BUCHER automation

User Manual



60883708_03

JC-975MC

Controller with Integrated Industrial PC

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	1.15.6
Date of issue	12/11/2024

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 ... 11
3.2 Design ... 12
3.3 Product features ... 13
3.4 Axis engineering guidelines ... 14
3.5 Status indication ... 14
3.5.1 LED states during the boot process ... 15
3.5.2 PWR and RSQ LED ... 15
3.6 Nameplate ... 16
3.7 Scope of Delivery ... 16
4 Technical Specifications ... 17
4.1 Dimensions ... 17
4.2 Mechanical specifications ... 18
4.3 Electrical properties ... 18
4.4 Real-time clock ... 21
4.5 Environmental conditions ... 22
4.6 EMI values ... 23
4.6.1 Housing ... 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 the controller ... 25
5.2 Removing the controller ... 25
5.3 Installing JI-PCIE-Exx riser cards ... 26
6 Electrical Connection ... 30
6.1 Improving noise immunity ... 30
6.2 Jack X101 – power supply ... 31

6.3 Jack X102 – Ethernet ... 31
6.4 Jack X103 – EtherCAT® ... 32
6.5 Jack X104 – Ethernet ... 32
6.6 Ports X105 and X106 – USB ... 32
6.7 Ports X107 and X108 – USB ... 33
6.8 Port X109 - DisplayPort ... 34
6.9 The SDCARD interface ... 34
6.10 Port X110 – serial interface RS-232 ... 35
6.11 PCI express slots X81, X82, X91, X92 ... 35
6.11.1 JX2 system bus interface ... 35
6.11.2 JX6-IO submodules for local I/Os ... 35

7 Control elements ... 38
7.1 Mode selector S11 ... 38
7.2 PWR button ... 39
7.3 RSQ button ... 39

8 Identification ... 40
8.1 Electronic Data Sheet (EDS) ... 40
8.1.1 EDS file "eds.ini" ... 40
8.1.2 EDS registers ... 42
8.2 Version registers ... 43
8.2.1 Hardware revisions ... 43
8.2.2 Operating system version ... 43

9 Configuration ... 45
9.1 Default values ... 45
9.2 Configuration file "config.ini" ... 46
9.2.1 Changing the configuration using the configuration file ... 47
9.2.2 Section [IP] ... 48
9.2.3 Section [HOSTNAME] ... 50
9.2.4 Section [PORTS] ... 50
9.2.5 Section [FILES] ... 51
9.3 Non-volatile configuration settings via registers ... 51
9.3.1 Setting a static route ... 52
9.4 Setting the IP address automatically via USB flash drive ... 54
9.5 Setting the IP address during runtime ... 55
9.6 Setting a route during runtime ... 56
9.7 Changing the IP address using the JetIPScan command line tool ... 56
9.8 Operating system ... 58
9.8.1 Operating system update of the controller ... 58

9.8.2 OS update of the Windows IPC component ... 60
9.9 File System ... 61
9.9.1 Properties ... 61
10 Initial commissioning ... 63
11 Programming ... 66
11.1 Abbreviations, module register properties and formats ... 66
11.2 Storage options - Overview ... 67
11.2.1 Operating system memory ... 67
11.2.2 File system memory ... 67
11.2.3 Application Program Memory ... 67
11.2.4 Memory for volatile application program variables ... 68
11.2.5 Memory for non-volatile application program registers ... 69
11.2.6 Memory for non-volatile application program variables ... 70
11.2.7 Special registers ... 71
11.2.8 Registers on I/O modules ... 72
11.2.9 Flag ... 72
11.2.10 Digital inputs and outputs ... 73
11.3 Saving and loading an application program ... 74
11.4 Addressing of registers and I/Os ... 74
11.5 Local JX6-IO module register slot numbering ... 75
11.5.1 Register numbers and I/O numbers of the JX6-IO16CB digital submodule ... 76
11.5.2 Register numbers of the JX6-SV1-ES counter module ... 80
11.6 Addressing expansion modules ... 86
11.6.1 Numbering registers and I/Os ... 87
11.6.2 Expansion modules connected to an Ethernet bus node ... 87
11.6.3 Expansion modules connected to an EtherCAT® bus node ... 89
11.7 System commands ... 91
11.8 Real-time clock ... 91
11.9 Startup delay time ... 92
11.10 System runtime registers ... 92
11.11 Monitoring interface activities ... 92
11.12 Ethernet system bus ... 93
11.13 JCF-SV1 "simple servo function" ... 93
11.14 E-mail feature ... 94
11.15 Modbus/TCP ... 95
11.15.1 Modbus/TCP server ... 95
11.15.2 Modbus/TCP client ... 98
11.16 User-programmable IP interface ... 99

11.17 Motion control 99
11.18 FTP server 101
11.19 HTTP server 102
11.19.1 Device Dashboard 103
11.20 FTP client 105
11.21 Automatic copying of controller data (AutoCopy) 106
11.22 MQTT client 106
11.23 OPC UA server 107
11.24 OPC UA client 107
12 Registers - Overview 108
13 Maintenance 131
13.1 Repairs 131
13.2 Storage and Shipment 131
13.3 Return and Disposal 132
14 Service 133
14.1 Technical support 133
15 Spare parts and accessories 134
15.1 Accessories 134
15.1.1 Ethernet Cable 134
15.1.2 PCI express expansion cards 134
15.1.3 USB flash drives 135

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 | www.bucherautomation.com

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

ⓘ 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 www.bucherautomation.com.

ⓘ Info

EtherCAT®

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

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 JC-975MC Controller is a PC-based high-end controller delivering excellent performance for control tasks.

It is logically divided into a controller unit (hereinafter referred to as JetControl or real-time partition) and an industrial PC (IPC) running a Windows® operating system.

The industrial PC performs non-real-time tasks, such as visualization, storing data such as recipes, or database applications.

The controller unit, on the other hand, handles hard real-time tasks, such as motion control. Motion control allows the user to program axis groups and complex path control setups. Special attention was given to straightforward and efficient realization of complex path control setups.

The controller can be integrated into a network via its Ethernet port.

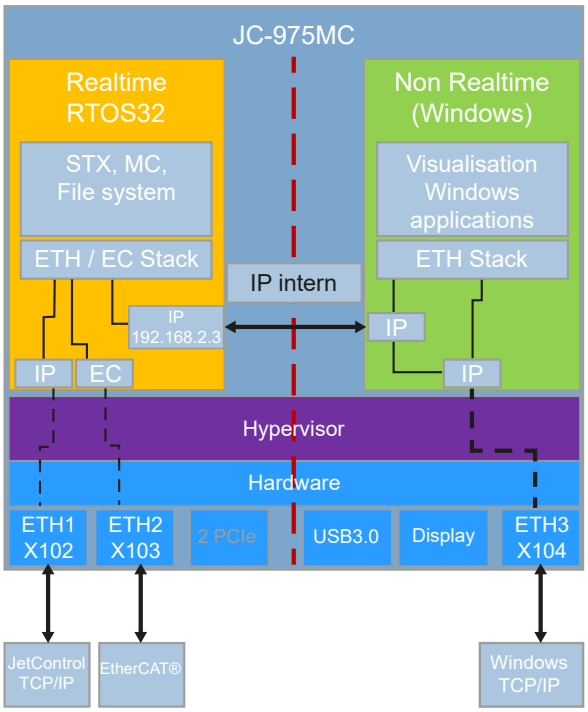


Fig. 1: Internal architecture

3.1 System overview

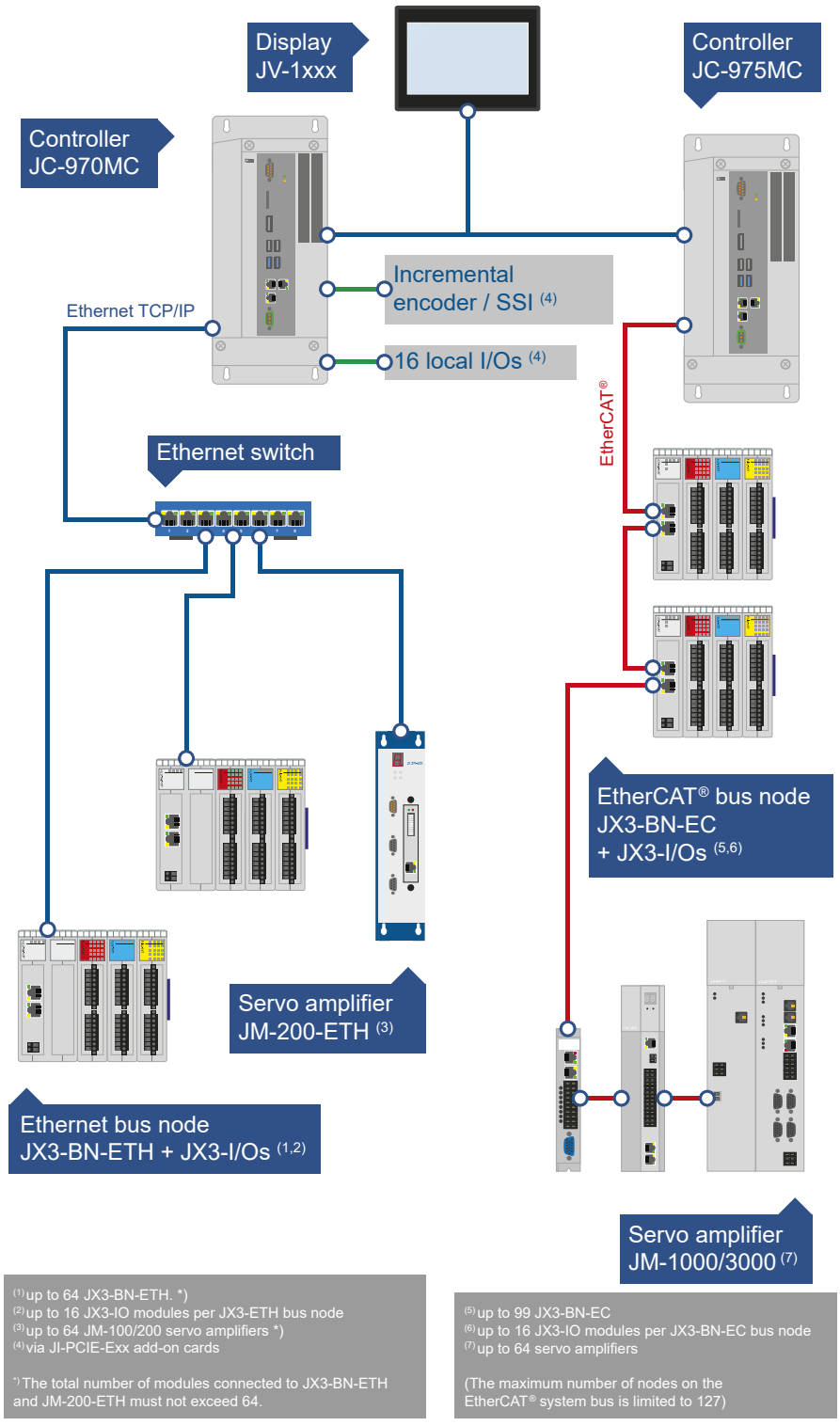


Fig. 2: JC-970MC and JC-975MC system overview

3.2 Design

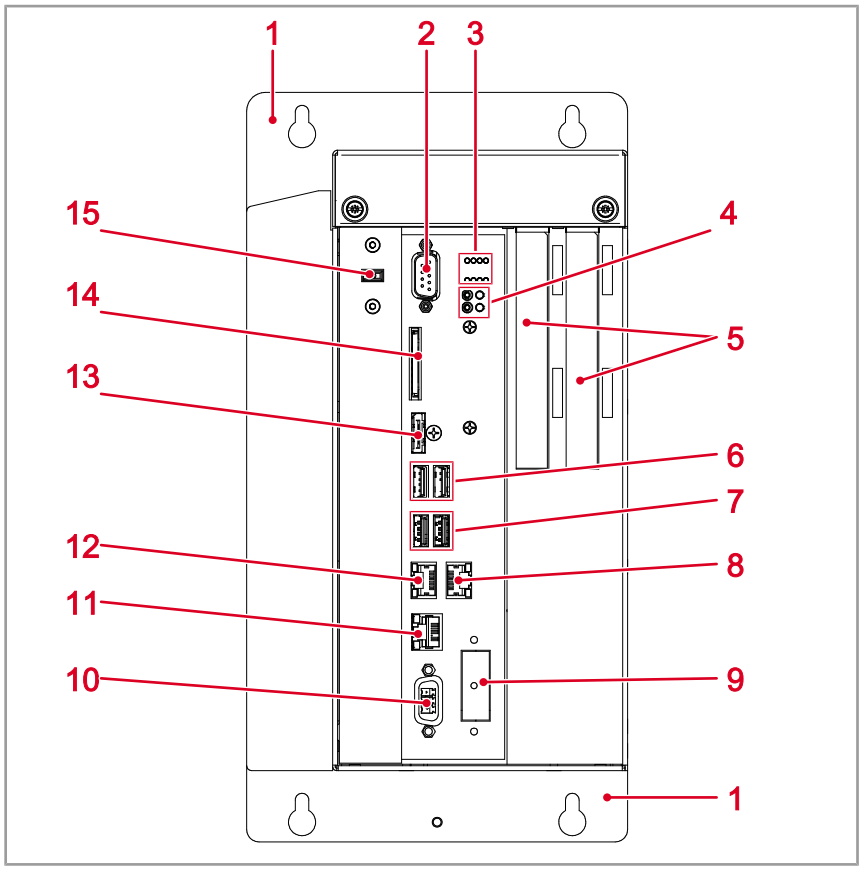


Fig. 3: Front

1	Fastening lugs with 2 keyholes each	
2	X110: RS-232 serial interface. Available only to the Windows OS!	
3	LED status indicators	
4	PWR button and LED RSQ button and LED	
5	X92: n. c.*	
	X91: n. c.*	
	X82:	Model variant -E02: JX6-SV1; SV - Counter module connecting to an incremental or absolute (SSI) rotary encoder
	X81:	Model variant -E02: JX6-IO16CB interface for digital inputs and outputs
6	X107, X108: USB 2.0 ports Accessible only from the controller side!	
7	X105: USB 3.0 port	Bootable USB port (USB 3.0/2.0) Accessible only from the IPC (Windows) side!
	X106: USB 3.0 port	Accessible only from the IPC side!
8	X104 (ETH3): Ethernet port of the Windows OS (= IPC side. Not accessible from the controller side).	
9	X205: n. c.*	

10	X101: Voltage supply (DC 24 V)
11	X102 (ETH1): Ethernet port
12	X103 (ETH2): EtherCAT® port
13	X109: n. c.*
14	SDCARD: Slot for SD memory cards
15	Selector S11: Mode selector
*n. c. = not connected (= without function)	

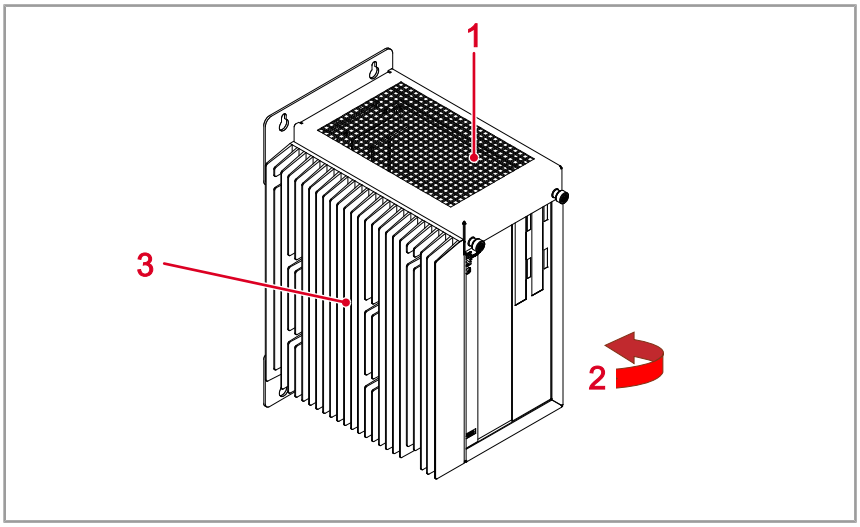


Fig. 4: Housing

1	Cover
2	Nameplate (on the right-hand side of the housing)
3	Cooling fins

3.3 Product features

- Hypervisor technology is used to separate the controller's OS from Windows.
- 64 MC/PtP axes via EtherCAT®
- 2 Ethernet ports (1 x Windows, 1 x RTOS)
- 1 EtherCAT® port
- 4 USB ports (2 x Windows, 2 x RTOS)
- Powerful programming language JetSym STX
- Non-volatile registers (NVRAM): 120,000
- SDRAM: 32 MB (16 MB data storage max.)
- Flash drive: 32 MB (RTOS)
- SSD 60 GB (Windows)
- Real-time clock
- Integrated Web server/email feature
- Modbus/TCP

3.4 Axis engineering guidelines

– The JC-975MC controller connects to a maximum of 64 MC/PtP axes via EtherCAT®.

3.5 Status indication

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

The status LEDs are located in the upper part of the JC-975MC front panel.

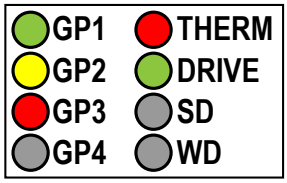


Fig. 5: Status indication

LED	Function	Color	Description
THERM	Temperature	Red, flashing	System is shutting down due to overheating
DRIVE	SSD/HDD drive is busy	Green	Drive is busy
SD	SD memory card	Green	SD card reader activity
WD	Watchdog	Red, flashing	Watchdog timeout
GP1	RUN	Green	Application program is running
		Green, flashing (1 Hz)	Application program is not running
		OFF ---	No power supply or failure
GP2	D1	OFF ---	Normal operating condition
		Yellow	Special states
GP3	ERR	OFF ---	No error
		Red	Error; refer to error register
GP4		n. c.*	
*n. c. = not connected (= without function)			

Tab. 1: Status indication

3.5.1 LED states during the boot process

If the following requirements are met, the controller boots without error:

– There is a valid OS.

– There is a valid application program.

The **GP1** and **GP3** LED flashing patterns indicate the different stages of the boot process.

RUN GP1	D1 GP2	GP3 ERROR	State
Phase 1			**Reset**
OFF	OFF	Red ON	Reset
Phase 2			**Runtime environment**
OFF	Yellow ON	OFF	Initializing the runtime environment of the application program and real-time communication.
Phase 3			**Motion control**
OFF	Yellow ON	OFF	Motion control start
Phase 4			**Additional features**
Green ON	Yellow ON	OFF	Initializing additional functions (Web, Modbus/TCP etc.)
Phase 5			**Normal operating condition**
Green ON	OFF	OFF	Normal condition; the application program is running

Tab. 2: LED states during the boot process

3.5.2 PWR and RSQ LED

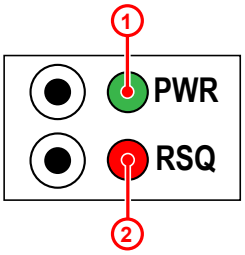


Fig. 6: PWR and RSQ LED

Position	Element	Description
1	PWR LED	– Permanently green for normal operation – Flashes green after shut-down via PWR button
2	RSQ LED	Flashes red if the backup flash has been selected for booting.

3.6 Nameplate

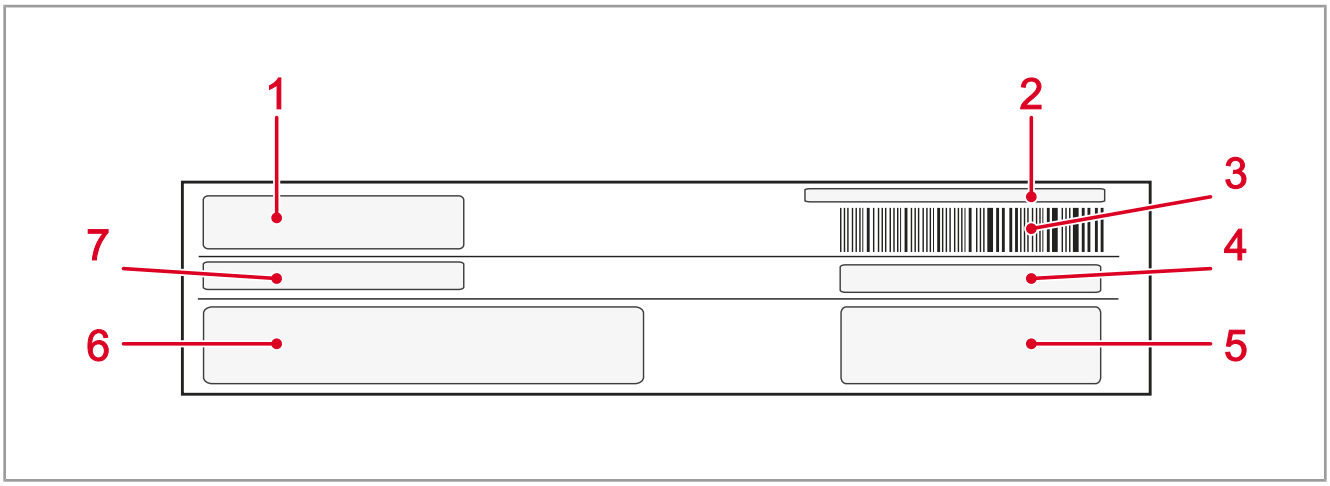


Fig. 7: Example of a nameplate

1	Company logo
2	Serial number
3	Bar code
4	Hardware revision
5	Approvals and certification mark
6	Power supply specifications
7	Model code number

3.7 Scope of Delivery

Scope of delivery

Quantity	Description
1	JC-975MC
1	Power Subcon connector, 3-pin PSC 1.5/3-F
1	Microsoft® software license agreement
1	Installation manual

Ordering options

Item no.	Device designation	Description
10001699	JC-975MC-E03-2	64 MC axes; no JI-PCIE-Exx expansion
10002011	JC-975MC-E02	64 MC axes; equipped with 2x JI-PCIE-Exx expansion (JX6-SV1 + JX6-IO16CB)

4 Technical Specifications

This chapter contains information on both electrical and mechanical data as well as on operating data of the JC-975MC device.

4.1 Dimensions

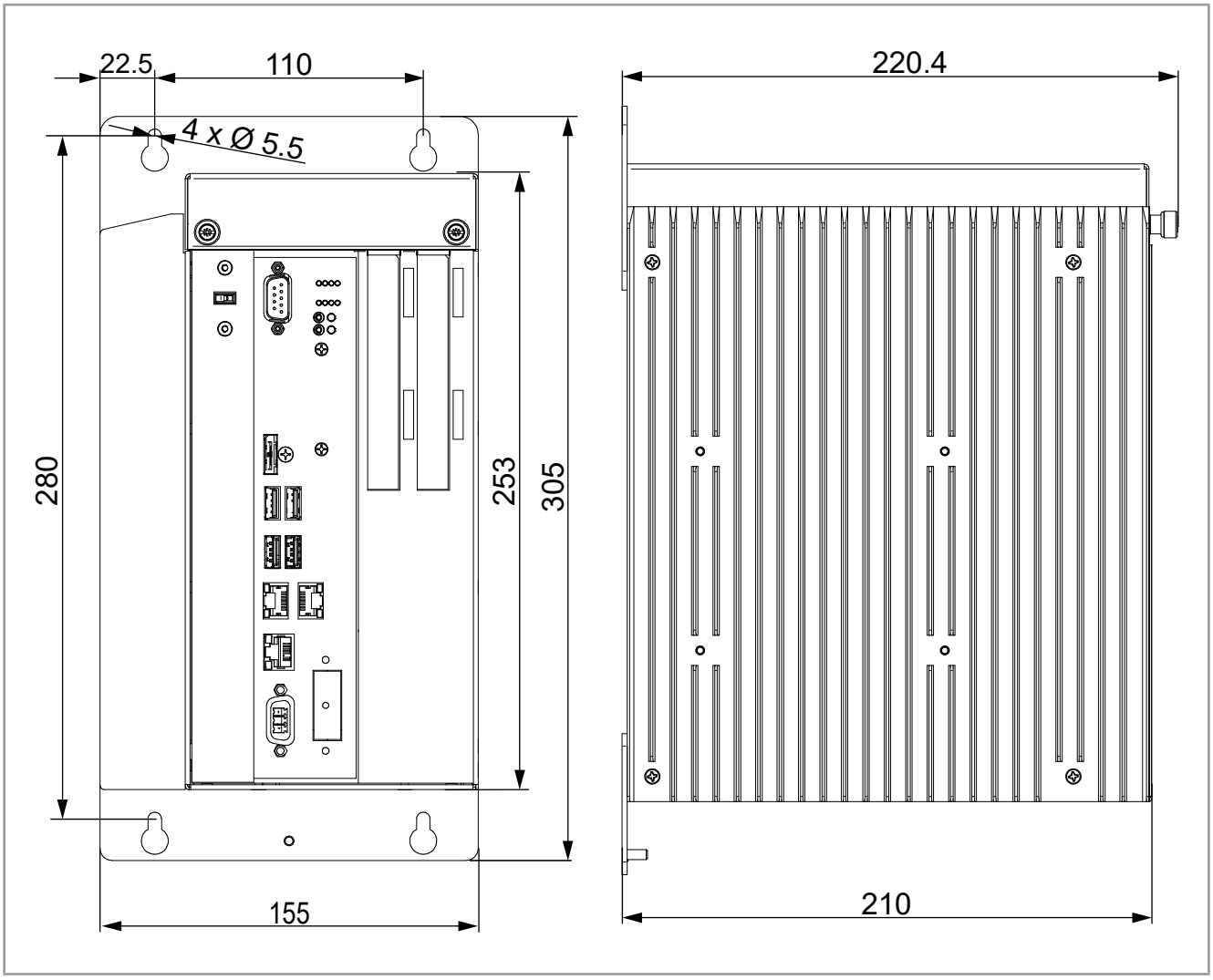


Fig. 8: Dimensions in mm

ⓘ Info

CAD data

For CAD data go to www.bucherautomation.com and navigate to the product page *JC-975MC > Downloads*.

4.2 Mechanical specifications

Parameter	Description	Standards
Mounting orientation	Vertical	
Cooling	Fanless; natural convection	
Weight	4.9 kg	
Vibration resistance		
Frequency sweeps	1 octave/minute, sinusoidal	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 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. 3: Mechanical specifications

4.3 Electrical properties

Power supply (port X101)

Parameter	Description
Supply voltage	DC 24 V
Permissible voltage range	-15 % … +20 %
Input current	6 A max., typically 1.2 A (without USB)
Power consumption	144 W max
Voltage interruption	Duration of voltage interruption ≤ 10 ms Time interval between two voltage dips ≥ 1 s Severity level PS2

Tab. 4: Power supply (port X101)

ETH Ethernet port (X102, X104)

Parameter	Description
Connecting	RJ45 jack
Port count	2
Bit rate	10 Mbit/s, 100 Mbit/s, 1,000 Mbit/s
Auto-crossover	Yes

Tab. 5: ETH Ethernet port (X102, X104)

EtherCAT® port (X103)

Parameter	Description
Connecting	RJ45 jack
Port count	1
Bit rate	100 Mbit/s
Auto-crossover	Yes

Tab. 6: EtherCAT® port (X103)

USB ports

NOTICE



Data loss due to connection of unsuitable devices

The USB ports must NOT be used to charge external devices (e.g. mobile phones), as this will cause the supply power to be insufficient to store non-volatile application data (%V, %RL) when powering the system down or in the case of a voltage drop.

► Only connect USB flash drives to the USB ports.

ⓘ Info

Relocating the USB port

If required, USB ports may be routed to the exterior of the control cabinet, e.g. Using a Weidmüller USB coupling IE-FCM-USB-AB.

To do so, use a Lindy USB cable of 2 m length.

For more information, go to lindy.com/en/.

USB port (X105, X106)

Parameter	Description
Maximum output current	1 A Observe the above note “*Data loss due to connection of unsuitable devices*”
USB type	Type A (host)
Specification	USB 3.0
Maximum permissible cable length	3 m

Tab. 7: USB port (X105, X106)

USB port (X107, X108)

Parameter	Description
Maximum output current	0.5 A Observe the above note “*Data loss due to connection of unsuitable devices*”
USB type	Type A (host)

Parameter	Description
Specification	USB 2.0
Maximum permissible cable length	5 m

Tab. 8: USB port (X107, X108)

USB1 and USB2 port assignment for USB flash drive

– While there are 4 USB ports, the JC-975MC controller is able to process only 2 USB flash drives at a time.
– The port where the first USB flash drive is plugged is automatically designated USB1. When connecting a second USB flash drive, the corresponding port will be designated USB2.

NOTICE



Be careful to observe the correct USB flash drive assignment for the AutoCopy feature [▶ 106].

For the AutoCopy function to work properly, it is mandatory that the USB flash drive be connected to the port specified in the */System/config.ini* file.

► Therefore, ensure that the USB flash drive for the AutoCopy function is the only flash drive connected at the given time.

PCI express slots (X81, X82, X91, X92)

The JC-975MC features 2 PCI express x4 plug-in card slots. These are meant for the JI-PCIe expansion cards which are available separately. PCI express expansion cards [▶ 134].

For a description of the installation of the plug-in cards refer to chapter Installing JI-PCIE-Exx riser cards [▶ 26], for a description of the interface connectivity see chapter PCI express slots X81, X82, X91, X92 [▶ 35].

Parameter	Description
Terminal type	Sub-D socket
Total of pins	15

Tab. 9: Port JX6-SV1-ES

Parameter	Description
Terminal type	Sub-D socket
Total of pins	25
Electrical isolation	Yes
Voltage rating	DC 24 V
Total of inputs	8
Total of outputs	8
Peak current per output	0.5 A

Tab. 10: Port JX6-IO16CB

DisplayPort interface – (X109)

– Connects to an external (digital) display

SD memory card (SDCARD)

Parameter	Description
Plug-in card type	Standard SD memory card
Mechanical operation	push/push
Maximum memory size	32 MB ... 32 GB
Drive format	FAT
Anti-misplug protection	Yes

Tab. 11: SD memory card specification

RS-232 interface (X110)

Parameter	Description	
Baud rates	Up to 115,200 Bd	8N1
This interface is available only to Windows!		

Tab. 12: RS-232 interface specification

Electrical safety

Parameter	Description	Standard
Class of protection	III	DIN EN 61131-2
Dielectric test voltage	Functional ground is internally connected to chassis ground.	
Overvoltage category	II	

Tab. 13: Electrical safety

4.4 Real-time clock

Parameter	Description	
Deviation	Maximum	1 min per month
Power reserve (if the controller has been running for at least 1 hour.)	Minimum	1 week
	Typically	3 weeks

Tab. 14: Real-time clock

4.5 Environmental conditions

Parameter	Description	Standards
Operating temperature	0 … +50 °C	DIN EN 61131-2
Storage temperature	-40 … +70 °C	DIN EN 60068-2-1
Humidity	5 ... 93 %, non-condensing	IEC 60068-2-78
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, alcaline 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. 15: Environmental conditions

4.6 EMI values

4.6.1 Housing

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. 16: 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 … 1 GHz	DIN EN 61131-2 DIN EN 61000-6-2 DIN EN 61000-4-3
	1.4 … 2.0 GHz 3 V/m	
	2.0 … 2.7 GHz 1 V/m	
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. 17: Immunity to interference

4.6.2 Shielded data and I/O lines

Immunity to interference

Parameter	Values	Standards
RF field, asymmetric, amplitude modulated		

Parameter	Values	Standards
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	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. 18: Immunity of shielded data and I/O lines

4.6.3 DC power supply inputs and outputs

Immunity to interference

Parameter	Values	Standards
Radio frequency, asymmetric, amplitude-modulated		
Frequency band	0.15 … 80 MHz	DIN EN 61131-2 DIN EN 61000-6-2 DIN EN 61000-4-6
Test voltage	10 V	
	AM 80 % with 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		
Differential-mode coupling	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 coupling	tr/th 1.2/50 µs	DIN EN 61131-2 DIN EN 61000-6-2 DIN EN 61000-4-5
	1 kV	

Tab. 19: DC power supply inputs and outputs

5 Mechanical installation

This chapter describes how to install and remove the JC-975MC.

⚠ WARNING



Non-observance of safety instructions may result in injuries or physical damage

- ► Prior to assembling and installing the device, read and follow the safety precautions and restrictions of use as set out in the Safety chapter.
- ► Only authorized qualified personnel is allowed to perform any kind work on the device.

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 the controller

To install the JC-975MC, proceed as follows:

1. Place the JC-975MC vertically on the mounting plate of the control cabinet.
2. Ensure that the clearance above and below the controller is 100 mm each. On the side of the heat sink, a minimum clearance of 50 mm is required.
3. On the mounting plate, mark the positions of the 4 fastening bores. For the dimensions, refer to the Dimensions [▸ 17] illustration.
4. Drill the holes and tap the threads.
5. Tighten the fastening bolts down halfway.
6. To hang the controller, place the keyholes in the rear panel over the fastening bolts.
7. Tighten the bolts down completely.

5.2 Removing the controller

To remove the JC-975MC, proceed as follows:

1. De-energize the JC-975MC.
2. Loosen the 4 fastening bolts halfway.
3. Slightly lift the JC-975MC and remove it from the mounting bolts.

5.3 Installing JI-PCIE-Exx riser cards

The JC-975MC controller features 2 PCI Express (PCIe) bus slots providing for additional expansion options. Each PCIe bus slot can accommodate 1 JI-PCIE-Exx riser card.

Pre-assembled PCIe cards are available as Accessory [▶ 134].

NOTICE

Electrostatic discharge can damage or destroy a device's electronic or microelectronic components. Therefore, observe the following when installing JI-PCIE-Exx cards:

- ▶ Be careful to work at an ESD-safe station.
- ▶ At all times, wear ESD-safe clothing, anti-static safety shoes and an anti-static wrist band.
- ▶ Do not remove an JI-PCIE-Exx card from the antistatic packaging until you are ready to install it.
- ▶ At all times, hold PCBs by the edges only. Never touch a PCB's tin-plated terminals and components.
- ▶ If you must put down the device, place it on the anti-static packaging.

Addressing JI-PCIE-Exx cards

The address of the JI-PCIE-Exx card depends on the number of the PCI express bus slot where the JI-PCIE-Exx card is plugged.

To set the card's address, use the rotary switch on the card. Observe the following table and the information provided under "*Inserting the JI-PCIE-Exx riser card(s) and setting the address*".

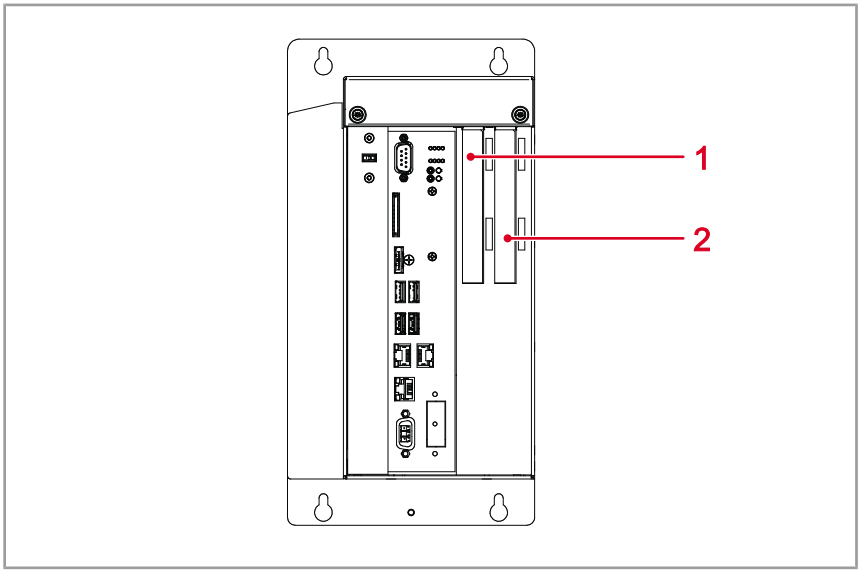


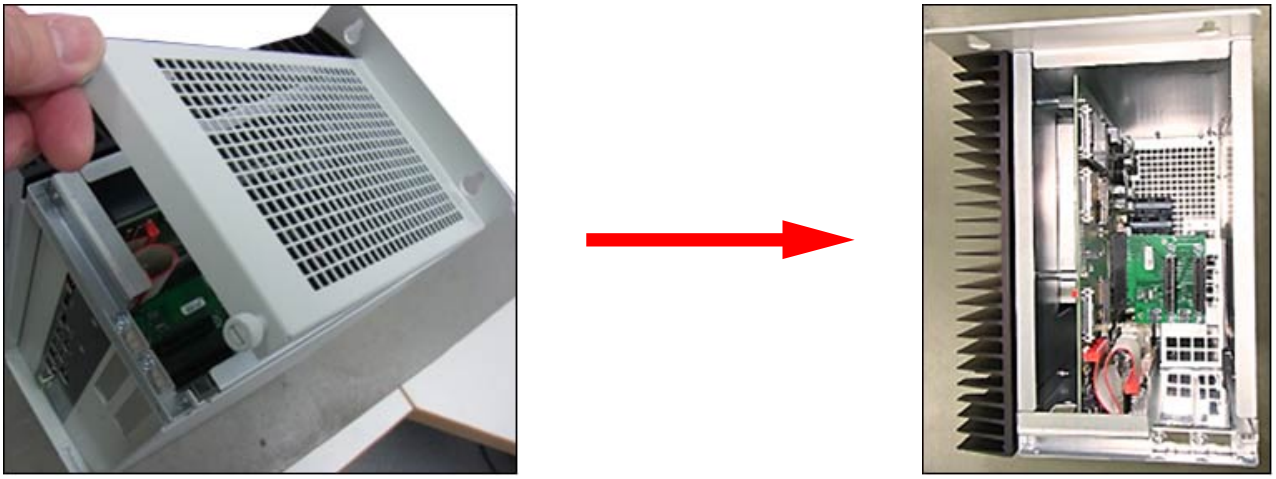
Fig. 9: PCI express bus slots

Number	PCI express bus slot number	Rotary switch position
1	1	0
2	2	1

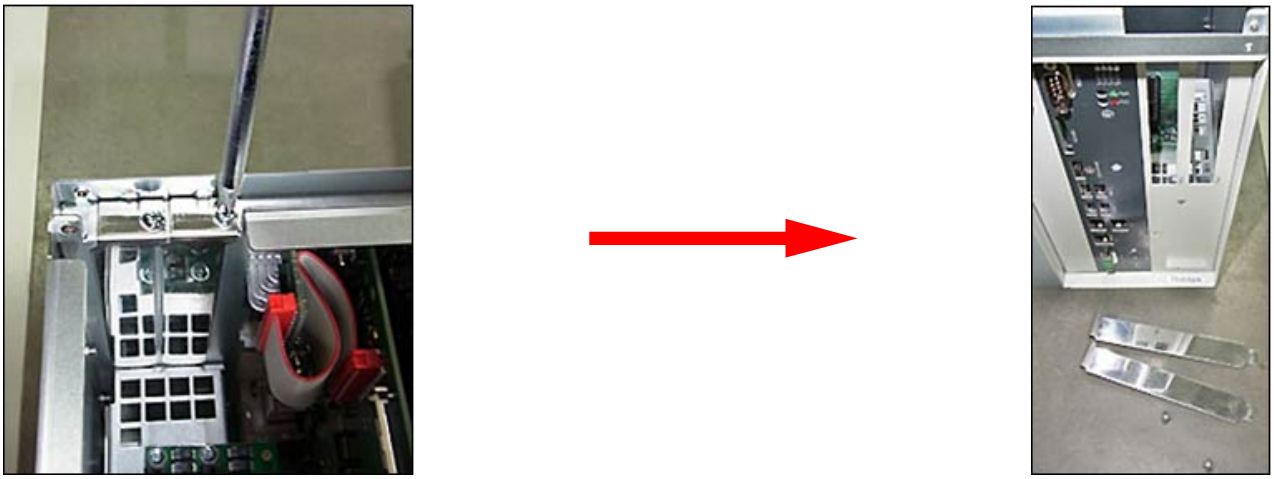
Tab. 20: Rotary switch position

Installation

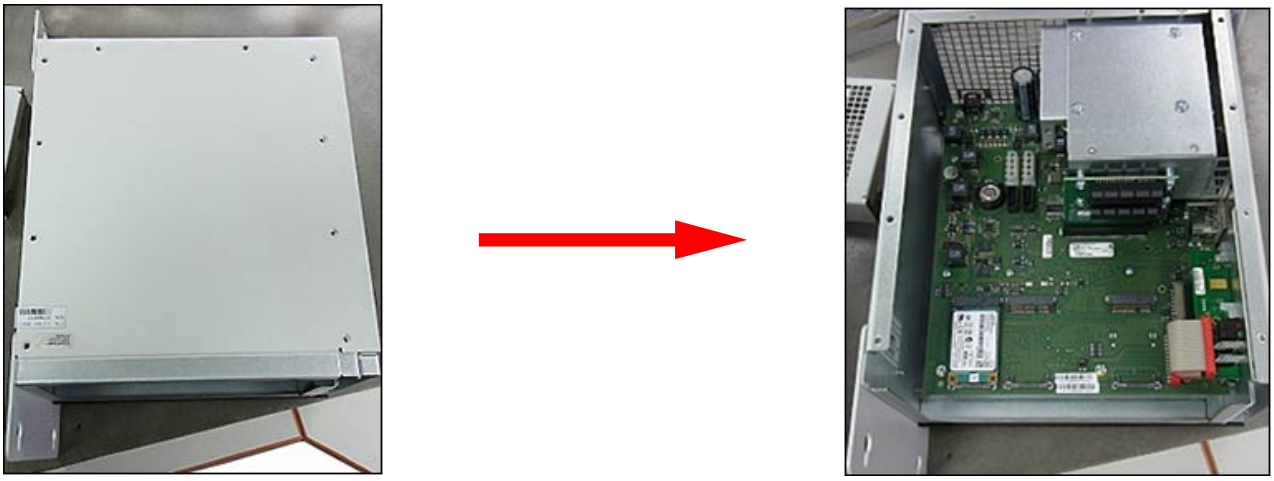
To plug a JI-PCIE-Exx card into a JC-975MC PCI bus slot, proceed as follows:



1. Unscrew the lid and put it aside.

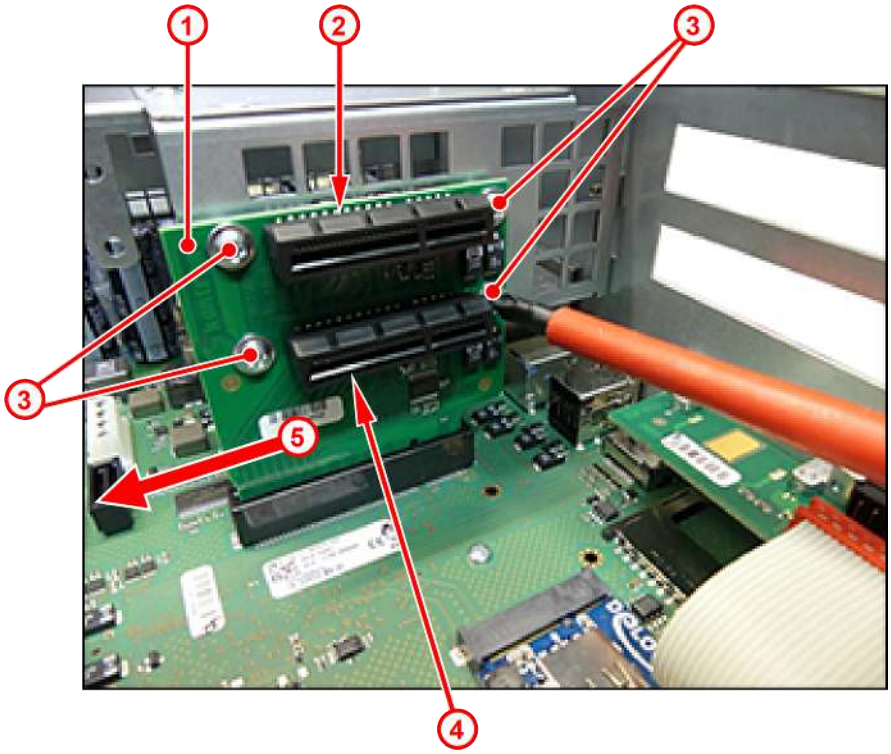


2. Remove the slot brackets.



3. Place the JC-975MC controller on the table with the cooling fins facing down and remove the side cover.

Adjusting the adapter card

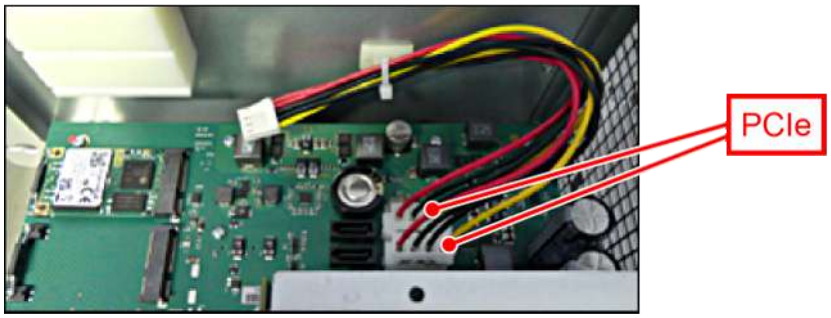


1. Remove the 4 fastening bolts (3).
2. Insert the adapter card (1) in the direction of the arrow (5) as far as it will go and hold it in place.
3. Tighten the fastening bolts (3) down.

Installing the JI-PCIE-Exx riser card(s)

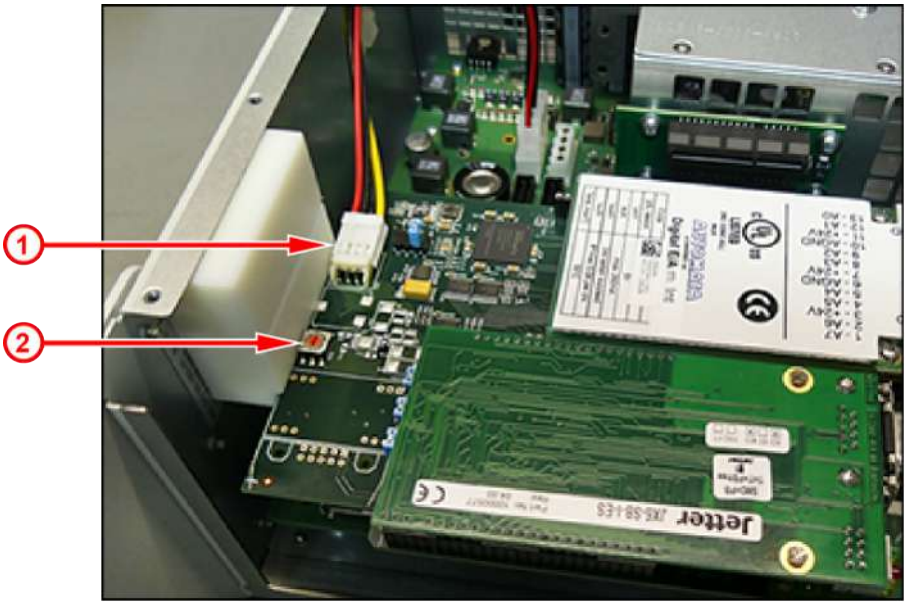
► Install the JI-PCIE-Exx riser card(s) from the bottom up.

Connecting the riser card(s)



1. Connect the JI-PCIE-Exx riser card(s).
2. If necessary, remove the cable tie and re-apply later.

Inserting the JI-PCIE-Exx riser card(s) and setting the address



1. Plug the card into the corresponding slot (1).
2. Set the position of the rotary switch (2) on the JI-PCIE-Exx card (see rotary switch position at the beginning of this chapter).
3. Re-install the side cover.
4. Screw the lid down tightly.

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 noise immunity

A system's immunity to noise is determined by its weakest component. Key factors are correct connections, lines and proper shielding. Observe the measures set out in this chapter.

ⓘ 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 www.bucherautomation.com.

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 a 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 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.

6.2 Jack X101 – power supply

Ports and interfaces

The following devices connect to port X101:

– Power supply for the JC-975MC controller

Pinning

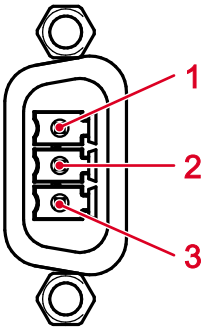


Fig. 10: Power supply, port X101

Pin	Description
1	DC 24 V supply voltage
2	Functional Earth
3	Reference potential

6.3 Jack X102 – Ethernet

Ports and interfaces

The following devices connect to port X102:

– Programming PCs (JET-IP)

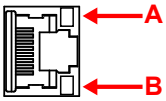
– Modbus/TCP (server, client)

– User-programmable TCP/IP devices (IP-PRIM)

– Devices featuring JetSync/PubSub

– Devices featuring NetConsistency-based protocols, e.g. JX3-BN-ETH and JX3-COM-xxx

Pinning



Socket	LED	Color	Description
X102	A SPEED		Shows the connection speed
		OFF	10 Mbit/s
		Green	100 Mbit/s
		Yellow	1000 Mbit/s
	B ACT/LINK	Green	Connected to the network
		Green, flashing	Data transfer

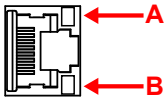
6.4 Jack X103 – EtherCAT®

Ports and interfaces

The following devices connect to port X103:

– One EtherCAT® slave device with 100 Mbit EtherCAT® port

Pinning



Socket	Signal	LED	Color	Description
X103	BUS OUT	A SPEED		Shows the connection speed
			OFF	10 Mbit/s
			Green	100 Mbit/s
		B ACT/ LINK	Green	Connected to the network
			Green, flashing	Data transfer

6.5 Jack X104 – Ethernet

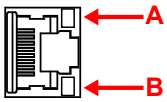
Ports and interfaces

The following devices connect to port X104:

– the integrated Windows IPC to an Ethernet network

NOTICE! The X104 port is available only from the Windows operating system of the IPC.

Pinning



Socket	LED	Color	Description
X104	A SPEED		Shows the connection speed
		OFF	10 Mbit/s
		Green	100 Mbit/s
		Yellow	1000 Mbit/s
	B ACT/LINK	Green	Connected to the network
		Green, flash- ing	Data transfer

6.6 Ports X105 and X106 – USB

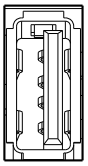
NOTICE

Data loss due to connection of unsuitable devices

The USB ports must NOT be used to charge external devices (e.g. mobile phones), as this will cause the supply power to be insufficient to store non-volatile application data (%V, %RL) when powering the system down or in the case of a voltage drop.

► Only connect USB flash drives to the USB ports.

Ports and interfaces



Ports X105 and X106 connect to the following devices:

– Standard USB drives
  – Ports X105 and X106: USB 3.0

NOTICE! Ports X105 and X106 are only available from the Windows operating system of the IPC.

6.7 Ports X107 and X108 – USB

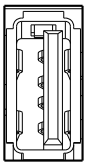
NOTICE

Data loss due to connection of unsuitable devices

The USB ports must NOT be used to charge external devices (e.g. mobile phones), as this will cause the supply power to be insufficient to store non-volatile application data (%V, %RL) when powering the system down or in the case of a voltage drop.

► Only connect USB flash drives to the USB ports.

Ports and interfaces



Ports X107 and X108 connect to the following devices:

– Standard USB flash drives
– Ports X107 and X108: USB 2.0

The controller actively uses data stored on the USB flash drive serving as a file system extension.

The operating system supports a maximum of 2 flash drives, /USB1 and /USB2, at a time. See Electrical properties [▸ 20]

6.8 Port X109 - DisplayPort

Interface

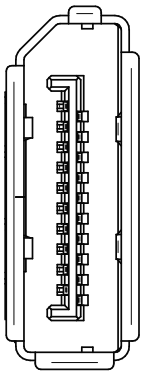


Fig. 11: DisplayPort interface

The following devices connect to port X109:

– External (digital) display

NOTICE! Ports X109 – DisplayPort is only available from the Windows operating system of the IPC.

ⓘ Info

Note on EMC

If the connected display screen is subject to image interference (stripes in the image), the display is exposed to increased EMC noise. We recommend the use of high-quality cables and ferrite cores, which can also be retrofitted into the display cable.

Ferrite cores for retrofitting are available separately (item number **60240700**).

ⓘ Info

Connecting cable

We recommend using DisplayPort cables from Lindy with a total length of 2 m.

For more information, go to lindy.com/en/.

6.9 The SDCARD interface



Fig. 12: SD memory card

– Slot for SD/SDHC/SDXC memory cards.

– The **SD** LED indicates card reader activity.

– This port supports hot-plugging of SD memory cards and system booting.

NOTICE! SD memory cards are accessible only from the Windows OS.

NOTICE

Loss of data!

Removing the SD memory card while it is active causes data loss.

► Do not remove the SD memory card unless the **SD** LED has turned off.

6.10 Port X110 – serial interface RS-232

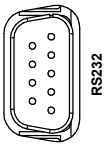


Fig. 13: Serial interface RS-232

NOTICE! The serial interface is available only to the Windows OS.

6.11 PCI express slots X81, X82, X91, X92

The JC-975MC features 2 PCI express slots

allowing for installation of 2 JI-PCIE-Exx type expansion cards (see PCI express expansion cards [▶ 134]). These, in turn, provide interfaces to a variety of peripherals.

The JC-975MC-E02 comes with the following:

Slot	Peripheral	Description
X81	JX6-IO16CB	I/O module featuring 8 digital inputs and 8 digital outputs
X82	JX6–SV1	I/O module featuring a 5 V/24 V dual-channel counter input or SSI encoder input

Tab. 21: Slots interfacing with peripherals

6.11.1 JX2 system bus interface

The JC-975MC does no longer support the JX2 system bus interface. Please use EtherCAT® to connect distributed peripheral devices.

See also

PCI express expansion cards [▶ 134]

6.11.2 JX6-IO submodules for local I/Os

This chapter describes JX6-IO submodules connecting directly to the JC-975MC controller's PCIe bus. As they are installed in the controller, they are called local submodules.

Maximum number of local JX6-IO submodules

Up to 2 riser cards (JI-PCIE-Exx) can be plugged into the controller's PCIe bus. Each riser card accommodates up to 2 local JX6-IO submodules. Thus, the controller can be equipped with a maximum of 4 JX6 submodules.

The riser card converts the controller-internal PCIe bus to the local JX6-IO submodules.

Supported Modules

JI-PCIE-Exx riser cards support the following JX6-IO submodules:

– JX6-SV1: Counter module for connecting an incremental or absolute (SSI) rotary encoder

– JX6-IO16CB: Digital 24 V inputs/outputs

Possible combinations

The JI-PCIE-Exx riser card accommodates 1 or 2 JX6-IO submodules. However, mechanical constraints allow for only certain combinations.

For a summary of the PCI express expansion cards, see PCI express expansion cards [▶ 134].

JX6-SV1-ES features

The Sub-D connector of the JX6-SV1 I/O submodule interfaces with the following I/O functions:

– 1 incremental encoder input 5 V differential or

– 1 incremental encoder input 24 V or

– 1 absolute encoder input SSI

Sub-D connector pinout – JX6-SV1-ES

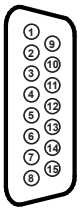


Fig. 14: Sub-D connector

Pin	Incremental encoder signal 5 V, differential	Incremental encoder signal 24 V	Absolute encoder signal SSI
1	GND	GND	GND
2	K0+	K0	n. c.
3	K0-	n. c.	n. c.
4	K1+	K1	Data+
5	K1-	n. c.	Data-
6	K2+	K2	n. c.
7	K2-	n. c.	n. c.
8	n. c.	n. c.	Clock-
9	n. c.	n. c.	Clock+
10	DC 5 V (50 mA)	n. c.	n. c.

Tab. 22: JX6-SV1-ES pin assignment

JX6-IO16CB interfaces

The Sub-D connector of the JX6-IO16CB I/O module is the interface to digital inputs and outputs.

JX6-IO16CB sub-D connector pinout

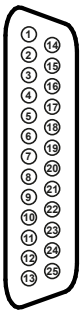


Fig. 15: Sub-D connector

Pin	Signal	Description
1	Output 8	
2	Output 7	
3	DC 24 V	Power supply of the outputs
4	Output 6	
5	Output 5	
6	GND_A	Reference potential of the outputs
7	DC 24 V	Power supply of the outputs
8	Output 4	
9	Output 3	
10	GND_A	Reference potential of the outputs
11	DC 24 V	Power supply of the outputs
12	Output 2	
13	Output 1	
14	Input 8	
15	GND_E	Reference potential of the inputs
16	Input 7	
17	Input 6	
18	Input 5	
19	DC 24 V	Power supply of the outputs
20	Input 4	
21	Input 3	
22	Input 2	
23	Input 1	
24	GND_A	Reference potential of the outputs
25	n. c.	

Tab. 23: JX6-IO16CB pin assignment

7 Control elements

The JC-975MC controller features the following control elements:

– Mode selector S11
– PWR button for switching the controller OFF
– RSQ button for booting the controller from backup flash

7.1 Mode selector S11

The position of mode selector S11 is identified while the controller is booting up. The post-booting behavior of the controller depends on the mode selector position. Changing the mode selector position during operation has no effect on the operating mode.

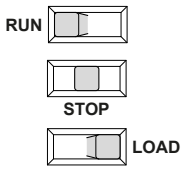


Fig. 16: S11 mode selector positions

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 – After completion of the AutoCopy function, the controller must be restarted.

Tab. 24: S11 mode selector positions and operating mode

Mode selector functions

The JC-975MC identifies the position of mode selector S11 **only** during start-up proceeding as follows:

Step	Description	
1	Power supply of the controller is at port X101.	
2	The boot loader of the controller checks the position of mode selector S11.	
	If ...	**... then ...**
	... mode selector S11 = *RUN* or *STOP*,	... the OS boots; --> proceed with step 3.
	... mode selector S11 = *LOAD*,	... the controller starts the AutoCopy function
3	The controller checks the position of selector S11.	
	If ...	**... then ...**
	... mode selector S11 = *RUN*,	... the application program starts.
	... mode selector S11 = *STOP*,	... the application program does not start.
4	**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.

7.2 PWR button



Fig. 17: PWR button

Element	Function	Description
PWR button	ON/OFF switch	– PWR = POWER

Function of the pushbutton

- The PWR button is an OFF switch.
- Pressing the PWR button briefly has no effect.
- As soon as power is supplied to port X101 the device boots automatically and starts running the functions corresponding to the mode selector position (see Mode selector S11 [▶ 38]).
- Pressing the PWR button for more than **4 s** forces the controller/device to shut down.
- Any variables declared to be non-volatile (%VL, %RL) in the application program are saved.
- The PWR LED flashes green after the device has shut down.

NOTICE! The device is not de-energized!

7.3 RSQ button

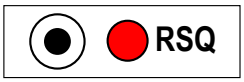


Fig. 18: RSQ button

Element	Function	Description
RSQ button	Boots the system	– RSQ = RESCUE – In the case of a corrupted BIOS flash, it boots the system to a defined and safe state to allow for troubleshooting.

Function of the pushbutton

If the system does not properly boot or gets hung-up and cold booting it does not remedy the issue, the RSQ function allows for the system to be booted from the backup flash.

- Pressing the RSQ button causes the system to boot using the BIOS (UEFI) from the backup flash.
- The RSQ LED flashes red if the backup flash was selected for booting.

► To this end, press the RSQ button for more than 5 seconds.

⇨ Whether or not the system is running, it will re-boot using the backup flash.

✓ To make the system use the default boot flash agian, perform a cold boot.

► To this end, press the PWR button or disconnect and reconnect the power supply voltage.

NOTICE

Rescue function

The rescue function is not intended for use within an application.
It is intended to force the controller in the case of a corrupted BIOS flash to boot into a defined and safe state to allow for troubleshooting.

8 Identification

This chapter describes how to identify the JC-975MC 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

8.1 Electronic Data Sheet (EDS)

Each JC-975MC 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.

8.1.1 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.
– This file is read-only.
– Formatting the flash drive 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-975MC:

```
;JC-975MC Electronic Data Sheet

[IDENTIFICATION]
Version = 2
Code = 2400
Name = JC-975MC
PcbRev = 00
PcbOpt = 00
OSVersionMin = 0.0.0.0
BLVersionMin = 0.0.0.0

[PRODUCTION]
Version = 0
SerNum = 20230325070900
Day = 25
Month = 03
Year = 2023
TestNum = -1
```

```
TestRev = 255.255.255.255

[FEATURES]
Version = 8
STX = 1
NVRegs = 120000
JCF-SV1 = 16
USB = 2
```

[IDENTIFICATION] section

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

Name	Example	Function
Version	2	Version of this section
Code	2400	Module code for JC-975MC
Name	JC-975MC	Corresponds to the information on the nameplate
PcbRev	00	PCB revision
PcbOpt	00	PCB option
OSVersionMin	0.0.0.0	Minimum required OS version
BLVersionMin	0.0.0.0	Minimum required bootloader version

Tab. 25: [IDENTIFICATION] section

[PRODUCTION] section

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

Name	Example	Function
Version	0	Version of this section
SerNum	20230325070900	Corresponds to the information on the nameplate
Day	25	Production date: Day
Month	3	Production date: Month
Year	2023	Production date: Year
TestNum	7	Internal usage
TestRev	255,255,255,255	Internal usage

Tab. 26: [PRODUCTION] section

[FEATURES]
section

In the [FEATURES] section, special properties of the controller are specified. The OS of the controller interprets properties not specified here as non-existing.

Name	Example	Function
Version	8	Version of this section
STX	1	Runtime environment for the application program is available
NVRegs	120000	Total of non-volatile registers
JCF-SV1	16	Axis count for simple SV function
USB	2	Total of available USB ports

Tab. 27: [FEATURES] section

8.1.2 EDS registers

Use the EDS registers to retrieve Electronic Data Sheet (EDS) information. 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-975MC	100000	100500 ... 100827

Tab. 28: EDS register numbers

Register 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**. This register array displays the EDS of the controller. EDS data can be retrieved from MR 600 onwards. Enter 0 in MR 500.

Registers	Section in eds.ini	Name in eds.ini	Description
MR 500	-	-	Functional group 0: CPU
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

Registers	Section in eds.ini	Name in eds.ini	Description
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
MR 800	FEATURES	Version	Version of this section
MR 805		STX	Runtime environment for the application program
MR 806		NVRegs	Total of non-volatile registers
MR 825		USB	Total of available USB ports
MR 829		JCF-SV1	Axis count for simple SV function

Tab. 29: EDS register summary

8.2 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.

8.2.1 Hardware revisions

The device features special registers allowing for hardware identification.

Register overview

Use the registers listed below to retrieve the hardware revisions:

Registers	Description
200170	Controller model
100613	PCB revision
100614	PCB options

Tab. 30: Overview of hardware revision registers

8.2.2 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	Represents major or main version number
2	Represents minor or secondary version number
3	Represents branch or intermediate version number
4	Represents build version number

Tab. 31: Software version number format

Register overview

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

Registers	Description
200169	Operating system version (decimal in IP address notation, e.g. 1.20.0.00)
210001	Version of the execution unit (JetVM) for the STX application program
59997901	MCX version

Tab. 32: Register overview

9 Configuration

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

- IP configuration
  - IP address of the controller
  - Subnet mask
  - IP address of the default gateway
  - Static route of the IP address
  - Static route of the subnet mask
  - Static route of the gateway
  - IP address of the DNS server
- Controller name
- Port configuration
  - IP port number for the JetSym debugger
  - Basic port number for communication via JetIP
- Name of the AutoCopy command file

ⓘ Info

Bucher Automation Ethernet system bus

ETH1 (X102) is the only interface to support the Bucher Automation Ethernet system bus (JetIPScan, JetSync, Publish/Subscribe), and thus synchronous transfer of data packets.

9.1 Default values

The JC-975MC controller is shipped with a default setting of various parameters. These parameters are customizable.

Passwords

admin user password: admin *

system user password: system *

* Changing the passwords requires admin access to the /system/users.ini file.

- The ETH2/X103 EtherCAT® interface is configured by the EtherCAT® master and not customizable.
- The ETH3/X104 Ethernet interface is assigned to Windows and requires configuration via the windows system settings.

Default values

Parameter	Factory settings	Fall-back value
ETH1 / X102: IP address	192.168.1.1	192.168.10.150
ETH1 / X102: Subnet mask	255.255.255.0	255.255.255.0
ETH1 / X102: IP address of the default gateway	0.0.0.0	0.0.0.0
IP address of the DNS server	0.0.0.0	0.0.0.0

Parameter	Factory settings	Fall-back value
ETH1 / X102: Static route of the IP address	0.0.0.0	0.0.0.0
ETH1 / X102: Static route of the subnet mask	0.0.0.0	0.0.0.0
ETH1 / X102: Static route of the gateway	0.0.0.0	0.0.0.0
Controller name	JC-975MC	JC-975MC
Suffix type of the name	0	0
Debugger IP port number (JVMDebug, XCOM protocol)	52000	52000
JetIP IP port number (JetIPBase, PCOM protocol)	50000	50000
Name of AutoCopy command file	/USB1/autocopy.ini	/USB1/autocopy.ini

Tab. 33: Default values

9.2 Configuration file "config.ini"

Properties

– The file is accessible through the JC-975MC file system.
– With an FTP connection, the user must have administrator or system rights.
– This file is located in the **System** folder.
– You cannot delete the file, but overwrite it.
– Formatting the flash drive leaves the file unchanged.

File structure

The configuration file is a text file and its entries grouped into several sections. The JC-975MC will use the factory settings to replace any missing IP configuration parameters.

Example

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

```
;System Configuration

[IP]
;ETH1 X102
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-975MC

[PORTS]
JetIPBase = 50000
JVMDebug = 52000

[FILES]
AutoCopyIni = /USB1/autocopy.ini
```

9.2.1 Changing the configuration using the configuration file

The **config.ini** file allows the user to make adjustments to the configuration.
To do so, proceed as follows:

1. Connect the PC and the JC-975MC controller via FTP(S).
2. Login as a user with administrator or system rights.
   Default login information:
   User: admin; password: admin
   User: system; password: system
3. Navigate to */System* folder of the JC-975MC.
4. Copy the **config.ini** file to your PC.
5. Adjust the settings as necessary and save the changes.
6. Copy the modified **config.ini** file back to the */System* folder of the JC-975MC.
7. Disconnect the FTP session.
8. Reboot the controller.

⇨ The new configuration is active.

Alternatively, use the configuration registers (see Non-volatile configuration settings via registers [▶ 51]) to make adjustments to the IP configuration.

9.2.2 Section [IP]

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

Address

Property	Description	
In the given example	192.168.50.1	
Function	IP address	
Valid values	> 1.0.0.0	< 223.255.255.255
Invalid values	Network address, broadcast address	
In the event of an invalid value	All 4 values will be reset to default.	

Tab. 34: Address

SubnetMask

Property	Description
In the given example	255.255.255.0
Function	Specifies the subnet mask
Valid values	≥ 128.0.0.0
Invalid values	1 and 0 mixed
In the event of an invalid value	All 4 values will be reset to default.

Tab. 35: SubnetMask

DefGateWay

Property	Description	
In the given example	192.168.50.11	
Function	IP address of the gateway to other subnets; set to 0.0.0.0, if no other nodes are available via the address/subnet mask. Only one default gateway can be set at a time. If the configuration memory of the controller includes several default gateways, the controller will use the first default gateway with a value unequal to 0.0.0.0 it has detected. The controller proceeds in the order of the ports.	
Valid values	≥ 0.0.0.0	< 223.255.255.255
Invalid values	– Network address – Broadcast address – If no other nodes are available via the address/subnet mask – The address value	
In the event of an invalid value	Will be set to 0.0.0.0.	

Tab. 36: DefGateWay

RouteIP1

Property	Description	
In the given example	0.0.0.0	
Function	IP address of the static route	
Valid values	> 1.0.0.0	< 223.255.255.255
Invalid values	Network address, broadcast address	
In the event of an invalid value	All 4 values will be reset to fallback.	

Tab. 37: RouteIP

RouteMask1

Property	Description
In the given example	0.0.0.0
Function	Subnet mask of the static route
Valid values	≥ 128.0.0.0
Invalid values	1 and 0 mixed
In the event of an invalid value	All 4 values will be reset to fallback.

Tab. 38: RouteMask

RouteGateway1

Property	Description	
In the given example	0.0.0.0	
Function	IP address of the gateway to other subnets; mask of the static route	
Valid values	≥ 0.0.0.0	< 223.255.255.255
Invalid values	– Network address – Broadcast address – If no other nodes are available via the address/subnet mask – The address value	
In the event of an invalid value	All 4 values will be reset to fallback.	

Tab. 39: RouteGateway

DNSServer

Property	Description	
In the given example	192.168.1.44	
Function	IP address of the server for the DNS (domain name system)	
Valid values	≥ 0.0.0.0	< 223.255.255.255
In the event of an invalid value	Will be set to 0.0.0.0.	

Tab. 40: DNSServer

9.2.3 Section [HOSTNAME]

The [HOSTNAME] section specifies the name of the JC-975MC. If desired, the controller automatically generates an individual name. Currently, the host name is only used for displaying **JetIPScan**.

SuffixType

Property	Description	
In the given example	0	
Feature	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. 41: SuffixType

Name

Property	Description	
In the given example	JC-975MC	
Feature	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-975MC	

Tab. 42: Name

9.2.4 Section [PORTS]

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

JetIPBase

(PCOM)

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

Tab. 43: JetIPBase

JVMDebug

(XCOM)

Property	Description
In the given example	52000
Feature	IP port for debugger/setup in JetSym

Property	Description
Allowed values	1024 ... 65535
In the event of an illegal value	52000

Tab. 44: JVMDebug

9.2.5 Section [FILES]

The [FILES] section specifies the name and path of the command file for the AutoCopy function.

AutoCopyIni

Property	Description
In the given example	/USB1/autocopy.ini
Feature	Command file for the AutoCopy function
Allowed values	Allowed path and file name
In the event of an illegal value	/USB1/autocopy.ini

Tab. 45: AutoCopyIni

9.3 Non-volatile configuration settings via registers

The ETH1 IP interface is initialized during the boot phase according to the settings in the configuration memory.

The registers allow for adjustments to the following settings. The values will be saved to a non-volatile memory:

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

Register overview

Register (range)	Description
101200	ETH1: IP address
101201	ETH1: Subnet mask
101202	ETH1: IP address of the default gateway
101203	IP address of the DNS server
101216	ETH1: Static route of the IP address
101217	ETH1: Static route of the subnet mask

Register (range)	Description
101218	ETH1: Static route of the gateway
101232	Host name suffix type
101233 … 101251	Host name
101264	Port number of JetIP
101265	Port number of the JetSym debugger
101280 … 101298	Name of AutoCopy command file
101299	Save settings (0x77566152)

Tab. 46: Configuration register overview

ⓘ Info

Note on configuration via registers

Writing the value 0x77566152 to register 101299 will overwrite the configuration file (/System/config.ini). The default formatting used by the controller is irrespective of comments or indentations present in the existing file.

ETH1 setting example

To make the settings of the ETH1 IP addresses, subnet mask, and default gateway permanent, proceed as follows:

1. Write the desired IP address of port ETH1 to register 101200.
2. Write the desired subnet mask value to register 101201.
3. Write the desired IP address of the default gateway to register 101202.
4. Applying the values to the controller requires a password.
   To do so, write the value 2002149714 (0x77566152) to register 101299.
5. Boot the controller.

⇨ The settings are complete. Communication has been re-established.

9.3.1 Setting a static route

Introduction

The operating system of the JC-975MC allows for the Ethernet port ETH (X102) to be configured for simple routing of IP packets.

If the JC-975MC requires access to networks that are not directly accessible to it, the user must set up static routes to these networks. Static routes are a must, in particular, where the default gateway cannot reach the destination network.

Saving routes to the configuration memory

The configuration memory holds entries reserved for Ethernet port ETH (X102). To change these entries, use the non-volatile registers 101216 ... 101218. All entries saved to the configuration memory are activated when rebooting the controller.

Modifying routes during runtime

Using register set 104550 ... 104555 allows you to make additional entries to the routing table or delete existing routes during runtime. This register set can hold up to five entries.

R 104550 – Status

Description	Register properties	Values / Description	
This register shows status information when adding or deleting routes during runtime.	Reading values	0	No error
		-1	Routing table is full
		-2	Entry not found
		-3	Port is not active
		-4	TCP/IP stack not initialized

Tab. 47: Status

R 104551 – Command

Description	Register properties	Values / Description	
This command adds or deletes a route.	Writing values	1	Add route
		2	Delete route

Tab. 48: Command

R 104552 – port number

Description	Register properties	Values / Description	
Use this register to define the Ethernet port.	Writing values	1	ETH1 (X102)

Tab. 49: Port number

R 104553 – IP address of the static route

Description	Register properties	Values
Write the IP address of the route to this register.	Writing values	0.0.0.0 …°255.255.255.255

Tab. 50: IP address of the static route

R 104554 – subnet mask of the static route

Description	Register properties	Values
Use this register to define the Ethernet port.	Writing values	0.0.0.0 …°255.255.255.255

Tab. 51: Subnet mask of the static route

R 104555 – gateway of the static route

Description	Register properties	Values
Write the gateway IP address of the route to this register.	Writing values	0.0.0.0 …°255.255.255.255

Tab. 52: Gateway of the static route

Setting a static route

To either add the route to the routing table or delete it from there during runtime, proceed as follows:

1. Write the port number to register 104552.
2. Write the IP address to register 104553.

3. Write the subnet mask to register 104554.
4. Write the IP address of the gateway to register 104555.
5. Write the command to register 104551.

⇨ If register 104550 has got value 0, the command has been executed successfully. The route settings are active.

Example

The JC-975MC is connected to your in-house network, for example, through the Ethernet port X102 and a router.

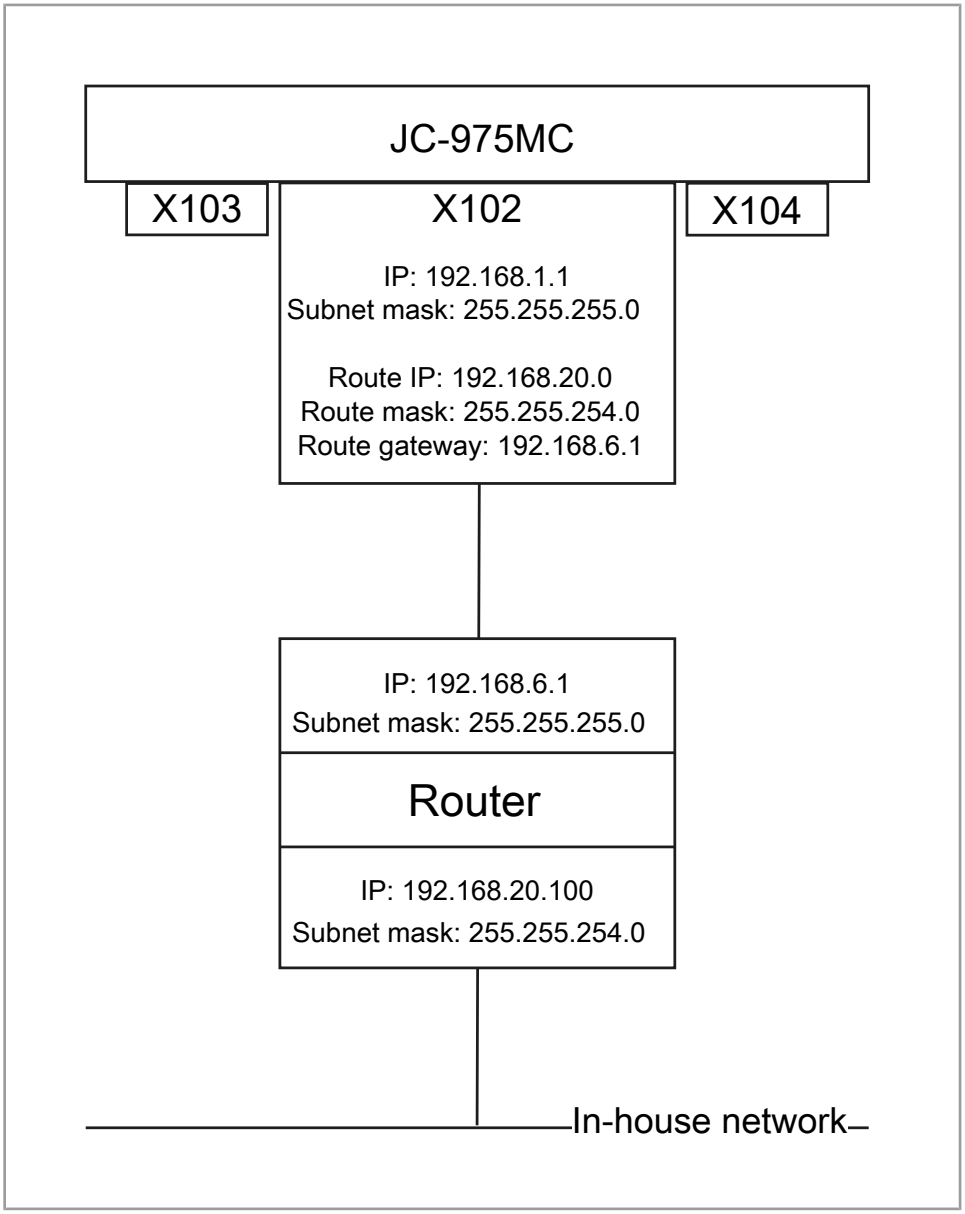


Fig. 19: Router – in-house network

9.4 Setting the IP address automatically via USB flash drive

The AutoCopy function automatically applies the IP configuration settings stored in a USB flash drive to the controller. For details see Automatic copying of controller data (AutoCopy) [▶ 106]. The AutoCopy function uses the registers described in the chapter “Setting the IP address by non-volatile registers”.

Requirements

You are familiar with the AutoCopy function. Please also refer to Automatic copying of controller data (AutoCopy) [▶ 106].

AutoCopy command file

The example below shows a command file of the AutoCopy function:

```
[OPTIONS]
CommandCount = 1
LogFile = /USB1/autocopy.log
LogAppend = 0

[COMMAND_1]
Command = FileCopy
Source = /USB1/config.ini
Destination = /System/config.ini
```

9.5 Setting the IP address during runtime

The IP interface is initialized during the boot phase according to the settings in the configuration memory . The registers allow for adjustment of the following settings. 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 file. When de-energizing the controller, your settings will be lost.

Register overview

Registers	Description
104531	ETH1 / X102: IP address
104532	ETH1 / X102: Subnet mask
104533	ETH1 / X102: IP address of the default gateway

Example

To make changes to the IP addresses and the subnet mask of the ETH1 Ethernet port, proceed as follows:

✓ While settings are being made, communication via IP interface is disabled, to prevent loss of data.
✓ The values to be written must be valid. This can be ensured, e.g. by including a validity check in the application program.
This is important, because no such check is performed if you set the parameters during runtime.

1. Write the value 0.0.0.0 to register 104533.
2. Write the value 0.0.0.0 to register 104532.
3. Write the desired IP address to register 104531.
4. Write the desired subnet mask to register 104532.

5. Write the desired IP address of the default gateway to register 104533.

⇨ The settings are complete. Communication has been re-established.

9.6 Setting a route during runtime

Using a different register set allows you to make further additions to the routing table or to delete existing routes during runtime.

After the controller has powered-up, there are 5 entries available per port.

Register number	Function
104550	Status 0 = No error -1 = Routing table is full -2 = Entry not found -3 = Port is not active -4 = TCP/IP stack not initialized
104551	Command 1 = Add route 2 = Delete route
104552	Port number 1 = ETH1 (X102)
104553	IP address
104554	Subnet mask
104555	Gateway

To set the route, proceed as follows:

1. Write the port number to register 104552.
2. Write the desired IP address to register 104553.
3. Write the subnet mask value to register 104554.
4. Write the IP address of the gateway to register 104555.
5. Write the command to register 104551.

⇨ If register 104550 shows the value 0, the command has been executed successfully and the route settings have been applied.

9.7 Changing the IP address using the JetIPScan command line tool

Introduction

The JetIPScan program changes the IP address, subnet mask and the IP address of the default gateway of the JC-975MC ETH1 (X102) interface.

ⓘ Info

Downloading JetIPScan

The JetIPScan program is available for download on www.bucherautomation.com.

Contents of the ZIP file

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

- Executable: JetIPScan_V_1-11-00.exe
- Help file: jetipscan_01_help_en.png
- Batch file for determining the IP address: read_IP_via_JETIPSCAN.bat
- Batch file for setting the controller IP address to 192.168.10.150: write_IP_via_JETIPSCAN_10_150.bat

The batch files launch the program JetIPScan.

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

Possible commands of the JetIPScan software

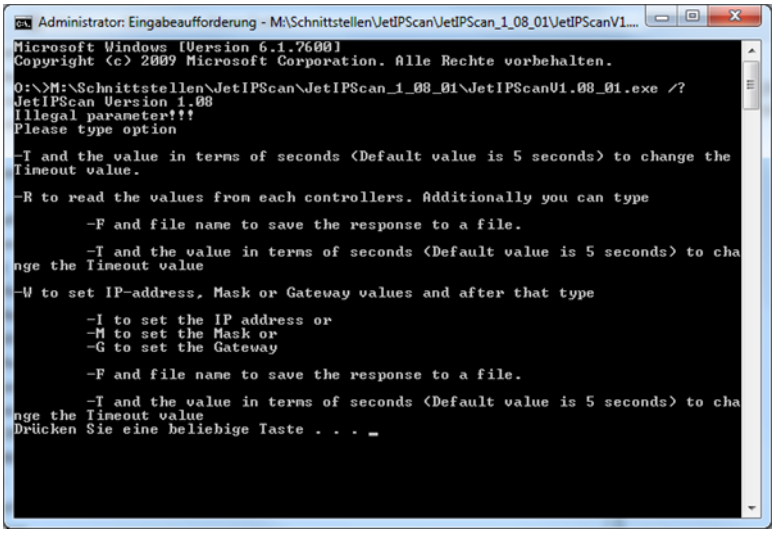


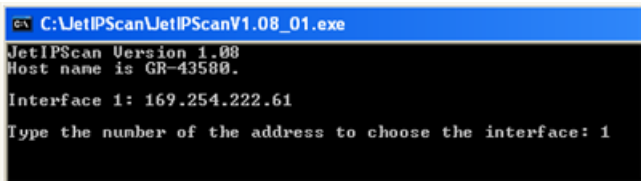
Fig. 20: JetIPScan command options

Changing the IP address

✓ The PC and the JC-975MC are connected by Ethernet.

1. Launch the JetIPScan program on your PC, for example by executing the batch file **write_IP_via_JETIPSCAN_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) where the device is connected whose IP address you want to adjust.



⇨ 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-975MC to 192.168.10.150.

Changing the subnet mask

1. Write a batch file. The content of the batch file is e.g. **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 e.g. **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**.

9.8 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 from our website on the pages of the respective product.

ⓘ Info

Further information

More information on this subject is available on our website.

Start | www.bucherautomation.com

9.8.1 Operating system update of the controller

This chapter describes how to carry out an OS update of the JC-975MC on the controller side.

There are several ways of transferring the OS file to the controller:

– Using the programming tool JetSym
– Via an FTP connection
– From a USB flash drive
– 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-975MC 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-975MC controller is available.
✓ A UDP/IP and a TCP/IP connection between programming tool and JC-975MC 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-975MC 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-975MC 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-975MC.
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.

OS update from a USB flash drive

An automatic OS update of the controller from the USB flash drive can be carried out using the AutoCopy function.

ⓘ Info

Further information

For in-depth information, see the *File System* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

Updating the OS from within the application program

The file functions included in the STX language set allow you to carry out a program-controlled OS update of a JC-975MC from within an OS file.

Performing the update

- ✓ An OS file is available in the JC-975MC file system.
- ✓ The operating system and the application program are running.
- ✓ The controller is and remains switched ON.

1. Copy the OS file, give it a name of your choice, add the extension ***.os** and store in the */System/OS* directory.
2. Re-boot the controller to activate the updated OS, for example by writing to the system command register.

STX Program

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

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

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;
```

9.8.2 OS update of the Windows IPC component

– To update or install a windows operating system, proceed as with any Windows PC.

– This is even possible while the controller is active.

– It is also save to shutdown or restart Windows without compromising the controller unit functions as long as power continues to be supplied to the controller.

9.9 File System

Controller unit

The file system lets you access files located on the internal flash disk or an USB flash drive. Some files may be protected against read/write access or deletion. This is normal behavior. 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

System directories cannot be deleted. System directories withstand formatting.

Directory	Description
/System	– System configuration – System information
/USB1 /USB2	– Root directory of the USB flash drive

Tab. 53: System directories

Formatting and checking of data

The JC-975MC is able to format only the flash drive. Formatting or checking the USB flash drive is not possible.

ⓘ Info

Further information

For in-depth information, see the *File System* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

9.9.1 Properties

The following properties apply to the internal flash disk and USB flash drive:

– 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:

– 32 MB

Properties

The internal flash drive has the following properties:

– Supports up to 7 directory levels and 1 file level.
– Is case-sensitive.
– Supports directory and file names with up to 63 characters.
– Supports all characters except "/" and ".." for directory and file names.
– User/access management supports 31 locks max for 33 users.

USB flash drive - Properties

Size

The available memory size depends on the USB flash drive being used. Tested size:

– 2 ... 64 GB

Properties

The USB flash drive has the following properties:

– The USB flash drive must be compatible with FAT12, FAT16, or FAT32.
– No case sensitivity.
– Directory and file name length must not exceed 63 characters.
– The following characters are not allowed: "/", "\", ":", "*", "?", "<", ">" and "|"
– The number of subdirectory levels depends on the formatting.
– There is no user/access management.

10 Initial commissioning

This chapter summarizes the initial commissioning of the Controller JC-975MC using the example of *Creating and executing a program that increments a variable*.

Requirements

– The controller is connected to a PC either directly via Ethernet, or through a switch or hub.
– The JetSym programming software with version 5.1.3 or higher is installed on the PC.
– Mode selector S11 [▶ 38] is in STOP position.
– The controller is connected with the JX3-BN-ETH bus node via Ethernet.

Switching the system on

Energizing the JC-975MC controller also starts a set of diagnostic and self-test routines.

Windows unit of the IPC

– Standard Windows power-ON self test (POST)
– In addition, default diagnostic options of Windows are available, including memory and CPU usage.

Controller unit

– File system check of the internal USB flash drive (error register 200008, bit 0: Flash drive error)
– Check if the non-volatile registers were saved correctly during the last shutdown (error register 200008, bit 4: Error: non-volatile registers)

To energize and boot the controller, proceed as follows:

1. Set the mode selector S11 to **STOP** position.
   ⇨ This prevents the application program from being loaded.
2. Connect the DC 24 V supply voltage to port X101.
   ⇨ The controller and the integrated industrial PC are booting up.
   ⇨ The green LED GP1 (RUN) is flashing with a frequency of 1 Hz.
3. Configure the controller as described below.
4. If you want to start the controller in normal operation, set switch S11 to **RUN**.

⇨ The controller loads and executes the user application.

Ethernet connection to the controller

The default IP address (factory setting) of the JC-970MC controller is 192.168.1.1. It provides access to the controller, e.g. from an external programming PC via Ethernet port X102.

► Configure the Ethernet interface of your PC in way that it allows the PC to communicate with the controller via this IP address.

Internal IP address of the controller

You can also access the controller unit internally from the Windows unit of the JC-975MC. In this case, the IP address of the controller is 192.168.2.3.

Boot-up behavior of the controller

NOTICE! The application program will not launch, if the mode selector is in STOP position during controller boot-up.

LEDs

The following LED status indicates that the system was has been commissioned without errors:

LED	Function	Color	Description
GP1	RUN	Green, flashing (1 Hz)	Logic voltage supply is OK; application program is not running
GP2	D1	OFF ---	Normal operating condition
GP3	ERR	OFF ---	No error

Tab. 54: Status indication

Configuration example

The initial commissioning outlined below is based on the following configuration:

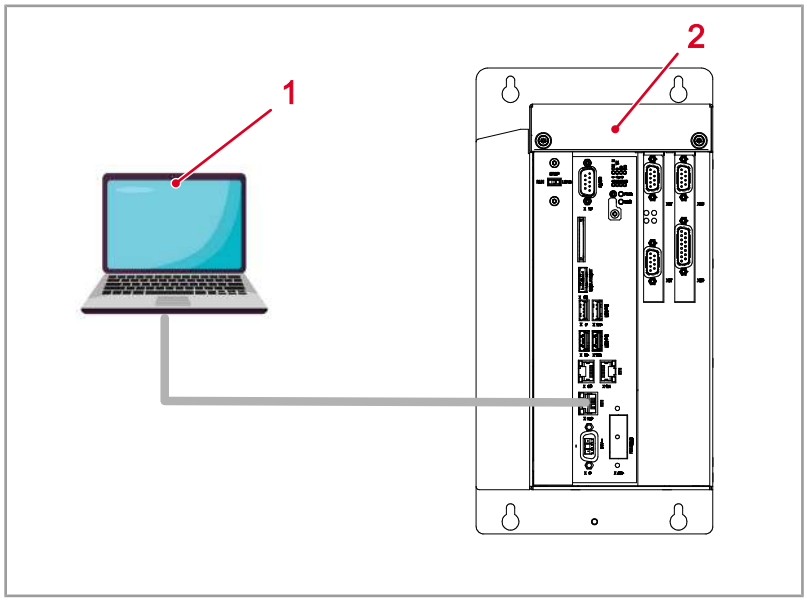


Fig. 21: Configuration example

Position	Part	Description
1	PC	Programming system
2	JC-975MC	Controller

Software versions

The sample program requires the following software versions:

– JetSym in version 5.1.3

– JC-975MC Controller with OS version 1.07

NOTICE! For code samples with recent software releases refer to JetSym online help.

Creating a program

To create and check the program, proceed as follows:

1. Launch the JetSym programming environment.
2. Create a new project.
3. Set the controller model (JC-975MC).
4. Set the IP address of the controller.
5. Open the program editor.

6. Enter the program specifications.
7. Upload the project to the controller.
8. Open a setup pane.
9. Enter the variable name (Count).
10. Activate the setup.

JetSym STX program

```
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	Current content of the variable	The content of the variable increments by 1 every 2 s

11 Programming

Programming of the JC-975MC is performed using the JetSym programming tool.

11.1 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. 55: 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. 56: 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. 57: Numerical formats

JetSym sample programs

The notation for sample programs used in this document is listed in the table below:

Notation	Format of numerical values
`Var, When, Task`	Keyword
`BitClear();`	Commands
`100 0x100 0b100`	Constant numerical values
`// This is a comment`	Comment
`// ...`	Further program processing

Tab. 58: JetSym sample programs

11.2 Storage options - Overview

The controller JC-975MC 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.

11.2.1 Operating system memory

The OS is stored to a non-volatile flash memory in the CPU. It boots immediately after the device was switched on.

Features

– Internal flash memory for storing OS data
– Internal volatile RAM for storing OS data

Access

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

11.2.2 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: 32 MB
– Size of the USB flash drive: 2 GB ... 64 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

11.2.3 Application Program Memory

By default, the application program is uploaded from JetSym to the controller and stored there.

Properties

– Stored as file within the file system
– Default directory */app*
– Files may also be stored to other directories (or USB flash drive)

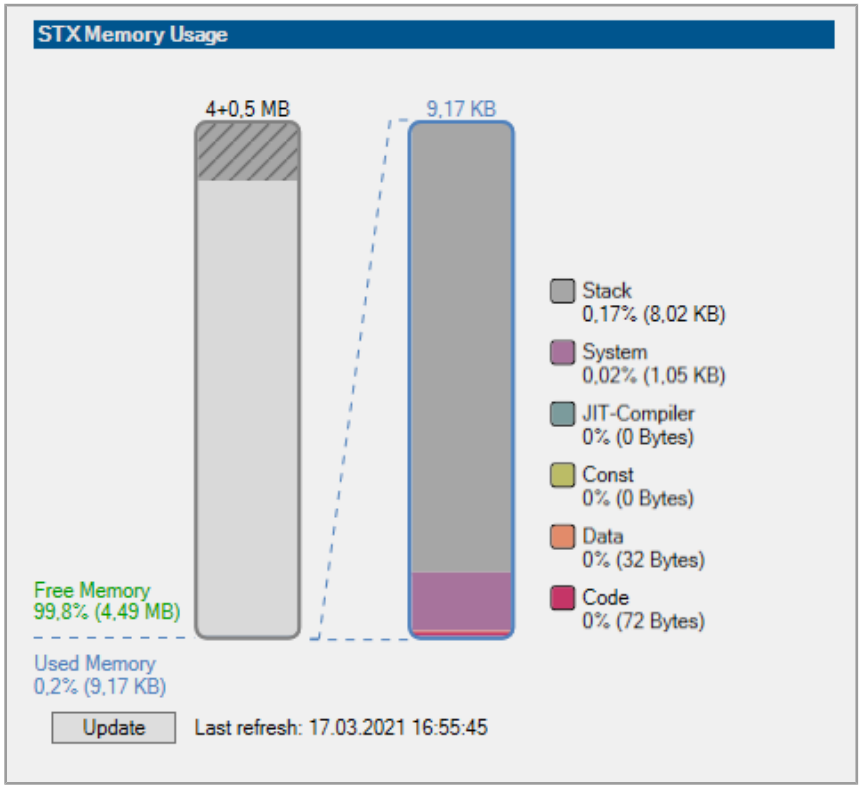
Type of access

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

STX memory utilization

After the program has started, JetSym lets you determine the memory utilization by the application program.

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.

11.2.4 Memory for volatile application program variables

Volatile variables are used to store data that need not be maintained when de-energizing the JC-975MC.

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

11.2.5 Memory for non-volatile application program registers

Non-volatile registers let you store data which must be saved when the JC-975MC is de-energized.

Properties

– Global variables which are permanently assigned to addresses (%VL)
(see also *Localization of Variables* in JetSym Help).
– Register variables always occupy 4 bytes.
– Register variables are not initialized by the operating system.
– Number of register variables: 120,000
– Register numbers: 1000000 ... 1119999 (identical with %RL area)

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 local-
ized 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.

11.2.6 Memory for non-volatile application program variables

Non-volatile variables let you store data which must be maintained when the JC-975MC is de-energized.

Properties

– Global variables which are assigned to permanent registers (%RL)
– Variables are stored in a compact way.
– Size: 480,000 bytes
– Register numbers: 1000000 ... 1119999 (identical with %VL area)

Types of access

– Via 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.

11.2.7 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

11.2.8 Registers on I/O modules

These registers are located on I/O modules connected via system bus.

Properties

– Global variables with dedicated addresses (%VL)
– The type is depending on the module.
– Register numbers of local JX6-IO modules:
20SJ0000 ... 20SJ19999 (S = slot 1 ... 2 JI-PCIE-Exx; J = submodule slot JX6-I/O 1 ... 2 on JI-PCIE-EXX)
– JX3-IO module register numbers on a JX3-BN-ETH: 1nnn020000 ... 1nnn179999 (nnn = GNN)
– JX3-IO module register numbers on a JX3-BN-EC: 12nn020000 ... 12nn339999 (nn = ENN)

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

11.2.9 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

11.2.10 Digital inputs and outputs

Inputs and outputs are 1-bit variables. This means they can be either TRUE or FALSE. Inputs and outputs are addressed through immediate read/write cycles.

Virtual digital inputs and outputs

– Global variables assigned to permanent addresses (%IX, %QX)
– Used for RemoteScan via Modbus/TCP
– Quantity: 16,000
– I/O numbers: 20001 ... 36000

Local digital inputs and outputs

– Global variables assigned to permanent addresses (%IX, %QX)
– Located on local JX6-IO16CB modules, which are plugged into JI-PCIe-Exx expansion cards, which, in turn, can be installed in the PCI express slots
– Quantity: 32 I/Os max. allocated to max. 2 x JX6-IO16CB modules
– I/O numbers: 20SJ001zz (see Local JX6-IO module register slot numbering [▶ 75])

Distributed inputs and outputs via Ethernet system bus

– Global variables assigned to permanent addresses (%IX, %QX)
– Located on JX3-IO modules connected via JX3-BN-ETH
– I/O numbers of remote devices connected to a JX3-BN-ETH: 1nnn010201 ... 1nnn011716 (nnn = GNN)

Distributed input and outputs via EtherCAT®

– Global variables assigned to permanent addresses (%IX, %QX)
– Located on JX3-IO modules connected via JX3-BN-EC
– I/O numbers of distributed devices connected to the JX3-BN-EC: 12nn000201 … 12nn003316 (nn = ENN)

Types of access

– By JetSym
– By the email client

– Via a browser (via the HTTP server)
– From HMIs
– From within the application program

11.3 Saving and loading an application program

Downloading the application program from JetSym to the device by default stores the program as a file to the internal flash drive.

If you want the device to write the application program to an SD memory card or USB flash drive, you need to configure the file path.

Restarting the application program in JetSym or booting the device loads and executes the application program via the file system. The user determines the program to be executed.

ⓘ Info

Further information

For in-depth information, see the *File System* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

11.4 Addressing of registers and I/Os

Bucher Automation AG controllers and modules offer a host of functions which can be accessed by the user via registers. A unique number is assigned to each register, input and output.

I/O expansion module connectivity options:

– JX6-IO modules via the local PCIe bus using JI-PCIE-Exx expansion cards
– JX3-IO modules via
  – Ethernet system bus using JX3-BN-ETH bus heads, or
  – EtherCAT® using JX3-BN-EC bus heads

Purpose of register numbers

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.

Module registers – Definition

Use module registers to read process, configuration and diagnostics data from the module, or write such data to it. The module register number within a module is unique.

Registers - definition

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.

Purpose of I/O numbers

I/O numbers are applied in the following cases:

– Read access to a digital input in the JetSym Setup pane.
– Read/write access 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.

Also refer to Installing JI-PCIE-Exx riser cards [▶ 26].

See also

Addressing expansion modules [▶ 86]

11.5 Local JX6-IO module register slot numbering

Local JX6-IO expansion modules are located on JI-PCIE-Exx expansion cards available separately. JI-PCIE-Exx expansion cards require assembly in the PCIe slots by the customer (see Installing JI-PCIE-Exx riser cards [▶ 26]).

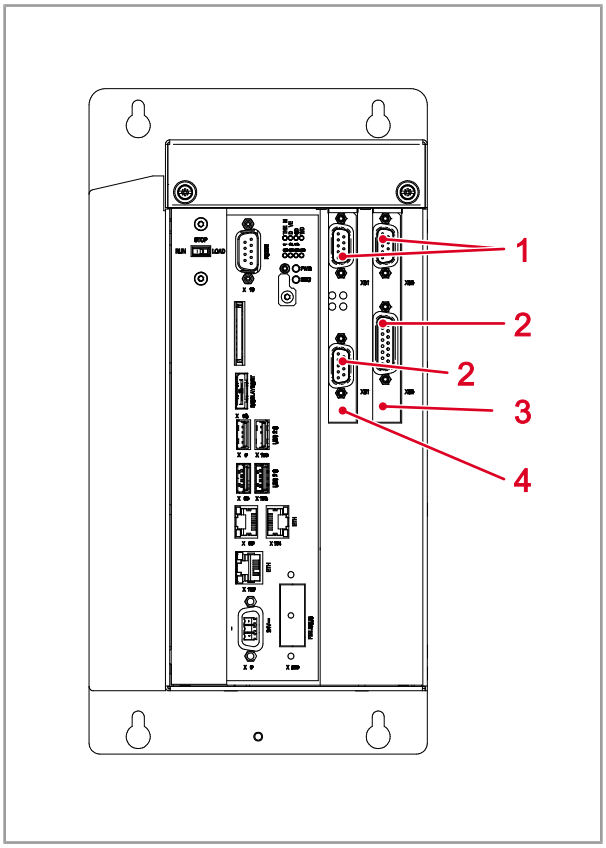


Fig. 22: Plug-in slots

1	Upper riser card slot (J=2)
2	Lower riser card slot (J=1)
3	PCIe slot (S=2) on the right-hand side of the JI-PCIE-Exx riser card
4	PCIe slot (S=1) on the left-hand side of the JI-PCIE-Exx riser card

Numbering

The slot number breaks down as follows:

– The riser card on the left-hand side of the PCIe bus is numbered 1 (S=1)
– The riser card on the right-hand side of the PCIe bus is numbered 2 (S=2)
– The lower module on the riser card is numbered 1 (J=1)
– The upper module on the riser card is numbered 2 (J= 2.)

Register numbers of local JX6-IO submodules

The register number of the local JX6-IO submodules connected to the JC-975MC breaks down as follows:

2	0	S	J	0	0	z	z	z

Element	Description	Value range
S	Number of the riser card	1 ... 2
J	Number of the module located on the riser card	1 ... 2
zzz	Module register number	100 … 999

Tab. 59: Register numbers

I/O numbers of local JX6-IO submodules

The I/O number of the local JX6-IO submodules connected to the JC-975MC breaks down as follows:

2	0	S	J	0	0	1	z	z

Element	Description	Value range
S	Number of the riser card	1 ... 2
J	Number of the module located on the riser card	1 ... 2
zz	Module-specific I/O number	01 … 08

Tab. 60: I/O numbers

11.5.1 Register numbers and I/O numbers of the JX6-IO16CB digital submodule

Introduction

The digital input/output module is equipped with 8 galvanically isolated 24 V inputs, and 8 galvanically isolated 24 V outputs with a maximum current of 0.5 A each.

It is pre-assembled on the JI-PCIE-E03 and JI-PCIE-E06 riser cards.

I/O numbers of local JX6-IO submodules

The I/O number of the local JX6-IO submodule breaks down as follows:

2	0	S	J	0	0	1	z	z

Element	Description	Value range
S	Number of the riser card	1 ... 2
J	Number of the JX6-IO submodule located on the riser card	1 ... 2
zz	Module-specific I/O number	01 … 08

Tab. 61: I/O numbers of local JX6-IO submodules

Register numbers

In this chapter, only the module register number is specified.

► To derive the register number itself, add the corresponding controller's basic register number.

The basic register number combines the riser card number and the number of the submodule on the riser card.

2	0	S	J	0	0	z	z	z

Element	Description	Value range
S	Number of the riser card	1 ... 2
J	Number of the JX6-IO submodule located on the riser card	1 ... 2
zz	Module register number	100 … 999

Tab. 62: Register numbers of local JX6-IO submodules

Register overview

The local JX6-IO submodule can be programmed using a set of module registers.

Registers	Description
MR 100	State of the digital inputs
MR 101	Access to the digital outputs
MR 102	Error state of the digital outputs
MR 103	Control register
MR 104	Filter interval of inputs 1 ... 4
MR 105	Filter interval of inputs 5 ... 8

Tab. 63: Local JX6-IO module register overview

MR 100
Digital inputs

– Use MR 100 to acquire the input state.
– If the voltage exceeds 12 V, the bit assigned to this input is set in this register (= 1).
– If the voltage is below 12 V, the assigned bit is not set (= 0).

Bit assignment

Bit 0	Input 1
Bit 1	Input 2
…	
Bit 7	Input 8

Module register properties

Access	Read
Value after reset	State of the inputs

MR 101
Digital outputs

– Use MR 101, to acquire the output state and to (de-)energize the outputs.
– When the bit is set, the assigned output is energized.
– When the bit is cleared, the assigned output is de-energized.

Bit assignment

Bit 0	Output 1
Bit 1	Output 2
…	
Bit 7	Output 8

MR 102
Error state of the digital outputs

– User bit 7 in MR 102 to acquire the error state of the digital outputs.
– The other bits of this register remain cleared (= 0).
– Even if the error is acknowledged, bit 7 remains set.
► To clear bit 7 , write any value to this register.

Meaning of the bits

Bit 7	**Collective error message of digital outputs**
	0 = No error
	1 = Error from at least one output

Module register properties

Values	0.128
Value after reset	128

Detects the following errors:

– Insufficient voltage of the 24 V output supply
– Short-circuit of the output
– Overloading of the output

MR 103
Control register

The control register allows for configuration of the JX6-IO module.

Meaning of the bits

Bit 0	**Bit 7 in MR 101 output register**
	0 = The error state read from the digital outputs is written to bit 7 of MR 101.
	1 = The state read from output 8 is written to bit 7 of MR 101.

Bit 1	**Error event behavior of digital outputs**
	0 = In the event of an error, only the faulty output is switched off.
	1 = In the event of an error, all outputs are switched off.
Bit 3, 2	**Filter interval for the error signal of the outputs**
	00 = 1.5 … 2.0 ms
	01 = 0.4 … 0.5 ms
	10 = approx. 0.1 ms
	11 = approx. 0.02 ms

Module register properties

Value after reset	1

MR 104
Filter interval of inputs 1 ... 4

– Use this register to set the filter interval for inputs 1 ... 4.
– Each input is assigned 2 bits of this register.

Meaning of the bits

Bit 1, 0	**Filter interval of input 1**
	00 = 1.5 … 2.0 ms
	01 = 0.4 … 0.5 ms
	10 = approx. 0.1 ms
	11 = approx. 0.02 ms
Bit 3, 2	**Filter interval of input 2**
	00 = 1.5 … 2.0 ms
	01 = 0.4 … 0.5 ms
	10 = approx. 0.1 ms
	11 = approx. 0.02 ms
Bit 5, 4	**Filter interval of input 3**
	00 = 1.5 … 2.0 ms
	01 = 0.4 … 0.5 ms
	10 = approx. 0.1 ms
	11 = approx. 0.02 ms
Bit 7, 6	**Filter interval of input 4**
	00 = 1.5 … 2.0 ms
	01 = 0.4 … 0.5 ms
	10 = approx. 0.1 ms
	11 = approx. 0.02 ms

MR 105
Filter interval of inputs 5 ... 8

– Use this register to set the filter interval for inputs 5 ... 8.
– Each input is assigned 2 bits of this register.

Meaning of the bits

Bit 1, 0	**Filter interval of input 5**
	00 = 1.5 … 2.0 ms
	01 = 0.4 … 0.5 ms
	10 = approx. 0.1 ms
	11 = approx. 0.02 ms
Bit 3, 2	**Filter interval of input 6**
	00 = 1.5 … 2.0 ms
	01 = 0.4 … 0.5 ms
	10 = approx. 0.1 ms
	11 = approx. 0.02 ms
Bit 5, 4	**Filter interval of input 7**
	00 = 1.5 … 2.0 ms
	01 = 0.4 … 0.5 ms
	10 = approx. 0.1 ms
	11 = approx. 0.02 ms
Bit 7, 6	**Filter interval of input 8**
	00 = 1.5 … 2.0 ms
	01 = 0.4 … 0.5 ms
	10 = approx. 0.1 ms
	11 = approx. 0.02 ms

11.5.2 Register numbers of the JX6-SV1-ES counter module

Introduction

The JX6-SV1-ES counter module features a counter input for the incremental encoders or SSI absolute encoders and is pre-assembled on the JI-PCIE-E04 or JI-PCIE-E05 riser cards.

Register numbers

In this chapter, only the module register number is specified.

► To derive the register number itself, add the corresponding controller's basic register number.

The basic register number combines the riser card number and the number of the submodule on the riser card.

2	0	S	J	0	0	z	z	z

Element	Description	Value range
S	Number of the riser card	1 ... 2
J	Number of the JX6-IO submodule located on the riser card	1 ... 2

Element	Description	Value range
zzz	Module register number	100 … 999

Tab. 64: Module register numbers of the JX6-SV1-ES counter module

Register overview

– The local JX6-SV1 module can be programmed using a set of module registers.
– The function of module registers 103 ... 110 depends on whether an incremental encoder or an absolute encoder has been configured.

Registers	Description
MR 100	Module ID
MR 101	Hardware configuration
MR 103	Strobe value/encoder value
MR 104	Preset/clock generator
MR 105	Count value
MR 106	Reference/offset
MR 107	Control register
MR 108	Status register
MR 110	Sampling timer

Tab. 65: JX6-SV1 module register overview

MR 100
Module ID

This register shows, if a JX6-SV1 is plugged in.

Module register properties

Access	Read
Value after reset	1 (for JX6-SV1)

MR 101
Hardware configuration

1. Use this register to set the module hardware.
2. Always write 0 to undocumented bits.

Meaning of the bits

Bit 0	**Relay**
	0 = De-energized; pin 14 not connected to pin 15
	1 = Enabled; pin 14 is connected to pin 15

Bit 2	**CHD signal on the counter chip**
	0 = Counter zeroing/reference has been disabled
	1 = Counter zeroing/reference has been enabled
Bit 5	**Encoder type**
	0 = Incremental encoder
	1 = SSI absolute encoder

MR 103
Incremental encoder: Strobe register

Module register properties

Values	24 bits
Write access	Write the actual count value to this register
Read	Read the count value last written to the register

SSI absolute encoder: Encoder value

– This register shows the position value output by the SSI encoder.
– Only if bit 0 is set in the status register (MR 108), are there valid values to be read.

Module register properties

Access	Read
Values	32 bits

MR 104
Incremental encoder: Preset

When writing a value to register MR 105, this value is used as new count value.

Module register properties

Values	24 bits

SSI absolute encoder: Clock generator

Use this register to configure the connected SSI encoder.

Meaning of the bits

Bit 9 … 0	**Frequency (= factor) of the SSI clock signal**
	Module clock/(SSI clock * 2) - 1 Example: Given: Module clock = 8 MHz (defined by JX6-SV1 module) SSI clock = 500 kHz (desired SSI clock frequency to be applied when reading data from the encoder) Clock generator value = 8,000,000 / (500,000 * 2) - 1 = 7

Bit 15 … 10	Clock pulse configuration
	Number of SSI encoder bits *2 + 1 Example: Number of SSI encoder bits = 24 = 24 * 2 + 1 = 49 (generates 25 pulses at clock output)

Module register properties

Value range	0 ... 65,535
Value taken from the above example	Data length of the SSI encoder (bits 10 ... 15): 49 * 1,024 (2^10) = 50,176 Clock generator value (bit 0 ... 9: 7 * 1(2^0) = 7 ==>Register value: 50,176 + 7 = 50,183

MR 105
Incremental encoder: Count value

Use this register to access the present count value of the incremental encoder.

Module register properties

Values	24 bits
Write access	The value stored in MR 104 is copied to this register.
Read	Present count value

MR 106
Reference/offset

Use this register as both reference or offset register, depending on the settings in the control register.

Module register properties

Values	24 bits

MR 107
Control register

Use this register to set the operating modes and input filters.

Meaning of the bits

Bit 0	Strobe
	0 = Strobe not allowed
	1 = Strobe allowed
Bit 1	**Reference strobe**
	0 = Reference strobe not allowed
	1 = Reference strobe allowed (with /CHA * CHB * CHD)

Bit 2	**Verification against reference value**
	0 = No verification against reference value
	1 = Verification of the count value against the reference value. If the values match, bit 2 is set in MR 008.
Bit 3	**Counter zeroing**
	0 = No counter zeroing
	1 = Counter zeroing with each CHC * CHD
Bit 4	**Repeated counter zeroing**
	0 = No counter zeroing
	1 = Counter zeroing with each CHC * CHD
Bit 5	**Counting inputs**
	0 = Counter inputs are active
	1 = Counter inputs are disabled
Bit 6	**Set by reference**
	0 = Set-by-reference is disabled
	1 = The counter is loaded with the reference value with CHC * CHD.
Bit 9, 8	**Pulse edge selection**
	00 = Pulse direction mode
	01 = Single evaluation
	10 = 2-fold evaluation
	11 = 4-fold evaluation
Bit 10	**Addition of offset**
	0 = Addition disabled
	1 = Adding an offset to the count and strobe values is allowed
Bit 11	**File format using an SSI encoder**
	0 = Binary code
	1 = Gray format
Bit 12	**Parity using an SSI encoder**
	0 = No parity bit
	1 = Parity check
Bit 13	**Polarity of the parity bit**
	0 = Even parity
	1 = Odd parity
Bit 23 … 14	**Input filters**
	Minimum signal duration required for signal detection: = (T_{sig} / T_{sys}) - 1 T_{sig} = minimum signal duration T_{sys} = Period of a system cycle (125 ns @ 8 MHz)

Module register properties

Values	24 bits
Value after reset	0x300

MR 108
Status register

After read access to this register, all bits, except for bit 3, are cleared.

Meaning of the bits

Bit 0	**Strobe**	
	Incremental encoder	0 = Strobe value not yet detected
		1 = Strobe value detected
	Absolute encoder	0 = Invalid position value
		1 = Valid position value
Bit 1	**Reference strobe**	
	1 = Reference strobe value detected	
Bit 3	**Counter zeroing**	
	1 = Counter was zeroed	
Bit 4	**Strobing value has been overwritten**	
	Incremental encoder	1 = The value in the strobe register was overwritten
	Absolute encoder	1 = The position value was overwritten
Bit 5	**Lost reference value**	
	1 = A new reference value was lost, as the previous value has not been read out yet.	
Bit 6	**Z track error**	
	1 = Z track errors, or maximum travel speed exceeded, or inadmissible Z track change.	
Bit 7	**Parity error**	
	1 = A parity error has occurred	

MR 110
Sampling timer of the SSI encoder

This register value determines the reading intervals of the position value sampled by the encoder. If set to "0", no sampling is carried out.

Writing a value greater than 0 to this register reads the encoder value according to following sequence:

The **first** reading of the encoder value takes place **halfway** through the sampling frequency period. In the example below, where the sampling frequency is 1 kHz, the reading takes place after 500 µs. After that, the encoder value is read after each full period.

Reading the encoder continues until a new value is written to MR 110.

Module register properties

Value range	0 ... 65,535

Calculation	Module frequency / (sampling frequency * 2) - 1
Example	Module frequency = 8 MHz Sampling frequency = 1 kHz = 8,000,000 / (1,000 * 2) - 1 = 3,999

Application example

Task:

– The value of a SSI encoder position is required to be read in sync with a cyclic user task.
– The task is called in 1 ms intervals.
– Reading the encoder is supposed to take place 600 µs after the cyclic task was stared.

Solution:

The period (1/sampling frequency) of the sampling timer must be 1200 µs, because the first reading of the encoder takes place halfway through the period.

Calculation: (Module frequency * length of the first period) - 1

Application example with pseudo code:

```
task SSiEncoderReadTask cycle t#1ms
    <MR110_SamplingTimer> := 0; // stop timer
    Read <MR108_StatusRegister>;
// read all status bits. (this will reset all bits except bit 3)
    Read <MR103_absoluteEncoderValue>;
    <MR110_SamplingTimer> := <(8MHz * 600µs) - 1 >;// restart timer
// => 8.000.000 * 0.0006 - 1 => RegValue = 4799.
// check status, and if ok further process the encoder raw-value
end_task;
```

11.6 Addressing expansion modules

The JX3 IO module connects to sensors and actuators. EtherCAT® bus nodes (JX3-BN-EC) or Ethernet system bus nodes (JX3-BN-ETH) transfer data from the JX3 system bus to the controller.

Each JX3 IO module provides 10,000 module registers

that are mapped to registers in the controller. The only way of writing and reading process, configuration, and diagnostic data to and from an expansion module is via controller registers. Controller registers can be accessed in the application program of the controller, in a setup pane of JetSym or via user interface.

The number of a controller register or IO being mapped to a module register depends on the following parameters:

– Controller model
– Type of connection between controller and expansion module
  – Remote connection via Ethernet
  – Remote connection via EtherCAT®
– Position and number of JX3 IO expansion modules in the system
  The maximum number of JX3 IO expansion modules connecting to the bus node depends on the expansion module type.

ⓘ 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.

You can find the configurator as a download at www.bucherautomation.com under product page *JX3*-Modules > Downloads.

ⓘ Info

Further information

For in-depth information, see the *JX3 System* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

11.6.1 Numbering registers and I/Os

Module registers - Definition

Module registers are the data interface of a JX3 module. Module registers let you read process, configuration and diagnostics data from the JX3 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.

Registers - Definition

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 in-depth information, see the *JX3 System* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

11.6.2 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

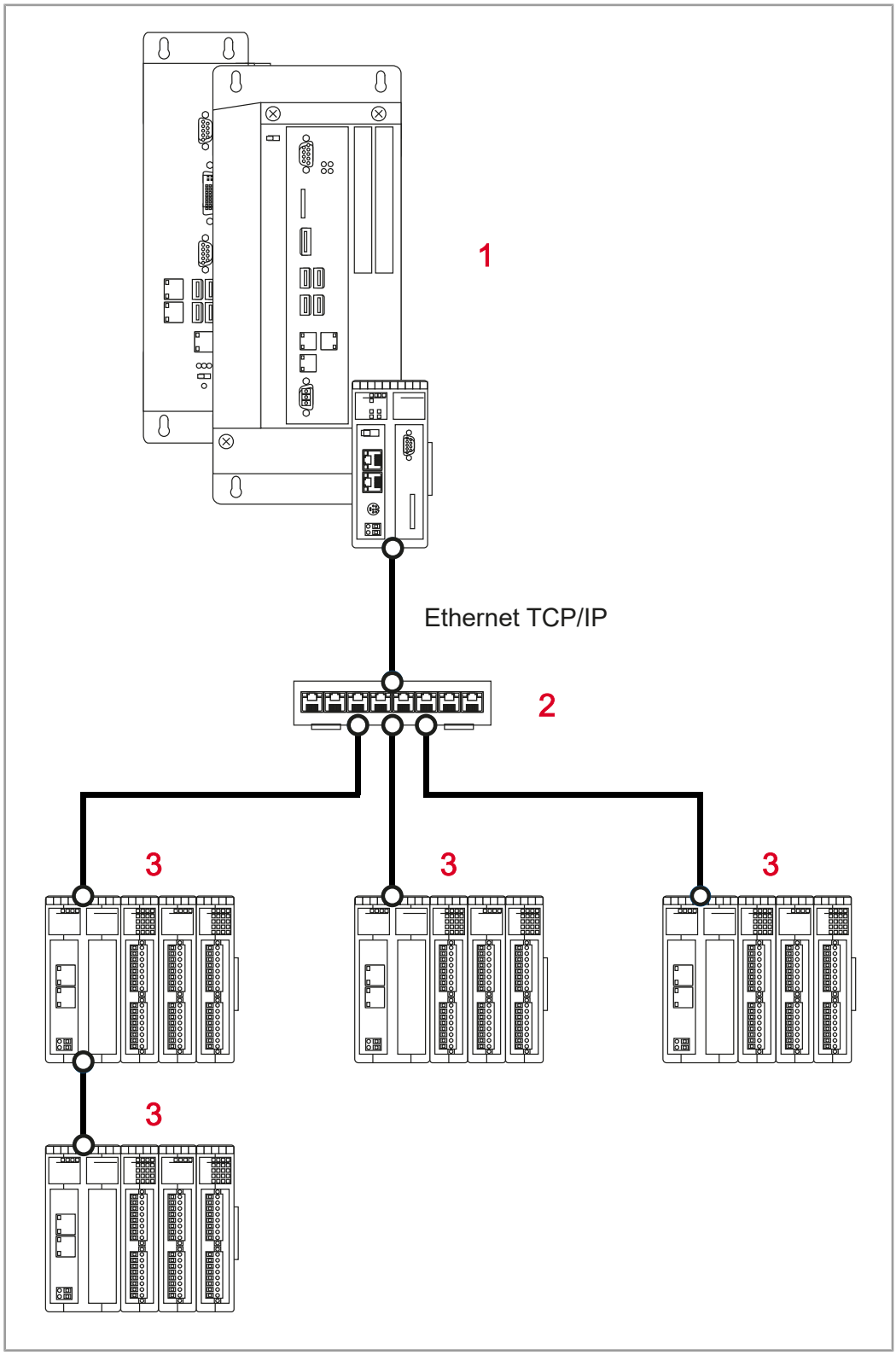


Fig. 23: System overview

1	Controller
2	Ethernet switch
3	Up to 199 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. 24: Example of a register number

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. 25: Example of an I/O number

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 in-depth information, see the *Ethernet System Bus* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

11.6.3 Expansion modules connected to an EtherCAT® bus node

EtherCAT® bus node and controller communicate via EtherCAT® system bus. When addressing the expansion modules via EtherCAT® bus node, the EtherCAT® Node Number (ENN) becomes part of the register number. The ENN is used to identify controllers and EtherCAT® bus nodes within an EtherCAT® network. The ENN is automatically assigned by JetSym.

System overview

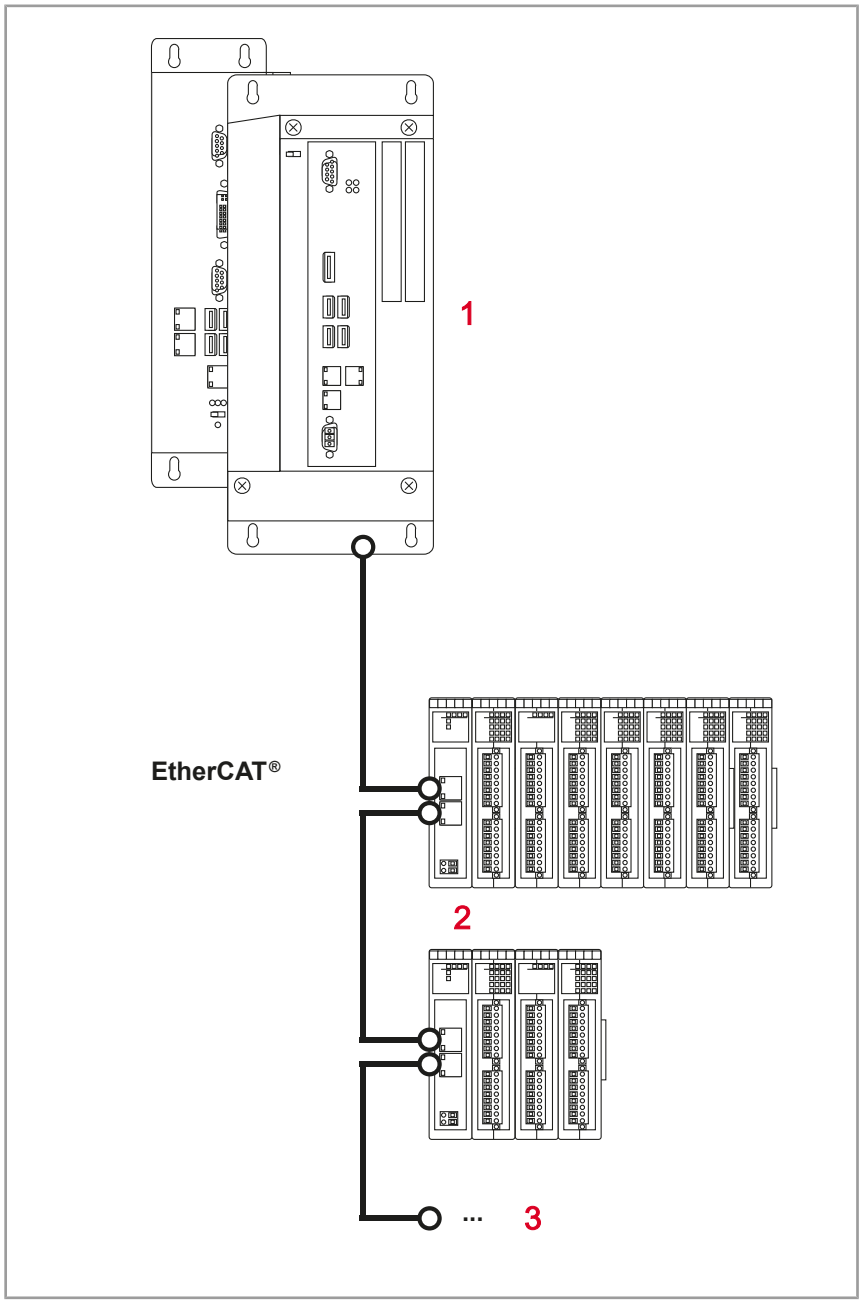


Fig. 26: System overview

1	EtherCAT® master
2	EtherCAT® bus node with up to 32 JX3 I/O modules
3	Up to 99 JX3-BN-EC

Register numbering system

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

12NNXXZZZZ

Fig. 27: Example of a register number

Digits	Description	Value range
12	Prefix	
NN	Bus node ID, ENN	01 … 99
XX	Position of the module within the station	02 … 33
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 ENN, prefix 2, the module position in the system and the module-specific I/O number.

12NN00XXZZ

Fig. 28: Example of an I/O number

Digits	Description	Value range
12	Prefix 1	
NN	Bus node ID, ENN	01 … 99
00	Prefix 2	
XX	Position of the module within the station	02 … 33
ZZ	Module-specific I/O number	01 … 16

11.7 System commands

The controller supports system commands and system registers.

ⓘ Info

Further information

For in-depth information, see the *System Registers* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

11.8 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).

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 in-depth information, see the *System Registers* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

11.9 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 in-depth information, see the *System Registers* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

11.10 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 in-depth information, see the *System Registers* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

11.11 Monitoring interface activities

The controller supports the *Monitoring interface activities* feature. This features allows the user to monitor communication between the Ethernet port and the controller from within the application program.

Monitored interface activities

The following interface activities can be monitored:

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

Purpose

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

– Facilities requiring process visualization to ensure safe operation. They can transition to a save state if communication fails.
– If a service technician connects an HMI, the application program automatically displays additional status information.

ⓘ Info

Further information

For in-depth information, see the *System Registers* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

11.12 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 in-depth information, see the *Ethernet System Bus* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

11.13 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.

Configuration

Access to process variables, such as target position, actual position, reference switch and limit switch, is provided by the JX3-IO modules. These connect remotely via JX3-BN-ETH or JX3-BN-EC bus nodes.

Engineering environment

To create the application program, use JetSym version 5.7 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 in-depth information, see *JCF-SV1 OS Function* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

11.14 E-mail feature

The user creates template files for emails, where the controller enters the corresponding variables when an email is 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. 66: Web status

ⓘ Info

Further information

For in-depth information, see the *File System* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

11.15 Modbus/TCP

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. 67: Web status

11.15.1 Modbus/TCP server

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

Booting the controller auto-starts the Modbus/TCP server.

Maximum number of connections

A maximum of 4 connections can be open at the same time.

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. 68: 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. 69: 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. 70: 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-975MC 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. 71: 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*.
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. 72: 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. 73: Supported commands – Class 2

11.15.2 Modbus/TCP client

The Modbus/TCP client integrated in the device supports Class 0 Conformance along with the fc23 function code of the Class 2 conformance requirement. These functions use 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.

This chapter describes how to carry out non-cyclic or cyclic transmission to a Modbus/TCP server using STX functions.

ⓘ Info

Outdated system features

As of JetSym version 5.0, the system features are outdated. Instead, apply the corresponding JetSym STX functions. For further information, refer to the JetSym online help.

Maximum number of connections

Modbus/TCP supports simultaneous connections to 36 different servers.

Noncyclic data transmission

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

Cyclic data transmission

Cyclic data transmission is made through the configurable function `RemoteScanConfig()`. Data is cyclically transmitted from and to the Modbus/TCP servers by means of STX variables.
The controller establishes a connection to each Modbus/TCP server (IP address and port) regardless of how many communication units are configured on this server.
If several communication units are configured on one Modbus/TCP server, access is serialized, because many servers 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.

ⓘ Info

Further information

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

11.16 User-programmable IP interface

The user-programmable IP interface allows for any data to be sent or received via the device's ETH1/X102 Ethernet interface using TCP/IP or UDP/IP, with the data being fully processed in the application program.

Applications

The user-programmable IP interface allows for data exchange via Ethernet connections that 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 in-depth information, see the *User-programmable Interface* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

11.17 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

Technology group

– Virtual leading axis
– External encoder as leading axis
– Cascaded groups
– Programming the axis group via the leading axis
– Dynamic coupling and decoupling of follower axes
– Print mark for high-precision position correction
– Up to 100 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 lead or follower 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 50 active path groups

ⓘ Info

Further information

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

11.18 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
– USB flash drive
– External SD memory card

Enabling the FTP server feature

On this device, the *FTP server* feature is permanently enabled.

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

FTPS

From OS version 1.22.0.00, “admin” users must log in via FTPS.

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. 74: Web status

ⓘ Info

Further information

For in-depth information, see the *File System* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

11.19 HTTP server

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

The browser allows you to read and display files that have been transfered 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.

Prerequisite

– For the HTTP server to be active, the basic application directory of the device must include a "*.html*" or "*.htm*" file.
– Only if either of the files is present, can the HTTP server answer any requests.
– Whether or not an .html or .htm file is present, is irrelevant for bit 1 in R202930.

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. 75: Web status

ⓘ Info

Further information

For in-depth information, see the *File System* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

11.19.1 Device Dashboard

The controller comes with the Device Dashboard installed.

The Device Dashboard is a browser-based software application designed to display in a user-friendly way key information and settings of a JC-4xx generation and higher Bucher Automation control system. The purpose of this application is to provide machine manufacturers and operators with additional first-level support resources.

ⓘ Info

The values displayed are read-only and may not be adjusted from within the application. To do this, the JetSym software tool is required.

System requirements

To use the Device Dashboard, the following requirements must be met:

– PC or mobile device running a standard browser.
– The control system and dashboard client are connected to the same IP network.
– The IP address of the controller is known.

Starting the Device Dashboard

ⓘ Info

Adjusting the IP address

Following initial call-up, the default IP address of a Bucher-Automation control system (192.168.1.1) usually needs to be replaced by the customer-specific IP address.

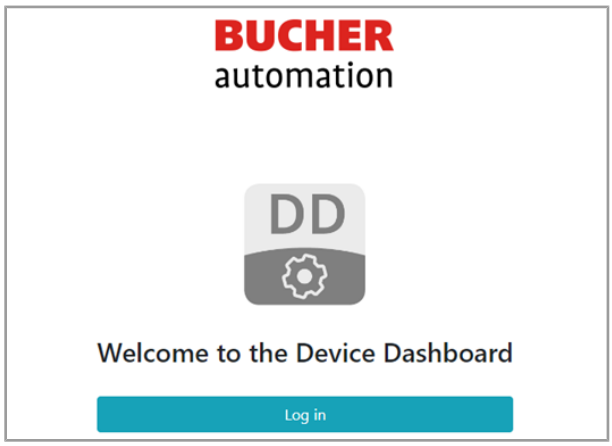
ⓘ Info

Protection against unauthorized access

It is recommended that the controller log-in credentials be changed to prevent unauthorized access to the Device Dashboard. To change the Device Dashboard log-in credentials, use the /system/users.ini file.

To open the Device Dashboard proceed as follows:

1. Enter the following URL in your browser's address bar:
   http://192.168.1.1/index.html
   ⇨ The login page is shown:



2. Click Log in.
3. Enter the login credentials. These are by default:
   User name: service
   Password: service
4. Click OK.

⇨ You are logged into the system. The homepage of the Device Dashboard is shown. By default, this is the Status page.

User interface layout

The main areas of the application's user interface are:

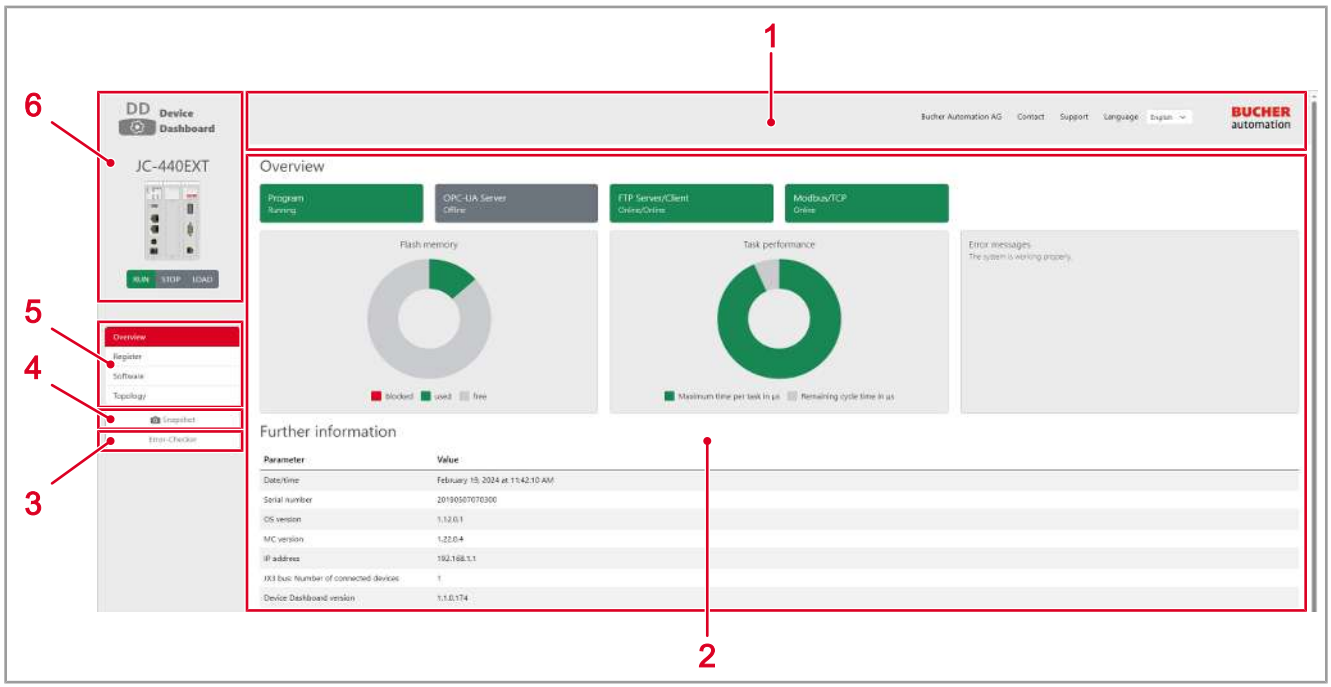


Fig. 29: User interface of the application

1	Header bar
2	Content area
3	Error checker feature
4	Snapshot feature
5	Menu
6	Status graphic

ⓘ Info

Further information

For further information, see the *Device Dashboard* application manual on www.bucher-automation.com.

ⓘ Info

Further information

For more information on this topic refer to the online help in our software products or to the application-oriented manuals on www.bucherautomation.com.

11.20 FTP client

The controller supports the *FTP client* feature. 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.

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. 76: Web status

ⓘ Info

Further information

For in-depth information, see the *File System* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

11.21 Automatic copying of controller data (AutoCopy)

The controller supports the *AutoCopy* feature. This feature allows for controller data to be automatically copied within one controller, and/or between the controller and an FTP server, any connected expansion modules and other controllers sharing the same network.

Use cases

AutoCopy allows for a new application program or operating system to be transfered from a USB flash drive to the controller.

Requirements

The AutoCopy feature is run only during boot-up of the JC-975MC. Prior to booting the device, the following requirements must be met:

– Mode selector S11 is in *LOAD* position.

– Only one USB flash drive is plugged in.

– The USB flash drive includes a valid command file containing the required copy commands.

  – In the **config.ini** file of the controller, the path and file name of the command file in the *[FILES]* section must be entered in **AutoCopyIni**. Access to the configuration file requires user credentials with system rights.

```
[FILES]
AutoCopyIni = /USB1/autocopy.ini
```

ⓘ Info

Further information

For in-depth information, see the *File System* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

11.22 MQTT client

The JC-975MC controller can be used as MQTT client. The MQTT client must be programmed via the STX application program.

MQTT protocol

The MQTT protocol is an open message protocol that has established itself as a standard in machine-to-machine communication and the Internet of Things. It is based on the publisher/subscriber architecture where a broker provides clients with messages about specific topics.

ⓘ Info

Further information

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

11.23 OPC UA server

The OPC UA server and client allow the user to access the following controller objects:

– Selected system registers
– Globally declared STX variables (non-localized)
– Globally declared STX variables (localized in the application register array of the controller)
– Method call-up

ⓘ Info

Further information

For in-depth information, see the *OPC UA* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

11.24 OPC UA client

Access to the OPC UA server is ensured via the OPC UA client. It is possible to configure the client as STX program in JetSym. It is based on the PLCopen standard.

ⓘ Info

Further information

For in-depth information, see the *OPC UA* application-oriented manual.

For application-oriented manuals open www.bucherautomation.com and go to *product page JC-975MC > Product-independent documentation*.

12 Registers - Overview

This register overview gives a condensed summary of the registers and flags of the JC-975MC device.

General overview – registers

Register range	Description
100000 ... 100999	Electronic Data Sheet (EDS)
101000 … 101999	Configuration
102000 … 102999	Real-time clock
104000 ... 104999	Ethernet
107500 ... 107599	Flash drive
108000 ... 108999	CPU
109000 … 109020	USB flash drives
200000 ... 209999	General system registers
210000 ... 219999	Application program
230000 ... 239999	JetIP networking
240000 ... 249999	JetSync
250000 ... 259999	Ethernet system bus
260000 ... 269999	RemoteScan
270000 ... 279999	Modbus/TCP
290000 ... 299999	Email
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 Axis count: yy = 00 … 16
1000000 ... 1119999	Application registers (non-volatile; integer/float)
50000000 ... 59999999	Motion control
Networking via Ethernet system bus GNN: nnn = 000 ... 199	
1nnn020000 ... 1nnn179999	JX3 module register
1nnn980000 ... 1nnn980199	Indirect access via local register 236xxx
1nnn990000 ... 1nnn999999	Indirect access with varying destination windows
JX3-IO module connected to JX3-BN-EC via EtherCAT® system bus	
12nn020000 ... 12nn339999	JX3-IO module register via JX3-BN-EC (ENN: nn = 01 ... 99)

I/Os – general overview

I/O numbers	Description
20001 ... 36000	Virtual I/Os for RemoteScan
20SJ00101 ... 20SJ00108	JX6-IO16CB

I/O numbers	Description
1Nnn010201 ... 1nnn011716	JX3-IO modules via JX3-BN-ETH (GNN: nnn = 000 ... 199)
12Nn000201 ... 12nn003316	JX3-IO modules via JX3-BN-EC (ENN: nn = 01 ... 99)

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)
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 … 100817	**[Features]** JC-975MC
100800	Internal version number
100801	MAC address (Bucher Automation)
100802	MAC address (device)
100803	Serial port
100804	Mode selector RUN-STOP-LOAD
100805	STX
100806	Non-volatile registers
100807	JX3 bus
100808	CAN bus
100810	Motion control
100811	Axes
100812	HTTP
100813	Modbus/TCP
100816	User-defined LEDs
100817	RTC
100825	USB
100826	LED for USB
100829	JCF-SV1 axis count

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
101180 ... 101198	File name for AutoCopy
Used by the system	
101200	IP address
101201	Subnet mask
101202	Default gateway
101203	DNS server

Registers	Description
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

Ethernet

Registers	Description
Link status	
104002	**Link status ETH1**
0	No link
10	10 MBit/s, half duplex
20	10 MBit/s, full duplex
100	100 MBit/s, half duplex
200	100 MBit/s, full duplex
1000	1000 MBit/s, half duplex
2000	1000 MBit/s, full duplex
104004	**Link status ETH2**
0	No link
10	10 MBit/s, half duplex
20	10 MBit/s, full duplex
100	100 MBit/s, half duplex
200	100 MBit/s, full duplex
1000	1000 MBit/s, half duplex
2000	1000 MBit/s, full duplex
104006	**Link status ETH3**
0	No link
10	10 MBit/s, half duplex
20	10 MBit/s, full duplex
100	100 MBit/s, half duplex
200	100 MBit/s, full duplex
1000	1000 MBit/s, half duplex
2000	1000 MBit/s, full duplex
MAC	
104111	ETH1 MAC address (manufacturer)
104112	ETH1 MAC address (device)
104121	ETH2 MAC address (manufacturer)
104122	ETH2 MAC address (device)
104131	ETH3 MAC address (manufacturer)
104132	ETH3 MAC address (device)
ARP	
104250	IP address for ARP request
104251	ARP cache: IP address
104252	ARP cache: MAC address (manufacturer)
104253	ARP cache: MAC address (device)
104254	ARP cache: TTL

Registers	Description
104350	GNN
IP	
104531	ETH1: Current IP address (rw)
104532	ETH1: Current subnet mask (rw)
104533	ETH1: Current default gateway (rw)
104534	IP address of DNS server (rw)

Registers	Description
Link status	
104002	**Link status ETH1**
0	No link
10	10 MBit/s, half duplex
20	10 MBit/s, full duplex
100	100 MBit/s, half duplex
200	100 MBit/s, full duplex
1000	1000 MBit/s, half duplex
2000	1000 MBit/s, full duplex
104004	**Link status ETH2**
0	No link
10	10 MBit/s, half duplex
20	10 MBit/s, full duplex
100	100 MBit/s, half duplex
200	100 MBit/s, full duplex
1000	1000 MBit/s, half duplex
2000	1000 MBit/s, full duplex
MAC	
104111	ETH1 MAC address (manufacturer)
104112	ETH1 MAC address (device)
104121	ETH2 MAC address (manufacturer)
104122	ETH2 MAC address (device)
ARP	
104250	IP address for ARP request
104251	ARP cache: IP address
104252	ARP cache: MAC address (manufacturer)
104253	ARP cache: MAC address (device)
104254	ARP cache: TTL

Registers	Description
104350	GNN
IP	
104531	ETH1: Current IP address (rw)
104532	ETH1: Current subnet mask (rw)
104533	ETH1: Current default gateway (rw)
104534	IP address of DNS server (rw)

Routing table	
104550	**Status**
0	No error
-1	Routing table is full
-2	Entry not found
-3	Port is not active
-4	TCP/IP stack not initialized
104551	**Command**
1	Add route
0	Delete route
104552	**Port number**
1	ETH1 (X102)
104553	IP address
104554	Subnet mask
104555	Gateway

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

Register range	Description
107521	Used
107522	Blocked
107523	Free

LEDs

Registers	Description
108002	**All LEDs on/off (bit-coded)**
Bit 0	RUN LED
Bit 1	ERR LED
Bit 2	D1 LED
108003	**RUN LED**
0	OFF
1	Flashing slowly
2	Flashing fast
3	ON
108004	**ERR LED**
0	OFF
1	Flashing slowly
2	Flashing fast
3	ON
108005	**D1 LED**
0	OFF
1	Flashing slowly
2	Flashing fast
3	ON

CPU

Registers	Description
108015	**Mode selector**
1	LOAD
2	RUN
3	STOP

General system registers

Registers	Description
200000	OS version (major * 100 + minor)
200001	**Application program is running (bit 0 = 1)**
0/2	Stops program
1	Starts program
3	Resumes program
200008	**Error register 1 (identical with 210004)**
Bit 0	Error on flash drive
Bit 2	Error on JX2 system bus
Bit 3	Error on Ethernet system bus
Bit 4	Error in application register
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 zero
Bit 13	Stack overflow
Bit 14	Stack underflow
Bit 15	Illegal stack
Bit 16	Error while loading the application program
Bit 17	Memory protection violated
Bit 24	Cycle time timeout
Bit 25	Task lock timeout
Bit 31	Unknown error
200009	**Enhanced error register 1 (bit-coded)**
Bit 1	Error in the bus system
Bit 2	DeviceManager error
Bit 3	Error in ModConfig.da
Bit 4	RPC error
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

Registers	Description
Bit 20	Internal memory error
Bit 21	Application program error
Bit 22	System logger is active (R 209700 = 213)
Bit 29	SNTP client error
Bit 30	DeviceManager register error
Bit 31	License violation
200010	**Enhanced error register 2 (bit-coded)**
Bit 1	Error in the MC object
Bit 2	JCF-SV1 error
Bit 3	NV machine data error
Bit 7	File system error
Bit 16	EtherCAT® initialization error
Bit 17	EtherCAT® master prompted error message
Bit 18	Error in EtherCAT® cycle
Bit 24	OPC UA server error
Bit 25	OPC UA client error
200051	JetIPScan **error numbers**
0	No error or warning
5	The user has terminated the function
1001	The response received first does not match responses 2 and 3
1002	The response received second does not match responses 1 and 3
1003	The response received third does not match responses 2 and 3
-1	All three responses are dissimilar
-2	The IP settings of at least one node are dissimilar
-3	The JetIPScan function was called while already active
-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 exists multiple times
-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	Error numbers of NetConsistency, see R 470040
200169	OS version (IP format)
200170	Controller type (940/970)
200300	Currently available heap
200301	Available heap at system launch
200302	Available heap ahead application program boot-up
201000	Runtime register in milliseconds (rw)
201001	Runtime register in seconds (rw)
201002	Runtime register in register 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 is available
Bit 1 = 1	HTTP server is available
Bit 2 = 1	E-mail is available
Bit 3 = 1	Data file function is available
Bit 4 = 1	Modbus/TCP has been licensed
Bit 5 = 1	Modbus/TCP is available
Bit 7 = 1	FTP client is available
202936	**Control register – file system**
0xc4697a4b	Format the flash drive
202960	Password for system command register (0x424f6f74)
202961	**System command register**
102	Controller restart (reboot)
103	Application register test
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	Resume task time slice OFF
171	Resume task time slice ON
301	Save the flash drive
310	Load the configuration files
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
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	Resume task time slice ON
202970	Password for start delay (0x424f6f74)
202971	Start delay in steps of 100 ms
203000	Interface monitoring: JetIP
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	System components enable

Networking via JetIP

Registers	Description
TCP Autoclose for JetIP**/TCP server**	
230000	Total of open connections
230001	Mode
230002	Time
Other registers for networking via JetIP	
232708	Timeout in milliseconds
232709	Response time in milliseconds
232710	Count of network errors
232711	**Error code of latest 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
232718	Total 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
1nnn980000 ... 1nnn980199	Indirect access via local register 236xxx
1nnn990000 ... 1nnn999999	Indirect access with varying destination windows

Ethernet system bus

Registers	Description
Subscriber	
250000	**Status (bit-coded)**
Bit 0 = 1	No CRC
Bit 1 = 1	Subscription error

Registers	Description
Bit 7 = 1	Subscriber is running
250001	**Command**
102	Restart
105	Stop
110	Acknowledge error
250002	Subscription ID of the latest error
250003	Total of subscriptions
250004	CRC of configuration file
250005	Start of communication (timeout register [ms])
250010	Selection via command
250011	Selection via ID
Subscription	
250020	Status
250021	Mode
250022	Total of elements
250023	Multicast group
250024	Hash
250025	Current sequence number
250026	Size (bytes)
250027	Timeout
250028	Total of received publications
250029	Total of timeout errors
250030	Total of sequence number errors
250031	Troubleshooting enabled
250032	State
250033	Current cycle time [ms]
250034	Minimum cycle time [ms]
250035	Maximum cycle time [ms]
250100 ... 250999	9 additional subscriber register blocks
Address of the timed-out bus node (or controller)	
254001	GNN
254002	IP address
254003	Port number
Publisher	

Registers	Description
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 latest error
255003	Total of publications
255004	CRC of configuration file
255010	Selection via command
255011	Selection via ID
Publication	
255020	Status
255021	Mode
255022	Total of elements
255023	Multicast group
255024	Hash
255025	Current sequence number
255026	Size (bytes)
255027	Cycle time
255028	Total of publications sent
255029	Total of retries
255030	Total of transmit errors
255033	Current cycle time [ms]
255034	Minimum cycle time [ms]
255035	Maximum cycle time [ms]
255100 ... 255999	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	Total of bytes sent
350007	Total of bytes received
350008	Total of bytes discarded
350009	Total of packets discarded
350016	Total of bytes sent last
350017	Total of bytes received last
350090	Maximum number of connections
350091	Total of open connections
350092	Total of open connections
350093	Total of closed connections
350099	Version number

Error history

Register	Description
380000	**Status**
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

Register	Description
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

Ethernet system bus 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 latest error
Address of bus node 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
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

Registers	Description
Bit 2 = 1	An instance has been initialized
Bit 3 = 1	Execution in process
471011	**Command**
0	There are no commands

DNS client

Registers	Description
510000	**Status**
Bit 0 = 0	**/etc/hosts** file was not read or is not existing
Bit 0 = 1	**/etc/hosts** file was read and entries were stored to the cache
Bit 1 = 0	Access to the DNS server failed
Bit 1 = 1	Access to the DNS server was successful
Bit 2 = 0	Access to the DNS server is not active
Bit 2 = 1	Access to the DNS server is active
Bit 3 = 0	No error on DNS server access
Bit 3 = 1	Error on DNS server access
510001	**Command**
1	Select the first entry in the cache
2	Select the next entry in the cache
3	Clear the entry from the cache
510002	Current IP address of DNS server
510003	**Error code**
-1	DNS server access error, e.g. fails to respond
-2	Invalid response of the DNS server
-3	No IP address could be retrieved from the response
510009	Total of entries in the DNS cache
510010	Timeout
510011	IP address of the cache entry called by command 1 or 2
510012 ... 510029	Displays the name of the cache entry selected by command 1 or 2 as register string.

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

SyncMaster

Registers	Description
530000	Status (bit-coded) Bit 0: SyncMaster started Bit 1: Component cycle error Bit 2: Fatal component cycle error Bit 3: Error: Component is not ready Bit 4: Fatal error: Component is not ready
530002	Length of scheduler table
530009	Software version
530012	Global error index

NV machine data

Registers	Description
560000	Status (bit-coded) Bit 0: feature enabled Bit 1: NV feature was initialized successfully Bit 2: File feature was initialized successfully Bit 3: Machine offsets in NV RAM are valid Bit 4: Encoder overflows in NV RAM are valid Bit 13: File feature is running Bit 16: Error while writing file Bit 17: Error while reading file Bit 18: Fatal error in file task
560002	Current file command
560003	File command state
560005	Last file command
560006	Outcome of latest file command
560007	Latest file command with error
560008	Outcome of latest file command with error
560009	Software version
560010	Machine offset: Total of data blocks
560011	Machine offset: Total of values per data block
560012	Machine offset: Data width of values
560013	Encoder overflows: Total of data blocks
560014	Encoder overflows: Total of values per data block
560015	Encoder overflows: Data width of values
560016	Total of read file operations
560017	Total of write file operations
560018	Duration of latest file operation [µs]

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
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

Register range	Description
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	

Register range	Description
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 ... 1119999	32-bit integer or floating point number (non-volatile)

Basic register for initializing the PCIe bus

Registers	Description
201100000	**Global status register**
Bit 30	Initialization completed
Bit 31	Error while initializing
201100001	**Command register**
1	Initialize the bus
201100002	Total of riser cards detected
201100005	**Bus initialization status**
0	Initialization is running
1	Initialization OK
-1	Error while initializing
201100006	Total of JX2 modules
201100007	JX6-**IO submodule type**
73	JX6-SB and JX6-SB-I
5	JX6-SV1
16	JX6-IO16CB
201100010	Timeout while waiting for command acknowledge

Registers	Description
201100011	Timeout while waiting for semaphore
201100012	Timeout in the interpreter for overall access

Digital I/O module JX6-IO16CB

Registers	Description
20SJ01100	State of the digital inputs
20SJ01101	State of the digital outputs
20SJ01102	Error state of the digital outputs
20SJ01103	Control register
20SJ01104	Filter time of inputs 1 … 4
20SJ01105	Filter time of inputs 5 … 8

Counter module JX6-SV1

Registers	Description
20SJ01100	Module ID
20SJ01101	Hardware configuration
20SJ01102	Analog output
20SJ01103	Strobe value/encoder value
20SJ01104	Preset/pulse generator
20SJ01105	Count value
20SJ01106	Reference/offset
20SJ01107	Control register
20SJ01108	Status register
20SJ01110	Sampling timer

32 combined inputs

Registers	Description
JX3-BN-ETH network: 1nnn910000 (nnn = GNN) JX3-BN-EC (EtherCAT®): n.a.	

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
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-BN-ETH network: 1nnn910000 (nnn = GNN) JX3-BN-EC (EtherCAT®): n.a.	

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
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

Regis-ters	Description
JX3-BN-ETH network: 1nnn910000 (nnn = GNN) JX3-BN-EC (EtherCAT®): n.a.	

4120	101 ... 108
4121	109 ... 116
4122	201 ... 208
4123	209 ... 216
4124	301 ... 308
4125	309 ... 316
4126	401 ... 408
4127	409 ... 416
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
4164	2301 ... 2308
4165	2309 ... 2316
4166	2401 ... 2408
4167	2409 ... 2416

32 combined outputs

Regis-ters	Description
JX3-BN-ETH network: 1nnn910000 (nnn = GNN) JX3-BN-EC (EtherCAT®): n.a.	

Example	Use the register 1001914202 to access outputs 1 ... 8 and 9 ... 16 of the JX3 module at positions 2 and 3 on a JX3-BN-ETH with GNN 001.			
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
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

Example	Use the register 1001914262 to 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
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-BN-ETH network: 1nnn910000 (nnn = GNN) JX3-BN-EC (EtherCAT®): n.a.	

Example	Use the register 1001914322 to access outputs 1 ... 8 of the JX3 module at position 2 on a JX3-BN-ETH with GNN 001.
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
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	JetIP networking error

Special flags – publish/subscribe

Flag	Description
2080	Enable for publishing an error
2081	Collective error message of the subscriber

Special flags – interface monitoring

Flag	Description
2088	JetIP OS flag
2089	JetIP user flag
2098	Debug server OS flag
2099	Debug server user flag

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

Registers	Description
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
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

Registers	Description
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

Registers	Description
1000043	1632 ... 1663
1000044	1664 ... 1695
1000045	1696 ... 1727
1000046	1728 ... 1759
1000047	1760 ... 1791
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	Sort 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	Initialize RemoteScan
81	Start RemoteScan
82	Stop RemoteScan
90	Write a data file
91	Append a data file
92	Read a data file
96	Delete a data file
150	Configure NetCopyList
151	Delete NetCopyList
152	Send 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: TNet-CopyLinstL): Int;
151	Function NetCopyListSend(Handle: int): Int;
152	Function NetCopyListDelete(Handle: int): Int;

13 Maintenance

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

13.1 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.

13.2 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.

13.3 Return and Disposal

Disposal options

Return your Bucher Automation AG product to us for proper disposal. Find detailed information along with the required Return Delivery Form on our website.

Meaning of WEEE icon



Fig. 30: 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.

14 Service

14.1 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 | www.bucherautomation.com

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.

15 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.

15.1 Accessories

ⓘ Info

Ordering Accessories

The accessories are not part of the scope of delivery.

Suitable accessories can be obtained from Bucher Automation AG.

15.1.1 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

15.1.2 PCI express expansion cards

The following PCIe expansion cards (JI-PCIE-Exx) are available as accessory for the JC-975MC.

For a description of the installation refer to chapter Mechanical installation [▶ 26].

Order reference	Description	Item number
JI-PCIE-E01*	PCIe riser card assembled with: 1 x JX2 system bus interface (JX6-SB-I-ES) connecting to up to 15 JetMove 100/200 servo amplifiers	10001522
JI-PCIE-E02*	PCIe riser card assembled with: 2 x JX2 system bus interface (JX6-SB-I-ES) connecting to up to 30 JetMove 100/200 servo amplifiers	10001523
JI-PCIE-E03*	PCIe riser card assembled with: 1 x JX2 system bus interface (JX6-SB-I-ES) connecting to up to 15 JetMove 100/200 servo amplifiers 1 x JX6-IO16CB featuring 16 digital, local I/Os for fast I/O processing	10001524

Order reference	Description	Item number
JI-PCIE-E04*	PCIe riser card assembled with: 1 x JX2 system bus interface (JX6-SB-I-ES) connecting to up to 15 JetMove 100/200 servo amplifiers 1 x JX6-SV1-ES with local interface connecting to an incremental or SSI encoder	10001525
JI-PCIE-E05	PCIe riser card assembled with: 1 x JX6-SV1-ES with local interface connecting to an incremental or SSI encoder	10001959
JI-PCIE-E06	PCIe riser card assembled with: 1 x JX6-IO16CB featuring 16 digital, local I/Os for fast I/O processing	10001962

***JetControl models with EtherCAT® do no longer support the JX6-SB-I-ES submodules. It is NOT possible to control JM-100 and JM-200 type servo amplifiers.**

15.1.3 USB flash drives

Component	Item number
USB flash drive XMORE, 4 GB, XQC8	60876836

Tab. 77: Accessories – USB flash drives

List of figures

Fig. 1 Internal architecture 10
Fig. 2 JC-970MC and JC-975MC system overview 11
Fig. 3 Front 12
Fig. 4 Housing 13
Fig. 5 Status indication 14
Fig. 6 PWR and RSQ LED 15
Fig. 7 Example of a nameplate 16
Fig. 8 Dimensions in mm 17
Fig. 9 PCI express bus slots 26
Fig. 10 Power supply, port X101 31
Fig. 11 DisplayPort interface 34
Fig. 12 SD memory card 34
Fig. 13 Serial interface RS-232 35
Fig. 14 Sub-D connector 36
Fig. 15 Sub-D connector 37
Fig. 16 S11 mode selector positions 38
Fig. 17 PWR button 39
Fig. 18 RSQ button 39
Fig. 19 Router – in-house network 54
Fig. 20 JetIPScan command options 57
Fig. 21 Configuration example 64
Fig. 22 Plug-in slots 75
Fig. 23 System overview 88
Fig. 24 Example of a register number 89
Fig. 25 Example of an I/O number 89
Fig. 26 System overview 90
Fig. 27 Example of a register number 90
Fig. 28 Example of an I/O number 91
Fig. 29 User interface of the application 104
Fig. 30 WEEE icon – crossed out trash can 132

List of tables

Tab. 1 Status indication 14
Tab. 2 LED states during the boot process 15
Tab. 3 Mechanical specifications 18
Tab. 4 Power supply (port X101) 18
Tab. 5 ETH Ethernet port (X102, X104) 18
Tab. 6 EtherCAT® port (X103) 19
Tab. 7 USB port (X105, X106) 19
Tab. 8 USB port (X107, X108) 19
Tab. 9 Port JX6-SV1-ES 20
Tab. 10 Port JX6-IO16CB 20
Tab. 11 SD memory card specification 21
Tab. 12 RS-232 interface specification 21
Tab. 13 Electrical safety 21
Tab. 14 Real-time clock 21
Tab. 15 Environmental conditions 22
Tab. 16 Emitted interference 23
Tab. 17 Immunity to interference 23
Tab. 18 Immunity of shielded data and I/O lines 23
Tab. 19 DC power supply inputs and outputs 24
Tab. 20 Rotary switch position 26
Tab. 21 Slots interfacing with peripherals 35
Tab. 22 JX6-SV1-ES pin assignment 36
Tab. 23 JX6-IO16CB pin assignment 37
Tab. 24 S11 mode selector positions and operating mode 38
Tab. 25 [IDENTIFICATION] section 41
Tab. 26 [PRODUCTION] section 41
Tab. 27 [FEATURES] section 42
Tab. 28 EDS register numbers 42
Tab. 29 EDS register summary 42
Tab. 30 Overview of hardware revision registers 43
Tab. 31 Software version number format 44
Tab. 32 Register overview 44
Tab. 33 Default values 45
Tab. 34 Address 48
Tab. 35 SubnetMask 48
Tab. 36 DefGateWay 48
Tab. 37 RouteIP 49
Tab. 38 RouteMask 49
Tab. 39 RouteGateway 49

Tab. 40 DNSServer 49
Tab. 41 SuffixType 50
Tab. 42 Name 50
Tab. 43 JetIPBase 50
Tab. 44 JVMDebug 50
Tab. 45 AutoCopyIni 51
Tab. 46 Configuration register overview 51
Tab. 47 Status 53
Tab. 48 Command 53
Tab. 49 Port number 53
Tab. 50 IP address of the static route 53
Tab. 51 Subnet mask of the static route 53
Tab. 52 Gateway of the static route 53
Tab. 53 System directories 61
Tab. 54 Status indication 64
Tab. 55 Abbreviations 66
Tab. 56 Module register properties 66
Tab. 57 Numerical formats 66
Tab. 58 JetSym sample programs 66
Tab. 59 Register numbers 76
Tab. 60 I/O numbers 76
Tab. 61 I/O numbers of local JX6-IO submodules 76
Tab. 62 Register numbers of local JX6-IO submodules 77
Tab. 63 Local JX6-IO module register overview 77
Tab. 64 Module register numbers of the JX6-SV1-ES counter module 80
Tab. 65 JX6-SV1 module register overview 81
Tab. 66 Web status 94
Tab. 67 Web status 95
Tab. 68 Register offset 96
Tab. 69 Input offset 96
Tab. 70 Output offset 96
Tab. 71 Supported commands – Class 0 97
Tab. 72 Supported commands – Class 1 97
Tab. 73 Supported commands – Class 2 98
Tab. 74 Web status 101
Tab. 75 Web status 102
Tab. 76 Web status 105
Tab. 77 Accessories – USB flash drives 135

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

www.bucherautomation.com