



User Manual

JVM-104-O08

HMI

60884475_01

This document has been compiled by Jetter 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. Jetter 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.



Jetter AG

Graeterstrasse 2
71642 Ludwigsburg
Germany

Phone

Switchboard	+49 7141 2550-0
Sales	+49 7141 2550-663
Technical hotline	+49 7141 2550-444

E-mail

Technical hotline	hotline@jetter.de
Sales	sales@jetter.de

www.jetter.de

Translation of the german original User Manual

Revision	4.12.1
Date of issue	4/3/2023

Table of contents

- 1 Introduction 5**
 - 1.1 Information on this document 5
 - 1.2 Typographical conventions 5
- 2 Safety 6**
 - 2.1 General Information 6
 - 2.2 Purpose 6
 - 2.2.1 Intended use 6
 - 2.2.2 Usage other than intended 6
 - 2.3 Warnings used in this document..... 7
- 3 Product description 8**
 - 3.1 Design..... 8
 - 3.2 Product features 8
 - 3.3 Nameplate 9
 - 3.4 Scope of delivery 9
- 4 Technical data..... 10**
 - 4.1 Dimensions 10
 - 4.2 Mechanical specifications 10
 - 4.3 Electrical properties 11
 - 4.3.1 Ports and interfaces..... 11
 - 4.4 Environmental conditions..... 12
 - 4.5 Display 12
 - 4.6 Acoustic signal generator 12
 - 4.7 EMI values 13
- 5 Mechanical installation 14**
 - 5.1 Requirements for the installation location..... 15
 - 5.2 Preparing for installation 16
 - 5.3 Installing the HMI 17
- 6 Electrical connection 19**
 - 6.1 Pin assignment 20
 - 6.1.1 M12 male connector – voltage supply, CAN, ignition 20

7	Identification and Configuration	21
7.1	Operating system.....	21
7.1.1	Operating system update of the HMI	21
7.2	File system.....	24
7.2.1	Features	24
8	Programming	26
8.1	Abbreviations, module register properties and formats	26
8.2	CANopen STX API	27
8.2.1	STX Functions	27
8.2.2	CANopen object directory.....	28
8.3	Storage options - Overview	29
8.3.1	Operating system memory	29
8.3.2	File system memory	29
8.3.3	Application program memory.....	30
8.3.4	Special registers	30
8.3.5	Flag.....	30
8.3.6	Storing registers and variables	31
8.4	Control elements.....	32
8.4.1	Input keys	32
8.4.2	Digipot	33
8.5	Ignition and OFF delay	34
8.6	Saving and loading an application program	36
9	Registers - Overview.....	37
10	Maintenance	45
10.1	Repairs	45
10.2	Return and disposal.....	45
10.3	Storage and shipment.....	46
11	Service	47
11.1	Customer service.....	47
12	Spare parts and accessories	48
12.1	Accessories	48

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 is 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 | Jetter - We automate your success.](#)

For further information refer to the following information products:

- JetSym software Online Help
Detailed description of software functions with application examples
- Application-oriented manuals
Cross-product documentation
- Version updates
Information about new versions of software products or of the operating system of your device

1.2 Typographical conventions

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

- ✓ This symbol indicates requirements which have to be met before executing the following action.
- ▶ This sign or a numbering at the beginning of a paragraph marks an action instruction that must be executed by the user. Execute the instructions one after the other.
- ⇒ The target after a list of instructions indicates reactions to, or results of these actions.

 INFO

Further information and practical tips

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

2 Safety

2.1 General Information

When placed on the market, this product corresponds to the current state of science and technology.

In addition to the operating instructions, the laws, regulations and guidelines 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 accepted safety rules.

RoHS 2

The device conforms to the EU directive 2011/65/EU (RoHS 2).

2.2 Purpose

2.2.1 Intended use

The JVM-104-O08 is intended for installation in commercial vehicles and self-propelled machines. It is an HMI with integrated controller for exchange of data with peripheral devices.

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/PELV

If this device is not used in a vehicle or mobile machine, a SELV or PELV power supply unit must be used to supply the device.

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 mal-functions or material damage.

3 Product description

The JVM-104-O08 is a graphical HMI. Thanks to its compact design and integrated controller, the HMI is versatile and has been specially developed for harsh applications in commercial vehicles and self-propelled machines.

3.1 Design

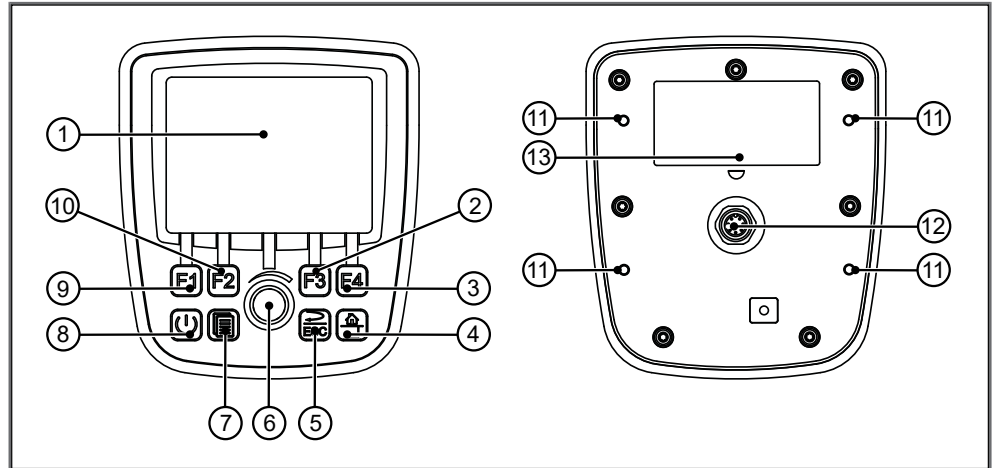


Fig. 1: Design

1	TFT display
2	F3 key
3	F4 key
4	Home key
5	ESC key
6	Digipot
7	Scroll key
8	Power key
9	F1 key
10	F2 key
11	Screw hole
12	M12 male connector
13	Nameplate

3.2 Product features

- Display: 3.5" TFT, 320 x 240 px
- CPU: ARM11, 500 MHz, 128 MB RAM, 512 MB Flash
- Surface mount model with connector M12x1
- Input: 8 keys, 1 DigiPot
- Ports and interfaces: 1x CAN
- Front/rear degree of protection: IP65/IP65

3.3 Nameplate

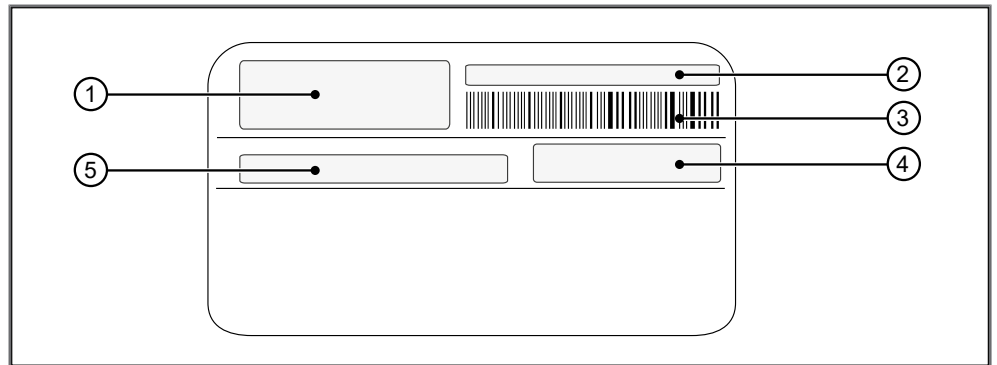


Fig. 2: Nameplate

1	Logo
2	Serial number
3	Barcode
4	Article number and hardware revision
5	Model code number

3.4 Scope of delivery

Scope of delivery	Item number	Quantity
JVM-104-O08	10001558	1

4 Technical data

This chapter contains electrical, mechanical data and operating data of the JVM-104-O08.

4.1 Dimensions

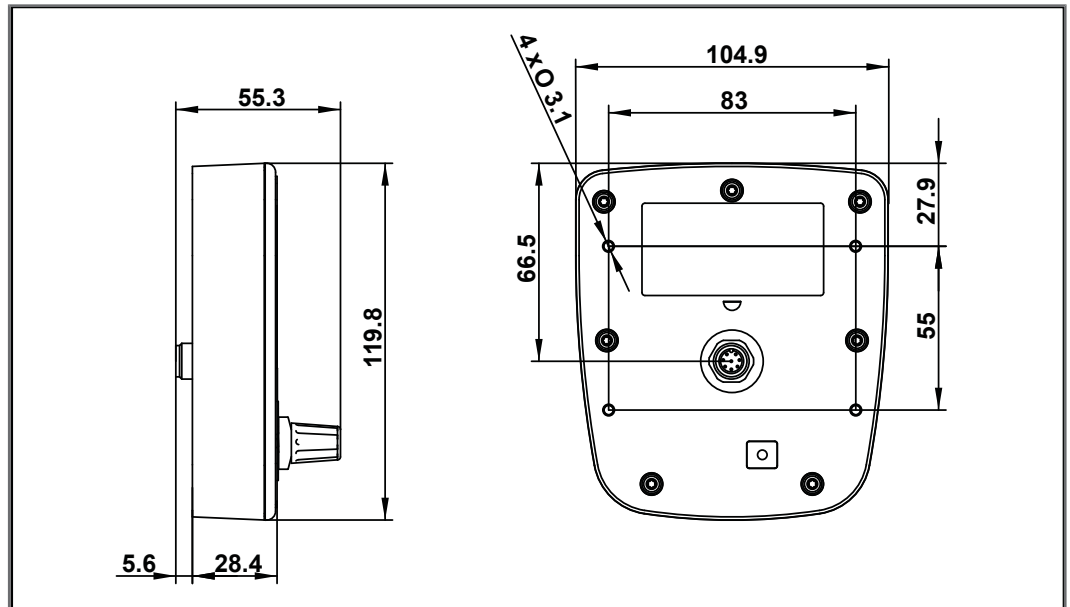


Fig. 3: Dimensions in mm

4.2 Mechanical specifications

Parameter	Description	Standards
Weight	350 g	
Vibration		
Broadband noise	10 Hz / 0.005 (m/s ²) ² /Hz	DIN EN 60068-2-64
	200 Hz / 0.02 (m/s ²) ² /Hz	
	300 Hz / 0.01 (m/s ²) ² /Hz	
	350 Hz / 0.002 (m/s ²) ² /Hz	
Duration	3x 24 h	
Shock resistance		
Type of shock	Half-sine wave	DIN EN 60068-2-27
Intensity and duration	30 g (300 m/s ²) for 18 ms	
Number and direction	18 shocks in all 6 directions	
Degree of protection		
Front panel	IP65	DIN EN 60529
Rear panel	IP65	

Tab. 1: Mechanical specifications

4.3 Electrical properties

Power supply VBAT_ECU

Parameter	Description
Nominal voltage	DC 12 V or 24 V
Maximum current	2 A
Typical logic current consumption (VBAT_ECU)	170 mA at DC 12 V 90 mA at DC 24 V
Power consumption	Approx 2 W
Integrated protective functions	Reverse polarity protection, overvoltage, short voltage pulses
Permissible voltage range VBAT_ECU	DC 8 V ... 32 V
Permissible voltage range at terminal 31 (ignition)	DC 8 V ... 32 V

Tab. 2: Power supply VBAT_ECU

4.3.1 Ports and interfaces

CAN interface

Parameter	Description
Baud rate	125 kBaud, 250 kBaud, 500 kBaud
Protocol	CANopen
Default node ID on the CANopen bus	127 (0x7F)
Terminating resistance	Does not exist. Must be connected externally.
Cable specification	Twisted pair conductors, unshielded

Tab. 3: CAN interface specification

CAN bus cable

Parameter	Description
Wire cross-section	500 kBaud: 0.34 mm ² ... 0.50 mm ² 250 kBaud: 0.34 mm ² ... 0.60 mm ²
Cable capacitance	60 pF/m max.
Resistivity	500 kBaud: Max. 60 Ω/km 250 kBaud: Max. 60 Ω/km
Number of cores	2
Twisting	CAN_L and CAN_H cables are twisted pairwise

Tab. 4: Specification - CAN bus cable

The maximum permitted cable length depends on the baud rate being used and the number of CANopen devices being connected.

Baud rate	Max. cable length	Max. stub length	Total cable length
500 kBaud	100 m	5 m	30 m
250 kBaud	250 m	10 m	60 m

Tab. 5: Cable lengths

4.4 Environmental conditions

Parameter	Description	Standards
Operating temperature	-20 °C ... +65 °C	ISO 16750-4
Climatic conditions	Humid heat	
Storage temperature	-20 °C ... +70 °C	ISO 16750-4 DIN EN 60068-2-1 DIN EN 60068-2-2
Relative humidity	10 % ... 95 %	DIN EN 61131-2
Pollution degree	2	DIN EN 61131-2

Tab. 6: Environmental conditions

4.5 Display

Parameter	Description
Type	TFT LCD flat screen monitor
Resolution	320 x 240 Pixel
Size	3.5"
Backlighting	LED, typ. 350 cd/m ² , dimmable
Horizontal viewing angle	70° to each side
Vertical viewing angle	50° from above, 70° from below

Tab. 7: Technical data – display

4.6 Acoustic signal generator

Parameter	Description	
Type	Loudspeaker	Adjustable frequency and volume.
Volume	83 dB	10 cm distance and resonance frequency 2,670 Hz

Tab. 8: Acoustic signal generator

4.7 EMI values

The JVM-104-O08 has E1 approval according to ECE R10 Rev. 5 and CE conformity according to ISO 14982.

Pulses ISO 7637-2

Test pulse	Values	Function class
1	-450 V	C
2 a	+37 V	A
2b	+20 V	C
3 a	-150 V	A
3b	+150 V	A
4	Ua1: -12 V / 50 ms Ua2: -5 V / 500 ms	A (24 V systems)
	Ua1: -6 V / 15 ms Ua2: -2.5 V / 1,000 ms	C (12 V systems)
5b	Load dump, capped 70 V / 2 Ω	A

Tab. 9: Pulses ISO 7637-2

Irradiation ISO 11452

Parameter	Values	Function class
Irradiation	20 MHz ... 2 GHz 30 V/m	A

Tab. 10: Irradiation ISO 11452

ESD EN 61000-4-2

Parameter	Values	Function class
Contact discharge	±4 kV (to conductive surfaces)	A
Discharge through air	±8 kV (to insulating surfaces)	A

Tab. 11: ESD EN 61000-4-2

5 Mechanical installation

NOTICE



Damages to material or functional impairment due to welding

Welding on the chassis may damage the device material, or impair device functions.

- ▶ Before you start welding, disconnect all connections between the device and the electric system of the vehicle.
- ▶ Protect the device from flying sparks and welding beads (splatter).
- ▶ Do not touch the device with the welding electrode or earth clamp.

NOTICE



Dirt and moisture can affect the electrical connections

- ▶ Protect unused pins using blanking plugs.
- ▶ Protect all electrical connections with appropriate single wire seals.
- ▶ Clean the area around a connector prior to removing the mating connector.

5.1 Requirements for the installation location

The installation location must meet the following requirements:

- The installation location must allow air to circulate.
- The installation location must be of sufficient size.
- The device must be easily accessible to allow for service work.

Space required for installation and service

It should be possible to disconnect the connectors at any time.

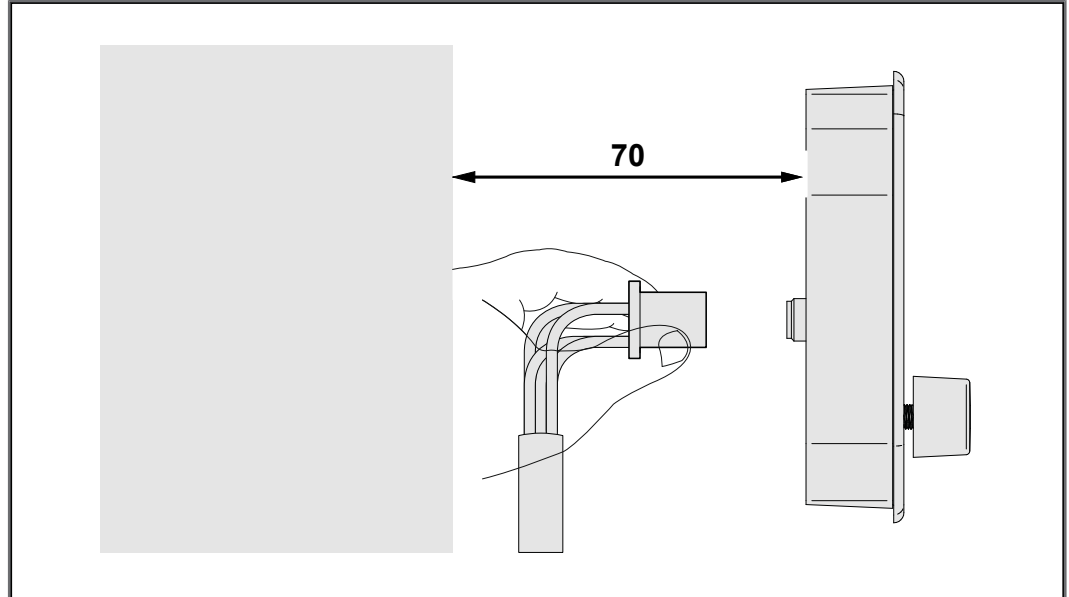


Fig. 4: Space requirements for installation work (in mm)

Avoiding unsuitable installation locations

The following installation locations are unsuitable for mounting the device:

Unsuitable installation location	Reason
Outdoor installation	The device must not be exposed to rain or a jet of water. Do not use a steam jet or other such devices to clean the device.
Installation location close to heat-sensitive materials	The materials could become warped or misshapen as a result of heat produced by the device.

Tab. 12: Unsuitable installation locations

5.2 Preparing for installation

Mounting accessories

Use the following accessories for installation:

Accessories	Item number
Mounting plate for RAM Mount ball consisting of mounting plate and screws for housing with Deutsch or M12 connector, without RAM Mount attachments	10001621
OR	
Mounting plate for RAM mount arm with suction cup consisting of mounting plate and screws for housings with Deutsch/M12 connector including RAM Mount arm with suction cup	10001551

Screw holes

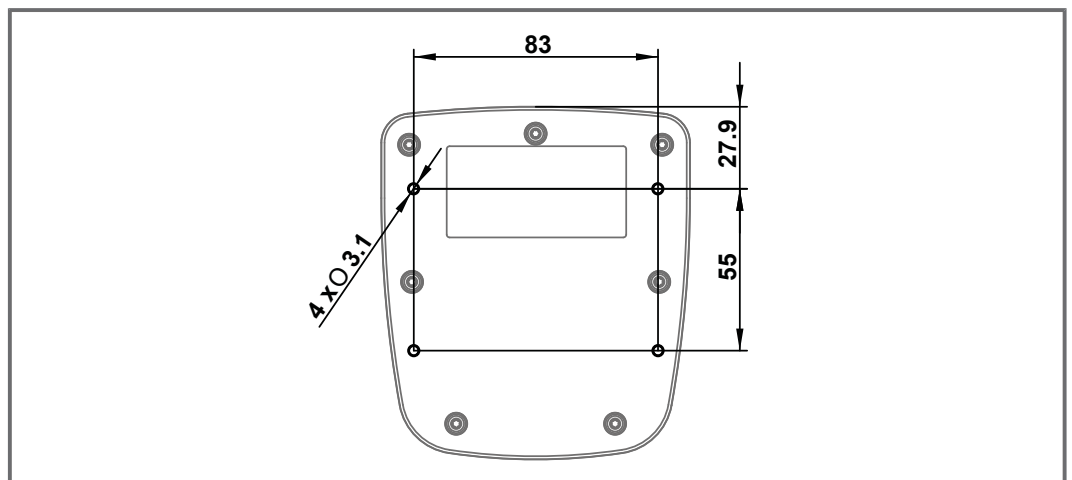


Fig. 5: Screw holes, dimensions in mm

5.3 Installing the HMI

The illustration below shows how to install the device:

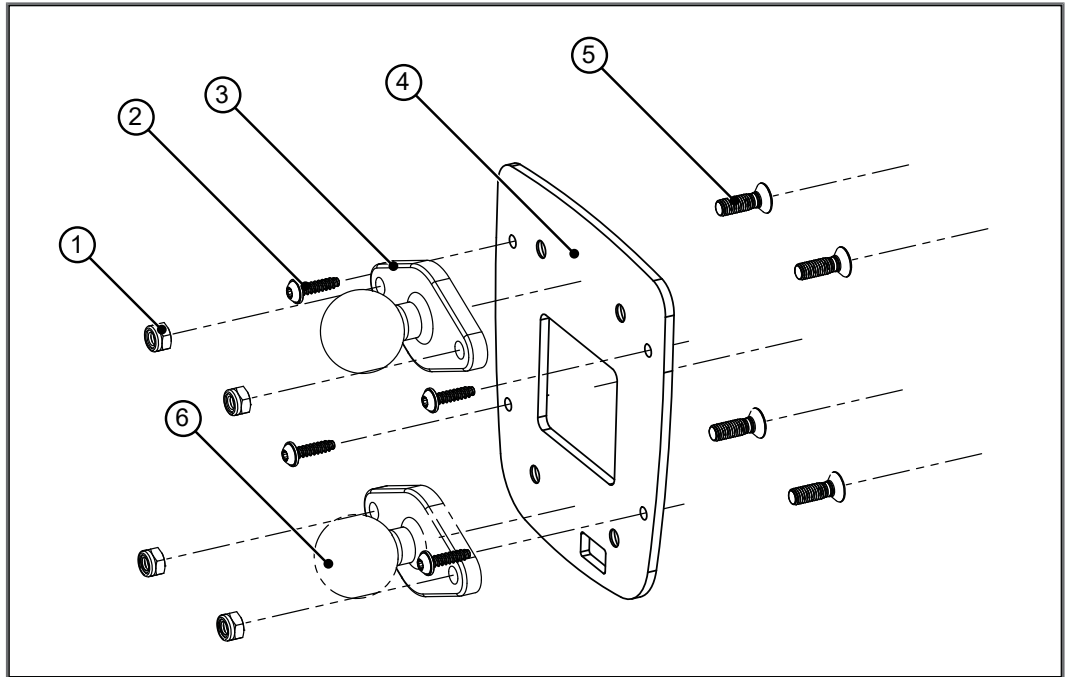


Fig. 6: Installation drawing

1	2 x self-locking nuts
2	4 x screw for fixing to the JVM-104-O08
3	RAM Mount ball
4	Mounting plate with opening for connector
5	2 x countersunk screws for mounting a RAM Mount ball
6	Alternate position of the RAM Mount ball

1. Screw the desired RAM Mount attachments onto the mounting plate.
2. Hold the JVM-104-O08 against the mounting plate from behind. The connectors must be accessible through the openings in the mounting plate.
3. Screw the mounting plate onto the JVM-104-O08.

Installing the strain relief

Install strain reliefs for the connecting cables.

- Ensure that there is sufficient clearance between the strain reliefs and the connectors.
- Connectors must not be obstructed, so that they can be removed in the event of service.

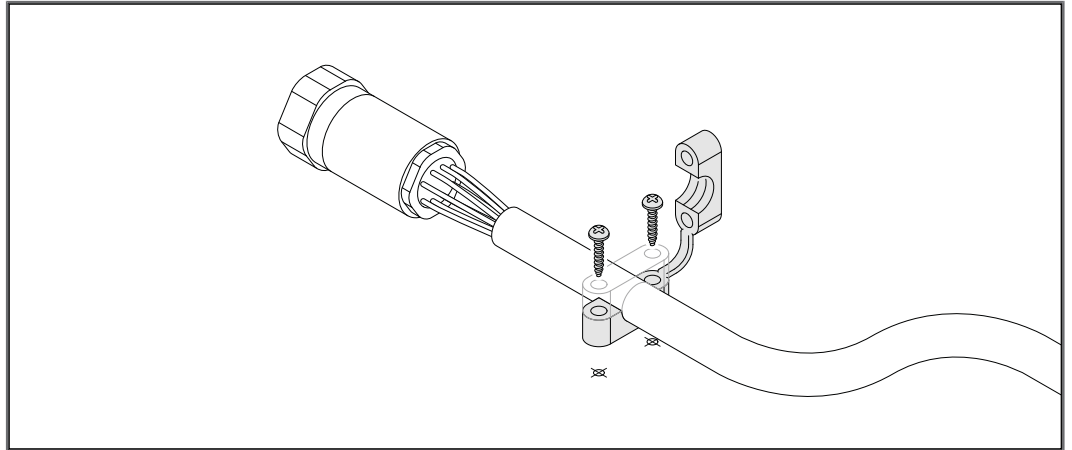


Fig. 7: Installing the strain relief

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.

NOTICE



Surges resulting from missing protection or fusing

Surges may cause malfunctions or damage to the product.

- ▶ Protect the voltage inputs from surges according to the requirements.
- ▶ Ensure that the device is handled in accordance with ESD regulations.

6.1 Pin assignment

6.1.1 M12 male connector – voltage supply, CAN, ignition

Function

The M12 connector has the following function:

- Power supply to the JVM-104-O08
- CANopen bus interface: CAN 1
- Recognition of the ignition signal

i INFO

Ignition

To launch the JVM-104-O08, pin 3 (Ignition+) must be connected with pin 1. The ignition control signal (Ignition+) is issued when the key is in position *Ignition ON*.

i INFO

Current consumption

When the device is energized, the current consumption is temporarily higher. To ensure a reliable start-up of the device, provide at least 3 times the typical current required.

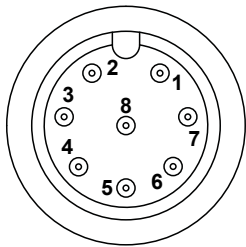


Fig. 8: 8-pin M12 male connector

Pin	Signal
1	VBAT_ECU
2	n. c.
3	IGN (ignition)
4	n. c.
5	CAN_L
6	GND
7	CAN_H
8	SHLD

Mating part

The following jack is a mating part to the 8-pin M12 connector:

Parameter	Description
Manufacturer	e.g. BELDEN Lumberg automation
Manufacturer's item no.	RKCN 8/9
Wire size range	0.5 mm ² (AWG 20)

Tab. 13: Compatible M12 connector (female)

7 Identification and Configuration

7.1 Operating system

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

INFO

Further information

More information on this subject is available on our website.

[Start | Jetter - We automate your success.](#)

7.1.1 Operating system update of the HMI

This chapter describes how to update the operating system of a JVM-104-O08 HMI. There are several options to transfer the operating system file to the HMI:

- Using the programming tool JetSym
- Via JetEasyDownload

Operating system update with programming tool

The programming tool JetSym offers an easy way to transfer an OS file to the JVM-104-O08.

Carrying out the update

- ✓ An OS file for the JVM-104-O08 is available.
- ✓ The device is connected to the PC via CAN.
- ✓ In JetSym the following parameters are set:
Device type, interface type, node ID, CAN baud rate
- ✓ The device is running during the operating system update.
- 1. Select in the JetSym menu **Build** menu item **Update OS...**
 - ⇒ A file selection window 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. To activate the newly installed OS, re-boot the device.

OS update via JetEasyDownload

To update the operating system of this device, use a PEAK CAN dongle and the Jetter command line tool JetEasyDownload (version 1.00.0.15 or higher).

JetEasyDownload Parameters

To call JetEasyDownload you need specific parameters.

Parameter	Description	Values	
-H<Num>	Hardware	0=	PCAN_ISA1CH
		1=	PCAN_ISA2CH
		2=	PCAN_PCI_1CH
		3=	PCAN_PCI_2CH
		4=	PCAN_PCC_1CH
		5=	PCAN_PCC_2CH
		6=	PCAN_USB_1CH
		7=	PCAN_USB_2CH
		8=	PCAN_Dongle Pro
		9=	PCAN_Dongle
		10=	PCAN_NET Jetter
		11=	PCAN_DEV default device
		20=	IXXAT V2.18
22=	IXXAT V3		
100=	CAN hardware detected first		
-T<nodeID>	Target node ID	The node ID is decimal.	
-B<Num>	Baud rate Observe the permissible baud rates of your device!	0=	10 kB
		1=	20 kB
		2=	50 kB
		3=	100 kB
		4=	125 kB
		5=	250 kB
		6=	500 kB
7=	1 MB		
-S<Num>	SDO timeout	De- fault	300 ms
-L<name>	OS filename	e.g. JVM-104-O08_Vx.xx.x.xx.os	

Tab. 14: JetEasyDownload Parameters

Carrying out the update

JetEasyDownload -H100 -T127 -B5 -S8000 -Ljvm_ce0_X.XX.X.XX.os

- ✓ JetEasyDownload and PEAK CAN dongle are ready for use.
- ✓ There is a CAN connection between PEAK CAN dongle and JVM-104-O08.
- 1. Call up JetEasyDownload with the above parameters and a valid OS file.
 - ⇒ The device carries out a reset.
 - ⇒ The device starts in boot loader mode with a single heartbeat in init state (data = 0x00).
- 2. Wait for approx. 7 seconds while the device formats the flash memory.
 - ⇒ The device starts the download process.
- ⇒ The device starts automatically with the new firmware.

7.2 File system

The file system allows for access to the files on the internal flash disk. 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.

Division

The file system distinguishes between the following file types:

- System directories/files used by the operating system
- Files used by the user

System directories

System directories cannot be deleted. They are still available after formatting.

Directory	Description
\System	<ul style="list-style-type: none"> ■ System configuration ■ System information ■ Boot screen (bootupscreen.bmp) ■ Screenshot
\App	Directory for applications
\Data	Folder for storing data
\Windows	Windows CE system directory
\	RAM disk drive

INFO

Further information

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

7.2.1 Features

The following properties apply to the internal flash disk:

- Open a maximum of 8 files at the same time.
- Separate directory names with a slash "/" rather than a backslash "\".
- Any file created by the controller contains the date and time of the device.
- 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:

Parameter	Value
Flash disk capacity	512 MB
Of which folder App	64 MB
Of which folder Data	368 MB

Features

The internal flash disk drive has got the following properties:

- Up to 7 directory levels and 1 file level are allowed.
- No case sensitivity.
- Directory and file names with a length of up to 63 characters are possible.
- All characters except "/" and ".." are permitted for directory and file names.
- The location of the directories "App" and "Data" is on the flash disk drive.

8 Programming

8.1 Abbreviations, module register properties and formats

Abbreviations

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

Abbreviation	Description
R 100	Register 100
MR 150	Module register 150

Tab. 15: 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
Types of access	Read/write
Value after reset	0 or undefined (e.g. revision/version number)
Takes effect	Immediately
Write access	Always
Data type	Integer

Tab. 16: 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. 17: 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
<code>Var, When, Task</code>	Keyword
<code>BitClear();</code>	Commands
<code>100 0x100 0b100</code>	Constant numerical values
<code>// This is a comment</code>	Comment
<code>// ...</code>	Further program processing

Tab. 18: JetSym sample programs

8.2 CANopen STX API

The CANopen STX API is a platform allowing users to send and receive CANopen messages via STX functions.

The device uses the STX functions to communicate with other CANopen nodes.

The CANopen standard

CANopen is an open standard for networking and communication, used e.g. in automotive applications. The CANopen protocol is being further developed by the CiA e.V. (CAN in Automation). On the physical layer it uses CAN high-speed in accordance with ISO 11898.

Specifications

The CANopen specifications can be obtained from the **CiA e.V.** homepage <http://www.can-cia.org>.

The key specification documents are:

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

INFO

Further information

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

8.2.1 STX Functions

Application

STX functions are used to enable communication between the JVM-104-O08 and other CANopen nodes.

The JVM-104-O08 supports the following STX functions:

Function	Description
CanOpenInit()	Calling up the CanOpenInit() function initializes one of the available CAN buses.
CanOpenSetCommand()	The function CanOpenSetCommand() lets you change the heartbeat status of the device itself and of all other devices (NMT slaves) on the CAN bus.
CanOpenUploadSDO()	The function CanOpenUploadSDO() lets you access a particular object in the Object Directory of the message recipient and read the value of the object. Data is exchanged in accordance with the SDO upload protocol.

Function	Description
CanOpenDownloadSDO ()	The function CanOpenDownloadSDO () lets you access a particular object in the Object Directory of the message recipient and specify the value of the object. Data is exchanged in accordance with the SDO download protocol.
CanOpenAddPDORx ()	Calling up the CanOpenAddPDORx () function allows for process data coming from other CANopen devices to be entered for reception.
CanOpenAddPDOTx ()	By calling up the CanOpenAddPDOTx () function, process data can be placed on the bus.

Tab. 19: Supported STX functions

i INFO Further information

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

i INFO Further information

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

8.2.2 CANopen object directory

The operating system of the JVM-104-O08 supports the following objects:

Index (hex)	Object (Abbreviation)	Object name	Data type	Access
1000	VAR	Device type	Unsigned32	RO (read only)
1001	VAR	Error Register	Unsigned8	RO
1002	VAR	Manufacturer status	Unsigned32	RO
1003	ARRAY	Pre-defined error field	Unsigned32	RO
1008	VAR	Manufacturer Device name	String const	1008
1009	VAR	Manufacturer Hardware version	String const	1009
100 A	VAR	Manufacturer Software Version	String const	100 A
100 B	VAR	Node ID	Unsigned32	RO
1017	VAR	Producer heartbeat time	Unsigned16	R/W (read & write)
1018	RECORD	Identity	Identity	RO

Index (hex)	Object (Abbreviation)	Object name	Data type	Access
1200	RECORD	Server 1 – SDO parameter	SDO parameter	RO
1201	RECORD	Server 2 – SDO parameter	SDO parameter	R/W
1203	RECORD	Server 3 – SDO parameter	SDO parameter	R/W
1203	RECORD	Server 4 – SDO parameter	SDO parameter	R/W

i INFO

Further information

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

8.3 Storage options - Overview

The JVM-104-O08 features several types of program and data memory. 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. This type of memory is also referred to as remanent memory, and it is located directly in the CPU or in separate I/O modules.

This chapter gives an overview of the available memory.

8.3.1 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.

8.3.2 File system memory

Data and program files are stored in the file system memory.

Features

- Non volatile
- Size of the internal flash disk: 368 MB

Access

- By the operating system
- By JetSym
- By means of file commands from within the application program

8.3.3 Application program memory

By default, the application program is uploaded from JetSym to the JVM-104-O08 and is stored there.

- Properties**
 - Stored as file within the file system
 - Default directory *lapp\program name*
- Type of access**
 - By operating system
 - By JetSym
 - By means of file commands from within the application program

8.3.4 Special registers

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

- Features**
 - 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**
 - By JetSym
 - By browser (via HTTP server)
 - From HMIs
 - From within the application program
 - From other controllers

8.3.5 Flag

Flags are one-bit operands. This means they can either have the value 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: 1,792
 - 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

Memory access

- By JetSym
- From HMI's
- From within the application program

8.3.6 Storing registers and variables

Storing to a non-volatile (remanent) memory

Registers and variables of the application program can be globally stored to a non-volatile memory with dedicated registers.
 The register variables %VL are always 4 bytes in size. They are not initialized by the operating system.
 These %RL registers always occupy 480,000 bytes.
 The number of %VL register variables is limited to 120,000.
 The register array for %VL or %RL ranges from 1000000 to 1119999

Storing to volatile (non-remanent) memory

Variables of the application program can be stored to volatile memory.
 Global variables that do not have a static assignment to addresses and are stored compactly.
 Their register number starts with the value 0.

8.4 Control elements

This chapter describes how to program the control elements of the JVM-104-O08.

8.4.1 Input keys

The HMI JVM-104-O08 has the following 8 input keys: **[F1]**, **[F2]**, **[F3]**, **[F4]**, **[POWER]**, **[SCROLL]**, **[ESC]** and **[HOME]**. These input keys are user-programmable.

Special registers

In register 361000 of the JVM-104-O08 a bit-coded map of the input keys is available. You can use this map for programming.

The following registers are available for programming the input keys:

Register	Description
361000	Bit-coded map of the input keys
361000.0	[F1] key Bit 0 = 1: [F1] key is pressed.
361000.1	[F2] key Bit 1 = 1: [F2] key is pressed.
361000.2	[F3] key Bit 2 = 1: [F3] key is pressed.
361000.3	[F4] key Bit 3 = 1: [F4] key is pressed.
361000.4	[POWER] key Bit 4 = 1: [POWER] key is pressed.
361000.5	[SCROLL] key Bit 5 = 1: [SCROLL] key is pressed.
361000.6	[ESC] key Bit 6 = 1: [ESC] key is pressed.
361000.7	[HOME] key Bit 7 = 1: [HOME] key is pressed.

Tab. 20: Input key registers

Virtual key codes

The following table shows the symbolic hexadecimal values for the virtual key codes used by the system.

Input key	Value
[F1]	0x70
[F2]	0x71
[F3]	0x72
[F4]	0x73
[POWER]	0x74
[SCROLL]	0x75

Input key	Value
[ESC]	0x1B
[HOME]	0x24

Tab. 21: Virtual key codes

8.4.2 Digipot

The JVM-104-O08 has a rotary dial (digipot) with pushbutton feature which offers a convenient input option. The following provides details of the digipot's special registers with a corresponding sample program.

Register

Register	Description
363000	This register holds the current count value. If you turn the digipot, the count value increments or decrements. Where: <ul style="list-style-type: none"> ■ Turning the digipot clockwise increments the count value ■ Turning the digipot counter-clockwise decrements the count value
363001	Bit 0: 0 = Pushbutton not pressed Bit 0: 1 = Pushbutton pressed
363002	This register lets you specify the lower limit for the count value of the digipot. If you continue turning the digipot counter-clockwise, register 363000 remains at this minimum value.
363003	This register lets you specify the upper limit for the count value of the digipot. If you continue turning the digipot clockwise, register 363000 remains at this maximum value.

JetSym STX program

In the following sample program, the background lighting of the JVM-104-O08 is dimmed using the digipot. An upper and lower limit for the digipot is specified for this purpose. If you press the pushbutton, full background lighting is activated.

```

Var
  Digipot_Count : Int At %VL 363000;
  Digipot_Limit_min: Int At %VL 363002;
  Digipot_Limit_max: Int At %VL 363003;
  Digipot_Button : Int At %VL 363001;
  BackgroundLighting: Int At %VL 364000;
End_Var;

Task Main Autorun
  Digipot_Count := 0;
  Digipot_Limit_max := 17;
  Digipot_Limit_min := 0;
  Loop
    If Digipot_Button Then
      BackgroundLighting := 255;
    Else BackgroundLighting := Digipot_Count * 15;
    End If
  End Loop
End_Task;
    
```

i INFO	Further information For more information on this subject, refer to the JetSym Online Help.
---------------	--

8.5 Ignition and OFF delay

This chapter covers the ignition and the function `Shutdown()`.

Special registers

The special register 361100 of the JVM-104-O08 is responsible for prompting the state of the ignition. Where:

If then ...
Bit 0 = 0	Ignition is switched on and voltage is present at terminal 15 Ignition (+)
Bit 0 = 1	Ignition is off and no voltage is present at terminal 15 Ignition (+)

Default ignition function

The HMI has the following default settings in connection with ignition:

If and then ...
... voltage is applied to the HMI, the ignition is off,	... the HMI does not boot up.
... voltage is applied to the HMI, the ignition is on,	... the HMI boots up.
... the HMI is powered on, the ignition is switched off (not the power supply),	... the HMI remains switched on.

Shutdown() function - Options

Unlike the default ignition function, the `Shutdown()` function offers the following options:

- The HMI can be explicitly shut down.
- The HMI can be restarted.

Function declaration

```
Function Shutdown(Reboot:Bool) :Bool;
```

Function parameters

The `Shutdown()` function has the following parameter.

Parameter	Description	Value
Reboot	System restart	True
	The system is shut down	False

Return value

This function transmits the following return values to the higher-level program.

Return value	Description
0	OK
-1	Ignition is still switched on

i INFO

If the ignition is still switched on, the device will not be switched off. However, the HMI can be restarted. Such a restart is carried out irrespective of the fact that the ignition is on.

If you switch off the vehicle's ignition, the sample program executes the `Shutdown()` function after a delay of 3 seconds. The **Reboot** parameter for the `Shutdown()` function has the value **false**. This means that the device will switch off.

```
Var
  Ignition: Int At %VL 361100;
End_Var;
```

```
Task Ign Autorun
  Loop
    When Ignition Continue;
    Delay(3000);
    Shutdown(False);
  End_Loop;
End_Task;
```

i INFO**Further information**

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

8.6 Saving and loading an application program

The user determines the program that is to be executed. When uploading the application program to the controller, this program is stored as a file to the internal flash disk. The device enters the path and file name into the `\App\start.ini` file.

Path and file name

By default, JetSym creates in the `/app` directory a subdirectory and assigns the project name to it. Then, JetSym stores the application program to this subdirectory assigning the extension `*.es3` to it. Path and file names are always converted into lower case letters.

\App\start.ini - Structure

This file is a text file with one section holding 2 entries:

Element	Description
[Startup]	Section name
Project	Path to the application program file. This path is relative to <code>l app</code> .
Program	Name of the application program file

Example

```
[Startup]
Project = test_program
Program = test_program.es3
```

The application program is loaded from the file `\App\test_program\test_program.es3`.

Loading the application program

When the application program is restarted via JetSym or after booting the JVM-104-O08, the application program is loaded via the file system and executed.

The application program is loaded by the OS of the JVM-104-O08 as follows:

Step	Description
1	The operating system reads the file <code>\App\start.ini</code> from the internal flash disk.
2	The OS evaluates the Project entry. It contains the path leading to the application program file.
3	The OS evaluates the Program entry. This entry contains the program name.
4	The OS loads the application program from the file <code><Project>/<Program></code> .

9 Registers - Overview

This register overview gives a condensed summary of the registers of the JVM-104-O08 device running OS version 4.12.0.11.

Default address on the CANopen bus

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

Node ID	127 (0x7F)
---------	------------

General overview - Registers

Register range	Description
100000 ... 100999	Electronic Data Sheet (EDS)
106000 ... 106999	CAN
108000 ... 108999	CPU/backplane
200000 ... 209999	General system registers
210000 ... 219999	Application program
310000 ... 319999	File system/data files
360000 ... 369999	Display
1000000 .. . 1029999	Application registers (non-volatile)

I/Os - General overview

Register range	Description
361000 ... 361007	Bit-coded map of input keys

Flags - General overview

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

Electronic Data Sheet (EDS)

EDS data can be retrieved for the CPU or the entire device.

Register range	Description
100500	Interface
0	CPU
1	Baseboard
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 ... 100802	Features
100800	Internal version number
100801	MAC address (Jetter)
100802	MAC address (JVM-104-O08)

Electronic nameplate (entire device)

Register range	Description
100708	Day
100709	Month
100710	Year
100900 ... 100907	Production
100900	Internal version number
100901 ... 100907	Serial number (register string)
100950 ... 100993	Identification
100950	Internal version number
100951	Module ID
100952 ... 100962	Module name (register string)
100965	Config ID
100966	Vendor ID
100967	Variant ID
100968	Type ID
100992	Navision ID
100993	FBG version

CAN

Register	Description
106000	Baud rate CAN 1
106001	Node ID CAN 1

Flash memory

Register range	Description
107510 ... 107513	Sector statistics
107510	Available sectors
107511	Used sectors
107512	Blocked sectors
107513	Free sectors
107520 ... 107523	Byte statistics

Register range	Description
107520	Size of the flash disk in bytes
107521	Used memory in bytes
107522	Blocked memory in bytes
107523	Free memory in bytes

System information

Register	Description
108500 ... 108509	JetVM DII version string
108510 ... 108519	Version string of the host application
108520 ... 108529	File name of the host application
108530 ... 108539	OS version (string)
108570	CPU type
108571	Number of CPUs
108573	Physical RAM
108574	Free physical RAM
108575	Memory utilization (in %)
108581	Screen width (in pixels)
108582	Screen height (in pixels)
108590	HID version

General system registers

Register	Description
200000	OS version (major * 100 + minor)
200001	Application program is running (bit 0 = 1)
0/2	Stop program
1	Start program
3	Continue program
200008	Error register 1 (identical with 210004)
Bit 8	Illegal jump
Bit 9	Illegal call
Bit 10	Illegal index
Bit 11	Illegal opcode
Bit 12	Division by 0

Register	Description
Bit 13	Stack overflow
Bit 14	Stack underflow
Bit 15	Illegal stack
Bit 16	Error when loading the application program
Bit 24	Timeout - Cycle time
Bit 25	Timeout - Task lock
Bit 31	Unknown error
200168	Boot loader version (IP format)
200169	OS version (IP format)
201000	Runtime register in milliseconds (rw)
201001	Runtime register in seconds (rw)
201002	Runtime register in register 201003
201003	10 ms units for register 201002 (rw)
201004	Runtime register in milliseconds (ro)
201005	Runtime registers in microseconds (ro)
202960	Password for system command register (0x424f6f74)
202961	System command register
202980	Error history: Number of entries
202981	Error history: Index
202982	Error history: Entry
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 ... 209739	Enabling system components

Application program

Register	Description
210000	Application program is running (bit 0 = 1)
210001	JetVM version
210004	Error register (bit-coded)
Bit 8	Illegal jump
Bit 9	Illegal call
Bit 10	Illegal index
Bit 11	Illegal opcode
Bit 12	Division by 0
Bit 13	Stack overflow
Bit 14	Stack underflow
Bit 15	Illegal stack
Bit 16	Error when loading the application program
Bit 24	Timeout - Cycle time
Bit 25	Timeout - Task lock
Bit 31	Unknown error
210006	Highest task number
210007	Minimum program cycle time
210008	Maximum program cycle time
210009	Current program cycle time
210011	Current task number
210050	Current program position within an execution unit
210051	ID of the execution unit being processed
210056	Desired total cycle time in µs
210057	Calculated total cycle time in µs
210058	Maximum time slice per task in µs
210060	Task ID (for R210061)
210061	Priority for task [R210060]
210063	Length of scheduler table
210064	Index in scheduler table
210065	Task ID in scheduler table
210070	Task ID (for R210071)
210071	Timer number (0 ... 31)
210072	Manual triggering of a timer event (bit-coded)
210073	End of cyclic task (task ID)

Register	Description
210074	Command for cyclic tasks
210075	Number of timers
210076	Timer number (for R210077)
210077	Timer value in milliseconds
210100 ... 210199	Task state Please use the STX function TaskGetInfo() as described in the JetSymb online help.
210400 ... 210499	Task - Program address
210600	Task ID of a cyclical task (for R210601)
210601	Processing time of a cyclical task in per mil figure
210609	Task lock timeout in ms
-1	Monitoring disabled
210610	Timeout (bit-coded, bit 0 → timer 0 etc.)

File system/data file function

Register	Description
312977	Status of file operation
312978	Task ID

Application registers

Register	Description
1000000 ... 1005999	32-bit integer (remanent)

Display

Register range	Description
361000 ... 361007	Bit-coded map of input keys (e.g. bit 0 =1 → key 1 pressed)
361000.0	[F1]
361000.1	[F2]
361000.2	[F3]
361000.3	[F4]
361000.4	[POWER]
361000.5	[SCROLL]

Register range	Description
361000.6	[ESC]
361000.7	[HOME]
361100	Ignition (IGN)
Bit 0 = 0	The ignition is ON
Bit 0 = 1	Ignition is OFF
363000 ... 363003	Digipot
363000	Present count value
363001	Digipot key
363002	Minimum count value
363003	Maximum count value
364000 ... 364001	Illumination
364000	Backlighting
364001	Night-lighting of keys
365100	Visualization
365100	Language selection according to ID

32 combined flags

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

16 combined flags

Registers	Description
203108	0 ... 15
203109	16 ... 31
203110	32 ... 47
203111	48 ... 63
203112	64 ... 79
203113	80 ... 95

Registers	Description
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

Registers	Description
203146	2272 ... 2287
203147	2288 ... 2303

Application registers with overlaid flags

Registers	Description
1000000	256 ... 287
1000001	288 ... 319
1000002	320 ... 351
1000003	352 ... 383
1000004	384 ... 415
1000005	416 ... 447
1000006	448 ... 479
1000007	480 ... 511
1000008	512 ... 543
1000009	544 ... 575
1000010	576 ... 607
1000011	608 ... 639
1000012	640 ... 671
1000013	672 ... 703
1000014	704 ... 735
1000015	736 ... 767
1000016	768 ... 799
1000017	800 ... 831
1000018	832 ... 863
1000019	864 ... 895
1000020	896 ... 927
1000021	928 ... 959
1000022	960 ... 991
1000023	992 ... 1023
1000024	1024 ... 1055
1000025	1056 ... 1087
1000026	1088 ... 1119
1000027	1120 ... 1151
1000028	1152 ... 1183
1000029	1184 ... 1215
1000030	1216 ... 1247
1000031	1248 ... 1279
1000032	1280 ... 1311
1000033	1312 ... 1343

Registers	Description
1000034	1344 ... 1375
1000035	1376 ... 1407
1000036	1408 ... 1439
1000037	1440 ... 1471
1000038	1472 ... 1503
1000039	1504 ... 1535
1000040	1536 ... 1567
1000041	1568 ... 1599
1000042	1600 ... 1631
1000043	1632 ... 1663
1000044	1664 ... 1695

Registers	Description
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	Sorting register values
90	Writing a data file
91	Appending a data file
92	Reading a data file
96	Deleting a data file

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: STXBASETTYPE, SortMode: QSORTMODE): 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;

10 Maintenance

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

10.1 Repairs

Defective components could cause dangerous malfunctions and could compromise safety.

Only the manufacturer is allowed to repair the device. It is forbidden to open the device.

Modifications to the device

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

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

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

10.2 Return and disposal

How to dispose of waste equipment

Return your Jetter AG product to us for proper disposal. Visit our [homepage](#) for detailed information and to download the required Returns form.

Meaning of the WEEE icon

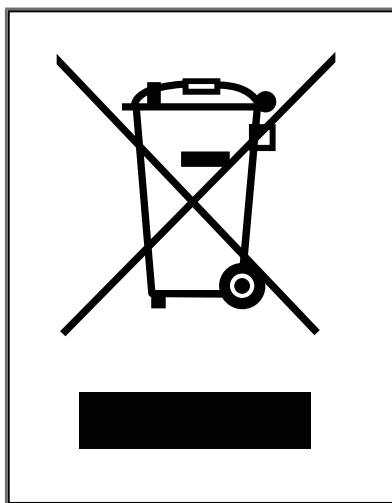


Fig. 9: WEEE icon – crossed-out trash can

Applicable local environmental directives and regulations must be complied with. 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.

Personal data

Customers are responsible for deleting personal data from waste equipment prior to its disposal.

10.3 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 Jetter AG of the damage caused during shipment. If the device is damaged or has been dropped, it is strictly forbidden to use it.

11 Service

11.1 Customer service

Should you have any questions, suggestions, or problems, please don't hesitate to contact our service representatives. To contact them, please call our technical hotline or use the contact form on our homepage:

[Technical hotline | Jetter - We automate your success.](#)

You are also welcome to send an e-mail to our technical hotline:

hotline@jetter.de

Please supply the following information when contacting our technical hotline:

- Hardware revision and serial number
For the hardware revision and serial number of your product, please refer to the nameplate.
- OS version
To determine the operating system version, use the functions of the development environment.

12 Spare parts and accessories

NOTICE



Inadequate accessories might cause damage to the product

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

- ▶ Only use accessories recommended by Jetter AG.

12.1 Accessories

INFO

Ordering accessories

The accessories are not part of the scope of delivery.
Suitable accessories can be obtained from Jetter AG.

Accessories	Item number
Mounting plate for RAM Mount ball consisting of mounting plate and screws for housing with Deutsch or M12 connector, without RAM Mount attachments	10001621
Mounting plate for RAM mount arm with suction cup consisting of mounting plate and screws for housings with Deutsch/M12 connector including RAM Mount arm with suction cup	10001551
Connecting cable Length 5 m, pre-assembled on one end with M12x1 connector, 8-pole, opposite end open	60882261
Interconnecting programming cable M12 on both ends, pre-assembled and tested, CAN1 brought out to Sub-D	60882086

List of figures

Fig. 1	Design	8
Fig. 2	Nameplate	9
Fig. 3	Dimensions in mm	10
Fig. 4	Space requirements for installation work (in mm)	15
Fig. 5	Screw holes, dimensions in mm	16
Fig. 6	Installation drawing.....	17
Fig. 7	Installing the strain relief.....	18
Fig. 8	8-pin M12 male connector	20
Fig. 9	WEEE icon – crossed-out trash can	45

List of tables

Tab. 1	Mechanical specifications	10
Tab. 2	Power supply VBAT_ECU	11
Tab. 3	CAN interface specification	11
Tab. 4	Specification - CAN bus cable	11
Tab. 5	Cable lengths.....	12
Tab. 6	Environmental conditions	12
Tab. 7	Technical data – display	12
Tab. 8	Acoustic signal generator	12
Tab. 9	Pulses ISO 7637-2	13
Tab. 10	Irradiation ISO 11452	13
Tab. 11	ESD EN 61000-4-2.....	13
Tab. 12	Unsuitable installation locations	15
Tab. 13	Compatible M12 connector (female)	20
Tab. 14	JetEasyDownload Parameters	22
Tab. 15	Abbreviations.....	26
Tab. 16	Module register properties.....	26
Tab. 17	Numerical formats	26
Tab. 18	JetSym sample programs.....	26
Tab. 19	Supported STX functions.....	27
Tab. 20	Input key registers	32
Tab. 21	Virtual key codes	32

Jetter AG
Graeterstrasse 2
71642 Ludwigsburg
www.jetter.de

E-mail info@jetter.de
Phone +49 7141 2550-0

We automate your success.