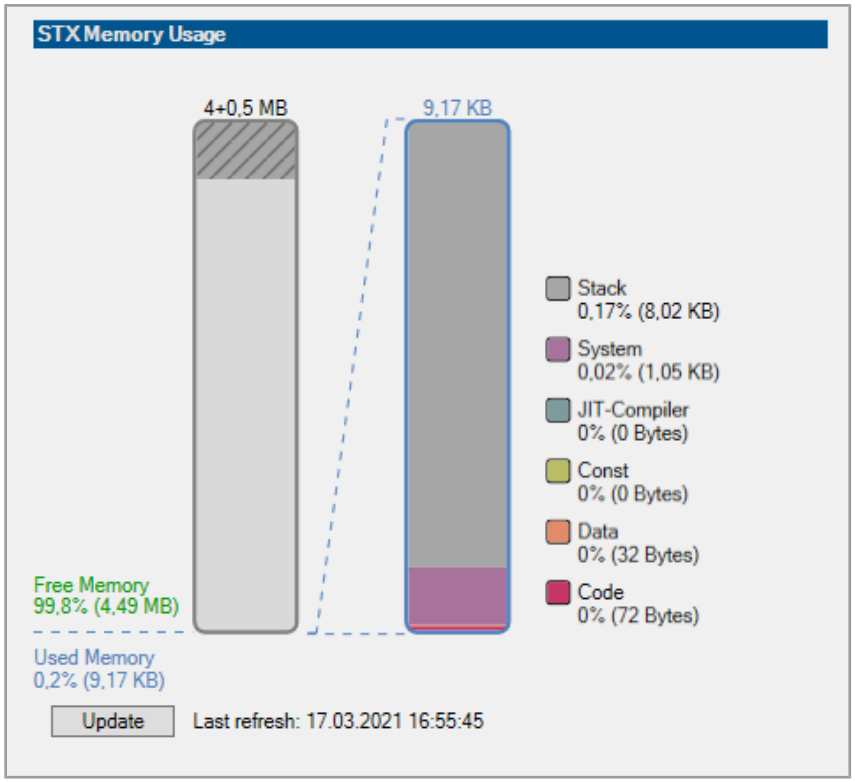
Types of access

– By the operating system
– By JetSym
– Via FTP connection
– Via file commands from the application program and through the AutoCopy function

STX memory utilization

During runtime, JetSym lets you determine the memory utilization by the application program in the SDRAM.

To view the memory utilization, go to the **Hardware** tab and click on **CPU**. On the right side you can see the STX memory usage.



Click the **Refresh** button and the current STX memory usage will be displayed.

8.2.4 Memory for volatile application program variables

Volatile variables are used to store data that need not be maintained when de-energizing the JC-365MC.

Properties

– Global variables which are not assigned to permanent addresses (not %VL or %RL)
– Local variables
– Variables are stored in a compact way according to their type size
– Variables are initialized with value 0 when they are created

Access

– Via JetSym
– From within the application program

JetSym STX program

The following program example increments a global volatile variable by 1 every 2 seconds.

```
Var
    Count: Int; //volatile, because non-localized
End_Var;

Task increment Autorun
    Loop
        Inc(Count);
        Delay(T#2s);
    End_Loop;
End_Task;
```

Setup pane

The JetSym setup pane shows the content of the variable:

	Name	Nummer	Inhalt	Typ
1	Count		39 1	
2				
3				
4				

Number	Description	Function
1	Present content of the variable	The content of the variable is incremented by 1 every 2 seconds

8.2.5 Memory for non-volatile application program registers

Non-volatile registers are used to store data that must be maintained when the JC-365MC is de-energized.

Properties

– Global variables which are permanently assigned to addresses (%VL) (see also *Localization of Variables* in JetSym Online Help)

– Register variables always occupy 4 bytes
– Register variables are not initialized by the operating system
– Number of register variables: 60,000 (120,000 with option -R)
– Register numbers: 1,000,000 ... 1,059,999 (up to 1,119,999 with option -R)

Access

– Via JetSym
– Via email client
– Via browser (via HTTP server)
– Via HMIs
– From within the application program
– From another JC controller using the NET_COPY command

JetSym STX program

The following program example increments a register variable by 1 with every start of the application program.

```
Var
    ProgramStartCounter: Int At %VL 1000000; //non-volatile, because localized
as %VL in the 1.000.000ff array
End_Var;

Task Work Autorun
    ProgramStartCounter := ProgramStartCounter + 1;
    Loop
        // ...
    End_Loop;
End_Task;
```

Setup pane

The JetSym setup pane shows the content of the register variable.

	Name	Nummer	Inhalt	Typ
1	ProgramStartCounter	1000000	10 1	
2				
3				

Number	Element	Description
1	Present content of the register variable	The content of the register variable is incremented by 1 every time the program is launched.

8.2.6 Memory for non-volatile application program variables

Non-volatile variables are used to store data that must be maintained when the JC-365MC is de-energized.

Properties

- Global variables which are assigned to permanent registers (%RL)
- Variables are stored in a compact manner, according to the size of their type
- Size: 240,000 bytes (480,000 with -R option)
- Register numbers: 1,000,000 ... 1,059,999 (up to 1,119,999 with option -R)

Types of access

- By JetSym
- From HMIs
- From within the application program

JetSym STX program

The following program example increments 4 non-volatile variables every second. The working range of the counters is between 0 and 255 (variable type: byte). For these 4 variables the 4 bytes of register 1000010 are used.

```
Var
    Cnt1, Cnt2, Cnt3, Cnt4: Byte At %RL 1000010;
End_Var;

Task Count4 Autorun
    Loop
        Inc(Cnt1);
        Inc(Cnt2, 2);
        Inc(Cnt3, 5);
        Inc(Cnt4, 10);
        Delay(T#1s);
    End_Loop;
End_Task;
```

Setup pane

The JetSym setup pane shows the content of the variable. As all 4 counters are typed byte, the counter will overflow after a relatively short period of time.

	Name	Nummer	Inhalt		Typ
1	Cnt1	1000010	209	1	
2	Cnt2	1000010	162	2	
3	Cnt3	1000010	21	3	
4	Cnt4	1000010	42	4	
5					
6					

Number	Element	Description
1	Current content of the variable Cnt1	The content of the variable is incremented by 1 every second.
2	Current content of the variable Cnt2	The content of the variable is incremented by 2 every second.
3	Current content of the variable Cnt3	The content of the variable is incremented by 5 every second.
4	Current content of the variable Cnt4	The content of the variable is incremented by 10 every second.

8.2.7 Registers on I/O modules

These registers are located on modules connected to the JX2 or JX3 system bus.

Properties

- Global variables with dedicated addresses (%VL)
- Volatile
- The type is depending on the module.
- Register numbers on JX2 system bus: 200020000 … 2000299999
- Register numbers on JX3 system bus: 100020000 ... 100179999
- Register numbers of remote devices connected to the JX3-BN-ETH: 1nnn020000 ... 1nnn179999 (nnn = GNN)

Access

- Via JetSym
- Via email client
- Via browser (via HTTP server)
- Via HMIs
- From within the application program
- From another JC controller using the NET_COPY command

JetSym STX program

The following program example calculates the set speed of a servo axis on the JX2 system bus (AxSpeed). This calculation is based on an analog value measured by a module on the JX3 system bus (SpeedIn).

```
Var
    AxSpeed: Float At %VL 200012103;
    SpeedIn: Int At %VL 100030002;
End_Var;

Task SetSpeed Autorun
    Loop
        If SpeedIn > 100 Then
            AxSpeed := SpeedIn * 0.35;
        End_If;
        Delay(T#100ms);
    End_Loop;
End_Task;
```

Setup pane

The JetSym setup pane shows the content of the register variable.

	Name	Nummer	Inhalt	Typ
1	SpeedIn	100030002	296 1	
2	AxSpeed	200012103	103.6000 2	
3				
4				

Number	Element	Description
1	Present content of the SpeedIn register variable	Analog value on channel 1 of the JX3-AI4 module on the JX3 system bus
2	Present content of the AxSpeed register variable	Set speed of the JetMove servo amplifier on JX2 the system bus

8.2.8 Memory for non-volatile registers on the backplane module

These registers can be used to store data that need to be maintained in the event that the electronic component of the controller requires replacement.

For example this applies to the register storing the station number which is permanently assigned to a specific part of the facility.

Properties

- Global variables with dedicated addresses (%VL)
- Integer registers
- Slow access: Milliseconds
- Limited number of write/erase cycles: Approx. 10,000
- Number of registers: 128
- Register numbers: 108100 ... 108227

Access

– Via JetSym
– Via email client
– Via browser (via HTTP server)
– Via HMIs
– From within the application program
– From another JC controller using the NET_COPY command

JetSym STX program

The following program example displays the station number the user has stored on the JX3 backplane module on the HMI when the program is launched:

```
Var
    StationNo: Int At %VL 108100;
End_Var;

Task DisplayStation Autorun
    DisplayText(0, 1, '_Station : ');
    DisplayValue(0, 0, StationNo);
    Loop
        // ...
    End_Loop;
End_Task;
```

8.2.9 Special registers

Special registers let you control OS functions and retrieve status information.

Properties

– Global variables with dedicated addresses (%VL)
– When the operating system is launched, special registers are initialized using their default values
– Register numbers: 100000 ... 999999

Access

– Via JetSym
– Via email client
– Via browser (via HTTP server)
– Via HMIs
– From within the application program
– From another JC controller using the NET_COPY command

JetSym STX program

The following program example copies the content of the special register containing the position of the S11 mode selector to a special register. This register can be used to trigger LEDs U1 ... U4.

```
Var
    UserLeds: Int At %VL 108008;
    RSLswitch: Int At %VL 108015;
End_Var;

Task UserLedDemo Autorun
    Loop
        UserLeds := RSLswitch;
    End_Loop;
End_Task;
```

8.2.10 Inputs and outputs

Inputs and outputs are 1-bit variables. This means they can either have the value TRUE or FALSE. Inputs and outputs are addressed in immediate read/write cycles.

Properties of virtual inputs/outputs

– Global variables assigned to permanent addresses (%IX, %QX)
– Used for RemoteScan via Modbus/TCP
– Quantity: 16000
– I/O numbers: 20001 ... 36000

Properties of digital inputs/outputs

– Global variables assigned to permanent addresses (%IX, %QX)
– Located on modules connected to the JX2 or JX3 system bus
– I/O numbers on the JX2 system bus: 200000201 … 200002416
– I/O numbers on the JX3 system bus: 100000201 ... 100001716
– I/O numbers of remote devices connected to a JX3-BN-ETH: 1nnn010201 ... 1nnn011716 (nnn = GNN)

Types of access

– By JetSym
– By the email client
– Via a browser (via the HTTP server)
– From HMIs
– From within the application program

JetSym STX program

The following program example triggers a running light on a digital output module connected to the JX3 system bus. The duration during which an output is activated can be selected through a digital input on the JX2 system bus:

```
Var
    Lights: Array[8] Of Bool At %QX 100000309;
    HighSpeed: Bool At %IX 200000601;
End_Var;

Task RunningLight Autorun
    Var
        Idx: Int;
    End_Var;
    Loop
        For Idx := 0 To 7 Do
            Lights[Idx] := True;
            If HighSpeed Then
                Delay(T#100ms);
            Else
                Delay(T#300ms);
            End_If;
            Lights[Idx] := False;
        End_For;
    End_Loop;
End_Task;
```

Setup pane

The JetSym setup pane displays the state of inputs and outputs.

	Name	Nummer	Inhalt	Typ
1	⊟ Lights	100000309	array[8] 1	
2	Lights[0]	100000309	0	
3	Lights[1]	100000310	0	
4	Lights[2]	100000311	1	
5	Lights[3]	100000312	0	
6	Lights[4]	100000313	0	
7	Lights[5]	100000314	0	
8	Lights[6]	100000315	0	
9	Lights[7]	100000316	1 2	
10	HighSpeed			
11				

Number	Element	Description
1	Present state of outputs	The outputs are set and reset one after another.
2	Present content of the input	When the input is set, the corresponding output is activated for 100 ms.

8.2.11 Flag

Flags are one-bit operands. This means they can be either TRUE or FALSE.

Properties of user flags

– Global variables with dedicated addresses (%MX)
– Non-volatile
– Quantity: 256
– Flag numbers: 0 ... 255

Properties of overlaid user flags

– Global variables with dedicated addresses (%MX)
– Non-volatile
– Overlaid by registers 1000000 ... 1000055
– Quantity: 1792
– Flag numbers: 256 ... 2047

Properties of special flags

– Global variables with dedicated addresses (%MX)
– When the operating system is launched, special flags are initialized using their default values
– Quantity: 256
– Flag numbers: 2048 ... 2303

Types of access

– By JetSym
– By the email client
– Via a browser (via the HTTP server)
– From HMIs
– From within the application program

JetSym STX program

The following program example increments or decrements the "Counter" variable when pressing the F1 or F2 HMI key. These two keys are mapped to the corresponding flags. As long as the key is pressed, the corresponding LED in this key is enabled.

```
Var
    CountUp: Bool At %MX 2201;
    CountDown: Bool At %MX 2202;
    LedUp: Bool At %MX 2224;
    LedDown: Bool At %MX 2225;
    Counter: Int;
End_Var;

Task FlagCount Autorun
    Loop
        When CountUp Or CountDown Continue;
        If CountUp Then
            LedUp := True;
            Inc(Counter);
            When Not CountUp Continue;
            LedUp := False;
        Else
            LedDown := True;
            Dec(Counter);
            When Not CountDown Continue;
            LedDown := False;
        End_If;
    End_Loop;
End_Task;
```

Setup pane

The JetSym setup pane displays the state of the special flags and the content of the Counter variable.

	Name	Nummer	Inhalt	Typ
1	CountUp	2201	1	
2	CountDown	2202	0	
3	LedUp	2224	1	
4	LedDown	2225	0	
5	Counter		10	
6				

1
2
3

Number	Element	Description
1	Present state of key flags	Key F1 is pressed, key F2 is not pressed.
2	Present state of LED flags	LED in key F1 is enabled, LED in key F2 is disabled.
3	Present state of Counter variable	The counter is incremented when F1 is pressed, and decremented when F2 is pressed.

8.3 Saving and loading an application program

When uploading the application program from JetSym to the device, this program is stored as a file to the internal flash disk by default.

If you want the device to save the application program to an SD card or USB flash drive, you must configure the storage location.

When restarting the application program via JetSym or after booting the device, the application program is loaded and executed via the file system. The user determines the program that is to be executed.

ⓘ Info

Further information

For more information on this subject refer to the application-oriented manual *File System* available for download from our homepage.

8.4 Addressing expansion modules

Addressing expansion modules

Bucher Automation AG controllers and expansion modules offer a host of functions accessible to the user via registers. In physical terms, expansion modules connect to sensors and actuators that ensure connection to the automated equipment. On the controller, a unique number is assigned to each register and each digital input or output. The following information combines to make this unique number:

- Controller model
- Type of connection between controller and expansion module
  - Local direct connection
  - Remote connection via Bucher-Automation
  - Ethernet system bus: Position and number of expansion modules on the station

The module registers on the expansion modules are mapped to the controller registers. Each expansion module features 10,000 module registers. Controller registers are the only possible way of writing and reading process, configuration and diagnostic data to and from an expansion module.

Applying a register number

Register numbers are used for:

- Read/write access to a module register in the JetSym Setup pane.
- Declaration of a module register in a JetSym application program.
- Declaration of a module register as tag in JetViewSoft.

Applying an I/O number

I/O numbers are used in the following cases:

- Read from a digital input in the JetSym Setup pane.
- Read from/write to a digital output in the JetSym Setup pane.
- Declaration of a digital input/output as variable in the JetSym application program.
- Declaration of a digital input/output as tag in JetViewSoft.

8.4.1 Numbering registers and I/Os

Module registers - Definition

Module registers are the data interface of the expansion module. Module registers let you read process, configuration and diagnostics data from the expansion module, or write such data to it.

– The module register number within a module is unique.
– This unique register number lets you access a specific module register within the system.

Access options

There are several ways to access registers directly:

– From an application program
– From the JetSym setup pane
– From a visualization application

The register number within the system is unique.

ⓘ Info

Further Information

For further information, please refer to the JX3 *System* application-oriented manual in the download area of our website.

Expansion modules connected with a controller

Our products offer a host of features accessible to the user via registers. Each register is addressed by a unique number – the register number. Each digital input and output has a unique I/O number.

Register numbering system

Register numbers consist of a prefix, the module position in the system and the module register number. If the expansion modules are connected with the controller via JX3 system bus, the prefix is 100. In this case, the module register number is always a four-digit number.

100XXZZZZ

Fig. 22: Example: Register numbers

Digits	Description	Value range
100	Prefix	
XX	Position of the module in the system	02 … 17
ZZZZ	Module register number	0000 … 9999

I/O numbering system

I/O numbers provide direct access to the inputs or outputs of the module. These I/O numbers consist of a five-digit prefix, the module position in the system and the I/O number of the module. The I/O numbers always start with the constant prefix **10000**.

10000XXZZ

Fig. 23: Example: I/O numbers

Digits	Description	Value range
10000	Prefix	
XX	Position of the module in the system	02 … 17
ZZ	Module-specific I/O number	01 … 16

Expansion modules connected to an Ethernet bus node

The Ethernet bus node and the controller communicate via Ethernet system bus. When addressing expansion modules via Ethernet bus node, the Global Node Number (GNN) becomes part of the register number.

System overview

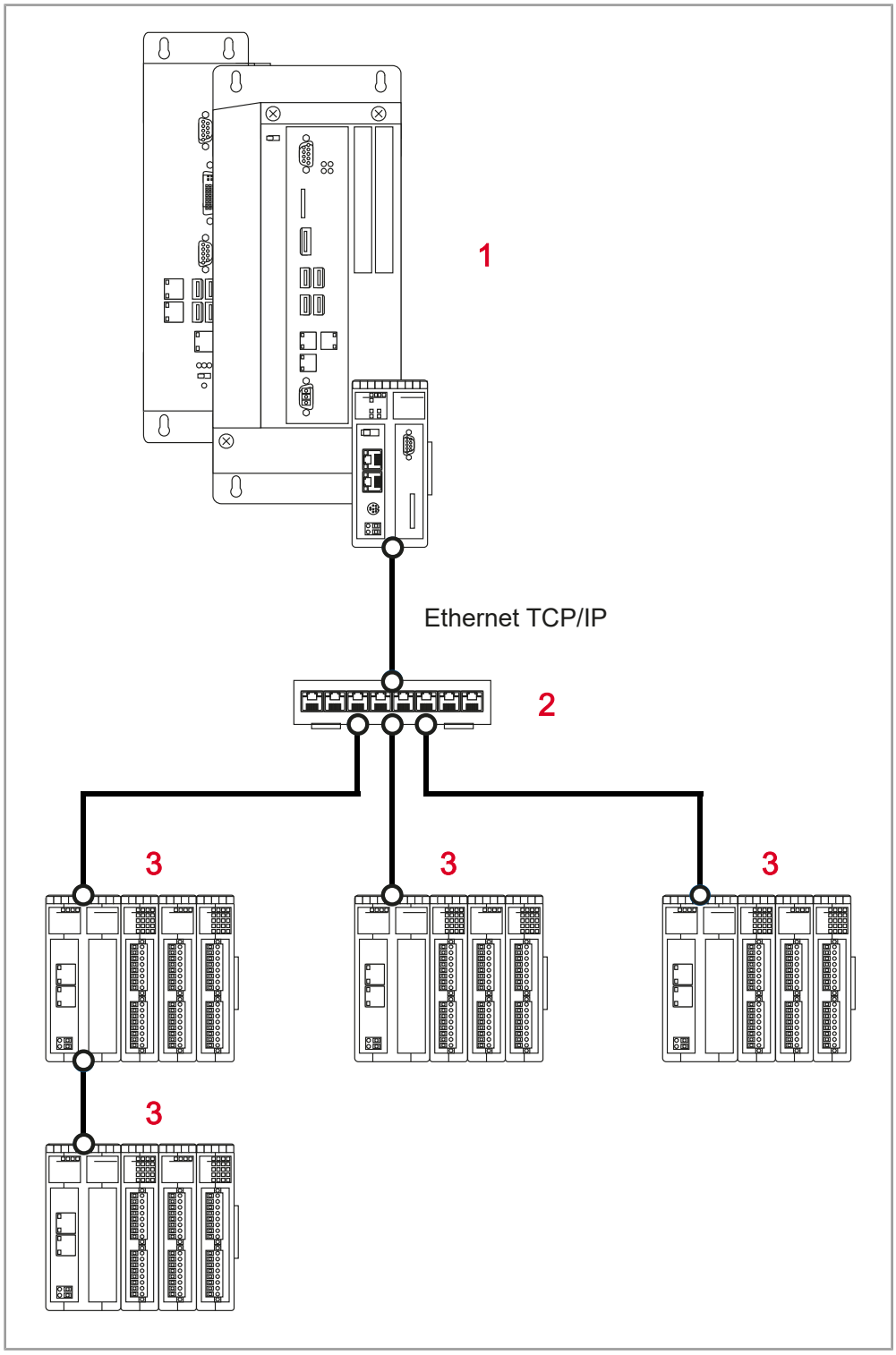


Fig. 24: System overview

1	Controller
2	Ethernet switch
3	Up to 64 Ethernet bus nodes per controller and up to 16 JX3-IO modules per Ethernet bus node

Definition of global node number

The Global Node Number (GNN) is used to identify Bucher Automation devices (e.g. controllers, bus nodes) within an Ethernet network:

– Within one network, each Bucher Automation device must have a unique GNN.
– The JetSym Hardware Manager automatically assigns the GNN during configuration.
– The value range of the GNN within a project is 000 ... 199.
– The GNN of the controller is always 000.

Register numbering system

Register numbers consist of a prefix, the GNN, the module position within the system and the module-specific register number.

1NNNXXZZZZ

Fig. 25: Example: Register numbers

Digits	Description	Value range
1	Prefix	
NNN	Bus node ID, GNN	001 … 199
XX	Position of the module within the station	02 … 17
ZZZZ	Module register number	0000 … 9999

I/O numbering system

I/O numbers provide direct access to the inputs or outputs of the module. The I/O numbers consist of prefix 1, the GNN, prefix 2, the module position in the system and the module-specific I/O number.

1NNN01XXZZ

Fig. 26: Example: I/O numbers

Digits	Description	Value range
1	Prefix 1	
NNN	Bus node ID, GNN	001 … 199
01	Prefix 2	
XX	Position of the module within the station	02 … 17
ZZ	Module-specific I/O number	01 … 16

ⓘ Info

Further information

For further information, please refer to the *Ethernet system bus* application-oriented manual in the download area of our website.

Register numbers of JX2 slave modules connected to the JX2 system bus

JX2 slave module numbers

To determine the slave module numbers of intelligent JX2 slave modules and JetMoves on the JX2 system bus of the JC-3xx, proceed as follows:

– Count the JX2 slave modules from left to right, starting from 2.
– Omit the power supply module JX2-PS1.
– Leave out non-intelligent JX2-I/O modules.

Register numbers of JX2 slave modules

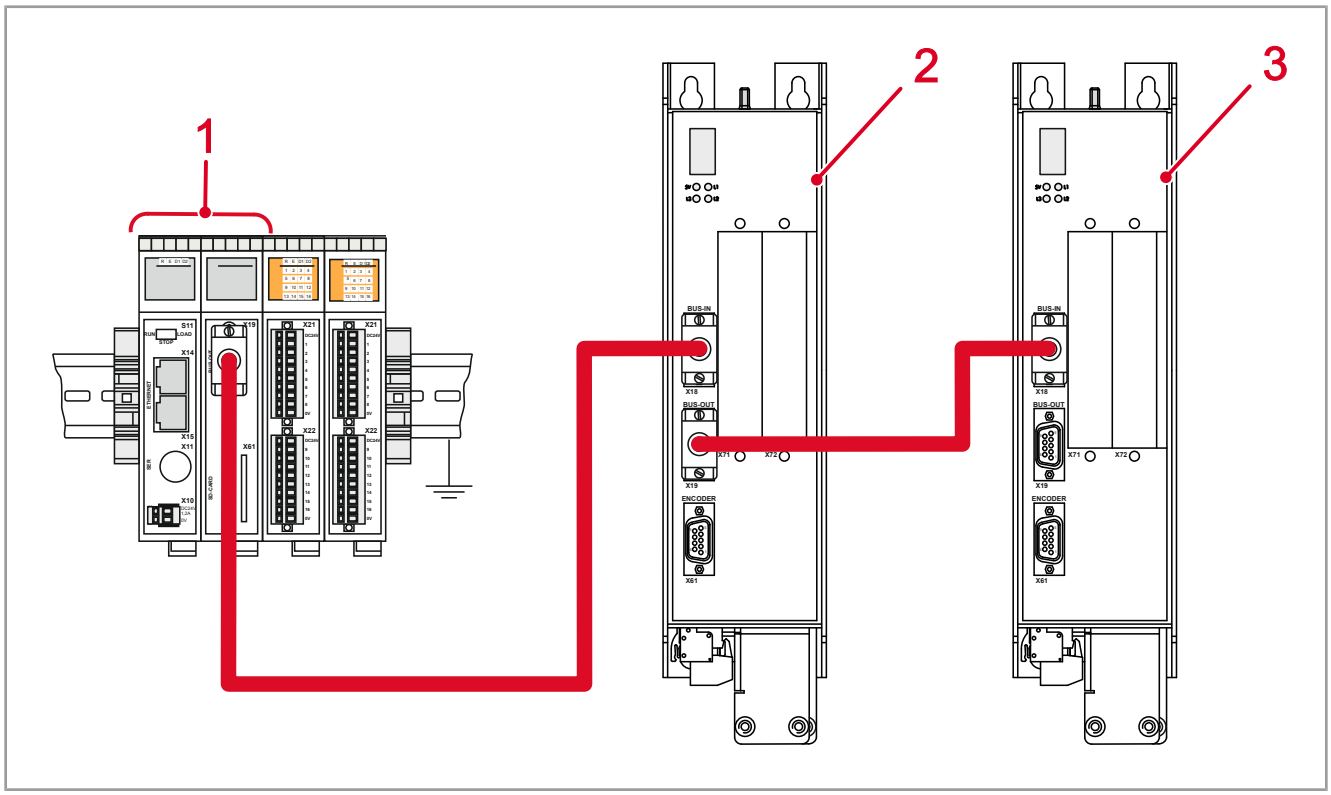
Register numbers for JX2 slave modules on the JX2 system bus of the JC-3xx break down as follows:

2	0	0	0	x	x	z	z	z	

Element	Description	Value range
xx	Slave module number + 10	12 ... 27
zzz	Module register number	000 ... 999

Example

Several JM-200 drives are connected to a JC-3xx controller.



Number	Module	Slave module number	Registers
1	JC-3xx	1	Refer to the JC- 3xx documentation
2	JM-206	2	200012zzz
3	JM-206	3	200013zzz

Registers and I/O numbers of JX2-I/O modules on the JX2 system bus

NOTICE

JX2-IO modules have been discontinued!

The following information is included to meet any servicing requirements on existing facilities.

I/O module numbers of JX2-I/O modules

To determine the I/O module numbers of JX2-I/O modules on the JX2 system bus of the JC-3xx, proceed as follows:

– Count the JX2-IO modules from left to right, starting from 2.
– Leave out the intelligent JX2 slave modules and JetMoves.
– Omit the power supply module JX2-PS1.

Register numbers for JX2-I/O modules

Register numbers for JX2-I/O modules connected to the JX2 system bus of a JC-3xx consist of the following elements:

2	0	0	0	0	3	x	x	z		

Element	Description	Value range
xx	I/O module number minus 2	00 ... 22
z	Module register number	0 ... 9

I/O numbers for JX2-I/O modules

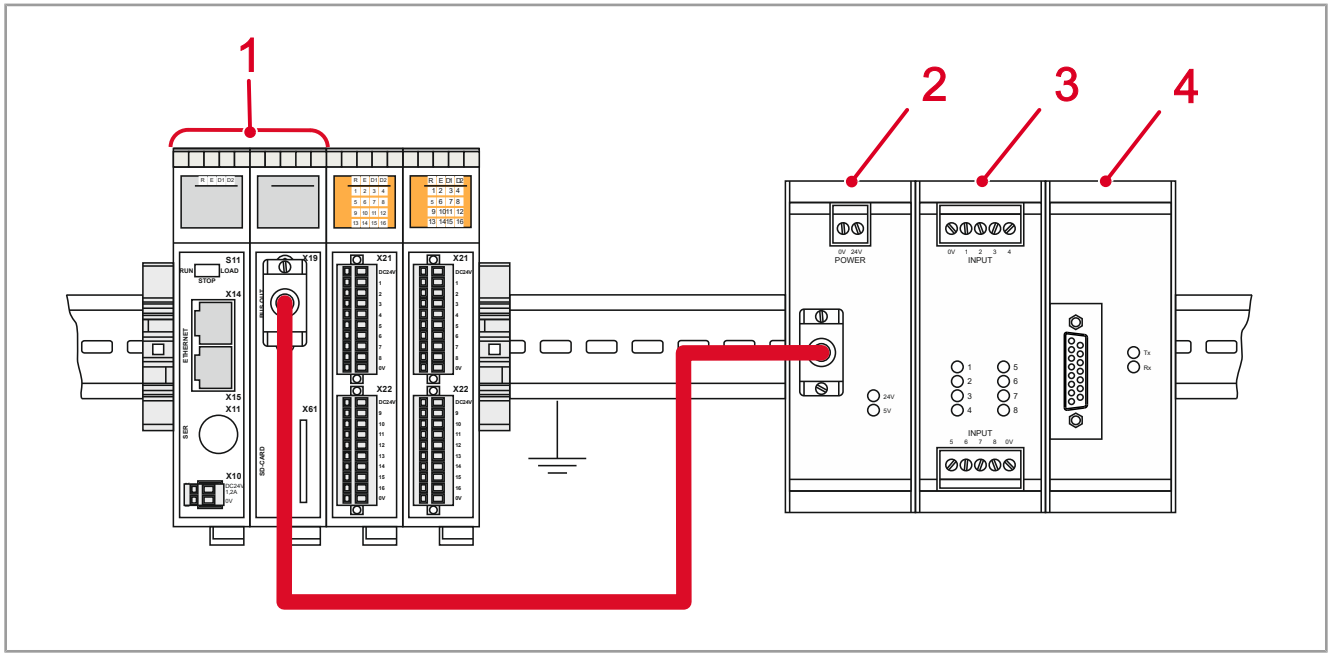
I/O numbers for JX2-I/O modules connected to the JX2 system bus of a JC-3xx consist of the following elements:

2	0	0	0	0	x	x	z	z		

Element	Description	Value range
xx	Module-specific I/O module number	02 ... 24
zz	Module-specific I/O number	1 ... 16

Example

Several JX2-I/O modules are connected to a JC-3xx controller.



Number	Module	I/O module number	Registers	I/O
1	JC-3xx	1	Refer to the JC-3xx documentation	
2	JX2-PS1	-	-	-
3	JX2-ID8	2	20000300z	2000002zz
4	JX2-CNT	3	20000301z	2000003zz

8.5 System commands

The controller supports system commands and system registers.

Info

Further information

For more information on this subject refer to the application-oriented manual *System Registers* available for download from our homepage.

8.6 Real-time clock

The controller is equipped with a real-time clock (RTC) that continues to work for a certain duration after the controller was de-energized (power reserve [▶ 22]).

Usage

Take advantage of the RTC when using the following feature:

– File date and time when creating a log file with time stamp

ⓘ Info

Restrictions

When using the real-time clock, the following restrictions apply:

– When the device is de-energized the power reserve is limited.

– The real-time clock has no automatic daylight savings time function.

Factory settings

At the end of the controller manufacturing process, the real-time clock is set to the actual date and time. As the power reserve corresponds to the typical delivery time, the as delivered condition is undefined.

ⓘ Info

Further information

For more information on this subject refer to the application-oriented manual *System Registers* available for download from our homepage.

8.7 Startup delay time

The controller supports a time delay of the boot process. You can enter the delay time in a register.

ⓘ Info

Further information

For more information on this subject refer to the application-oriented manual *System Registers* available for download from our homepage.

8.8 System runtime registers

The controller has several system runtime registers. The operating system of the controller increments the register value at fixed time intervals.

Application

The system runtime registers let you easily carry out time measurements in the application program.

ⓘ Info

Further information

For more information on this subject refer to the application-oriented manual *System Registers* available for download from our homepage.

8.9 Monitoring interface activities

The controller supports the function *Monitoring of interface activity*. This feature allows the application program to determine if the serial and Ethernet interfaces are communicating with the servers.

Monitored interface activities

The following interface activities can be monitored:

– pcomX server via serial interface
– JetIP server via Ethernet interface
– STX debug server via Ethernet interface

Purpose

The monitoring function for interface activities can be used, amongst others, for the following scenarios:

– Plants requiring process visualization to ensure safe operation. They can be transferred into a save state if communication fails.
– When the service technician connects an HMI, the application program automatically displays additional status information.

ⓘ Info

Further information

For more information on this subject refer to the application-oriented manual *System Registers* available for download from our homepage.

8.10 Ethernet system bus

The controller uses the Ethernet system bus as interface. The bus is based on TCP, UDP/IP and can therefore be used in parallel with other TCP, UDP/IP protocols.

Application

The Ethernet system bus has been designed for data exchange between the following devices via standard Ethernet:

– Programming unit
– Controllers
– Bus node
– Communication modules

Functions

The following functions of the bus are implemented in the controller:

– Acyclic (explicit) data exchange
– Cyclic (implicit) data exchange
– Error handling
– NetConsistency
– JetIPScan
– Connection management

ⓘ Info

Further information

For further information, please refer to the *Ethernet system bus* application-oriented manual in the download area of our website.

8.11 Control of alphanumeric displays and HMIs

Connecting

HMIs connect to the serial interface (port X11) on the controller.

Control of HMIs

The controller provides the following display options:

– Displaying text strings
– Displaying the contents of variables
– Scanning the HMI keys
– Switching the HMI LEDs
– Monitor function

Display functions

The display features are part of the STX language set.

ⓘ Info

Further information

For more information on this topic refer to the application-oriented manual *Controlling alphanumeric HMI devices (LCD) and printers* available in the download area of our homepage.

8.12 Controlling printer and serial interfaces

Connecting

The devices connect to the serial interface (port X11) on the controller.

Controlling

The controller provides the following display options:

– Displaying text strings
– Displaying the contents of variables

Display functions

The display features are part of the STX language set.

ⓘ Info

Further information

For more information on this topic refer to the application-oriented manual *Controlling alphanumeric HMI devices (LCD) and printers* available in the download area of our homepage.

8.13 Project Engineering of a JX2 Station

JX2 station

A JX2 station consists of a JetControl controller and expansion modules connected to it. Communication among devices takes place via the JX2 system bus.

Interface

The JX2 system bus is located on interface CAN 1 at terminal X19.

Configuration

The JX2 system bus needs not be configured. The controller or the bus node automatically detects connected modules and commissions them.

If third-party CAN or CANopen modules are to be accessed, configuration can be performed via R 200002077 [▶ 111].

Modules to connect

– Drives JetMove 1xx, JetMove 2xx, and JetMove 6xx

8.14 Project Engineering of a JX3 Station

JX3 station

A JX3 station consists of a JetControl controller or a JX3-BN-xxx bus node and JX3-IO modules connected to it.

Configuration

Access to- IO modules is completely transparent, as configuration of the JX3 system bus is not required. The controller or the bus node automatically detects connected modules and commissions them.

Modules to connect

You can connect all JX3-IO modules except JX3-COM-xxxx modules to a JX3 station.

ⓘ Info

Further Information

For further information, please refer to the JX3 *System* application-oriented manual in the download area of our website.

8.15 JCF-SV1 "simple servo function"

The JCF-SV1 function allows the user to implement a simple, yet convenient positioning of servo drives or frequency inverter drives using analog set values.

It is possible to operate up to 16 axes independently of each other.

Type	Servo axis count	Minimum HW revision	From OS version
JC-365MC	16	1.05	1.30

Configuration

Access to process variables, such as target position, actual position, reference switch and limit switch, is provided by the JX3-IO modules. They connect either directly via the local JX3 system bus of the controller, or via the JX3-BN-ETH bus node in a distributed setup.

Engineering environment

To create the application program, use JetSym version 5.x and higher.

JCF-SV1 feature

The register interface provides access to the JCF-SV1 feature.

Commands, such as POS, and AXARR, or a Motion API are not available.

ⓘ Info

Further information

For more information on this subject refer to the application-oriented manual *JCF-SV1 OS Function* available for download from our homepage.

8.16 E-mail feature

The user creates template files for emails, where the controller enters any variables when they are being sent. This allows for automated transfer of reports or other production or maintenance information via email.

The controller sends emails to an email server that will then forward the message to the addressee.

Activating the email feature

To activate the email feature in the controller, the following requirements must be met:

– A valid email configuration file **/EMAIL/email.ini** exists while the controller is booting.

If this requirement has been met, the corresponding bit in the web status register is set and the email feature is available.

R 202930

Web status

The registers shows available features in a bit-coded way.

Property	Description	
Type of access	Read	
Meaning of the bits		
Bit 0	FTP server	
	1 =	available
Bit 1	HTTP server	
	1 =	available
Bit 2	Email	
	1 =	available
	Bit 2 is set only if the **/EMAIL/email.ini** configuration file is available.	
Bit 3	Data file function	
	1 =	available
Bit 4	Modbus/TCP	
	1 =	available
Bit 5	Modbus/TCP	
	1 =	available
Bit 7	FTP client	
	1 =	available

Tab. 51: Web status

Info

Further information

For more information on this subject refer to the application-oriented manual *File System* available for download from our homepage.

8.17 Modbus/TCP

Enabling the Modbus/TCP feature

On the JC-365MC controller, the Modbus/TCP feature is permanently enabled.

This permanently sets bits 4 and 5 in the Web Status register 202930.

R 202930

Web status

The registers shows available features in a bit-coded way.

Property	Description	
Type of access	Read	
Meaning of the bits		
Bit 0	FTP server	
	1 =	available
Bit 1	HTTP server	
	1 =	available
Bit 2	Email	
	1 =	available
	Bit 2 is set only if the **/EMAIL/email.ini** configuration file is available.	
Bit 3	Data file function	
	1 =	available
Bit 4	Modbus/TCP	
	1 =	available
Bit 5	Modbus/TCP	
	1 =	available
Bit 7	FTP client	
	1 =	available

Tab. 52: Web status

Modbus/TCP server

With the Modbus/TCP feature enabled and the Modbus/TCP server launched, an external client can access registers, flags, inputs and outputs.

Modbus/TCP client

The Modbus/TCP client included in the device supports only Class 0 Conformance. This class uses commands for reading and writing multiple registers. One frame can transmit up to 125 registers with a width of 16 bits.

The protocol ID is "0". Assignment of sent and received frames is carried out using the transaction ID.

8.17.1 Modbus/TCP server

If a valid license exists, the Modbus/TCP function is activated and the Modbus/TCP server has been started, registers, flags, inputs and outputs can be accessed by an external client.

Number of possible connections

4 connections can be open at the same time.

Info

Restriction

Modbus/TCP only supports transmission of registers with a width of 16 bits. From this follows, that only the 16 least significant bits are transmitted when 32-bit registers are sent. When assigning incoming register values to the internal 32-bit registers no sign extension will be carried out.

Addressing

The addresses which have been received via Modbus/TCP can be modified locally in the Modbus/TCP server. There are 3 registers for this purpose. The basic addresses for accessing registers, inputs and outputs are entered into these registers. Then, the address contained in the Modbus/TCP frame specifies the address with reference to the basic address.

R 272702

Register offset

The basic address for accessing registers via Modbus/TCP is entered into R 272702.

Property	Description
Value after reset	1000000

Tab. 53: Register offset

R 272704

Input offset

The basic address for accessing inputs via Modbus/TCP is entered into register 272704.

Property	Description
Value after reset	100000000

Tab. 54: Input offset

R 272705

Output offset

The basic address for accessing outputs via Modbus/TCP is entered into R 272705.

Property	Description
Value after reset	100000000

Tab. 55: Output offset

Example 1

The Modbus/TCP server on the controller receives from a Modbus/TCP client the command **read multiple registers** starting from register number 100. The number of registers to be read is 5. Register 272702 *Register Offset* contains the value 1000000. Hence, registers 1000100 through 1000104 will be read.

Example 2

The Modbus/TCP server of the controller receives from a Modbus/TCP client the Modbus/TCP command **read input discretes** with the input number 210 and the instruction to read this input. Register 272704 *Input offset* contains the value 100000000. Hence, input 100000210 of a peripheral module, such as JX3-DI16, will be read.

Example 3

The Modbus/TCP server on the JC-365MC receives from a Modbus/TCP client the command **write coils** specifying output number 205 and the instruction to enable this output. Register 272705 *Output offset* contains the value 100000000.

Hence, output 100000205, for example of a peripheral module JX3-DO16, will be activated.

Supported commands

Class 0

Command	Description
fc 3	**read multiple registers**
	Reading register sets The starting register number within the controller is calculated as follows: Add the register number specified in the command to the content of R 272702 *Register Offset*.
fc 16	**write multiple registers**
	Writing register sets The starting register number within the controller is calculated as follows: Add the register number specified in the command to the content of R 272702 *Register Offset*.

Tab. 56: Supported commands – Class 0

Class 1

Command	Description
fc 1	**read coils**
	Reading outputs The output register number within the controller is calculated as follows: Add the output number specified in the command to the content of R 272705 *Output offset*.
fc 2	**read input discretes**
	Reading inputs The input register number within the controller is calculated as follows: Add the input number specified in the command to the content of R 272704 *Input offset*.

Command	Description
fc 4	**read input registers**
	Reading inputs blockwise in 16-bit words The starting register number within the controller is calculated as follows: Add the register number specified in the command to the content of R 272702 *Register Offset*.
fc 5	**write coil**
	Enabling/disabling an individual output The output register number within the controller is calculated as follows: Add the output number specified in the command to the content of R 272705 *Output offset*.
fc 6	**write single register**
	Entering values into the 16 least significant bits of a register The starting register number within the controller is calculated as follows: Add the register number specified in the command to the content of R 272702 *Register Offset*.

Tab. 57: Supported commands – Class 1

Class 2

Command	Description
fc 15	**force multiple coils**
	Enabling/disabling several outputs The output register number within the controller is calculated as follows: Add the output number specified in the command to the content of R 272705 *Output offset*.
fc 23	**read/write registers**
	Reading/writing registers simultaneously The starting register number within the controller is calculated as follows: Add the register number specified in the command to the content of R 272702 *Register Offset*.

Tab. 58: Supported commands – Class 2

8.17.2 Modbus/TCP client with STX functions

The Modbus/TCP client integrated in the controller supports Class 0 Conformance along with the Class 2 fc 23 command.
This class uses commands for reading and writing multiple registers. One frame transmits up to 125 registers of 16 bits width.
The protocol ID is "0". Assignment of sent and received frames is carried out using the transaction ID.

This chapter describes how to carry out noncyclical or cyclical transmission to a Modbus/TCP server using STX functions.

ⓘ Info

Obsolete system functions

As of JetSym 5.0, the system functions are outdated. Instead, use the corresponding JetSym STX functions.

Number of possible connections

Connections to 11 different Modbus/TCP servers can be open at the same time.

Noncyclical data transmission

The functions `ModbusReadReg()` and `ModbusWriteReg()` let you establish a noncyclical transmission channel to a Modbus/TCP server.
These functions copy data between registers of a Modbus/TCP server and STX variables. They establish a connection to the specified Modbus/TCP server, transmit the desired data and clear down the connection again.
If RemoteScan has already established a connection for cyclical data transmission, this connection will be used. Setting-up and clearing-down the connection is, therefore, not required.

Cyclical data transmission

Cyclical data transmission is made through the configurable function `RemoteScanConfig()`. The data are cyclically transmitted from and to the Modbus/TCP servers by means of STX variables.
To each Modbus/TCP server (IP address and port) a connection is established. The connection is established independent of how many communication units are configured on this server.
If several communication units are configured on one Modbus/TCP server, the accesses are serialized. This is because the servers often do not support **command pipelining**. If several servers have been configured, communication is carried out in parallel.

Unit ID

Converters from Modbus/TCP to Modbus RTU use the *Unit ID* for addressing the Modbus RTU servers. For this reason, the *Unit ID* can be set.

JetSym STX functions

This is a comparison between the system functions and the corresponding JetSym STX functions.

System function	Corresponding JetSym STX function
60	Function ModbusCRCgen(FramePtr: Int, Length: int): Int;
61	Function ModbusCRCcheck(FramePtr: Int, Length: int): Int;
65/67	Function ModbusReadReg(Const Ref MbParam: MODBUS_PARAM): Int;
66/68	Function ModbusWriteReg(Const Ref MbParam: MODBUS_PARAM): Int;
80/85	Function RemoteScanConfig(Protocol: RSCAN_PROTOCOL, Elements: Int, Const Ref Configuration: RSCAN_DSCR): Int;
81	Function RemoteScanStart(Protocol: int): Int;
82	Function RemoteScanStop(Protocol: int): Int;

Tab. 59: System functions vs. JetSym STX functions

ⓘ Info

Further information

For more information on this subject, refer to the JetSym Online Help.

8.18 User-programmable serial interface

The controller has a serial interface (mini DIN socket X11) which can be addressed from the application program to send and receive characters.

Applications

The user-programmable serial interface lets you connect devices which use communication protocols that are not supported by the OS of the controller. Fields of application, for example, are:

– Scales
– Scanners
– Display elements
– Frequency inverters
– Temperature controllers
– etc.

ⓘ Info

Further information

For more information on this subject refer to the application-oriented manual *User-Programmable Interfaces* available for download from our homepage.

8.19 User-programmable IP interface

The user-programmable IP interface lets you send or receive any data via Ethernet interface (jacks X14 and X15) using TCP/IP or UDP/IP. The data is processed in the application program entirely.

Applications

The user-programmable IP interface allows for data exchange via Ethernet connections which do not use standard protocols, such as FTP, HTTP, JetIP or Modbus/TCP. The following applications are possible:

– Server
– Client
– TCP/IP
– UDP/IP

ⓘ Info

Further information

For more information on this subject refer to the application-oriented manual *User-Programmable Interfaces* available for download from our homepage.

8.20 User-programmable CAN interface

CAN interface

The user-programmable CAN interface lets you send and receive CAN messages. The CAN messages are completely processed in the application program.

Advantage of CAN

This feature is not only apt for CANopen devices. It lets the user communicate with third-party devices which are based on a CAN protocol.

Applications

The user-programmable CAN interface can be used for the following applications:

– Devices which are equipped with a CAN interface can be controlled via proprietary protocols
– Controlling CANopen-capable devices

If your device supports the CANopen protocol, you should prefer to use the CANopen STX API for [▶ 111] communication.

Hardware requirements

The hardware required is a JC controller with CAN interface and/or JX2 system bus.

ⓘ Info

Further information

For more information on this subject refer to the application-oriented manual *User-Programmable Interfaces* available for download from our homepage.

8.21 CANopen STX API

The CANopen STX API provides a software platform that lets the user send and receive CANopen messages via STX functions.

The CANopen standard

CANopen is an open standard for networking and communication in the automation sector. The CANopen protocol is being further developed by CiA e.V. (CAN in Automation) and is based on the physical layer with CAN high-speed according to ISO 11898.

Specifications

The CANopen specifications can be obtained from the **CiA e.V.** homepage at http://www.can-cia.org.
The key specification documents are:

– CiA DS 301 - This document is also known as the communication profile and describes the fundamental services and protocols used under CANopen.
– CiA DS 302 - Framework for programmable devices (CANopen Manager, SDO Manager)
– CiA DR 303 - Information on cables and connectors
– CiA DS 4xx - These documents describe the behavior of a number of device classes in, what are known as, device profiles.

R 200002077

Enabling JX2 **system bus special functions**

The value of this register influences the initializing behavior of the JX2 system bus (CAN 1).

Property	Description	
Value after reset	Remanent; factory setting: 0	
Takes effect	Next time when the controller is launched	
Meaning of the individual bits		
Bit 2, bit 3	Enabling the user-programmable CAN interface in addition to the JX2 system bus	
	01 =	The user-programmable CAN interface and the JX2 system bus are initialized the next time the JX2 system bus is started. This requires a restart of the controller. This function allows for the connection of JX2 expansion modules.
Bit 2, bit 3	Enabling only the user-programmable CAN interface or CANopen STX API	
	1x =	At the next restart, the JX2 system bus is not initialized. The user-programmable CAN interface can be used. All node IDs can be used **without** any restrictions. The controller does not initialize any JX2 expansion modules of the JX2 system bus. For this reason, JX2 expansion modules **cannot** be connected. The CANopen STX API can be used.
Bit 6	1 =	The CANopen functionality in the JX2 system bus driver is disabled. This requires a restart of the controller. For any connection to the JX2 bus system of JX2-IO modules and JX2 slave modules along with CANopen devices, the communication of which is programmed via CANopen STX API (mixed operation), bit 6 must be set. The JX2 system bus driver supports certain CANopen devices. If you intend to connect unsupported CANopen devices the communication of which you intend to program via CANopen STX API, this bit needs to be set in order to completely disable the support by the JX2 system bus driver. If you intend to program communication with CANopen devices via CANopen STX API yourself, only WAGO I/O System 750, JX-SIO, LioN-S modules, etc. can be connected to the JX2 system bus. Bit 6 makes sense only if bit 3 has not been set.

Only CANopen devices are connected to the JX2 system bus (CAN 1)

The JX2 system bus (CAN 1) can be disabled completely, if only CANopen devices are connected and CANopen STX API is used.

The steps to be taken for this are described below:

1. Set bit 3 in R 200002077.
   ```
   Regs[200002077] := 0x08;
   ```
2. To have the changes accepted, relaunch the controller once.

⇨ Now, all node IDs can be used by CANopen **without** any restrictions.

Restrictions at mixed operation

Mixed operation means that the following devices are connected to the JX2 system bus:

– JX2 I/O module

– JX2 slave modules

– CANopen devices which are not supported by the JX2 system bus driver and which must be programmed via CANopen STX API

In this case, the following restrictions apply:

– Only the ranges 50 ... 59 and 70 ... 79 are permitted to be node IDs for CANopen devices!

– If the node ID range 50 ... 59 must be used, a maximum of **9** JX2 IO modules can be connected to the JX2 system bus. Otherwise, the CAN IDs used will overlap with the PDO IDs.
Important note: There is no scanning by the controller.

– To be able to connect more than 9 JX2 IO modules, the COB IDs of TX and RX PDOs used by CANopen devices must be mapped to the ID range 70 ... 79.
Important note: Whether it is possible to make changes to the COB ID of PDOs depends on the corresponding CANopen device.

– The non-volatile register 200002077 lets you disable the CANopen support by JX2 system bus driver. For more information refer to the sample codes below. If you intend to program communication with CANopen devices via CANopen STX API yourself, only WAGO I/O System 750, JX-SIO, LioN-S modules, etc. can be connected to the JX2 system bus.

```
// JX2 system bus not supporting the CANopen function
// of the JX2 system bus driver and CAN-PRIM disabled
Regs[200002077] := 0x40;
// JX2 system bus not supporting the CANopen function
// of the JX2 system bus driver and CAN-PRIM enabled
Regs[200002077] := 0x44;
```

– To apply the changes, restart the controller once.

ⓘ Info

Further information

For more information on this subject refer to the application-oriented manual *CANopen STX API* available for download from our homepage.

8.22 Motion control

Motion control is programmed in the Motion API interface in the programming language STX.

Seamless integration of axis functions into the controller is an essential feature of Bucher Automation AG control systems. This allows for easy implementation of point-to-point (PtP) positioning, technology functions (MC), path control and certain robot functions. This universal motion control system is called **motion control eXtended (MCX).**

MCX features

– Centralized generation of all command positions for single axes and axis groups
– Seamless integration into the controller
– Jerk control to minimize machine vibration
– Online generation of cam discs
– Combination of technology and path groups
– Cascading of axis groups
– Programming using SI units
– Consideration of mechanical parameters

Amount of axes

JX2 system bus (CAN 1, X19):

– Up to 16 servo amplifiers JM-100/JM-200 in PtP mode
– 6 axes max. in MCX mode

Bucher Automation Ethernet system bus (X14/X15):

– Up to 64 servo amplifiers JM-200-ETH in PtP mode
– 12 axes max. in MCX mode

Prerequisite:

The total number of axes must not exceed the maximum possible number of axes of the controller.

Mixed operation of MCX axes via Ethernet and JX2 system bus is not supported.

Technology group

– Virtual leading axis
– External encoder as leading axis
– Cascaded groups
– Programming the whole group via leading axis
– Dynamic coupling and decoupling of following axes
– Print mark for high-precision position correction
– Up to 6 active technology groups operating up to 24 cam discs with 100 segments each

Electronic gearing

– Gear ratio can be changed dynamically
– Superimposed motions in the electronic gearbox

Electronic cam disc

– Segments: Polynomials 1, 3, 5, 7 (jerk-free transition); sine
– Definition via coefficients or marginal conditions
– Flexible enabling of segments depending on the positions of the leading and following axes or immediate
– Scaling and shifting the areas of definition during operation

Path group (Tool Center Point programming)

– Cartesian (3 spatial axes, 2 axes for orientation)
– SCARA (4 axes)
– 2D delta
– Buffered operation (total of buffers: 96)
– Look-ahead function
– Smoothened coupling (optimum velocity profile)
– Block-synchronous events
– Coupling an axis group to a leading axis
– Linear
– 2D circle
– 3D circle
– Helix
– Spline interpolation
– Gantry axis pair
– Shifting in axial direction via offset
– Up to 4 active path groups

Required OS version of the servo amplifiers

If servo amplifiers are operated as MC axes:

As of operating system 1.28 of the JC-365MC, the OS version installed in the JM-100 and JM-200 servo amplifiers must be at least 2.16.

ⓘ Info

Further information

For more information on Motion Control programming, refer to the JetSym Online Help.

8.23 FTP server

The controller supports the *FTP server* feature. The FTP server handles directories and files using an FTP client.

The files can be stored to the following storage media:

– Internal flash drive
– External SD card (slot X61)

Enabling the FTP server feature

On this device, the *FTP server* feature is permanently enabled. This is due to the device configuration upon purchase.

Bit 0 in the Web Status register 202930 is permanently set.

R 202930

Web status

The registers shows available features in a bit-coded way.

Property	Description	
Type of access	Read	
Meaning of the bits		
Bit 0	FTP server	
	1 =	available
Bit 1	HTTP server	
	1 =	available
Bit 2	Email	
	1 =	available
	Bit 2 is set only if the **/EMAIL/email.ini** configuration file is available.	
Bit 3	Data file function	
	1 =	available
Bit 4	Modbus/TCP	
	1 =	available
Bit 5	Modbus/TCP	
	1 =	available
Bit 7	FTP client	
	1 =	available

Tab. 60: Web status

Info

Further information

For more information on this subject refer to the application-oriented manual *File System* available for download from our homepage.

8.24 HTTP server

The controller supports the *HTTP server* feature. A standard browser is sufficient for accessing the HTTP server.

The browser is for reading and displaying files which have been downloaded to the controller via FTP.

Using SSI (Server Side Includes) allows for register content and state information on inputs, outputs, and flags to be integrated into HTML pages at runtime.

Enabling the HTTP server feature

On this device, the *HTTP server* feature is permanently enabled. This is due to the device configuration upon purchase.

Bit 1 in the Web Status register 202930 is permanently set.

R 202930

Web status

The registers shows available features in a bit-coded way.

Property	Description	
Type of access	Read	
Meaning of the bits		
Bit 0	FTP server	
	1 =	available
Bit 1	HTTP server	
	1 =	available
Bit 2	Email	
	1 =	available
	Bit 2 is set only if the **/EMAIL/email.ini** configuration file is available.	
Bit 3	Data file function	
	1 =	available
Bit 4	Modbus/TCP	
	1 =	available
Bit 5	Modbus/TCP	
	1 =	available
Bit 7	FTP client	
	1 =	available

Tab. 61: Web status

Info

Further information

For more information on this subject refer to the application-oriented manual *File System* available for download from our homepage.

8.25 FTP client

The FTP client allows a remote network device to access files and directories from within the application program by communicating with the FTP server of this network device.

Function
Enabling the FTP client

The *FTP client* function is always enabled on this device. It is enabled when you order this feature.

Bit 7 in the Web Status register 202930 is permanently set.

R 202930

Web status

The registers shows available features in a bit-coded way.

Property	Description	
Type of access	Read	
Meaning of the bits		
Bit 0	FTP server	
	1 =	available
Bit 1	HTTP server	
	1 =	available
Bit 2	Email	
	1 =	available
	Bit 2 is set only if the **/EMAIL/email.ini** configuration file is available.	
Bit 3	Data file function	
	1 =	available
Bit 4	Modbus/TCP	
	1 =	available
Bit 5	Modbus/TCP	
	1 =	available
Bit 7	FTP client	
	1 =	available

Tab. 62: Web status

ⓘ Info

Further information

For more information on this subject refer to the application-oriented manual *File System* available for download from our homepage.

8.26 Automatic copying of controller data (AutoCopy)

The controller supports the *AutoCopy* function which automatically copies controller data. The AutoCopy function lets you copy data within the controller and/or between the controller and an FTP server, the connected expansion modules and a controller within the network.

Use cases

AutoCopy can be used to copy a new application program or operating system from the SD card to the controller.

Prerequisites

AutoCopy is executed only when the JC-365MC is booting. Before switching on, the following conditions must be met:

– The mode selector S11 is in the *LOAD* position.
– There is an SD card in the SD card slot X61.
– The SD card contains a valid command file with the corresponding copy commands.

ⓘ Info

Further information

For more information on this subject refer to the application-oriented manual *File System* available for download from our homepage.

9 Registers - Overview

This register overview gives a condensed summary of the registers and flags of the JC-365MC device.

Default address on the CANopen bus

The pre-set default address on the CANopen bus is:

Node ID	127 (0x7F)

Maximum number of CANopen interfaces

– Maximum number of CAN interfaces (CANopen-STX-API): 2
– CANMAX: 1

General overview - Registers

Register range	Description
100000 ... 100999	Electronic Data Sheet (EDS)
101000 … 101999	Configuration
102000 … 102999	Real-time clock
103000 … 103999	Serial port
104000 ... 104999	Ethernet
107000 … 107499	SD memory card
107500 ... 107599	Flash disk
108000 ... 108999	CPU/backplane
200000 ... 209999	General system registers
210000 ... 219999	Application program
220000 ... 229999	HMI control
230000 ... 239999	JetIP networking
240000 ... 249999	JetSync
250000 ... 259999	Ethernet system bus
260000 ... 269999	RemoteScan
270000 ... 279999	Modbus/TCP
290000 ... 299999	E-mail
310000 ... 319999	File system/data files
320000 ... 324999	FTP client
350000 ... 359999	User-programmable IP interface
380000 ... 389999	Error history
390000 ... 399999	I/O networking
470000 ... 479999	NetConsistency
510000 ... 519999	DNS server/DNS cache
520000 ... 529999	JetIPScan
6yy000 … 6yy999	SW module JCF-SV1 Number of axes: yy = 00 … 16
1000000 ... 1059999	JC-365MC: Application registers (non-volatile; integer/float)
1000000 ... 1119999	JC-365MC: Application registers (non-volatile; integer/float) with option -R
100xx0000 ... 100xx9999	JX3 IO modules (xx: 02 ... 17)
200002000 ... 200029999	JX2 system bus
Networking via Bucher Automation Ethernet system bus GNN: nnn = 000 ... 199	

Register range	Description
1nnn020000 ... 1nnn179999	JX3 module register
1nnn202000 ... 1nnn227999	JX2 module register
1nnn810000 ... 1nnn819999	JetMove registers
1nnn980000 ... 1nnn980199	Indirect access via local R 236xxx
1nnn990000 ... 1nnn999999	Indirect access with variable destination window

I/Os - General overview

I/O numbers	Description
20001 ... 36000	Virtual I/Os for RemoteScan
10000xx01 ... 10000xx16	JX3-IO modules (xx = 02 ... 17)
20000xx01 ... 20000xx16	JX2 IO modules (xx = 02 ... 24)
1nnn010101 ... 1nnn011716	JX3 IO modules on the JX3-BN-ETH (GNN: nnn = 000 ... 199)

Flags - General overview

Flag	Description
0 ... 255	Application flags (non-volatile)
256 ... 2047	Overlaid by registers R 1000000 through 1000055
2048 ... 2303	Special flags

Electronic Data Sheet (EDS)

Register range	Description
100500	Interface (0 = CPU, 1 = JX3 modules)
100501	Module number (2 ... 17) If <100500> = 0: The EDS of the controller is displayed. If <100500> = 1 and <100501> = 2 ... 17: The EDS of the selected JX3 module is displayed.
100600 ... 100614	**[Identification]**
100600	Internal version number
100601	Module ID
100602 ... 100612	Module name (register string)
100613	PCB revision
100614	PCB options
100700 ... 100712	**[Production]**
100700	Internal version number
100701 ... 100707	Serial number (register string)
100708	Day
100709	Month
100710	Year
100711	TestNum.
100712	TestRev.
100800 ... 100809	**[Features]** I/O module
100800	Internal version number
100801	Diagnostic configuration
100802	Digital inputs
100803	Digital inputs, inverted
100804	Digital outputs
100805	Digital outputs, inverted
100806	Cyclic inputs
100807	Cyclic outputs
100808	Features

Register range	Description
100809	Diagnostics mask
100800 ... 100817	**[Features]** JX3-BN-ETH/JC-365MC
100800	Internal version number
100801	MAC address (Bucher Automation)
100802	MAC Address (device)
100803	Serial port
100804	Switches
100805	STX
100806	Non-volatile registers
100807	JX3 bus
100808	CAN-bus
100809	SD memory card
100810	Motion Control
100811	Intelligent slave modules
100812	HTTP/e-mail
100813	Modbus/TCP
100815	LED for the SD card
100816	User-defined LEDs
100817	RTC

Configuration

Registers	Description
From file /system/ config.ini	
101100	IP address
101101	Subnet mask
101102	Default gateway
101103	DNS server
101132	Host name suffix type
101133 ... 101151	Host name (register string)
101164	Port number JetIP
101165	Port number for STX debugger
Used by the system	
101200	IP address
101201	Subnet mask
101202	Default gateway
101203	DNS server
101232	Host name suffix type
101233 ... 101251	Host name (register string)
101264	Port number JetIP
101265	Port number for STX debugger
101280 ... 101298	File name for AutoCopy
101299	Saving the settings (0x77566152)
101908	CRC of ModConfig.da

Real-time clock

Registers	Description
102910 … 102917	**Direct access**
102910	Milliseconds
102911	Seconds
102912	Minutes
102913	hours
102914	Day of the week (0 = Sunday)
102915	Day
102916	Month
102917	Year
102920 … 102928	**Buffer access**
102920	Milliseconds
102921	Seconds
102922	Minutes
102923	hours
102924	Day of the week (0 = Sunday)
102925	Day
102926	Month
102927	Year
102928	Read/write trigger

Serial port

Registers	Description
103000	**Error state (bit-coded)**
Bit 14 = 1	Framing error
Bit 13 = 1	Parity error
Bit 12 = 1	Overflow
103001	**Protocol**
1	System logger
2	Prim
3	pcomX
103002	Baud rate (1,200 … 115,200)
103003	Bits per character (5 ... 8)
103004	Stop bits (1, 2)
103005	**Parity**
0	None
1	Odd
2	Even
3	1
4	0
103006	0 = RS-232, 1 = RS-422, 3 = RS-485/2
103010	Sending buffer
103011	Sending buffer filling level
103012	Receiving buffer (without immediate clearing)
103013	Receiving buffer (with immediate clearing)
103014	Receive buffer filling level
103015	Receive buffer, 16-bit, little endian
103016	Receive buffer; 16-bit; big endian
103017	Receive buffer, 32-bit, little endian
103018	Receive buffer; 32-bit; big endian
103019	Error counter

Ethernet

Registers	Description
104100 ... 104156	MIB counter
ARP	
104200	Sent requests
104201	Received requests
104202	Sent responses
104203	Received responses
104204	Dynamic entries
104205	Static entries
104206	Obsolete entries
104250	Executing an ARP request
104350	GNN
IP	
104500	Sent packets
104501	Sent bytes
104502	Received packets
104503	Received bytes
104504	Invalid packets
104505	Discarded received packets
104506	Checksum error at reception
104507	Discarded transmit packets
104508	Sent fragments
104509	Received fragments
104531	Current IP address (rw)
104532	Current subnet mask (rw)
104533	Current default gateway (rw)
104534	IP address of DNS server (rw)
TCP	
104800	Sent packets
104801	Sent bytes
104802	Received packets
104803	Received bytes
104804	Invalid packets
104805	Discarded received packets
104806	Checksum error
104807	Connections
104808	Disconnections
104809	Discarded connections
104810	Repeated transmit packets

Registers	Description
UDP	
104900	Sent packets
104901	Sent bytes
104902	Received packets
104903	Received bytes
104904	Invalid packets
104905	Discarded received packets
104906	Checksum error

SD memory card

Registers	Description
107000	**Status**
Bit 0 = 1	SD card inserted
Bit 1= 1	SD card ready
107001	Write protection
1	Card is write-protected (only valid if R 107000 = 3)
107002	Memory size in MB

Flash disk

Register range	Description
107500	Status
107501	**Command**
30	Read statistics
107510 … 107513	**Sector statistics**
107510	Total
107511	Used
107512	Blocked
107513	Free
107520 … 107523	**Byte statistics**
107520	Total
107521	Used
107522	Blocked
107523	Free

LEDs

Registers	Description
108002	**All LEDs ON/OFF (bit-coded)**
Bit 0	"R" LED
Bit 1	"E" LED
Bit 2	LED D1
Bit 3	LED D2
108003	"R" LED
108004	"E" LED
108005	LED D1
108006	LED D2
Values	
0	OFF
1	Flashing slowly
2	Flashing fast
3	ON
108007	**"SD" LED**
0	OFF
3	ON
108008	**LED U1 ... U4 on/off (bit-coded)**
Bit 0	LED U1
Bit 1	LED U2
Bit 2	LED U3
Bit 3	LED U4

CPU/backplane module

Registers	Description
108010	DIP switch - All sliders
108011	DIP switch - Address
108012	DIP switch - Mode
108015	**Mode selector**
1	LOAD
2	RUN
3	STOP
108020	Backplane revision
108021	CPU revision
108099	Clear EEPROM (0x12345678)

Registers	Description
108100 ... 108227	EEPROM registers on backplane

General system registers

Registers	Description
200000	OS version (major * 100 + minor)
200001	**Application program is running (bit 0 = 1)**
0/2	Stop program
1	Start program
3	Continue program
200008	**Error register 1 (identical with 210004)**
Bit 0	Error on flash disk
Bit 1	Error on the JX3 system bus
Bit 2	Error on the JX2 system bus
Bit 3	Error on Ethernet system bus
Bit 7	Error in extended error register
Bit 8	Illegal jump
Bit 9	Illegal call
Bit 10	Illegal index
Bit 11	Illegal opcode
Bit 12	Division by 0
Bit 13	Stack overflow
Bit 14	Stack underflow
Bit 15	Illegal stack
Bit 16	Error when loading the application program
Bit 17	Memory protection violated
Bit 24	Timeout - Cycle time
Bit 25	Timeout - Task lock
Bit 31	Unknown error
200009	**Enhanced error register 1 (bit-coded)**
Bit 3	Error in ModConfig.da
Bit 5	JetVM error
Bit 10	A bus node (publish/subscribe client) has reported an error
Bit 12	JetIPScan has reported an error
Bit 16	NetConsistency has reported an error
Bit 20	Internal memory error
Bit 21	Application program error
Bit 22	System logger is active (R 209700 = 213)
Bit 24	IP address conflict detected
200010	**Enhanced error register 2 (bit-coded)**
Bit 1	Error in the MC object
Bit 7	File system error
200051	**Error numbers** of JetIPScan
0	No error or warning
5	The user has terminated the function
1001	The first received response does not match response 2 and 3
1002	The second received response does not match response 1 and 3
1003	The third received response does not match response 2 and 3
-1	All three responses are dissimilar
-2	The IP settings of at least one node are dissimilar
-3	The JetIPScan function has been invoked, although it is running
-10	The length of the set value list is <1 or >255, or the pointer to the list is invalid
-11	A GNN of the set value list is <1 or >255, or it is a multiple GNN
-20 ... -40	Internal error
-1001 ... -1199	The node has reported the wrong CtrlID or CtrlIDopt
-2001 ... -2199	The node has not called
-3001 ... -3199	Several nodes of the same GNN have called
200061	NetConsistency error numbers, see R 470040

Registers	Description
200169	OS version (IP format)
200170	Controller type (340/350/360)
200300	Currently available heap
200301	Available heap at system launch
200302	Available heap at pre-booting of the application program
201000	Runtime register in milliseconds (rw)
201001	Runtime register in seconds (rw)
201002	Runtime register in reg. 201003 (rw)
201003	10 ms units for register 201002 (rw)
201004	Runtime register in milliseconds (ro)
201005	Runtime registers in microseconds (ro)
202930	**Web status (bit-coded)**
Bit 0 = 1	FTP server available
Bit 1 =1	HTTP server available
Bit 2 = 1	E-mail available
Bit 3 = 1	Data file function available
Bit 4 = 1	Modbus/TCP has been licensed
Bit 5 = 1	Modbus/TCP available
Bit 6	Reserved
Bit 7 = 1	FTP client is available
202936	**Control register - File system**
0xc4697a4b	Formatting the flash disk
0xd364e64d	Formatting the SD card
0x2c9b3c94	Checking the SD card
202960	Password for system command register (0x424f6f74)
202961	**System command register**
102	Controller restart (reboot)
104	Reset non-volatile parameters
122	Wait for communication - OFF
123	Wait for communication - ON
160	Task switch on I/O access - OFF
161	Task switch on I/O access - ON
170	Continue task time slice - OFF
171	Continue task time slice - ON
310	Load the configuration data
311	Load the module configuration
312	Load process data configuration for Ethernet system bus
313	Stop process data configuration for Ethernet system bus
330	JetIPScan client OFF
331	JetIPScan client ON
410	JetSync blockage OFF
411	JetSync blockage for all ports ON
412	JetSync blockage for port X15 ON
202962	**System status register**
Bit 0 = 1	Task switch on I/O access
Bit 1 = 1	Without waiting for communication
Bit 2 = 1	JetIPScan client ON
Bit 3 = 1	Continue task time slice - ON
Bit 8 = 1	JetSync blockage ON
202970	Password for start delay (0x424f6f74)
202971	Start delay in steps of 100 ms
203000	Interface monitoring: JetIP
203001	Interface monitoring: SER
203005	Interface monitoring: STX debug server
203100 ... 203107	32-bit overlaying - Flag 0 ... 255
203108 ... 203123	16-bit overlaying - Flag 0 ... 255
203124 ... 203131	32-bit overlaying - Flag 2048 ... 2303
203132 ... 203147	16-bit overlaying - Flag 2048 ... 2303
209700	System logger: Global enable
209701 ... 209754	Enabling system components

Application program

Registers	Description
210000	**Application program is running (bit 0 = 1)**
0/2	Stop program
1	Start program
2	Continue program
210001	JetVM version
210004	**Error register (bit-coded)**
Bit 1	Error on the JX3 system bus
Bit 2	Error on the JX2 system bus
Bit 3	Error on Ethernet system bus
Bit 7	Error in extended error register
Bit 8	Illegal jump
Bit 9	Illegal call
Bit 10	Illegal index
Bit 11	Illegal opcode
Bit 12	Division by 0
Bit 13	Stack overflow
Bit 14	Stack underflow
Bit 15	Illegal stack
Bit 16	Error when loading the application program
Bit 24	Timeout - Cycle time
Bit 25	Timeout - Task lock
Bit 31	Unknown error
210006	Highest task number
210007	Minimum program cycle time
210008	Maximum program cycle time
210009	Current program cycle time
210011	Current task number
210050	Current program position within an execution unit
210051	ID of the execution unit being processed
210056	Desired total cycle time in µs
210057	Calculated total cycle time in µs
210058	Maximum time slice per task in µs

Registers	Description
210060	Task ID (for R210061)
210061	Priority for task [R210060]
210063	Length of scheduler table
210064	Index in scheduler table
210065	Task ID in scheduler table
210070	Task ID (for R210071)
210071	Timer number (0 ... 31)
210072	Manual triggering of a timer event (bit-coded)
210073	End of cyclic task (task ID)
210074	Command for cyclic tasks
210075	Number of timers
210076	Timer number (for R210077)
210077	Timer value in milliseconds
210091	Debugging - STX variable address
210093	Debugging - STX variable value
210100 ... 210199	Task state Apply the STX function TaskGetInfo() , as described in the JetSym Online Help.
210400 ... 210499	Task - Program address
210600	Task ID of a cyclical task (for R210601)
210601	Processing time of a cyclical task in per mil figure
210609	**Task lock timeout in ms**
-1	Monitoring disabled
210610	Timeout (bit-coded, bit 0 → timer 0 etc.)
211010 … 211024	**STX memory utilization**
211010	Total memory: Total
211011	Total memory: Used
211012	Total memory: Free
211013	System memory: Total
211014	System memory: Used
211015	System memory: Free
211016	Application memory: Total

Registers	Description
211017	Application memory: Used
211018	Application memory: Free
211019	Used memory: Program
211020	Used memory: Data
211021	Used memory: Constants
211022	Used memory: Stack
211023	Used memory: JIT compiler
211024	Used memory: System
TCP auto-close for the STX debug server	
212000	Number of open connections
212001	Mode
212002	Time

HMI control

Registers	Description
222804	Total number of display characters
222805	Number of characters per line
222806	Text selection (DisplayText2)
222808	Number of decimal places (UserInput)
222810	Number of decimal places (DisplayValue)
222811	Max. number of decimal places (UserInput)
222812	Field length (DisplayValue)
222813	Field length (UserInput)
222814	Indirect cursor position
222815	Default value for UserInput (Integer/Float)
222816	Displaying the sign
222817	Status UserInput
222818	Enable/disable monitor functions
222819	Display text - monitor function
222820	Switching over to monitor display
222821	Dialog language
222824	Indirect buffer number
Multi-display mode	
222825	Text buffer for display 1
222826	Text buffer for display 2
222827	Text buffer for display 3
222828	Text buffer for display 4
222829	Basic flag number for display 1
222830	Basic flag number for display 2
222831	Basic flag number for display 3
222832	Basic flag number for display 4
222833	Register number - LED display 1
222834	Register number - LED display 2
222835	Register number - LED display 3
222836	Register number - LED display 4
222837	Module number of PRN (display redirection)
222838	Module number of SER (display redirection)
222839	Character code for *Delete display*
222840	Character code for *Delete to end of line*

Networking via JetIP

Registers	Description
TCP Autoclose for JetIP/**TCP server**	
230000	Number of open connections
230001	Mode
230002	Time
Other registers for networking via JetIP	
232708	Timeout in milliseconds
232709	Response time in milliseconds
232710	Amount of network errors
232711	**Error code of last access**
0	No error
1	Timeout
3	Error message from remote station
5	Illegal network address
6	Invalid amount of registers
7	Illegal interface number
232717	Maximum number of retries

Registers	Description
232718	Number of retries
Network registers	
235000 ... 235399	IP addresses
235400 ... 235799	Port numbers
236000 ... 236399	Indirect register numbers GNN: nnn = 000 ... 199
1nnn020000 ... 1nnn179999	JX3 module register
1nnn202000 ... 1nnn227999	JX2 module register
1nnn810000 ... 1nnn819999	JetMove registers
1nnn980000 ... 1nnn980199	Indirect access via local register 236xxx
1nnn990000 ... 1nnn999999	Indirect access with variable destination window

Ethernet system bus

Register	Description
Subscriber	
250000	**Status (bit-coded)**
Bit 0 = 1	No CRC
Bit 1 = 1	Error in connection with a subscription
Bit 7 = 1	Subscriber is running
250001	**Command**
102	Restart
105	Stop
110	Acknowledge error
250002	Subscription ID of the last error
250003	Number of subscriptions
250004	CRC of configuration file
250005	Start of communication (timeout reg. [ms])

Register	Description
250010	Selection via command
250011	Selection via ID
Subscription	
250020	Status
250021	Mode
250022	Number of elements
250023	Multicast group
250024	Hash
250025	Current sequence number
250026	Size (bytes)
250027	Timeout
250028	Number of received publications
250029	Number of timeout errors
250030	Number of sequence number errors
250100 ... 250999	9 more subscriber register blocks
Address of the bus node (or controller) exceeding the timeout time	
254001	GNN
254002	IP address
254003	Port number
Publisher	
255000	**Status (bit-coded)**
Bit 0 = 1	No CRC
Bit 1 = 1	Error in connection with a publication
Bit 7 = 1	Subscriber is running
255001	**Command**
102	Restart
105	Stop
110	Acknowledge error
255002	Publication ID of the last error
255003	Number of publications
255004	CRC of configuration file
255010	Selection via command
255011	Selection via ID
Publication	

Register	Description
255020	Status
255021	Mode
255022	Number of elements
255023	Multicast group
255024	Hash
255025	Current sequence number
255026	Size (bytes)
255027	Cycle time
255028	Number of publications sent
255029	Number of retries
255030	Number of transmit errors
255100 ... 25 5999	9 more publisher register blocks

RemoteScan

Registers	Description
262965	Protocol type
262966	Amount of configuration blocks
262967	Status

Modbus/TCP

Registers	Description
272702	Register offset
272704	Input offset
272705	Output offset
278000 ... 278999	16-bit I/O registers overlaid by virtual I/Os 20001 ... 36000

E-mail

Registers	Description
292932	IP address of the SMTP server
292933	IP address of the POP3 server
292934	Port number of SMTP server
292935	Port number of POP3 server
292937	Status of e-mail processing
292938	Task ID - E-mail

File system/data file function

Register	Description
312977	Status of file operation
312978	Task ID

FTP client

Registers	Description
320000	Number of open connections
320001	Command
320002	Timeout
320003	Server port
320004	Selection via number
320005	Selection via handle
320006	Server socket: IP address
320007	Server socket: Port
320008	Client socket: IP address
320009	Client socket: Port
320100	Access status
320101	Task ID

User-programmable IP interface

Registers	Description
Reading out the connection list	
350000	Last result (-1 = no connection selected)
350001	1 = Client; 2 = Server
350002	1 = UDP; 2 = TCP
350003	IP address
350004	Port number
350005	Connection state
350006	Number of sent bytes
350007	Number of received bytes
350008	Number of discarded bytes
350009	Number of discarded packets

Error history

Register	Description
380000	**Status**

Register	Description
Bit 0 = 1	Recording
Bit 1 = 1	Stop if buffer is full
Bit 2 = 1	Stop on error code
Bit 3 = 1	Non-volatile memory
380001	**Command**
1	Clear error log
2	Start error log
3	Stop error log
4	Stop if error buffer is full
5	Circular buffer
6	Stop on error code ON
7	Stop on error code OFF
10	Non-volatile memory
11	Dynamic memory
380002	Buffer length
380003	Maximum buffer length
380004	Number of error entries
380005	Index to error list
380006	Error entry
380007	Error stop code
380008	Number of codes until stop
380029	Group index to error list
380030 ... 380093	64 error entries

I/O networking

Registers	Description
Status register	
390000 + node * 10	Error register
390001 + node * 10	Enhanced error register 1
390002 + node * 10	Enhanced error register 2
390003 + node * 10	JetSync status
390004 + node * 10	Subscriber status
390005 + node * 10	Subscription ID of the last error
Address of a bus node (not of a controller) having reported an error	
394001	GNN
394002	IP address
394003	Port number
Control register	
395000 + node * 10	Command

NetConsistency function

Registers	Description
Basic drivers	
470000 ... 470008	Cookie
470009	Version
470010	**Status**
Bit 0 = 1	Error
Bit 1 = 1	Alarms
Bit 2 = 1	Basic driver initialized
470011	**Command**
0	There are no commands
470020	Maximum possible number of instances
470021	Number of instances ready for operation
470030	Max. number of error messages for the logger
470031	Number of error messages transmitted to the logger
470032	Max. number of warnings for the logger
470033	Number of warnings forwarded to the logger
470034	Max. possible number of error history entries
470035	Number of entries in the error history
470040	Error numbers

Registers	Description
470041	Time of the error in ms
470042	Instance, at which the error occurred
470043	Number of error parameters
470044 ... 470048	Error parameters 1 through 5
470049	Number of characters of the error message
470050 ... 470157	Text of the error message
First instance	
471010	**Status**
Bit 0 = 1	Error
Bit 1 = 1	Alarms
Bit 2 = 1	An instance has been initialized
Bit 3 = 1	Execution in process
471011	**Command**
0	There are no commands

JetIPScan

Registers	Description
Global status information	
520000	Summary of status messages
520010	State of execution - corresponds to the return value *State*
520011	Number of cycles - corresponds to the return value *Count*
520012	Number of changes - corresponds to the return value *Changed*
520013	Result of the function - corresponds to the return value *Result*
Warnings and errors	
521000 ... 521006	All 3 responses are different
521010 ... 521016	Response 1 is different
521020 ... 521026	Response 2 is different
521030 ... 521036	Response 3 is different
521100 ... 521106	Wrong CtrlID or CtrlIDopt
521200 ... 521206	The node has not called
521300 ... 521306	Multiple call
521400 ... 521406	The IP settings could not be changed
Configuration	
522000	GNN
522010 ... 522015	Set configuration
522110 ... 522123	Actual configuration 1
522210 ... 522223	Actual configuration 2
522310 ... 522323	Actual configuration 3

SW function JCF-SV1

Register range	Description
Initializing the JCF-SV1 SW function	
600000	Status register
600001	Activation of instances
600002	Call interval
600003	Cycle time of all instances
600004 … 600009	Reserve
Configuring the JCF-SV1 SW function for individual axes	
600010 … 600029	**Axis 1**
600010	Source register number of the actual position
600011	Target register number of the manipulated variable
600012	Input number: Negative HW limit switch

Register range	Description
600013	Input number: Positive HW limit switch
600014	Input number: Reference switch
600015	Output number: Dig. neg. direction preset
600016	Output number: Dig. pos. direction preset
600017	Output number: Dig. ENABLE for the amplifier
600018 … 600029	Reserve
600030 … 600049	**Axis 2**
600030	Source register number of the actual position
600031	Target register number of the manipulated variable
…	…
600038 … 600049	Reserve
600050 … 600069	**Axis 3** → see axis 1
600070 … 600089	**Axis 4** → see axis 1
600090 … 600109	**Axis 5** → see axis 1
600110 … 600129	**Axis 6** → see axis 1
600130 … 600149	**Axis 7** → see axis 1
600150 … 600169	**Axis 8** → see axis 1
600170 … 600189	**Axis 8** → see axis 1
600190 … 600209	**Axis 10** → see axis 1
600210 … 600229	**Axis 11** → see axis 1
600230 … 600249	**Axis 12** → see axis 1
600250 … 600269	**Axis 13** → see axis 1
600270 … 600289	**Axis 14** → see axis 1
600290 … 600309	**Axis 15** → see axis 1
600310 … 600329	**Axis 16** → see axis 1
Operation of individual axes	
6yyzzz **6= prefix** **yy = axis number 01 … 16** **zzz = module register number 000 … 999**	
601000 … 601999	Axis 1
602000 … 602999	Axis 2
…	…
616000 … 616999	Axis 16

Application registers

Registers	Description
1000000 ... 1059999	32-bit integer or floating point number (non-volatile)
1000000 ... 1119999	32-bit integer or floating point number (non-volatile), with option -R

JX3 system bus registers

Registers	Description
100000000	**Bus status**
Bit 15 = 1	Data exchange takes place via JX3 system bus.
100002000	JX3 system bus hardware revision
100002008	**Error (bit-coded)**
Bit 3 =1	Error at module access
Bit 16 = 1	Fatal irrecoverable error has occurred. Data interchange has been aborted.

Registers	Description
100002011	I/O module number where error has occurred
100002013	Number of detected I/O modules
100002015	Index to module array
100002016	Module array
100002023	Dummy modules
100002034	Number of retries
100002072	Version of the JX3 system bus driver
100002111	Module register number where error has occurred
100002764	Timeout period for register access [ms]
100003xx0 ... 100003xx9	Registers on I/O modules (compatibility mode) xx: Module number - 2 (00 ... 15)
100004000 ... 100004367	Inputs/outputs mapped to registers
100xx0000 … 100xx9999	Registers on I/O modules (direct access) xx: Module number (02 ... 17)

JX2 System Bus Registers

Registers	Description
200002000	Version of JX2 system bus driver (IP)
200002008	**Error (bit-coded)**
Bit 3	I/O or CANopen module timeout
Bit 4	JX2 slave module timeout
Bit 9	I/O module error
Bit 12	Object length has not been set
Bit 13	Error during JX2 system bus initialization
Bit 14	Timeout of system registers
Bit 15	SDO abort
200002011	I/O module number at timeout
200002012	JX2 slave module number at timeout
200002013	Amount of connected I/O modules
200002014	Amount of connected JX2 slave modules
200002015	Index to module array
200002016	Module array
200002023	Dummy I/O module
200002024	JX2 slave dummy modules
200002028	Monitoring interval for I/O modules [10 ms]
200002029	Baud rate of JX2 system bus
200002032	ON delay
200002039	Faulty I/O module (bit-coded)
200002070	Number of CANopen modules
200002071	Actual I/O sum of modules on the JX2 system bus
200002072	Version of JX2 system bus driver (IP)
200002073	Timeout for register access CANopen modules
200002074	CANopen SYNC interval [ms]
200002077	**Enabling JX2 system bus special functions**
Bit 3, 2 = 01	User-programmable CAN interface in addition to the JX2 system bus
Bit 3, 2 = 10	Only CANopen interface (CANopen STX API)
Bit 3, 2 = 11	Only user-programmable CAN interface
Bit 4 = 1	CAN IDs 0x081 ... 9x09F for user-programmable CAN interface
Bit 6 = 1	CANopen functionality in the JX2 system bus driver is disabled Bit 6 makes sense only if bit 3 has not been set
200002080	CANopen module index for JX2 system bus application registers
200002085	SysBus application registers: Register number (65-89)
200002086	SysBus application registers: Object number
200002087	SysBus application registers: Subindex
200002088	SysBus application registers: Length
200002760	Max. number of I/O update retries

Registers	Description
200002761	Index to array of I/O retry counters
200002762	Array of I/O retry counters
200002763	Timeout for I/O update of I/O modules [ms]
200002764	Timeout for register access to I/O modules [ms]
200002765	Timeout for register access to JX2 slave modules [ms]
200002821	Write 1 to set the CAN error counters to 0
200002824	Counter for stuff errors
200002825	Counter for CRC errors
200002826	Counter for formal errors
200002827	Counter for acknowledge errors
200002828	Counter for bit errors
200002995	Bootloader version of JX2 system bus interface
200003xx0 ... 200003xx9	Registers on I/O modules xx: I/O module number - 2 (00...22)
200004000 ... 200004367	Inputs/outputs mapped to registers (see below)
200005x00 ... 200006x99	I/O registers for CANopen/JX-SIO x: I/O module number - 70 (0...9)
200007x00 ... 200007x99	Configuration registers for CANopen/JX-SIO x: I/O module number - 70 (0...9)
2000xx100 ... 2000xx999	JX2 slave registers xx: JX2 slave number + 10

User-programmable CAN interface

Registers	Description
200010500	Status register
Bit 1 = 1	CAN message has been received
Bit 2 = 0	11-bit CAN ID
Bit 2 = 1	29-bit CAN ID

Registers	Description
200010501	**Command register**
Direct access	
7	Clearing the Fifo buffer
8	Setting CAN ID to 11 bits
9	Setting CAN ID to 29 bits
10	Checking boxes for received messages
Indirect access	
1	Enabling the message box
2	Disabling the message box
3	Sending CAN messages
4	Clearing the NEW DAT bit
5	Clearing the OVERRUN bit
6	Clearing the sending error bit
7	Clearing the Fifo buffer
8	Setting CAN ID to 11 bits
9	Setting CAN ID to 29 bits
10	Checking boxes for received messages
200010502	Message box number (indirect access)
200010503	FIFO buffer occupancy
200010504	FIFO data
200010506	Global receive mask
200010507	Global receive ID
200010509	Version of the user-programmable CAN interface (IP)
Indirect access	
200010510	Message box status register
200010511	Message box configuration register
200010512	CAN-ID
200010513	Number of data bytes
200010514 ... 200010521	Data bytes 0 through 7
Direct access	

Registers	Description
200010530 + box number * 20	Message box status register
200010531 + box number * 20	Message box configuration register
200010532 + box number * 20	CAN-ID
200010533 + box number * 20	Number of data bytes
200010534 ... 200010541 + box number * 20	Data bytes
200010542 + box number * 20	CAN-ID mask
200010543 + box number * 20	Box command register
200010544 + box number * 20	Received CAN-ID

32 combined inputs

Registers	Description			
JX3 system bus: Register + 100000000 / Network: + 1GNN910000				
JX2 system bus: + 200000000				
Example	Register number 100004002 lets you access inputs 1 ... 8 and 9 ... 16 of the JX3 modules at positions 2 and 3.			
4000	101 ... 108	109 ... 116	201 ... 208	209 ... 216
4001	109 ... 116	201 ... 208	209 ... 216	301 ... 308
4002	201 ... 208	209 ... 216	301 ... 308	309 ... 316
4003	209 ... 216	301 ... 308	309 ... 316	401 ... 408
4004	301 ... 308	309 ... 316	401 ... 408	409 ... 416
4005	309 ... 316	401 ... 408	409 ... 416	501 ... 508
4006	401 ... 408	409 ... 416	501 ... 508	509 ... 516
4007	409 ... 416	501 ... 508	509 ... 516	601 ... 608
4008	501 ... 508	509 ... 516	601 ... 608	609 ... 616
4009	509 ... 516	601 ... 608	609 ... 616	701 ... 708
4010	601 ... 608	609 ... 616	701 ... 708	709 ... 716
4011	609 ... 616	701 ... 708	709 ... 716	801 ... 808
4012	701 ... 708	709 ... 716	801 ... 808	809 ... 816
4013	709 ... 716	801 ... 808	809 ... 816	901 ... 908
4014	801 ... 808	809 ... 816	901 ... 908	909 ... 916
4015	809 ... 816	901 ... 908	909 ... 916	1001 ... 1008
4016	901 ... 908	909 ... 916	1001 ... 1008	1009 ... 1016
4017	909 ... 916	1001 ... 1008	1009 ... 1016	1101 ... 1108
4018	1001 ... 1008	1009 ... 1016	1101 ... 1108	1109 ... 1116
4019	1009 ... 1016	1101 ... 1108	1109 ... 1116	1201 ... 1208
4020	1101 ... 1108	1109 ... 1116	1201 ... 1208	1209 ... 1216
4021	1109 ... 1116	1201 ... 1208	1209 ... 1216	1301 ... 1308
4022	1201 ... 1208	1209 ... 1216	1301 ... 1308	1309 ... 1316
4023	1209 ... 1216	1301 ... 1308	1309 ... 1316	1401 ... 1408
4024	1301 ... 1308	1309 ... 1316	1401 ... 1408	1409 ... 1416
4025	1309 ... 1316	1401 ... 1408	1409 ... 1416	1501 ... 1508

Registers	Description			
4026	1401 ... 1408	1409 ... 1416	1501 ... 1508	1509 ... 1516
4027	1409 ... 1416	1501 ... 1508	1509 ... 1516	1601 ... 1608
4028	1501 ... 1508	1509 ... 1516	1601 ... 1608	1609 ... 1616
4029	1509 ... 1516	1601 ... 1608	1609 ... 1616	1701 ... 1708
4030	1601 ... 1608	1609 ... 1616	1701 ... 1708	1709 ... 1716
4031	1609 ... 1616	1701 ... 1708	1709 ... 1716	1801 ... 1808
4032	1701 ... 1708	1709 ... 1716	1801 ... 1808	1809 ... 1816
4033	1709 ... 1716	1801 ... 1808	1809 ... 1816	1901 ... 1908
4034	1801 ... 1808	1809 ... 1816	1901 ... 1908	1909 ... 1916
4035	1809 ... 1816	1901 ... 1908	1909 ... 1916	2001 ... 2008
4036	1901 ... 1908	1909 ... 1916	2001 ... 2008	2009 ... 2016
4037	1909 ... 1916	2001 ... 2008	2009 ... 2016	2101 ... 2108
4038	2001 ... 2008	2009 ... 2016	2101 ... 2108	2109 ... 2116
4039	2009 ... 2016	2101 ... 2108	2109 ... 2116	2201 ... 2208
4040	2101 ... 2108	2109 ... 2116	2201 ... 2208	2209 ... 2216
4041	2109 ... 2116	2201 ... 2208	2209 ... 2216	2301 ... 2308
4042	2201 ... 2208	2209 ... 2216	2301 ... 2308	2309 ... 2316
4043	2209 ... 2216	2301 ... 2308	2309 ... 2316	2401 ... 2408
4044	2301 ... 2308	2309 ... 2316	2401 ... 2408	2409 ... 2416

16 combined inputs

Registers	Description	
JX3 system bus: Register + 100000000 / Network: + 1GNN910000		
JX2 system bus: Register + 2000000		
Example	Register number 100004062 lets you access inputs 1 ... 8 and 9 ... 16 of the JX3 module at position 2.	
4060	101 … 108	109 … 116
4061	109 … 116	201 … 208
4062	201 … 208	209 … 216
4063	209 … 216	301 … 308
4064	301 … 308	309 … 316
4065	309 … 316	401 … 408
4066	401 … 408	409 … 416
4067	409 … 416	501 … 508
4068	501 … 508	509 … 516
4069	509 … 516	601 … 608
4070	601 … 608	609 … 616
4071	609 … 616	701 … 708
4072	701 … 708	709 … 716
4073	709 … 716	801 … 808
4074	801 … 808	809 … 816
4075	809 … 816	901 … 908
4076	901 … 908	909 … 916
4077	909 … 916	1001 … 1008
4078	1001 … 1008	1009 … 1016
4079	1009 … 1016	1101 … 1108
4080	1101 … 1108	1109 … 1116
4081	1109 … 1116	1201 … 1208
4082	1201 … 1208	1209 … 1216
4083	1209 … 1216	1301 … 1308
4084	1301 … 1308	1309 … 1316
4085	1309 … 1316	1401 … 1408
4086	1401 … 1408	1409 … 1416
4087	1409 … 1416	1501 … 1508
4088	1501 … 1508	1509 … 1516

Registers	Description	
4089	1509 … 1516	1601 … 1608
4090	1601 … 1608	1609 … 1616
4091	1609 … 1616	1701 … 1708
4092	1701 … 1708	1709 … 1716
4093	1709 … 1716	1801 … 1808
4094	1801 … 1808	1809 … 1816
4095	1809 … 1816	1901 … 1908
4096	1901 … 1908	1909 … 1916
4097	1909 … 1916	2001 … 2008
4098	2001 … 2008	2009 … 2016
4099	2009 … 2016	2101 … 2108
4100	2101 … 2108	2109 … 2116
4101	2109 … 2116	2201 … 2208
4102	2201 … 2208	2209 … 2216
4103	2209 … 2216	2301 … 2308
4104	2301 … 2308	2309 … 2316
4105	2309 … 2316	2401 … 2408
4106	2401 … 2408	2409 … 2416

8 combined inputs

Registers	Description
JX3 system bus: Register + 100000000 / Network: + 1GNN910000	
JX2 system bus: Register + 200000000 / Network: + 1GNN910000	
Example	Register number 100004122 lets you access inputs 1 ... 8 of the JX3 module at position 2.
4120	101 ... 108
4121	109 ... 116
4122	201 ... 208
4123	209 ... 216
4124	301 ... 308
4125	309 ... 316
4126	401 ... 408
4127	409 ... 416

Registers	Description
4128	501 ... 508
4129	509 ... 516
4130	601 ... 608
4131	609 ... 616
4132	701 ... 708
4133	709 ... 716
4134	801 ... 808
4135	809 ... 816
4136	901 ... 908
4137	909 ... 916
4138	1001 ... 1008
4139	1009 ... 1016
4140	1101 ... 1108
4141	1109 ... 1116
4142	1201 ... 1208
4143	1209 ... 1216
4144	1301 ... 1308
4145	1309 ... 1316
4146	1401 ... 1408
4147	1409 ... 1416
4148	1501 ... 1508
4149	1509 ... 1516
4150	1601 ... 1608
4151	1609 ... 1616
4152	1701 ... 1708
4153	1709 ... 1716
4154	1801 ... 1808
4155	1809 ... 1816
4156	1901 ... 1908
4157	1909 ... 1916
4158	2001 ... 2008
4159	2009 ... 2016
4160	2101 ... 2108
4161	2109 ... 2116
4162	2201 ... 2208
4163	2209 ... 2216

Registers	Description
4164	2301 ... 2308
4165	2309 ... 2316
4166	2401 ... 2408
4167	2409 ... 2416

32 combined outputs

Registers	Description			
JX3 system bus: Register + 100000000 / Network: + 1GNN910000				
JX2 register + 200000000 / network: + 1GNN910000				
Example	Register number 100004202 lets you access outputs 1 ... 8 and 9 ... 16 of the JX3 modules at positions 2 and 3.			
4200	101 ... 108	109 ... 116	201 ... 208	209 ... 216
4201	109 ... 116	201 ... 208	209 ... 216	301 ... 308
4202	201 ... 208	209 ... 216	301 ... 308	309 ... 316
4203	209 ... 216	301 ... 308	309 ... 316	401 ... 408
4204	301 ... 308	309 ... 316	401 ... 408	409 ... 416
4205	309 ... 316	401 ... 408	409 ... 416	501 ... 508
4206	401 ... 408	409 ... 416	501 ... 508	509 ... 516
4207	409 ... 416	501 ... 508	509 ... 516	601 ... 608
4208	501 ... 508	509 ... 516	601 ... 608	609 ... 616
4209	509 ... 516	601 ... 608	609 ... 616	701 ... 708
4210	601 ... 608	609 ... 616	701 ... 708	709 ... 716
4211	609 ... 616	701 ... 708	709 ... 716	801 ... 808
4212	701 ... 708	709 ... 716	801 ... 808	809 ... 816
4213	709 ... 716	801 ... 808	809 ... 816	901 ... 908
4214	801 ... 808	809 ... 816	901 ... 908	909 ... 916
4215	809 ... 816	901 ... 908	909 ... 916	1001 ... 1008
4216	901 ... 908	909 ... 916	1001 ... 1008	1009 ... 1016
4217	909 ... 916	1001 ... 1008	1009 ... 1016	1101 ... 1108
4218	1001 ... 1008	1009 ... 1016	1101 ... 1108	1109 ... 1116
4219	1009 ... 1016	1101 ... 1108	1109 ... 1116	1201 ... 1208
4220	1101 ... 1108	1109 ... 1116	1201 ... 1208	1209 ... 1216
4221	1109 ... 1116	1201 ... 1208	1209 ... 1216	1301 ... 1308
4222	1201 ... 1208	1209 ... 1216	1301 ... 1308	1309 ... 1316
4223	1209 ... 1216	1301 ... 1308	1309 ... 1316	1401 ... 1408
4224	1301 ... 1308	1309 ... 1316	1401 ... 1408	1409 ... 1416
4225	1309 ... 1316	1401 ... 1408	1409 ... 1416	1501 ... 1508
4226	1401 ... 1408	1409 ... 1416	1501 ... 1508	1509 ... 1516
4227	1409 ... 1416	1501 ... 1508	1509 ... 1516	1601 ... 1608
4228	1501 ... 1508	1509 ... 1516	1601 ... 1608	1609 ... 1616
4229	1509 ... 1516	1601 ... 1608	1609 ... 1616	1701 ... 1708
4230	1601 ... 1608	1609 ... 1616	1701 ... 1708	1709 ... 1716
4231	1609 ... 1616	1701 ... 1708	1709 ... 1716	1801 ... 1808
4232	1701 ... 1708	1709 ... 1716	1801 ... 1808	1809 ... 1816
4233	1709 ... 1716	1801 ... 1808	1809 ... 1816	1901 ... 1908

Registers	Description			
4234	1801 ... 1808	1809 ... 1816	1901 ... 1908	1909 ... 1916
4235	1809 ... 1816	1901 ... 1908	1909 ... 1916	2001 ... 2008
4236	1901 ... 1908	1909 ... 1916	2001 ... 2008	2009 ... 2016
4237	1909 ... 1916	2001 ... 2008	2009 ... 2016	2101 ... 2108
4238	2001 ... 2008	2009 ... 2016	2101 ... 2108	2109 ... 2116
4239	2009 ... 2016	2101 ... 2108	2109 ... 2116	2201 ... 2208
4240	2101 ... 2108	2109 ... 2116	2201 ... 2208	2209 ... 2216
4241	2109 ... 2116	2201 ... 2208	2209 ... 2216	2301 ... 2308
4242	2201 ... 2208	2209 ... 2216	2301 ... 2308	2309 ... 2316
4243	2209 ... 2216	2301 ... 2308	2309 ... 2316	2401 ... 2408
4244	2301 ... 2308	2309 ... 2316	2401 ... 2408	2409 ... 2416

16 combined outputs

Registers	Description	
System bus JX3: Register + 100000000 / Network: + 1GNN910000		
System bus JX2: Register + 200000000 / Network: + 1GNN910000		
Example	Register number 100004262 lets you access outputs 1 ... 8 and 9 ... 16 of the JX3 module at position 2.	
4260	101 ... 108	109 ... 116
4261	109 ... 116	201 ... 208
4262	201 ... 208	209 ... 216
4263	209 ... 216	301 ... 308
4264	301 ... 308	309 ... 316
4265	309 ... 316	401 ... 408
4266	401 ... 408	409 ... 416
4267	409 ... 416	501 ... 508
4268	501 ... 508	509 ... 516
4269	509 ... 516	601 ... 608
4270	601 ... 608	609 ... 616
4263	209 ... 216	301 ... 308
4271	609 ... 616	701 ... 708
4272	701 ... 708	709 ... 716
4273	709 ... 716	801 ... 808
4274	801 ... 808	809 ... 816
4275	809 ... 816	901 ... 908
4276	901 ... 908	909 ... 916
4277	909 ... 916	1001 ... 1008
4278	1001 ... 1008	1009 ... 1016
4279	1009 ... 1016	1101 ... 1108
4280	1101 ... 1108	1109 ... 1116
4281	1109 ... 1116	1201 ... 1208
4282	1201 ... 1208	1209 ... 1216
4283	1209 ... 1216	1301 ... 1308
4284	1301 ... 1308	1309 ... 1316
4285	1309 ... 1316	1401 ... 1408
4286	1401 ... 1408	1409 ... 1416
4287	1409 ... 1416	1501 ... 1508
4288	1501 ... 1508	1509 ... 1516
4289	1509 ... 1516	1601 ... 1608
4290	1601 ... 1608	1609 ... 1616
4291	1609 ... 1616	1701 ... 1708
4292	1701 ... 1708	1709 ... 1716
4293	1709 ... 1716	1801 ... 1808
4294	1801 ... 1808	1809 ... 1816
4295	1809 ... 1816	1901 ... 1908
4296	1901 ... 1908	1909 ... 1916
4297	1909 ... 1916	2001 ... 2008
4298	2001 ... 2008	2009 ... 2016
4299	2009 ... 2016	2101 ... 2108
4300	2101 ... 2108	2109 ... 2116
4301	2109 ... 2116	2201 ... 2208
4302	2201 ... 2208	2209 ... 2216

Registers	Description	
4303	2209 ... 2216	2301 ... 2308
4304	2301 ... 2308	2309 ... 2316
4305	2309 ... 2316	2401 ... 2408
4306	2401 ... 2408	2409 ... 2416

8 combined outputs

Registers	Description
JX3 system bus: Register + 100000000 / Network: + 1GNN910000	
JX2 system bus: Register + 200000000 / Network: + 1GNN910000	
Example	Register number 100004322 lets you access outputs 1 ... 8 of the JX3 module at position 2.
4320	101 ... 108
4321	109 ... 116
4322	201 ... 208
4323	209 ... 216
4324	301 ... 308
4325	309 ... 316
4326	401 ... 408
4327	409 ... 416
4328	501 ... 508
4329	509 ... 516
4330	601 ... 608
4331	609 ... 616
4332	701 ... 708
4333	709 ... 716
4334	801 ... 808
4335	809 ... 816
4336	901 ... 908
4337	909 ... 916
4338	1001 ... 1008
4339	1009 ... 1016
4340	1101 ... 1108
4341	1109 ... 1116

Registers	Description
4342	1201 ... 1208
4343	1209 ... 1216
4344	1301 ... 1308
4345	1309 ... 1316
4346	1401 ... 1408
4347	1409 ... 1416
4348	1501 … 1508
4349	1509 ... 1516
4350	1601 … 1608
4351	1609 ... 1616
4352	1701 ... 1708
4353	1709 … 1716
4354	1801 … 1808
4355	1809 … 1816
4356	1901 … 1908
4357	1909 … 1916
4358	2001 … 2008
4359	2009 … 2016
4360	2101 … 2108
4361	2109 ... 2116
4362	2201 … 2208
4363	2209 … 2216
4364	2301 … 2308
4365	2309 … 2316
4366	2401 … 2408
4367	2409 … 2416

Special flags for networks

Flag	Description
2075	Error in networking via JetIP
2080	Ethernet system bus error in R 200008
2081	Ethernet system bus error

Special flags - Interface monitoring

Flag	Description
2088	OS flag – JetIP
2089	User flag – JetIP
2090	OS flag - SER
2091	User flag - SER
2098	OS flag - Debug server
2099	User flag - Debug server

Special flags - HMIs

Flag	Description
does not apply to LCD 27	
2160	[0]
2161	[1]
2162	[2]
2163	[3]
2164	[4]
2165	[5]
2166	[6]
2167	[7]
2168	[8]
2169	[9]
2170	[SHIFT]+[0]
2171	[SHIFT]+[1]
2172	[SHIFT]+[2]
2173	[SHIFT]+[3]
2174	[SHIFT]+[4]
2175	[SHIFT]+[5]
2176	[SHIFT]+[6]
2177	[SHIFT]+[7]
2178	[SHIFT]+[8]
2179	[SHIFT]+[9]
2181	[SHIFT]+[F1]
2182	[SHIFT]+[F2]
2183	[SHIFT]+[F3]
2184	[SHIFT]+[F4]
2185	[SHIFT]+[F5]
2186	[SHIFT]+[F6]
2187	[SHIFT]+[F7]
2188	[SHIFT]+[F8]
2189	[SHIFT]+[F9]
2190	[SHIFT]+[F10]
2191	[SHIFT]+[F11]
2192	[SHIFT]+[F12]
2193	[SHIFT]+[←]
2194	[SHIFT]+[→]
2195	[SHIFT]+[R]
2196	[SHIFT]+[I/O]
2197	[SHIFT]+[=]
2198	[SHIFT]+[C]
2199	[SHIFT]+[ENTER]
2200	[SHIFT]
2201	[F1]
2202	[F2]
2203	[F3]
2204	[F4]
2205	[F5]
2206	[F6]
2207	[F7]
2208	[F8]
2209	[F9]
2210	[F10]
2211	[F11]
2212	[F12]
2213	[→]
2214	[←]
2215	[R]
2216	[I/O]
2217	[=]
2218	[C]
2219	[ENTER]
2220	[-]

Flag	Description
2221	[SHIFT]+[-]
2222	[.]
2223	[SHIFT]+[.]
2224	LED of [F1]
2225	LED of [F2]
2226	LED of [F3]
2227	LED of [F4]
2228	LED of [F5]
2229	LED of [F6]
2230	LED of [F7]
2231	LED of [F8]
2232	LED of [F9]
2233	LED of [F10]
2234	LED of [F11]
2235	LED of [F12]

Special flags for HMI LCD 27

Flag	Description
2209	[↑]
2210	[↓]
2211	[C]
2212	[ENTER]

Special flags for HMI NUM 25

Flag	Description
2186	[SHIFT]+[S1]
2187	[SHIFT]+[S2]
2188	[SHIFT]+[S3]
2189	[SHIFT]+[S4]
2190	[SHIFT]+[S5]
2206	[S1]
2207	[S2]
2208	[S3]
2209	[S4]
2210	[S5]

32 combined flags

Registers	Description
203100	0 ... 31
203101	32 ... 63
203102	64 ... 95
203103	96 ... 127
203104	128 ... 159
203105	160 ... 191
203106	192 ... 223
203107	224 ... 255

16 combined flags

Registers	Description
203108	0 ... 15
203109	16 ... 31
203110	32 ... 47
203111	48 ... 63
203112	64 ... 79
203113	80 ... 95
203114	96 ... 111
203115	112 ... 127
203116	128 ... 143
203117	144 ... 159
203118	160 ... 175
203119	176 ... 191
203120	192 ... 207
203121	208 ... 223
203122	224 ... 239
203123	240 ... 255

32 combined special flags

Registers	Description
203124	2048 ... 2079
203125	2080 ... 2111
203126	2112 ... 2143
203127	2144 ... 2175
203128	2176 ... 2207
203129	2208 ... 2239

Registers	Description
203130	2240 ... 2271
203131	2272 ... 2303

16 combined special flags

Registers	Description
203132	2048 ... 2063
203133	2064 ... 2079
203134	2080 ... 2095
203135	2096 ... 2111
203136	2112 ... 2127
203137	2128 ... 2143
203138	2144 ... 2159
203139	2160 ... 2175
203140	2176 ... 2191
203141	2192 ... 2207
203142	2208 ... 2223
203143	2224 ... 2239
203144	2240 ... 2255
203145	2256 ... 2271
203146	2272 ... 2287
203147	2288 ... 2303

Application registers with overlaid flags

Registers	Description
1000000	256 ... 287
1000001	288 ... 319
1000002	320 ... 351
1000003	352 ... 383
1000004	384 ... 415
1000005	416 ... 447
1000006	448 ... 479
1000007	480 ... 511
1000008	512 ... 543
1000009	544 ... 575
1000010	576 ... 607
1000011	608 ... 639
1000012	640 ... 671
1000013	672 ... 703
1000014	704 ... 735
1000015	736 ... 767
1000016	768 ... 799
1000017	800 ... 831
1000018	832 ... 863
1000019	864 ... 895
1000020	896 ... 927
1000021	928 ... 959
1000022	960 ... 991
1000023	992 ... 1023
1000024	1024 ... 1055
1000025	1056 ... 1087
1000026	1088 ... 1119
1000027	1120 ... 1151
1000028	1152 ... 1183
1000029	1184 ... 1215
1000030	1216 ... 1247
1000031	1248 ... 1279
1000032	1280 ... 1311
1000033	1312 ... 1343
1000034	1344 ... 1375
1000035	1376 ... 1407
1000036	1408 ... 1439
1000037	1440 ... 1471
1000038	1472 ... 1503
1000039	1504 ... 1535
1000040	1536 ... 1567
1000041	1568 ... 1599
1000042	1600 ... 1631
1000043	1632 ... 1663
1000044	1664 ... 1695
1000045	1696 ... 1727
1000046	1728 ... 1759
1000047	1760 ... 1791

Registers	Description
1000048	1792 ... 1823
1000049	1824 ... 1855
1000050	1856 ... 1887
1000051	1888 ... 1919
1000052	1920 ... 1951
1000053	1952 ... 1983
1000054	1984 ... 2015
1000055	2016 ... 2047

System Functions

For reasons of compatibility, the system functions are listed below.
In JetSym STX, use the corresponding JetSym STX functions instead of the system functions.

System function	Description
4	Converting BCD to HEX
5	Converting HEX to BCD
20	Square root
21	Sine
22	Cosine
23	Tangent
24	Arc sine
25	Arc cosine
26	Arc tangent
27	Exponential function
28	Natural logarithm
29	Absolute value
30	Separation of digits before and after the decimal point
50	Sorting register values
60	CRC generation for Modbus RTU
61	CRC check for Modbus RTU
65/67	Reading register block via Modbus/TCP
66/68	Writing register block via Modbus/TCP
80/85	Initializing RemoteScan
81	Starting RemoteScan
82	Stopping RemoteScan
90	Writing a data file
91	Appending a data file
92	Reading a data file
96	Deleting a data file
150	Configuring NetCopyList
151	Deleting NetCopyList
152	Sending NetCopyList

System function	Corresponding JetSym STX function
4	Function Bcd2Hex(Bcd: int): Int;
5	Function Hex2Bcd(Hex: int): Int;
50	Function QSort(DataPtr: Int, ElementCnt: Int, ElementSize: Int, SortOffset: Int, SortType: STXBASETYPE, SortMode: QSORTMODE): Int;
60	Function ModbusCRCgen(FramePtr: Int, Length: int): Int;
61	Function ModbusCRCcheck(FramePtr: Int, Length: int): Int;
65/67	Function ModbusReadReg(Const Ref MbParam: MODBUS_PARAM): Int;
66/68	Function ModbusWriteReg(Const Ref MbParam: MODBUS_PARAM): Int;
80/85	Function RemoteScanConfig(Protocol: RSCAN_PROTOCOL, Elements: Int, Const Ref Configuration: RSCAN_DSCR): Int;
81	Function RemoteScanStart(Protocol: int): Int;
82	Function RemoteScanStop(Protocol: int): Int;
90/91	Function FileDAWrite(Const Ref FileName: String, Const Ref Mode: String, VarType: DAWRITE_TYPE, First: Int, Last: int): Int;
92	Function FileDARead(Const Ref FileName: String): Int;
110	Function EmailSend(Const Ref FileName: String): Int;
150	Function NetCopyListConfig(IPAddr: Int, IPPort: Int, Const Ref List: TNetCopyLinstL): Int;
151	Function NetCopyListSend(Handle: int): Int;
152	Function NetCopyListDelete(Handle: int): Int;

10 Maintenance

This device is maintenance free. Therefore, for the operation of the device, no inspection or maintenance is required.

10.1 Repairs

Defective components could cause dangerous malfunctions and could compromise safety.
Only the manufacturer is allowed to repair the device.
It is forbidden to open the device.

Modifications to the device

Modifications and alterations to the device and its functions are not allowed. In the case of modifications to the device, any liability is excluded.

The original parts are specifically designed for the device. Parts and equipment from other manufacturers must, therefore, not be used.

Any liability for any damages resulting from the use of non-original parts and equipment is excluded.

10.2 Storage and Shipment

Storage

When storing the device observe the environmental conditions given in chapter “Technical specifications”.

Shipment and packaging

The device contains electrostatically sensitive components which can be damaged if not handled properly. Damages to the device may impair its reliability.

To protect the device from impact or shock, it must be shipped in its original packaging, or in an appropriate protective ESD packaging.
In case of damaged packaging inspect the device for any visible damage, and inform your freight forwarder and the Bucher Automation AG of the damage caused during shipment. If the device is damaged or has been dropped, it is strictly forbidden to use it.

10.3 Return and Disposal

Disposal options

Return your Bucher Automation AG product to us for proper disposal. Visit our website for detailed information and to download the required Returns form.

Meaning of WEEE icon



Fig. 27: WEEE icon – crossed out trash can

The product is waste electronic equipment and must be disposed of by a certified waste management facility. Do not dispose of the product with normal household waste. Applicable local environmental directives and regulations must be complied with.

Batteries

Prior to disposing of waste electronics, remove any batteries where this is possible in a safe and non-destructive way. Dispose of batteries properly.

Personal data

It is the responsibility of the customers to delete any personal data stored on waste electric and electronic equipment prior to disposal.

11 Service

11.1 Technical support

In case of questions, suggestions, or issues, please contact our experts from Technical Support. You may reach out by phone or through the contact form on our homepage:

Technical Support | Bucher Automation - We automate your success.

Or email us:

support@bucherautomation.com

Please supply the following information when contacting Technical Support:

– Hardware revision and serial number
  The hardware revision and serial number is printed on the nameplate of the product.
– OS version
  To determine the operating system version, use the functions of the development environment.

12 Spare parts and accessories

NOTICE



Inadequate accessories might cause damage to the product

Parts and equipment from other manufacturers might impede the function of the device and cause damage to the product.

► Only use accessories recommended by Bucher Automation AG.

12.1 Spare Parts

Component	Item number
Terminal labels	60870411
Key	60870410
Male connector in spring-cage technology, 2-pin	60870409

Tab. 63: Spare parts

12.2 Accessories

ⓘ Info

Ordering Accessories

The accessories are not part of the scope of delivery.

Suitable accessories can be obtained from Bucher Automation AG.

Component	Item number
Screwdriver	60871712
End clamp for DIN rail	60863970

Tab. 64: Accessories

12.2.1 CAN bus cable

Component	Item number
Cable assy # 530 0.2 m	10309001
Cable assy # 530 0.5 m	10309002
Cable assy # 530 1.0 m	10309003
Cable assy # 530 1.5 m	10309004
Cable assy # 530 2.0 m	10309006
Cable assy # 530 2.5 m	10309016
Cable assy # 530 3.0 m	10309015
Cable assy # 530 4.0 m	10309007
Cable assy # 530 5.0 m	10309008
Y-cable KAY_Breakout_Cable-second_CAN	60880710

12.2.2 Ethernet Cable

Component	Item number
Patch cable 1:1, 1 m, gray, Hirose, Cat 5e, shielded	60537500
Patch cable 1:1, 2 m, gray, Hirose, Cat 5e, shielded	60854512
Patch cable 1:1, 5 m, gray, Hirose, Cat 5e, shielded	60854514
Patch cable 1:1, 10 m, gray, Hirose, Cat 5e, shielded	60854515

12.2.3 Cable for the serial interface

Controller to modem

Component	Description	Item number
KAY_0576-0050	From controller to modem 9-pin Sub-D, length 0.5 m	60867209

From controller to PC

Component	Description	Item number
Cable assy no.196 2.5M	From controller to PC 9-pin Sub-D, length 2.5 m	60868359
Cable assy no.196 5M	From controller to PC 9-pin Sub-D, length 5 m	60860013
Cable assy no.196 8M	From controller to PC 9-pin Sub-D, length 8 m	60868956

Controller to LCD 16, 23, 25, 27, 110

Component	Description	Item number
Cable assy no. 192 2.5M	From controller to HMI 15-pin Sub-D, length 2.5 m	60860011
Cable assy no. 193 5M	From controller to HMI 15-pin Sub-D, length 5 m	60860012

Controller to LCD 60

Component	Description	Item number
KAY_0386-0250	From controller to LCD 60 15-pin Sub-D, length 2.5 m	60864359
KAY_0386-0500	From controller to LCD 60 15-pin Sub-D, length 5 m	60864360

Controller to LCD 52/54(Z)

Component	Description	Item number
KAY_0533-0025	From controller to LCD 52/54(Z) 15-pin Sub-D, length 0.25 m	60864897

Controller to JetView 200/300

Component	Description	Item number
Cable assy no. 197 5M	From controller to 200/300 9-pin Sub-D, length 5 m	60864257
Cable assy no. 197 12M	From controller to 200/300 9-pin Sub-D, length 12 m	60871930

List of figures

Fig. 1 System overview 10
Fig. 2 Design 11
Fig. 3 Status indication 12
Fig. 4 Sample nameplate 15
Fig. 5 Dimension in mm 18
Fig. 6 Installing the device on the DIN rail 26
Fig. 7 Removing the device from the DIN rail 27
Fig. 8 Removing the enclosure from the backplane module 28
Fig. 9 Strain relief on the Sub-D enclosure 30
Fig. 10 X11 port block diagram 33
Fig. 11 Ports X14, X15 34
Fig. 12 Y-cable (splitter cable) for the CAN bus interface 36
Fig. 13 Wiring diagram of the system bus cable 38
Fig. 14 Inserting the SD card 40
Fig. 15 Configuration 41
Fig. 16 44
Fig. 17 DIP switch sliders 1 ... 12 OFF 60
Fig. 18 JetIPScan - command options 61
Fig. 19 DIP switch sliders 1 ... 8 ON 63
Fig. 20 DIP switch sliders 1 ... 8 arbitrarily 65
Fig. 21 DIP switch GNN 69
Fig. 22 Example: Register numbers 93
Fig. 23 Example: I/O numbers 93
Fig. 24 System overview 94
Fig. 25 Example: Register numbers 95
Fig. 26 Example: I/O numbers 95
Fig. 27 WEEE icon – crossed out trash can 149

List of tables

Tab. 1 LED states during the boot process 14
Tab. 2 Mechanical specifications 19
Tab. 3 Power supply (terminal X10) 19
Tab. 4 Serial interface (socket X11) 20
Tab. 5 Ethernet ports (sockets X14, X15) 20
Tab. 6 CAN bus interface (socket X19) 20
Tab. 7 SD card (slot X61) 20
Tab. 8 Memory configurations 21
Tab. 9 Electrical safety 21
Tab. 10 System bus 21
Tab. 11 JX3 IO modules on the system bus 22
Tab. 12 Real-time clock 22
Tab. 13 Environmental conditions 22
Tab. 14 Emitted interference 23
Tab. 15 Immunity to interference 23
Tab. 16 Immunity of shielded data and I/O lines 23
Tab. 17 DC power supply inputs and outputs 24
Tab. 18 S11 mode switch positions 30
Tab. 19 Connector specification, 2-pin connector, spring-cage technology 32
Tab. 20 JX2 system bus cable specifications 37
Tab. 21 [IDENTIFICATION] section 47
Tab. 22 [PRODUCTION] section 48
Tab. 23 [FEATURES] section 48
Tab. 24 Register numbers of the EDS 49
Tab. 25 Overview of EDS registers 49
Tab. 26 Register overview - Hardware revisions 50
Tab. 27 Software version number format 51
Tab. 28 Registers - Overview 51
Tab. 29 Factory settings 52
Tab. 30 Configuration memory - Default values 54
Tab. 31 Address 55
Tab. 32 SubnetMask 55
Tab. 33 DefGateWay 56
Tab. 34 DNSServer 56
Tab. 35 SuffixType 56
Tab. 36 Name 57
Tab. 37 JetIPBase 57
Tab. 38 JVMDebug 57
Tab. 39 AutoCopyIni 57

Tab. 40 Register numbers of the configuration registers 58
Tab. 41 Overview of configuration registers 59
Tab. 42 Elements of the IP address 62
Tab. 43 IP address elements 64
Tab. 44 Overview over the configuration memory registers 66
Tab. 45 Elements of the IP address 68
Tab. 46 Name resolution 70
Tab. 47 System directories 77
Tab. 48 Abbreviations 79
Tab. 49 Module register properties 79
Tab. 50 Numerical formats 79
Tab. 51 Web status 104
Tab. 52 Web status 105
Tab. 53 Register offset 106
Tab. 54 Input offset 106
Tab. 55 Output offset 106
Tab. 56 Supported commands – Class 0 107
Tab. 57 Supported commands – Class 1 107
Tab. 58 Supported commands – Class 2 108
Tab. 59 System functions vs. JetSym STX functions 109
Tab. 60 Web status 116
Tab. 61 Web status 117
Tab. 62 Web status 118
Tab. 63 Spare parts 151
Tab. 64 Accessories 151

Bucher Automation AG
Thomas-Alva-Edison-Ring 10
71672 Marbach/Neckar, Germany
T +49 7141 2550-0
info@bucherautomation.com

www.bucherautomation.com